PUCRS

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DOUTORADO

DOUGLAS MATOS DE SOUZA

EFFICIENT AND MULTILINGUAL TEXT-TO-IMAGE SYNTHESIS: EXPLORING NOVEL ARCHITECTURES AND CROSS-LANGUAGE STRATEGIES

Porto Alegre
2024

PÓS-GRADUAÇÃO - *STRICTO SENSU*

Pontifícia Universidade Católica
do Rio Grande do Sul

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

EFFICIENT AND MULTILINGUAL TEXT-TO-IMAGE SYNTHESIS: EXPLORING NOVEL ARCHITECTURES AND CROSS-LANGUAGE STRATEGIES

DOUGLAS MATOS DE SOUZA

Doctoral Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Ph. D. in Computer Science.

Advisor: Prof. Duncan D. Ruiz

Porto Alegre
2024

Ficha Catalográfica

S729e Souza, Douglas Matos de

Efficient and multilingual text-to-image synthesis : Exploring novel architectures and cross-language strategies / Douglas Matos de Souza. – 2024.

97f.

Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz.

1. Cross-language text-to-image synthesis. 2. Generative adversarial networks. 3. Generative models. 4. Deep neural networks. I. Ruiz, Duncan Dubugras Alcoba. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS com os dados fornecidos pelo(a) autor(a).

Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

Douglas Matos de Souza

Efficient and Multilingual Text-to-Image Synthesis: Exploring Novel Architectures and Cross-language Strategies

This Doctoral Thesis has been submitted in partial fulfillment of the requirements for the degree of Doctor of Computer Science, of the Graduate Program in Computer Science, School of Technology of the Pontifícia Universidade Católica do Rio Grande do Sul.

Sanctioned on January 15th, 2024.

COMMITTEE MEMBERS:

Prof. Dr. Lucas Silveira Kupssinsku (PPGCC/PUCRS)

Prof. Dr. Adriano Alonso Veloso (UFMG)

Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho (USP)

Prof. Dr. Duncan Dubugras Alcoba Ruiz (PPGCC/PUCRS - Advisor)

ACKNOWLEDGMENTS

In order to better reach the honored individuals, the following acknowledgments are writen in Brazilian Portuguese.

Em primeiro lugar, gostaria de expressar a mais profunda gratidão à minha família, cujo apoio incondicional foi o pilar fundamental para a realização desta jornada. Em especial, à minha mãe, Maria Helena, que com seu amor, dedicação e sacrifícios, sempre me inspirou e me incentivou a perseguir meus sonhos. Sua força e sabedoria foram a luz que me guiou nos momentos mais desafiantes e suas orações me protegeram em todos os momentos. Não existem palavras que possam traduzir a importância do seu papel nesta conquista.

Não poderia deixar de agradecer ao meu orientador e grande amigo, Duncan Ruiz, por todos estes anos de parceria e colaboração. Sua dedicação e orientação foram essenciais para que esta pesquisa atingisse seus objetivos. Sua paciência, incentivo e paixão pelo conhecimento foram essenciais, e as lições aprendidas se estenderão para muito além destes anos de estudo. A confiança e liberdade que depositou em meu trabalho foram fundamentais para o meu desenvolvimento acadêmico e pessoal.

A jornada do doutorado é repleta de desafios e descobertas, e sem a presença, incentivo e compreensão destas pessoas, ela certamente não teria sido a mesma. A todos vocês, meu sincero muito obrigado.

SÍNTESE EFICIENTE E MULTI-IDIOMA DE IMAGENS A PARTIR DE TEXTO: EXPLORANDO NOVAS ARQUITETURAS E ESTRATÉGIAS MULTILÍNGUE

RESUMO

A síntese de texto para imagem é a tarefa de gerar imagens a partir de descrições textuais. Dada uma descrição textual, um algoritmo de síntese de imagens a partir de texto pode gerar várias imagens inéditas que contenham os detalhes descritos no texto. Estes algoritmos são atrativos para várias tarefas do mundo real. Com tais algoritmos, seria possível utilizar máquinas para criar imagens totalmente inéditas para geração de conteúdo ou para realizar desenhos assistidos, por exemplo. A estrutura geral das abordagens para síntese de imagens a partir de texto pode ser dividida em duas partes principais: i) um codificador de texto e ii) um modelo gerador para imagens, que aprende uma distribuição condicional sobre o texto codificado. Atualmente, as abordagens de síntese de imagens a partir de texto utilizam várias redes neurais para superar os desafios de aprender um modelo gerador sobre as imagens, aumentando a complexidade geral da abordagem, bem como a computação necessária para gerar imagens de alta resolução. Até o momento, nenhum trabalho explorou modelos que suportem múltiplos idiomas no contexto da geração de imagens a partir de texto, limitando as abordagens atuais a suportarem apenas o inglês. Esta limitação apresenta uma desvantagem significativa, pois restringe o acesso à tecnologia apenas para usuários familiarizados com a língua inglesa, deixando de fora um número substancial de pessoas que poderiam se beneficiar. Nesta tese, realizamos as seguintes contribuições para abordar cada uma das lacunas mencionadas anteriormente. Primeiramente, propomos uma nova abordagem de síntese de imagem a partir de texto, de ponta a ponta, que utiliza apenas uma rede neural para o modelo gerador de imagens, reduzindo a complexidade e a computação necessária. Em segundo lugar, propomos uma nova função de custo, que aprimora o treinamento e produz modelos mais precisos. Por fim, estudamos como os codificadores de texto afetam o desempenho geral da geração de imagens a partir de texto e propomos uma nova abordagem de múltiplas linaguagens para ampliar os modelos e suportar múltiplos idiomas simultaneamente.

Palavras-Chave: síntese multi-idiomas de imagens a partir de texto, redes geradoras adversárias, modelos geradores, redes neurais profundas, aprendizado profundo.

EFFICIENT AND MULTILINGUAL TEXT-TO-IMAGE SYNTHESIS: EXPLORING NOVEL ARCHITECTURES AND CROSS-LANGUAGE STRATEGIES

ABSTRACT

Text-to-image synthesis is the task of generating images from text descriptions. Given a textual description, a text-to-image algorithm can generate multiple novel images that contain the details described in the text. Text-to-image algorithms are appealing for various real-world tasks. With such algorithms, machines can draw truly novel images that can be used for content generation or assisted drawing, for example. The general framework of text-to-image approaches can be divided into two main parts: i) a text encoder and ii) a generative model for images, which learns a conditional distribution over encoded text. Currently, text-to-image approaches leverage multiple neural networks to overcome the challenges of learning a generative model over images, increasing the overall framework's complexity as well as the required computation for generating high-resolution images. Additionally, no works so far have explored cross-language models in the context of text-to-image generation, limiting current approaches to supporting only English. This limitation has a significant downside as it restricts access to the technology to users familiar with the English language, leaving out a substantial number of people who could benefit. In this thesis, we make the following contributions to address each of the aforementioned gaps. First, we propose a new end-to-end text-to-image approach that relies on a single neural network for the image generator model, reducing complexity and computation. Second, we propose a new loss function that improves training and yields more accurate models. Finally, we study how text encoders affect the overall performance of text-to-image generation and propose a novel cross-language approach to extend models to support multiple languages simultaneously.

Keywords: cross-language text-to-image synthesis, generative adversarial networks, generative models, deep neural networks.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

RNN – Recurrent Neural Network

VAE – Variational Auto Encoder

GAN – Generative Adversarial Network

VQ-VAE – Vector-Quantized Variational Auto Encoder

IS – Inception Score

FID – Fréchet Inception Distance

LSTM – Long Short-term Memory

CUB – Caltech-UCSD Birds Dataset

DAMSM – Deep Attentional Multimodal Similarity Model

GRU – Gated Recurrent Unit

CA – Conditioning Augmentation

SI – Sentence Interpolation

LIST OF SYMBOLS

CONTENTS

1. INTRODUCTION

Text-to-image synthesis is the task of generating images from text descriptions. Consider the example in Figure 1.1. In this example, a text-to-image algorithm has generated 100 images for the following text description: *"light tan colored bird with a white head and an orange beak."*. This visual example is useful for us to observe some important characteristics of text-to-image synthesis: i) all generated images are novel (*i.e.* none of the birds really exist); ii) all content present in generated images present a strong correlation with the text present in the description and iii) it is possible to generate a finite but huge number of novel images for a single text description.

All of these aspects make text-to-images algorithms appealing for several real world tasks. If we have such solution, we can use machines to draw truly novel images that can be used for content generation or for assisted drawing, for example. Although promising, text-to-image generation is both complex and challenging. Given the subjective characteristics present in text descriptions, there are an immense set of images that may satisfy what a simple statement describes. Furthermore, due to its multi-modal nature, *i.e.* involving text and images, it requires combining techniques from both computer vision and natural language processing.

The first method that tried to address text-to-image synthesis was proposed by Reed *et al.* [55] in 2016. This work extended the first Convolutional GAN [51] architecture to be conditioned on text descriptions. This method used an Recurrent Neural Network (RNN) [26, 6] to encode sentences to feature vectors so that this vectors could be used as a conditioning factor to train a conditional version of a Convolutional GAN. This was the first work that embraced the text-to-image pipeline in an end-to-end fashion, from characters to pixels. Despite generating low resolution and low quality images, Reed's work opened the community to the possibility of creating such models.

Since 2016, the research field of text-to-image synthesis quickly gained momentum and became one of the most researched topics in academia and industry. The timeline of evolution of text-to-image research can be divided in two main parts: the methods prior large-scale training (2020 and backwards) and methods post large-scale training (2020 onwards). Before large-scale training, text-to-image methods were based on a variation of a conditional GAN and a text encoder. After large-scale training, the paradigm shifted to large VQ-VAEs [47, 54] and Diffusion models [25, 57]. The topic of research of this thesis is based in the prior paradigm of small-scale training.

The general small-scale framework of text-to-image approaches can be divided in two main components: i) a text encoder and ii) a conditional GAN that learns a conditional distribution over encoded text. The text encoder is usually implemented using a type of RNN. Since the text encoder is another learning algorithm, there is not a clear definition of the best metrics or the best way to encode text so that quality of image synthesis is maximized. This is one of the challenges that is often overlooked in text-to-image synthesis. Most works pay little attention to the text encoding and end up reusing text encoder from previous works [78, 81, 49, 73].

The second part, which is a type of conditional GAN, presents its own issues. Since its first appearance in 2014, GANs [18] quickly received attention due its great capacity of learning

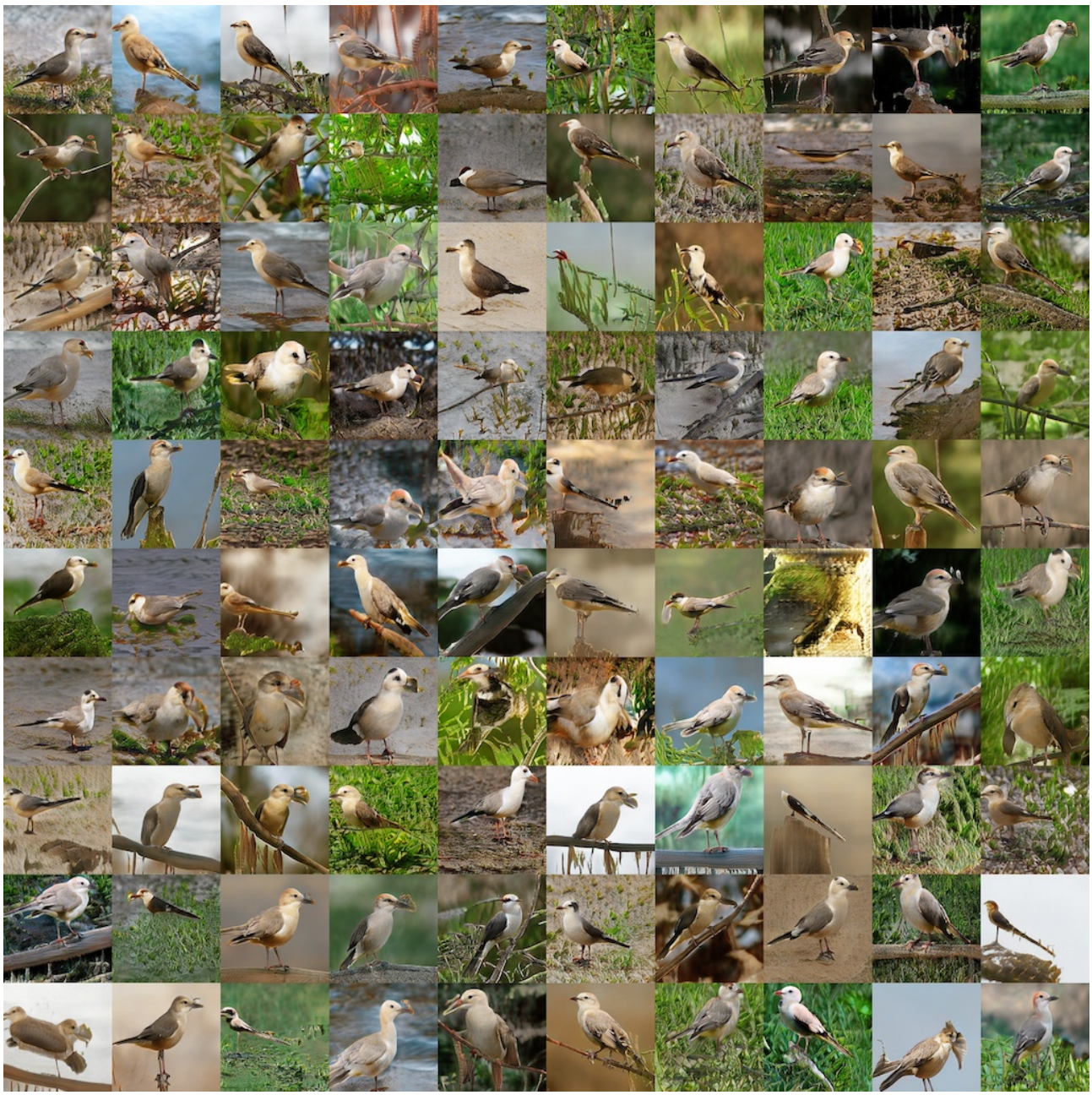


Figure 1.1 – Images generated for the caption *"light tan colored bird with a white head and an orange beak."* [59].

generative models over complex data, such as images. However, GANs are models highly unstable. Therefore, a great research effort has been made [58, 1, 20, 42] to make GANs more easily trainable and less prone to several drawbacks. Still, GANs are models highly sensitive to hyper-parameters, and successfully training is a difficult task. To this day, there is not a consensus on the best neural architecture and/or loss function and optimization strategies.

Another important aspect of text-to-image generation research is that it has been language restrictive, all research has been focusing on experimenting on the same datasets and in the same language: English. This lack of research involving other languages is problematic because it excludes a vast portion of global linguistic diversity, potentially reinforcing biases and limiting the accessibility

of this technology to non-English speakers. Furthermore, this monolingual focus overlooks the rich cultural nuances and visual symbolism inherent in different languages, which could inform and enhance the generative models. Expanding the scope to include multilingual datasets would not only democratize the technology, making it more inclusive, but also improve the robustness and generalization capabilities of these models. By incorporating a variety of languages and cultural contexts, these generative systems could be trained to better understand and interpret the intricacies of human communication, leading to more accurate and diverse visual representations.

The rest of this PhD thesis is structured as follows: Chapter 2 presents the problem statement as well as the specific aims of this study. Chapter 3 provides an introduction to the foundational machine learning concepts that are required for text-to-image synthesis. Chapter 4 presents the initial contribution of the thesis, which is a proposed novel and efficient architecture designed specifically for text-to-image synthesis tasks. A subsequent contribution is explored in Chapter 5, where a novel loss function is introduced that seeks to improve the training processes of text-to-image generative models. The third contribution is covered in Chapter 6, discussing the development of a technique that extends the functionality of text-to-image generative models to include multilingual support. An in-depth discussion on the contributions made by this research, as well as an analysis of the results obtained, is shown in Chapter 7. Chapter 8 presents a review of the literature and related works in the field of text-to-image synthesis. Finally, the conclusion is presented in Chapter 9, which summarizes the key findings and outlines potential avenues for future research.

2. THESIS OBJECTIVES

This Chapter presents the problem this thesis aims to solve as well as its general and specific objectives.

2.1 Problem Statement

The rapid-growing field of text-to-image synthesis has seen a substantial evolution with the advent of deep neural networks, enabling the generation of realistic and elaborate visual contents from text descriptions. However, as reported in a survey from Frolov *et al.* [16], the field still suffers with major deficiencies:

- State-of-the-art methods in small-scale text-to-image generation rely on complicated frameworks that require multiple networks to work.
- Major works make use of the same adversarial loss functions from class-conditioned models in the text-to-image context.
- No work so far has investigated the impact and the benefits of extending text-to-image generation to multiple languages.

This thesis proposes developing a streamlined and linguistically versatile text-to-image model, which integrates a novel neural architecture and an optimized loss function, will effectively address these challenges. The proposed research aims to make significant strides in each of the aforementioned areas of contribution.

2.2 General Objectives

This work aims to address each gap by first understanding the current limitations through a comprehensive review of existing methods and then by systematically addressing each limitation through the proposed research contributions. The ultimate goal of this research is to present a unified solution that advances the state-of-the-art in text-to-image synthesis and democratizes its access through a novel approach to support multiple languages simultaneously.

2.3 Specific Objectives

The specific objectives of this work are:

1. Designing a new efficient neural architecture for text-to-image synthesis: current models often rely on complex, multiple networks frameworks that are not suitable for extensibility and fast inference. This work proposes a new neural architecture that increases the computational efficiency. The architecture is designed to use a single pair of generator/discriminator that can be trained directly at target resolution, all while improving the visual quality of the generated images.

2. Proposing a new loss function to improve training for text-to-image models: the training of text-to-image models involves training a conditional GAN on text embeddings. The nature of text embeddings differ dramatically from discrete data – as the usual case of GANs conditioned on class labels – because they are vector of continuous numbers. In order to adapt this conditioning to text-to-image generation in a more suitable way, we introduce a novel loss function tailored to better capture the nuances of visual semantics and textual alignment, which guides the model to produce images that are both visually appealing and contextually appropriate to the input text.

3. Developing a new cross-lingual method for text-to-image models: since all major datasets contain captions only in English, all existing text-to-image models are predominantly monolingual. This setup is negative in two ways: i) it restricts the research of text-to-image comprising different languages, and ii) it restricts the access to text-to-image models to entire communities that are not familiar to the English language. To address this issue the proposed research includes the development of a cross-lingual method that enables a single model to understand and generate images from descriptions in multiple languages simultaneously. This multilingual ability expands the ise of text-to-image synthesis, allowing for broader applications in different linguistic contexts.

3. BACKGROUND

In this section, we lay out the context required to follow the study presented this thesis.

3.1 Generative Models For Images

The goal of a generative model is to infer a model that can map a known distribution $p_{\boldsymbol{z}}$ – also called the *the latent space* – to the data distribution $p_{\boldsymbol{x}}$ (training set). The known distribution is often defined as random distribution, such as $\mathcal{N}(0, 1)$. The generative model that learns the mapping $\boldsymbol{z} \mapsto \boldsymbol{x}$ is called *generator*. A generative model seeks to understand and replicate the distribution of real-world data, $p_{\boldsymbol{x}}$, by learning an intricate transformation from a simpler, predefined probability distribution, $p_{\boldsymbol{z}}$, which is the latent space. This latent space typically takes the form of a low-dimensional vector space, in which each vector $\boldsymbol{z}$ corresponds to a conceptual point or a seed that, through the generative process, can be transformed into a complex data instance similar to those found in the actual data distribution. The transformation is commonly implemented by a computational construct known as the *generator*, G, which represents a function that maps the latent vectors to data space, i.e., $G(\boldsymbol{z}) = \boldsymbol{x}'$. Where $\boldsymbol{x}'$ is the generated data instance that ideally should resemble a genuine sample from the true distribution $p_{\boldsymbol{x}}$.

During the training process, the generator adjusts its parameters to create samples that are increasingly indistinguishable from the real data. To guide this training, many generative models rely on another component known as the *discriminator* in adversarial frameworks, such as Generative Adversarial Networks (GANs). The discriminator evaluates the samples produced by the generator and tries to distinguish between real and generated data. Through an iterative process of competition, both the generator and discriminator improve, leading to more realistic synthetic samples.

Early generative models, were partially successful in learning the structure of low dimensional and often discrete data, like text. In the case of images, however, they fell short of the goal. The main reason for that is that images are a high dimensional and diversity-rich data. Images I are defined as 3D tensors $\mathbb{R}^{h \times w \times 3}$ of pixels, where h and w are the height and width of the image, respectively. The third dimension is the color channel, which is 3 for RGB images. An image of size $h \times w \times 3 = 224 \times 224 \times 3$, for instance, has $150,528$ pixels. Converting to a vector, this yields 150,528-dimensional vector, a size that was not only too large for most generative models but also very difficult to learn from, since the the flattening discards all the important spatial information.

Thanks to the advent of an entire machine learning area named *Deep Learning* that made heavy use of deep neural networks, handling high-dimensional data became feasible. Image-wise, we can mention the Convolutional Neural Networks (CNNs) [34] as the one of the most important developments in the area. CNNs became the start-of-the-art in several computer vision tasks, such as image classification [33], object detection [72], image segmentation [21] and many others. More recently, new breakthroughs were achieved by a new type of model called Transformer [63].

Transformer is a new feed-forward architecture that was designed mainly for text. Recently, it has shown state-of-the-art performance for images as well [12]. The success of Transformers rely on two main components: the attention module [63, 66] and large-scale training [63, 4, 50, 53].

In the context of generative models for images, Variational Autoencoders (VAEs) [31] and Generative Adversarial Networks (GANs) [18] were the pioneering architectures that changed the landscape of image generation across the machine learning field. Introduced in the early 2010s, they represented a significant leap forward in the ability of these models to emulate real-world visual data. VAEs use probabilistic inferences and optimization principles to create high-quality, diverse images. GANs, on the other hand, leverage a game theory-inspired approach, involving a duel between two neural networks – a generator and a discriminator. By jointly training these networks, GANs can generate strikingly realistic images. Their success in image synthesis has been pivotal in various applications, including art creation, image super-resolution, and domain adaptation. Their introduction marked a milestone in the evolution of machine learning, enabling the design and development of more reliable, efficient, and realistic generative models.

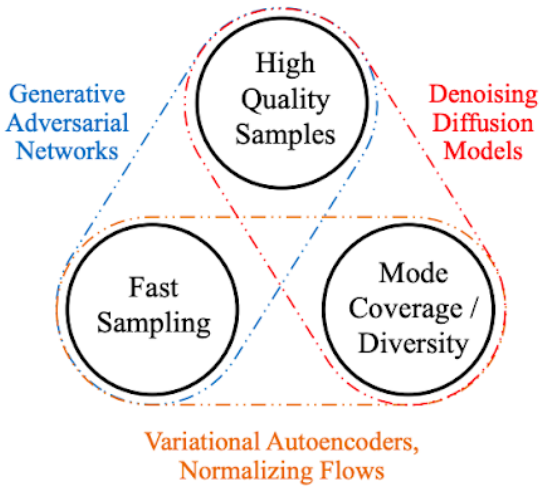


Figure 3.1 – Image synthesis trilemma. Figure from NVIDIA blog post [62].

An important breakthrough in image generation happened with the discovery of the impact of large-scale training. It has brought a whole new perspective to image generation research. Up to that point, most of the work focused on improving model architecture and training procedures. The first major work to report outstanding results with large-scale training was Big-GAN [4]. After that, other works followed, such as the second iteration of VQ-VAE [47], named VQ-VAE-2 [54], and the whole new class of generative models called Denoising Diffusion Probabilistic Models [25]. Although those approaches presented great results, each method has its own limitations.

All of the generative models available today are restricted by the image generation trillema, as shown in Figure 3.1, there is not a single model class that can cover the three pillars of image generation at once: high quality samples, fast sampling and mode coverage. Therefore, when choosing an approach, one of the pillars must be left out. For example, if image quality and sampling speed is paramount, then the best choice is GAN [18]. On the other hand, if sampling

speed and mode coverage are more important, then the obvious choice is a VAE [31] – or some of its variations. Finally, if sampling quality and mode coverage are most prioritized requirements, the best choice is a Diffusion model [25].

Besides model architectures and training procedures, a big breakthrough in image generation – and text-to-image generation as a byproduct – was achieved by scaling up training. Large-scale models are defined by the following components:

- Large training datasets: current large-scale models are training on datasets in the magnitude of tens of millions examples. As a comparison, small scale training uses datasets that contain the order tens of thousand examples.
- Large models: most recent models are huge in respect of number of trainable parameters, it is common to see models having a few billion trainable parameters, as opposed as the tens of thousands used in small scale training.
- Large computational requirement: current large-scale models require orders of hundreds high-end GPUs that contain the sums of 80GB of memory per GPU.

In the context of large-scale training, some important breakthroughs were made. Following on the success of CLIP [50], OpenAI's DALL-E [53] combined the GPT-3 [5] transformer model with a VQ-VAE-2[54]. DALL-E's demonstrated capabilities represent the significant advancements that large-scale training brings to the field of machine learning and image generation. After DALL-E [53], improved methods were proposed, such as the DALL-E-2 [52], which replaced the VQ-VAE-2 to a Denoising Diffusion Model and the Stable Diffusion [57], which shifted the diffusion process from the pixel space to a latent space, yielding a computationally more efficient model.

3.2 Deep Generative Models

In this section, we show, in-depth, the inner workings of the most popular generative models.

3.2.1 Generative Adversarial Networks

In recent developments in Deep Learning, Goodfellow et. al. [18] have introduced the Generative Adversarial Networks (GANs). This method is able to learn generative models over complex data distributions. Generative adversarial nets are composed by two differentiable functions (i.e. neural networks), namely a **generator** and a **discriminator**. The generator and the discriminator are set to play a two-player minimax game against each other. The discriminator is trained using usual supervised learning in order to distinguish between two classes (real/fake), while the generator is trained to fool the discriminator. The analogy usually made is that the generator is like a

Figure 3.2 – Figure generated for the prompt "*A computer scientist beaver working on his machine learning PhD research. High detailed drawing.*" by Dalle-3 [3].

counterfeiter that tries to make fake money, and the discriminator is like the police that wants to allow only legitimate money and catch fake ones. To win this game, the counterfeiter (generator) needs to learn how to generate fake money (fake data samples) that are indistinguishable from the legitimate money (real data samples).

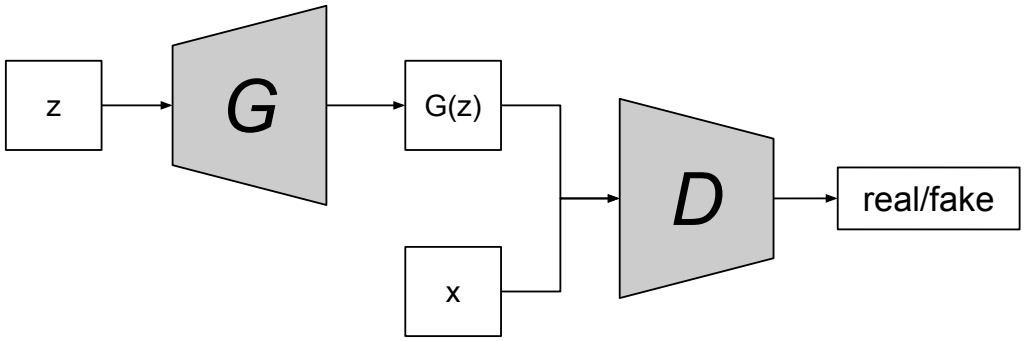


Figure 3.3 – Scheme of a regular Generative Adversarial Network.

Formally, the inner workings of a GAN is as follows: from an input noise $\boldsymbol{z}$ sampled from a prior $p_z(\boldsymbol{z})$, which is a known simple distribution (like a Gaussian or uniform), the generator maps a sample $G(\boldsymbol{z})$ to the data space aiming to learn its own distribution p_g over the real data $\boldsymbol{x}$. The discriminator D takes an input data $\boldsymbol{x}$ and outputs a scalar, which is the probability that the input came from the real data $\boldsymbol{x}$ rather than p_g. A general GAN architecture is shown in Figure 3.3. D is then trained to maximize the probability of assigning the correct class label for both the real data

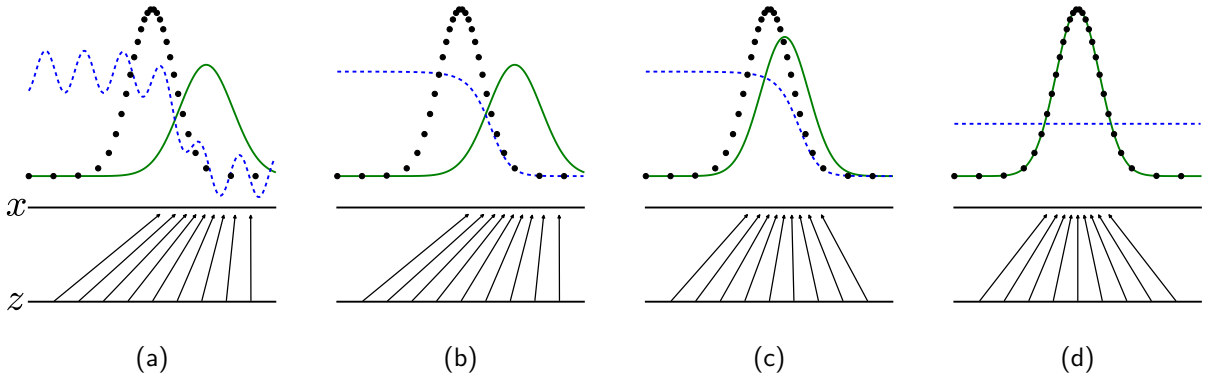


Figure 3.4 – An illustration of a GAN near convergence from Goodfellow et al. [18]. The lower horizontal line show the domain of $\boldsymbol{z}$, which in this case is uniform. The line above shows the domain of the real data $\boldsymbol{x}$, which is Gaussian. Up arrows represent the generator mapping function (notice the contraction needed to map a uniform distribution to a Gaussian). Black dotted line represents the distribution of the real data, the green line represents the distribution learned by the generator and, finally, the blue dashed line represent the decision boundary of the discriminator. Specifically: a) a GAN near convergence after a generator's update; b) after the generator's update, the discriminator is forced to move its decision boundary in order to discriminate better between real and fake distributions; c) the generator is updated, then its distribution is moved closer to the real distribution; d) this process is repeated until the discriminator is no longer able to distinguish between the distributions.

$\boldsymbol{x}$ and the fake data $G(\boldsymbol{z})$. G is trained simultaneously to minimize $log(1 - D(G(\boldsymbol{z})))$, so that the discriminator is fooled thinking the fake data $G(\boldsymbol{z})$ came from the real data $\boldsymbol{x}$. The complete loss function for a classical GAN is given by following value function:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})} [log(D(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})} [log(1 - D(G(\boldsymbol{z})))] \tag{3.1}$$

which is simply the binary cross entropy for the two classes: real and fake. Specifically, the discriminator is trained to predict the correct class labels, which is 1 for real samples and 0 for fake samples. The generator, on the other hand, is trained to minimize the discriminator's loss but with the labels flipped: 1 for fake samples and 0 for real samples. As training progresses, if G and D have enough capacity, training may converge to the point where G can produce samples that are indistinguishable from the real data $\boldsymbol{x}$. Figure 3.4 shows a visualization of the GAN learning process.

During training, the generator learns how to map a noise $\boldsymbol{z}$ to the data space generating a sample that resembles those from the training data. The $\boldsymbol{z}$ space (also known as the latent space), is a continuous space which allows us to perform arithmetic operations and, surprisingly, the result of such operations relates to the results in the data space. Figure 3.5, for example, shows a linear interpolation in latent space for a GAN trained in the MNIST dataset [34].

Although GANs have become the state-of-the-art among generative models, there are some drawbacks. GANs are models that tend to be difficult to train. Several issues may happen during

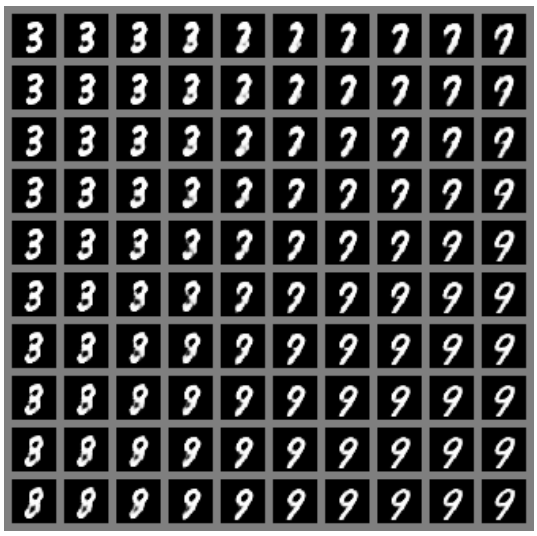


Figure 3.5 – Example of linear interpolation in latent space between the four corners. Figure adapted from Goodfellow et al. [18].

training that may prevent the model from converging. First, the GAN game is very sensitive, if the power between the generator and discriminator is not well balanced, one may win easily against the other, resulting in very poor learning. More importantly, GANs suffer from a phenomenon called mode collapse. Mode collapse is the scenario where the generator may end up learning just one mode of the data, because it is more likely to fool the discriminator. Mode collapse is the issue that prevented GANs from having success in datasets that contain many different concepts (modes) such as CIFAR10 [32] and ImageNet [11].

Mode collapse and training stability have been the most pursued issues in field of GANs in the academia. Since the first GAN was proposed, several improvements were made that makes training more stable and mitigate, at least partially, mode collapse. Following, we mention three important methods that presented a leap of improvement over theirs predecessors:

Wasserstein GAN

The Wasserstein GAN (or simply WGAN) [1] brought a new methodology for training GANs that showed several benefits. First, WGAN introduces a new adversarial loss function that behaves better than the traditional loss introduced by Goodfellow et al. [18]. The traditional GAN loss, which is simply the binary cross entropy, may saturate at some points, and consequently, produce poor gradients for both discriminator and generator. WGAN loss is based on the Earth-Mover distance (also known as Wasserstein-1), this loss function present better gradients. In fact, in a WGAN the loss function is better interpreted as game of cooperation rather than competition. This is why in a WGAN the discriminator is called *critic* as it is not trained to discriminate. The WGAN loss is given by:

$$\mathcal{L}_D = \mathbb{E}[D(\boldsymbol{x})] - \mathbb{E}[D(G(\boldsymbol{z}))] \tag{3.2}$$

$$\mathcal{L}_G = \mathbb{E}[D(G(\boldsymbol{z}))] \tag{3.3}$$

It is important to note that in this case the discriminator output is not a class probability as in a traditional GAN. In a WGAN the loss is computed over the discriminator *logits* (raw output). Also, it is shown that the optimal WGAN discriminator for a fixed generator lives under a Lipschitz continuity constraint. To enforce a Lipschitz constant constraint in the discriminator, weight clipping is performed so that its weights fall under a compact space. The authors acknowledge that this is not a very good solution to enforce a Lipschitz constraint, but it works in practice.

Improved Wasserstein GAN

The Improved Wasserstein GAN (also known as WGAN-GP, where GP stands for *gradient penalty*) [20] is a natural improvement to the original WGAN. As the WGAN authors pointed out, weight clipping is not a very good solution to enforcing a Lipschitz continuity constraint. In a WGAN-GP, the Lipschitz constraint in the discriminator is achieved by adding a gradient penalty term in the loss function rather than imposing restrictions directly to weights. The loss function in WGAN-GP is given by:

$$\mathcal{L}_D = \underbrace{\mathbb{E}[D(G(z))] - \mathbb{E}[D(x)]}_{\text{WGAN loss}} + \underbrace{\lambda_{GP}\mathbb{E}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]}_{\text{Gradient penalty term}} \tag{3.4}$$

$$\mathcal{L}_G = -\mathbb{E}[D(G(z))] \tag{3.5}$$

where the only difference to the WGAN loss is the gradient penalty term. The gradient penalty enforces a Lipschitz constraint by penalizing the discriminator gradient with respect to an input $\hat{x}$. The gradient penalty input $\hat{x}$ is given by a point sampled along straight lines between real and generated data:

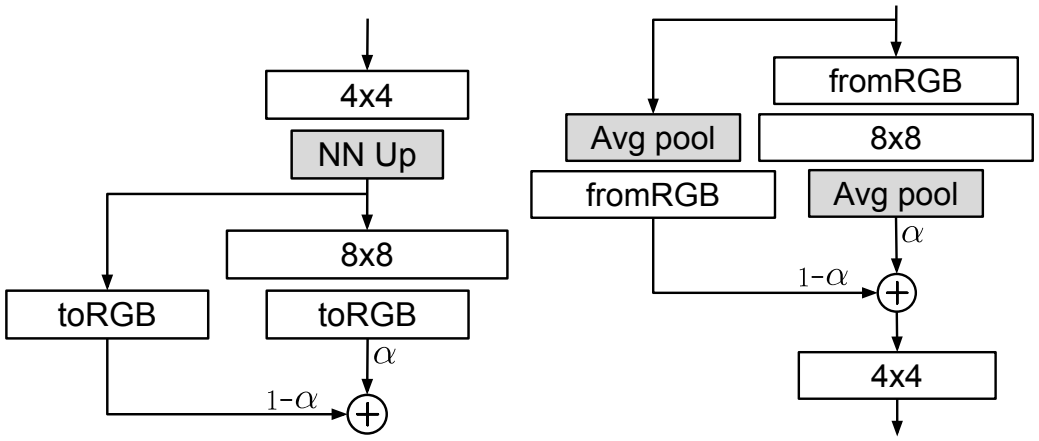
$$\hat{x} = \epsilon x + (1 - \epsilon)G(z) \tag{3.6}$$

where ϵ is sampled from a uniform distribution $U[0, 1]$. Finally, λ_{GP} control the weight put on the gradient penalty term during gradient descent.

Progressive Growing of GANs

Issues like non-convergence and mode collapse present in GAN training become even more evident when training models at high-resolutions. Most previous works [51, 1, 20, 17] were able to reach resolutions up to 128×128 pixels. Recently, a new methodology for GAN training introduced by Karras et. al. [29] improved on these issues, allowing training of GANs of resolutions up to 1024×1024 pixels. The key idea is to progressively grow the generator and discriminator as training progresses. Training starts at a low resolution as 4×4 pixels. As training progresses, new layers are

added on both discriminator and generator while all previous layers remain trainable. More layers are added until the target resolution for the model is reached. In a certain way, progressive growing, resembles layer-wise training of autoencoders [2].



(a) Fading in a new layer in the generator. (b) Fading in a new layer in the discriminator.

Figure 3.6 – Growing the progressive GAN networks.

Specifically, in progressive GAN, training alternates between two phases: *fade in* of new layers and *stabilization* of added layers. In order to preserve stability and not shock the networks, new layers are *faded in* smoothly. During a transition to a higher resolution, the networks operate at both the lower and higher resolution at the same time using a skip connection between layers. Fig. 3.6a and Fig. 3.6b show a transition from a 4×4 resolution to 8×8 resolution for the generator and discriminator, respectively. The weight α of the skip connection increases linearly until the transition is complete. After a transition is complete, the skip connection is discarded and the *stabilization* phase begins, where the networks are trained for more iterations before new layers could be added.

In the example of Fig. 3.6a, the *toRGB* layer projects the generator's output to 3 channels to form the RGB output image and *NN Up* is a layer that performs upsampling using nearest neighbor interpolation. In Fig. 3.6b, *fromRGB* is a layer that projects the RGB input image to the same number of channels as the next current convolutional layer and *Avg pool* is downsampling performed by average pooling. Both *toRGB* and *fromRGB* are usually composed by convolutions with filters of size 1×1.

3.2.2 Conditional Generative Adversarial Networks

GANs are, by default, unconditioned generative models. It means that it is not possible to impose any control over the generated samples. In some cases, however, it may be desirable to have control over the data generated by the GAN generator (e.g. generate a face with a given face expression). This can be achieved by restricting some dimensions of the latent space to hold meaningful information. For a GAN trained in a dataset of faces, for example, some dimension of latent space could control hair color, while other could control face expression, and so on. In order

to have this control, we need to learn a *disentangled* latent representation. Formally, we would like to provide a conditioning factor y for the generator alongside with the regular noise z and generate a sample $G(z|y)$ that correlates with the conditioning factor y. Next, we show the three most well-known approaches to GAN conditioning: Concat Conditioning, Auxiliary Classifier Conditioning and Projection conditioning.

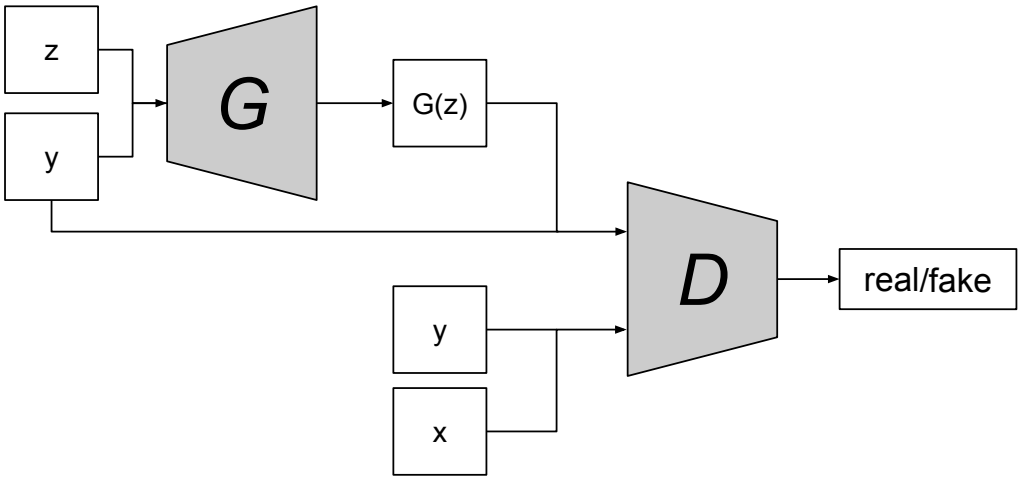


Figure 3.7 – CGAN Architecture.

Concat Conditioning

The Concat Conditioning is the first form of GAN conditioning. It was proposed by Mirza et al. [41]. Sometimes this approach is referred simply as Conditional GAN or CGAN. In a CGAN, no additional loss term is required. The only difference to regular GAN training is that both generator and discriminator are provided side information during training. In Fig. 3.7, it is shown the CGAN scheme. The generator is fed with a regular random noise z concatenated to a conditioning factor y and outputs a fake sample. The discriminator is trained to distinguish between the fake sample concatenated with the conditioning factor y and the real sample concatenated with the same conditioning factor y. Some variants [55] concatenate the y factor to an intermediate layer of the discriminator instead of its input. What happens in practice is that the discriminator has more information to work with and, in order to the generator fool the discriminator, it has not only to generate a realistic sample, but also generate samples that correlate with the conditioning factor y.

Auxiliary Classifier

The Auxiliary Classifier GAN (or AC-GAN) [46] approach differs from the CGAN approach in terms of how the side information provided to the discriminator. In a AC-GAN, the generator is the same as in CGANs. The discriminator, however, does not receive any side information as input during training. The conditioning behavior is achieved by the use of an auxiliary classifier. The scheme of an AC-GAN is shown in Figure 3.8. Specifically, in an AC-GAN the discriminator does

not only predicts if its input is real/fake but it also predicts the conditioning factor y. Therefore, the loss term for an AC-GAN is given by the adversarial loss plus the auxiliary classifier loss, which is given by:

$$\mathcal{L}_D = \mathcal{L}_{real} + \mathcal{L}_{fake} + \mathcal{L}_C \tag{3.7}$$

$$\mathcal{L}_G = -\mathcal{L}_{fake} + \mathcal{L}_C \tag{3.8}$$

where $\mathcal{L}_D$ and and $\mathcal{L}_G$ are the discriminator and the generator loss, respectively. $\mathcal{L}_C$ is the auxiliary classifier loss, which in a classification problem may be a binary cross entropy or a multiclass cross entropy.

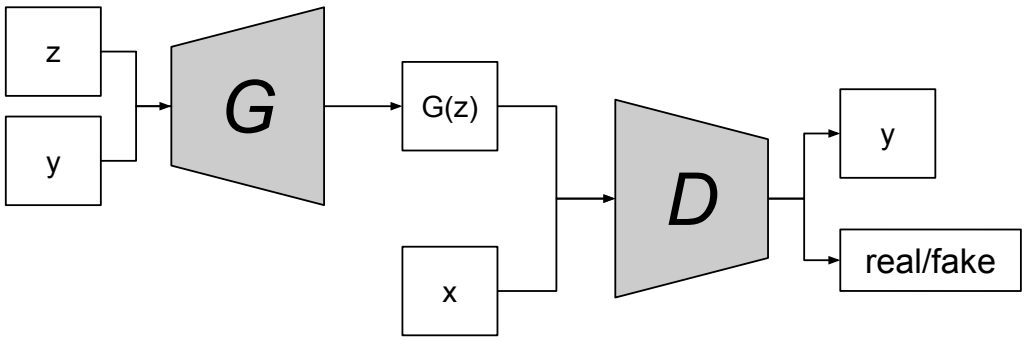


Figure 3.8 – AC-GAN Architecture.

Projection Conditioning

Miyato et al. [43] introduced a new form of inserting conditioning information on both generator and discriminator. In the generator, side information is provided through a conditional batch normalization [13, 10]. The learnable parameters of batch normalization are chosen according to the y factor fed to the generator. In other words, each class present in the set of discrete classes will have its own batch normalization parameters. Since the parameters of batch normalization are continuous, this setting also allows for interpolations between classes. In the discriminator, side information is provided by projecting the y using a dot product between y and some intermediate layer of the discriminator and then adding this product to the final discriminator output. The architecture of a Projection GAN is shown in Figure 3.9. Naturally, y is usually holds some discrete data that represents a class, therefore, in this case, for projection to work, y needs to be represented by a dense vector (*embedding*).

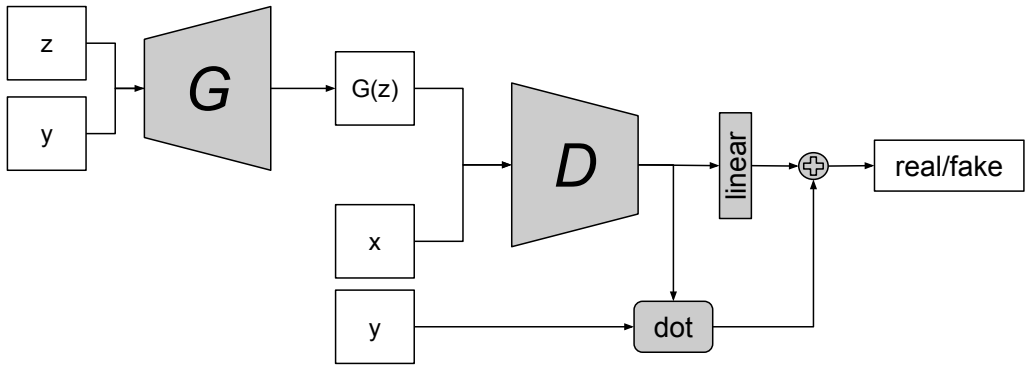


Figure 3.9 – Projection GAN Architecture.

3.2.3 Evaluation of Generative Models for Images

An important aspect related to generative models, especially to GANs, is evaluation. Evaluating generating models can be very complicated, mostly because there is not a direct way to quantitatively assess visual quality of generated samples. In fact, models that have good likelihood can generate bad samples while models with poor likelihood can generate good samples [17]. When evaluating a generative model there are two important aspects to be considered: *visual quality* and *variety* of generated samples. Some models might choose to pay less attention to one in favor of the other. Although, as of today, there is no proof that those are related. Currently there are three main approaches used to evaluate generative models:

Visual Inspection

The most straightforward form of evaluation is to have human annotators to judge the visual quality of generated samples. This approach was used in [58], where the authors employed the Amazon Mechanical Turk (MTurk) to ask users which images were real and which were generated. Visual inspection was the main form of evaluation until quantitative metrics were proposed. Currently, visual inspection is still used when the quantitative metrics are not well suited for the problem at hand.

Inception Score

In order to have a common quantitative measure to generative models, Salimans et al. [58] introduced the *inception score* (IS), which is a measure that seems to correlate well with human annotators and balances well between sample quality and variety. The measure is obtained by computing class probabilities $p(y|x)$ for some generated sample x using an inception model[1] [61] trained on the ImageNet dataset [11]. If the generated sample is realistic, we expect the entropy of class probabilities to be low. Also, we expect the entropy over class scores between samples to

[1]The model used for computing the inception score is available at http://download.tensorflow.org/models/image/imagenet/inception-2015-12-05.tgz.

be high, indicating that the model is producing a variety of distinct samples. The inception score is given by the following formula:

$$\exp(\mathbb{E}_x D_{KL}(p(y|x)||p(y))) \tag{3.9}$$

The inception score is usually used to measure performance of generative models trained on the ImageNet [11] and CIFAR10 datasets [32].

Fréchet Inception Distance (FID)

As pointed out by Heusel et al. [24], the inception score may have some drawbacks, it does not consider statistics of the real and statistics of the generated data. To this end, they propose an improved version of the inception score, which is called "Fréchet Inception Distance". To compute FID, the same inception model as in the inception score is used, but instead of computing class probabilities, the image features from the layer that precedes class predictions is computed. It is assumed that the image features follow a multidimensional Gaussian distribution. The distance between two Gaussian distributions is calculated by the Fréchet Distance [15], which is also known as Wasserstein-2 [68] distance. FID is computed according to the following formula:

$$d^2((\mu, \Sigma), (\mu_w, \Sigma_w)) = ||\mu - \mu_w||_2^2 + Tr(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2}) \tag{3.10}$$

where μ andΣ are the mean and covariance matrix of features of real images, respectively, μ_w and Σ_w are the mean and covariance matrix of features of generated images, respectively. Currently, FID alongside with Inception Score, have been the most used metrics to evaluate generative models.

In this Chapter, so far, we showed the basic concepts regarding machine learning, deep neural networks and generative adversarial networks, as well as some recent developments in GANs. For more details, however, we recommend the reader to go over the specific references, since this is a very broad topic that cannot be covered completely here.

3.3 Text-conditioned Generative Models for Images

Text-to-image synthesis is the task of generating images from text descriptions. Image generation, by itself, is a challenging task. When we combine image generation and text, we bring complexity to a new level: we need to combine data from two different modalities. In the most common setting, text-to-image methods are based on generative models that learn a text-conditioned distribution over images. Given a text description and some random variable, the algorithm produces a random image (controlled by the random variable) that correlates with the information present in the text. Text-to-image synthesis is a very recent research area. It has a great potential to aid several real-world applications, the list goes from automated content generation to assisted drawing.

Also, it points to the direction on how humans may interact with creative systems in the future using simply natural language. Moreover, it can help develop an understanding between the relationship of statistical distributions of different modalities (*e.g.* images and text).

Consider the example shown in Figure 3.10. It illustrates two important aspects of text-to-image models:

- Relationship between images and text: images generated should resemble all details described in the text. These details usually include shapes, colors, and specific location. Note how in the example the algorithm respects the details *"red crown adn throat"*, *"block eye ring"* and *"white ad pink belly"*. It is important to note also, the robustness to typos and grammar errors, which is a very desirable element in text-to-image models.
- Variability: How big is the set of images that can be considered "correct" for a single text description? The answer to this questions is unknown, but we know for sure that this set is far from being small. To address this behavior, text-to-image models need to generate images with a *high variability*. That is, for a single text description, the model should be able to produce several novel images while all images strongly present the details present in the text. Again, this can be seen in the examples show in Figure 3.10, where all images are different despite representing well the information in the text.

The first method that tried to address text-to-image synthesis was proposed by Reed *et al.* [55] in 2016. This work extended the first Convolutional GAN [51] architecture to be conditioned on text descriptions. This method used an RNN to encode sentences to feature vectors so that this vectors could be used as a conditioning factor to train a conditional version of a Convolutional GAN. This was the first work that embraced the text-to-image pipeline in an end-to-end fashion, from characters to pixels. Despite generating low resolution and low quality images, Reed's work opened the community to the possibility of creating such models.

Text-to-image generation models can be defined as a the special case of generative conditional models conditioned specifically on text descriptions. A typical framework of text-to-image generation is composed by two main components: a text encoder and a conditional generative model for images, as depicted in Figure 3.11. The text encoder's role is to convert the input text descriptions into a numerical representation that captures the semantic meaning of the text. This is usually achieved through natural language processing techniques, often employing models such as RNNs [26, 6] or Transformers [63, 12], which can understand context and nuances in language. The resultant vector representation, often called *embedding*, serves as the conditioning input for the image-generative model.

To train a text-to-image generative model, it is necessary first to have a trained text-encoder capable of mapping text descriptions to a vector representation. Text encoders are trained to align the text-vector representation to the image-vector representation of an image encoder. The training of image-text alignment language models encompasses a process known as multimodal learning. To train a language model for image-text alignment, typically two encoders are used: one

Text description: *"this colorful bird has a red crown adn throat with a black eye ring, and a white ad pink belly" (sic).*

Figure 3.10 – Example of synthetically generated images for a given text description [59]. Notice that this description belongs to the test set of the CUB Dataset [64] and it has some typos.

for images (image encoder) and one for text (text encoder). Both encoders aim to represent their respective inputs in a shared vector space where the corresponding image and text features are closely aligned.

The text encoder is trained first with the goal of learning how to encode text descriptions accurately. This is usually achieved by employing a pre-trained language model, such as BERT or GPT, and then fine-tuning it on a dataset comprised of text-captioned images. The language model learns to predict the probability of a word given its context, generating embeddings that effectively capture the nuances of the language. Subsequently, the image encoder is trained, often using a deep convolutional neural network (CNN) or a vision transformer (ViT), to transform visual inputs into feature vectors. The image encoder is trained to produce representations that can be matched or aligned with the representations produced by the text encoder.

To ensure that these encoders create compatible embeddings, a typical approach involves the use of contrastive loss functions or other alignment objectives during training. For example, the Contrastive Language-Image Pre-training (CLIP) approach uses a contrastive loss that encourages the distance between the correct pairings of text and image embeddings to be smaller than the distance between mismatched pairs. This reinforces the model's ability to cluster semantically similar representations of images and text closer together in the embedding space while pushing dissimilar ones apart. Through this training process, the representations of both images and text

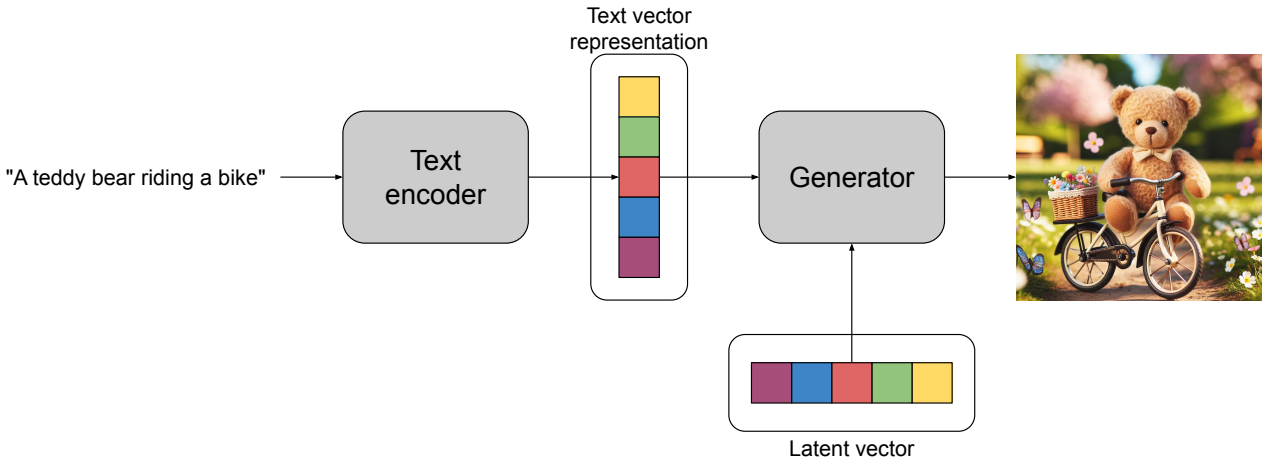


Figure 3.11 – General text-to-image generation framework.

become closely aligned in the same high-dimensional space. The result is a powerful multimodal model capable of understanding the relationship between images and text, which can be applied to a variety of tasks such as image captioning, text-to-image generation, or cross-modal retrieval.

The conditional generative model for images, on the other hand, uses the encoded text representations to generate images that correspond to the text description. Generative Adversarial Networks (GANs) [18] or Variational Autoencoders (VAEs) [31] are widely used in this role. In the GAN framework, the generator network attempts to create images that are indistinguishable from real images, while a discriminator network tries to classify images as real or synthetic. The conditioning happens when the generator takes both random noise and the text embedding as input, ensuring that the generated images are aligned with the semantic content of the text. Through iterative training, the generator learns to produce more accurate and high-quality images that match the input text descriptions. VAEs can be similarly conditioned by text embeddings to produce desired images, but they approach the problem from a probabilistic perspective, aiming to learn a latent space that encodes variations in the data in a structured manner.

3.4 Building Blocks

The general framework of text-to-image models is usually composed by a text-encoder and a variant of a conditional GAN. The diagram in Figure 3.12 shows an example of a text-to-image approach similar to the one presented in [55]. Next, we detail the inner workings of each component.

3.4.1 Text Encoder

There are several ways to encode text to a vector representation. However, most of them follow a similar approach. The idea is to have an image encoder and a text encoder that maps both images and text to the same semantic space, so that image features will be close to their descriptions

in vector space. The image encoder is a vision model, like ConvNet and the text encoder is sequence model, like a type of RNN, such as an LSTM [26].

An important aspect in text encoding is the level on which text is encoded. There are two levels:

- Sentence-level: in this level, an entire text description is encoded to a single feature vector. This is accomplished by using the last hidden state of the RNN, which encapsulates the meaning of the entire sentence.
- Token-level: in this level, each word – or sub-word, depending on the tokenization – is represented by its own feature vector. This approach presents a more fine-grained representation of the sentence. However, it imposes its own challenges. Since a sentence can be composed by an arbitrary number of tokens, their respective encoding will be composed by an arbitrary number of feature vectors.

Recent works have achieved better performance by incorporating word-level features in the framework. This is done by adding Attention modules [70] to the neural architecture. The first work that introduced this idea was AttnGAN [71]. It is important to note, however, that few studies were performed specific on text encoding approaches. Most ideas were just borrowed from other research areas, like multimodal information retrieval, for example.

3.4.2 Conditional GAN

Conditional GANs for text-to-image synthesis follow the same conditioning methods presented in the Subsection 3.2.2. There is one big difference, however. Most research on Conditional GANs used a different type of conditioning information. Those work employed a discrete set of labels (*i.e.* class labels) as condition. This way, it is possible to generate samples for a specific class (which is commonly done for ImageNet [11] and Cifar10 [32]). In the case of text-to-image, conditioning information is completely different. As shown in the previous Subsection, text descriptions are mapped to a *continuous vector space*. This presents impacts on hyper-parameters and overall training behavior.

3.4.3 Datasets

There are three widely used datasets for training and evaluating these models, we present each of them below.

Oxford-102 [44]: The Oxford-102 dataset is composed of 8,189 images of flowers of 102 categories. The dataset is split in 7034 images for training and 1154 images for testing. Each image

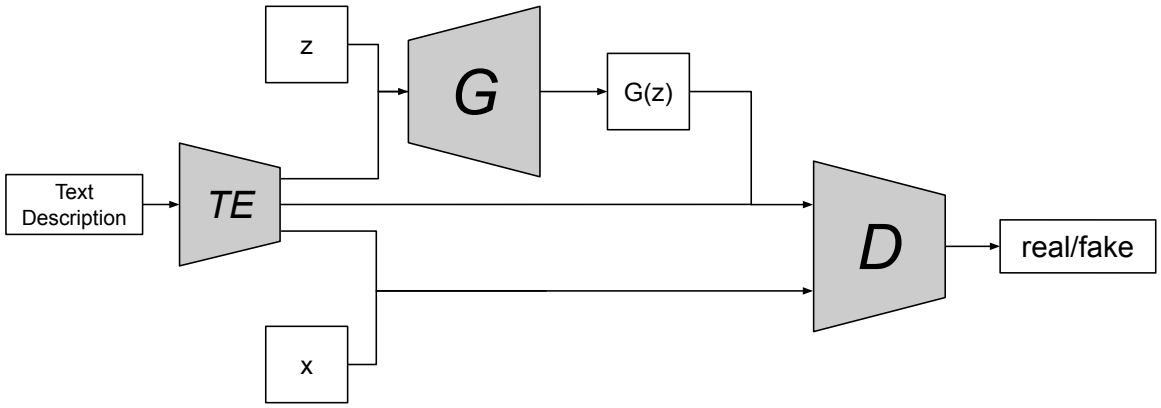


Figure 3.12 – General framework for text-to-image synthesis.

(a)

1. this flower has a tiny pistil and large leaves, that are colored with a mix of pink and white.
2. the petals of the flower are a mixture of pink and white and have a center made of green berries.
3. a flower with big red petals and dark green anther filaments.
4. this flower is white and pink in color, with petals that are multicolored.
5. the deep mauve petals are outlined with a light pink hue.
6. this flower has pink petals as well as a green stamen.
7. the petals are red with brown edges and have a wrinkled texture.
8. green stigma with pink fading into white petals
9. this flower has petals that are red and has green edges
10. this particular flower has petals that are red and white and sharp

(b)

Figure 3.13 – Example of image and its respective text descriptions taken from the Oxford-102 dataset [44]

.

contains 10 text descriptions.An example of image and its text descriptions for Oxford is shown in Figure 3.13.

Caltech-UCSD Birds (CUB) [64]: The CUB dataset is composed of 11,788 images of birds distributed among 200 class categories. The dataset is split in 8,855 images of 150 categories for training and 2,933 images of 50 categories for testing. Each image contains 10 text descriptions. An example of image and its text descriptions for CUB is shown in Figure 3.14.

MS Comon Objects in Context (COCO) [37]: The COCO Dataset is composed of $80k$ images for training and $40k$ images for testing. Unlike the CUB Dataset, COCO images are composed of scenes containing multiple objects, which makes text-to-image generation particularly challenging. In the COCO Dataset, each image contains 5 text descriptions. An example of image and its text descriptions for CUB is shown in Figure 3.15.

(a)

1. light tan colored bird with a white head and an orange beak.
2. the bird has a very thick, curved, and beige beak
3. this bird has a long neck that is grainy and a pastel orange/blue narrow beak that droops down at the tip
4. this bird is light brown, has a long hooked bill, and looks dumb.
5. this large white bird has a large curved bill and a brown eye
6. this bird is white with grey and has a long, pointy beak.
7. this bird is white with grey and has a long, pointy beak.
8. the crown of the bird is white, with light brown tones throughout.
9. the crown of the bird has distinctive tones of white and brown throughout.
10. this bird has a long neck and an orange bill

(b)

Figure 3.14 – Example of image and its respective text descriptions taken from the CUB dataset [64].

(a)

1. yellow school bus drives through the wet street
2. the school bus drives down the city street.
3. a big school bus drives down the street
4. a yellow school bus parked on the side of a road.
5. a yellow school bus is traveling down a road near a building.

(b)

Figure 3.15 – Example of image and its respective text descriptions taken from the COCO dataset [37].

3.5 Evaluation of Text-to-image Models

Evaluation of text-to-image models follow the same strategies as evaluating traditional GANs. Quantitative analysis are performed using the most widely used metrics: the Inception Score and Frechét Inception Distance. Details about how these metrics are computed are presented in Subsection 3.2.3. Both of them attempt to measure quality and variety of generated images. However, they do not measure if generated images relate well with its text descriptions.

To address the evaluation of the relationship of generated images and its text descriptions, previous methods introduced different strategies. The most used is the multimodal retrieval approach. This approach is quite simple: given a generated image, a retrieval algorithm attempts to retrieve the descriptions for that image, if the algorithm retrieves the correct description, then the result for that image is considered "correct", it is considered wrong otherwise. The final score is given by metrics commonly used for information retrieval, such as R-precision. While these evaluation protocols make more sense for evaluating text-to-image models, they are not still not well standardized and widely adopted.

4. EFFICIENT NEURAL ARCHITECTURE FOR TEXT-TO-IMAGE SYNTHESIS

4.1 Introduction

While investigating previous approaches to text-to-image synthesis, we developed a novel method for synthesizing single object images from text description. It was intended to be simpler and present superior performance than previous methods. It was published in the International Joint Conference on Neural Networks (IJCNN), its title is: Efficient Neural Architecture for Text-to-Image Synthesis [59].

First approaches to text-to-image synthesis [55, 55, 76, 75, 78] have simply extended GANs to be conditioned to sentence vectors. Naturally, results were not optimal. Most recent methods [71, 49, 73, 81] have proposed different strategies to circumvent the complex relationship between image and text. Most of those works, however, follow a similar pattern when it comes to neural architectures. Due to previously mentioned difficulties, plus the inherent difficulty of training GANs at high resolutions, most recent works have adopted a multi-stage training strategy. In a multi-stage setting, training is performed first at low resolutions (*i.e.* 32×32 and 64×64 pixels) and then refined to higher resolutions (128×128 and 256×256 pixels). Usually, multi-stage training is implemented using several generators and several discriminators, which makes training complex and slow. This architectural choice has been followed by most previous work, which have been adding small improvements, such as word-level features through Attention Mechanisms [71], Memory Networks [81], Siamese Networks [49] and a Mirror strategy [49].

In this work, we shift the architectural paradigm currently used in text-to-image methods and show that an effective neural architecture can achieve state-of-the-art performance using a single stage training directly at the target resolution. By doing so, we not only introduce a simpler method for text-to-image synthesis but also point a new direction in text-to-image research, which has not experimented with novel neural architecture recently.

Specifically, we introduce an adversarial training-based architecture that leverages full capacity of modern deep convolutional networks, alongside to an improved sentence embedding approach for generating photorealistic text-conditioned images. Both discriminator and generator networks draw inspiration from [4], though we provide important improvements on that architecture, allowing for the use of sentence embeddings rather than class labels as conditioning vectors. Results show that our models outperform multi-stage state-of-the-art methods without heavy hyperparameter optimization in two widely used benchmarks, namely CUB [64] and Oxford-102 [44] datasets, in terms of both Inception Score [58] and Fréchet Inception Distance [24]. Figure 4.1 shows samples generated by our method. Moreover, we provide an extensive set of experiments, in which we explore key components and abilities of our models.

Formally, in this work we make the following contributions:

Figure 4.1 – Images generated by our method.

- We introduce a novel sentence interpolation strategy that allows the generator to learn a smooth conditional space, and also work as a data augmentation procedure.
- We show how the use of a modern residual neural architecture enables single-stage training at the target image size, and generates state-of-the-art text-to-image models.
- We perform an extensive analysis of the properties of text-to-image models, both in quantitative and qualitative fashion.
- We demonstrate that our models enable image editing using natural language via arithmetic operations in the conditional space, being able to modify aspects of the image while keeping its overall structure.

4.2 Method

In this section we present in detail the proposed approach. Text-to-image synthesis methods have followed a similar design pattern regarding neural architectures: they make use of multi-stage training using several networks. This choice, however, increases training complexity and computational costs required to train such models. Our approach departs from this design altogether. We present evidence that the use of an adequate neural architecture plus a simple sentence interpolation strategy can produce state-of-the-art results. In addition, our method performs a single-stage training with a single generator and a single discriminator. Next, we detail every component of our proposed method: the text encoder, the sentence interpolation strategy and the neural architecture.

4.2.1 Text Encoder

We encode text descriptions into a vector representations by using a pre-trained Deep Attentional Multimodal Similarity Model (DAMSM) [71]. The DAMSM module, similarly to [14, 69, 36], learns image and text encoding functions, namely $\varphi(I)$ and $\phi(S)$, that map images I and textual descriptions S into the same semantic multimodal space. Such a space is trained so that correlated image-caption pairs lie close to each other, while non-correlated pairs must present larger distance than the correlated ones. By optimizing that space, the learned text representation is forced to closely resemble the content from images, and therefore can be as a condition vector $\boldsymbol{s} \in \mathbb{R}^{256}$ in our architecture.

Original image captions S are tokenized, and each token is represented by a specific vector $\mathbb{R}^{300}$. Those vectors feed a Bidirectional GRU network, which provides per-token hidden representations, as well as a final global vector. Hidden representations are used for learning fine-grained correlations with the spatial information of the images, while the global vector contains holistic high-level information of the caption. In this study, we use the global vector alone as textual condition vector $\boldsymbol{s}$, hence $\phi(S) = \boldsymbol{s}$.

4.2.2 Sentence Interpolation

In this section we detail a novel strategy for improving the smoothness of the conditional space, which we hereby call Sentence Interpolation (SI). This technique consists in using all the available captions for computing the general sentence embedding regarding an image during training. By doing so, we make the textual representation vector to be continuous in the projected space, rather than being discrete points in the manifold, as a traditional approach would generate.

Formally, let I_i be the i^{th} image from the training dataset, and $S_{ij} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_n\}$ be the set of n correlated sentence embeddings that describe that particular image. We sample an n-sized vector of weights $\mathbf{m} \sim \mathcal{U}(0, 1)$, and further normalize it with a softmax function. Those normalized values are used to weight each one of the sentence vectors, so their sum consists in an interpolated representation of the original sentences. Therefore, the vector $\dot{\boldsymbol{s}}$ that represents the interpolated textual embedding of a given image is calculated as follows:

$$\dot{\boldsymbol{s}} = \sum_{j=1}^{n} \left[S_j \times \left(\frac{e^{\mathbf{m}}}{\sum_{k=1}^{n} e^{\mathbf{m}_k}} \right)_j \right] \tag{4.1}$$

Such an approach makes a limited set of sentences to be represented by countless continuous points during the training process. The main implications of this technique are two-fold: (i) it makes the sentence embedding space to be more smooth; (ii) and also works as a data augmentation strategy, given that the same textual descriptions can assume different forms depending on the sampling of $\mathbf{m}$. In comparison to the Conditioning Augmentation (CA) module introduced by

StackGAN [76], the sentence interpolation has the advantage of being deterministic. This is due to the fact that it is not used during the test phase. CA, on the other hand, introduces randomness when encoding sentence vectors during training and testing.

4.2.3 Architecture

We follow the steps of Brock *et al.*[4], which introduced the state-of-the-art architecture for GANs, namely BigGAN-Deep. This architecture is based on residual blocks with bottleneck structure of He *et al.*[22], which makes deeper networks more computationally efficient and easier to train. Also, like SAGAN [74], BigGAN-Deep applies Spectral Normalization [42] and Non-local Blocks [66] in both generator and discriminator. Finally, BigGAN-Deep introduces conditioning information in the generator using Conditional Batch Normalization [13] and in the discriminator using the projection approach of Miyato *et al.*[43].

BigGAN-Deep [4] presented, at the time, a new state-of-the-art result in the ImageNet [11] dataset in the supervised setting. Therefore, it was designed to be conditioned on class labels. Since in this architecture class labels are represented by dense embeddings, we extended it to handle the sentence vector. Specifically, we replaced the trainable class embeddings by the fixed sentence vectors $\boldsymbol{s}$. In the discriminator, sentence vectors are linearly projected to be used in the projection conditioning. In the generator, sentence vectors are concatenated with the noise vector $\boldsymbol{z}$ and then linearly projected to form BatchNorm gains and biases, gains are one-centered while biases are zero-centered. By using the fixed sentence vectors, the generator and discriminator are forced to adapt to the conditional space learned by the DAMSM encoder, which yields interesting properties, such as the generator's ability to handle arithmetic operations in conditional space, which is presented in Section 4.5.

The BigGAN-Deep architecture was originally designed to be used in large scale training. Large scale training is done by using a big batch size (*e.g.* 2048) and training the models across several devices. In order to apply this architecture in a small scale, we need to make additional adaptations. First, we switch relu activation to leaky relu. This helps avoiding sparse gradients, which is helpful due to the second adaptation. Second, we reduce the number of parameters of both networks. We reduce the number of parameters in the generator and discriminator by reducing the channel multiplier ch to 96 instead of 128 in default BigGAN-Deep architecture. This reduction represents 43% less parameters in the discriminator and 36% less parameters in the generator. Finally, training is performed directly at the target resolution of 256×256 pixels. As far as we know, no previous text-to-image method was able to train directly at this resolution without relying on multiples generators and discriminators.

4.2.4 Objective Function

We adopt the so-called hinge adversarial loss. The hinge loss works similar to WGAN loss [1] but is more stable thanks to the margins introduced in the discriminator loss function. For the discriminator, the hinge loss is given by:

$$V_D(\hat{G}, D) = \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{s}\sim q_{\text{data}}} \left[\min\left(0, -1 + D(\boldsymbol{x}, \boldsymbol{s})\right)\right] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}},\boldsymbol{s}\sim q_{\text{data}}} \left[\min\left(0, -1 - D\left(\hat{G}(\boldsymbol{z}, \boldsymbol{s}), \boldsymbol{s}\right)\right)\right], \tag{4.2}$$

where $\boldsymbol{x}$ and $\boldsymbol{s}$ are real images and their corresponding sentence vectors, respectively. $\hat{G}(\boldsymbol{z}, \boldsymbol{s})$ is a fake image from the generator for a given random vector $\boldsymbol{z}$ and a sentence vector $\boldsymbol{s}$, respectively. Note that the hat in G means that, in this case, the generator's weights are not being updated.

Similarly, the loss function for the generator is given by:

$$V_G(G, \hat{D}) = - \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}},\boldsymbol{s}\sim q_{\text{data}}} \left[\hat{D}\left(G(\boldsymbol{z}, \boldsymbol{s}), \boldsymbol{s}\right)\right], \tag{4.3}$$

in this case, the hat in D means the discriminator's weights are not being updated.

4.3 Experiments

In this section we present our experimental setup. We conduct extensive experiments in the most used datasets for text-to-image generation. We also present an extensive quantitative and qualitative analysis of our findings.

4.3.1 Datasets

We have used two widely used datasets for training and evaluating our models, as follows.

Caltech-UCSD Birds (CUB) [64]: The CUB dataset is composed of 11,788 images of birds distributed among 200 class categories. The dataset is split in 8,855 images of 150 categories for training and 2,933 images of 50 categories for testing. Each image contains 10 text descriptions.

Oxford-102 [44]: The Oxford-102 dataset is composed of 8189 images of flowers of 102 categories. The dataset is split in 7034 images for training and 1154 images for testing. Each image contains 10 text descriptions.

4.3.2 Evaluation

In order to evaluate our method, we employ the two most widely used metrics to evaluate generative models: the Inception Score (IS) and the Fréchet Inception Distance (FID). The IS uses a pre-trained Inception Network [60] to compute class probabilities over generated samples. IS is both a measure of *objectness* and variety, therefore, the higher the score the better. In order to compute IS, and also be able to compare results, we use the same Inception Networks used to evaluate previous work. The networks are provided by StackGAN [76] and are finetuned for the CUB and Oxford-102 datasets.

A downside of the IS is that it does not consider the statistics present in the real data. A generative model that generates a few high quality examples for each class would have a very high IS score, despite its variety being low. To circumvent this issue, Heusel *et al.*[24] introduced the Fréchet Inception Distance (FID). FID considers the statistics present in the training data, so it possible to evaluate if the generative model learned a distribution that have similar statistics. Specifically, FID uses an Inception Network to compute activation features of both training set images and generated images. The Fréchet Distance is then computed over the features of real and fake images. FID gives a measure of how close the statistics of generated images are from those in the training set, hence, the lower the score the better.

4.3.3 Implementation Details

We use Adam optimizer [30] with a learning rate of 4×10^{-4} for D and 10^{-4} for G. We set $\beta_1 = 0$ and $\beta_2 = 0.999$ for both G and D. We train one D step per G step. We use synchronized implementation of BatchNorm, where statistics are aggregated across all devices. We keep an exponential moving average of the generator weights with a decay of 0.999 for sampling. Since BatchNorm statistics are not computed for averaged weights of the generator, we employ the "standing statistics" strategy of Brock *et al.*[4]. In other words, we first run 100 forward passes through G to update its BatchNorm statistics, making the generator invariant to batch sizes. Finally, we perform training using 3 GPUs with a batch size of 8 per GPU, making up for a batch of size 24. Most models take up to 3 days to train.

4.4 Comparison to state-of-the-art methods

In order to provide reassurance on the generative performance of our models, we compare their quantitative and qualitative results against current state-of-the-art methods [55, 76, 75, 9, 78]. Note that some of them have not reported FID results. Hence, we compare to the results publicly available.

Table 4.1 – Quantitative results for Efficient GAN.

Method	# Networks		Multi	IS↑		FID↓	
	D	G	Stage	CUB	Oxford-102	CUB	Oxford-102
GAN-INT-CLS [55]	1	1	No	2.88 ± 0.04	2.66 ± 0.03	-	-
GAWWN [56]	1	1	No	3.60 ± 0.07	-	-	-
StackGAN [76]	2	2	Yes	3.70 ± 0.04	3.20 ± 0.01	55.28	51.89
StackGAN++ [75]	3	3	Yes	4.04 ± 0.05	3.26 ± 0.01	15.30	48.68
TAC-GAN [9]	1	1	No	-	3.45 ± 0.05	-	-
HDGAN [78]	3	3	Yes	4.15 ± 0.05	3.45 ± 0.07	-	-
Ours	1	1	No	**4.23 ± 0.05**	**3.71± 0.06**	**11.17**	**16.47**

4.4.1 Quantitative Analysis

Table 4.1 depicts quantitative results, alongside to the number of discriminator and generator networks used in each work. It arguably shows that our approach is the preferred method, once it achieves top performance in all metrics while employing just a single discriminator and a single generator in the entire architecture. Notably, it outperforms all the baseline approaches by a margin across all datasets and metrics.

The largest improvement provided by our approach is on Oxford-102 dataset. It provides a relative improvement of $\approx 7\%$ IS and $\approx 300\%$ FID when compared to the strongest baseline with public results available. Clearly our approach also leads to a significantly better results on CUB dataset, allowing for a $\approx 24\%$ FID reduction.

4.4.2 Qualitative Analysis

Fig. 4.2 depicts qualitative results of models trained on CUB dataset. In that Fig., we compare our model to the baseline ones. One can observe that our model brings improvement on several aspects regarding the generated images. For instance, our images look more photorealistic, present better semantic correspondence of the generated images to the provided description, and also generate more fine-grained details in both foreground and background elements.

Results shown in Fig. 4.3 were generated using a model trained on Oxford-102 dataset. Once again, our model generates images with much richer detail level and photorealistic aesthetic. Such experiment supports our claims that our proposed single-stage architecture can be used for generating concepts across distinct datasets. It is worth noticing that despite Oxford-102 being a somewhat small dataset, our models were able to learn a proper distribution without suffering from mode collapse or additional training instabilities.

Figure 4.2 – Qualitative results in the CUB Dataset.

Figure 4.3 – Qualitative results in the Oxford-102 Dataset.

Figure 4.4 – Image generation based on condition space arithmetic of embedded textual descriptions.

4.5 Condition Space Arithmetic

In this section we explore the inherent capability of our approach to handle condition space arithmetic. This is a very interesting property and finds applications in many real world tasks, such as image manipulation via natural descriptions. This capability emerges from the fact that the employed sentence embedding vector $\boldsymbol{s}$ concatenated to the z vector lie in a smooth embedding space that present structural regularity. In that particular kind of space we can find semantic regularities regarding concepts learned by the model, i.e., they respect a semantic organization of concepts. We observed that the use of our novel sentence interpolation strategy during training is quite helpful to improve the learned condition space. It increases the model capacity of learning a smooth condition space, in which embedding regularities are more easily found.

Figure 4.4 showcases examples regarding regularities found in our trained models. For generating those images we hold z fixed, and embed captions into the multimodal space, which are used in simple vector operations, as follows. The uppermost example depicts an image generated by $G(z, \phi(\text{“}\textit{This is a red bird}\text{”}))$. We then subtract $\phi(\text{“}\textit{It is red}\text{”})$ from $\phi(\text{“}\textit{This is a red bird}\text{”})$, and

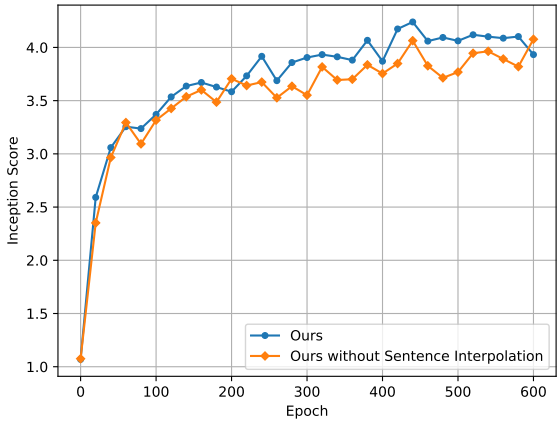


Figure 4.5 – Inception Score during training epochs for our model with and without Sentence Interpolation in the CUB dataset.

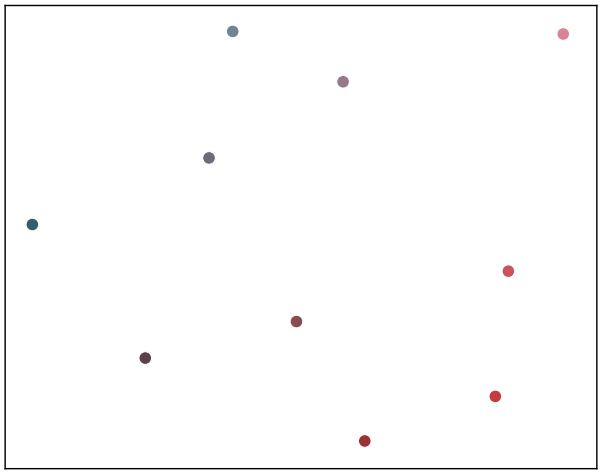
generate a novel image (in the center). One can see that such operation completely removed the red color from the generated bird. Finally, we add $\phi(\text{“}\textit{It is blue}\text{”})$ to the resulting embedding, and use it to generate the rightmost image. That image shows the same bird, though with its color changed from red to blue, using only simple vector-level operations.

Note that our models are able to edit images while preserving the main image structural content without even being explicitly trained to learn disentangled representations. Figure 4.4 also demonstrates that one can edit several aspects of the generated images, such as shape of the beak, and presence of colored crown.

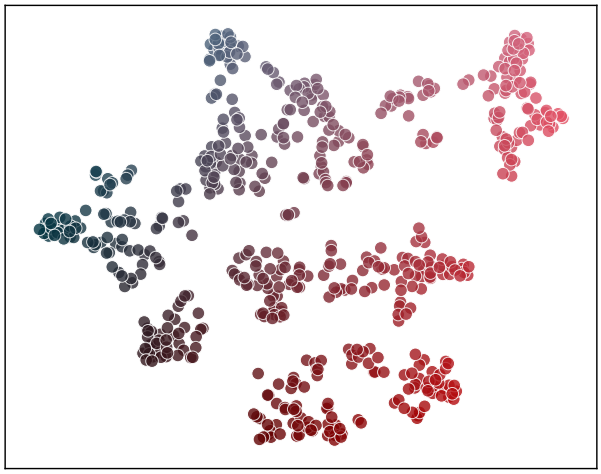
4.6 Impact of Sentence Interpolation

One of the contributions of this method is the introduction of a novel Sentence Interpolation procedure. In order to understand its effects, we have trained two models: (i) a default complete model that performs Sentence Interpolation; and (ii) a model with the same overall architecture, though without applying any interpolation between sentences. Fig. 4.5 shows per-epoch Inception Score values computed during the entirety of the training process. It arguably proves the importance of the proposed technique. During the early stages of training, results are indeed quite similar, the difference being more relevant after the 100^{th} epoch. Notably, after the 400^{th} epoch, IS results with Sentence Interpolation were consistently higher than 4.00, while the model without it surpassed that mark only twice throughout the training.

Effects of the SI approach also can be seen in Fig. 4.6. In this analysis, we plot ten sentence embeddings of a randomly chosen image during the entire training (*i.e.*, resulting in 600 embeddings). We plot the very same embeddings for the model trained with and without SI. We apply the t-SNE [38] technique on those embeddings so as to project $\mathbb{R}^{256}$ vectors onto a $\mathbb{R}^2$ space. Such a plot clearly shows that the proposed interpolation provides a much larger exploration of



(a) Sentence embeddings sampled without Sentence Interpolation.



(b) Sentence embeddings sampled with Sentence Interpolation.

Figure 4.6 – Manifold visualization of the sampled sentence embeddings during training. We visualize sentence embeddings by applying t-SNE [38] to project sentence embeddings from the original $\mathbb{R}^{256}$ space to a $\mathbb{R}^2$ space. We show 10 sentence embeddings of a randomly chosen image during the entire training (*i.e.*, resulting in 600 embeddings). In (a) is shown the regular sampling of a random sentence. In (b) is shown the sampling using the Sentence Interpolation.

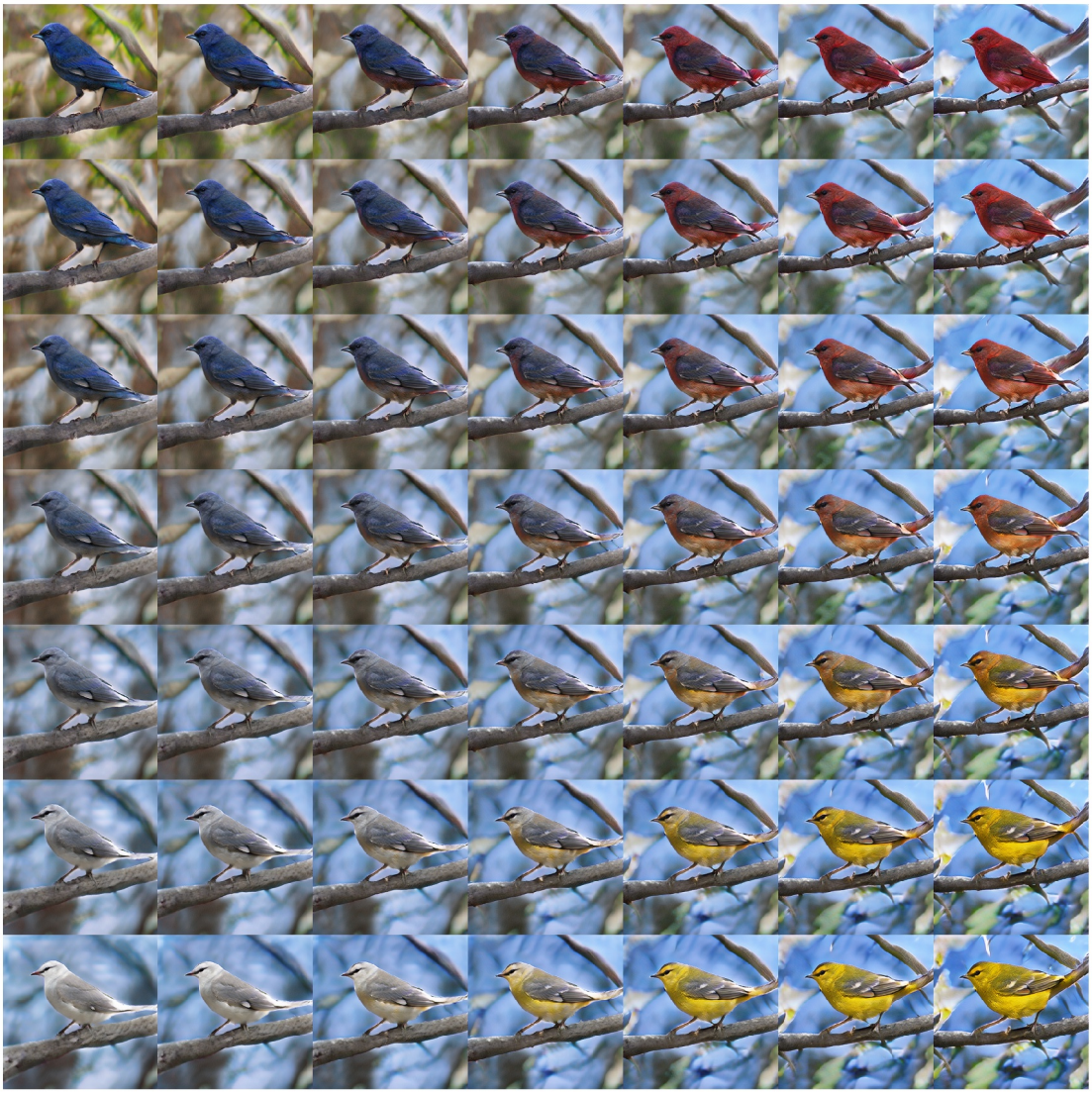


Figure 4.7 – Image generation with sentence embeddings linearly interpolated across all directions. There are four original embeddings, each one used to generate an image (those from the four corners), while all the remaining ones were generated using interpolated description embeddings. The upper-left position depicts an image generated with the description *"It is a blue bird"*, the bottom-left image was generated with *"It is a white bird"*, the upper-right image with *"It is a red bird"*, and the bottom-right image with *"It is a yellow bird"*.

the sentence embedding manifold, allowing for sampling continuous points from that space. That sampling region is obviously constrained by the ten points regarding the image descriptions chosen. We intend to further extend this technique for future work, so as to allow sampling points from outside of those boundaries, without loosing semantic context. When training without it, one can only sample fixed discrete points, which poses a considerable constraint on the information carried on the condition vector. This analysis corroborates with our hypothesis that SI works also as a data-augmentation scheme, providing better generation results for points present in a larger region of the manifold. Finally, Figure 4.7 presents the interpolation between captions, which shows the smoothness introduced by SI.

5. TEXT-TO-IMAGE GENERATION WITH TEXT AUXILIARY REGRESSOR GANS

5.1 Introduction

After introducing the work presented in Chapter 4, we noticed that the way text embeddings are introduced as condition for GAN training may not be optimal. The conditioning strategy used was originally intended for data of discrete nature (*e.g.* class labels). Therefore, we addressed this problem by introducing a new conditioning methodology specific for text-to-image synthesis.

In this Chapter we introduce a new approach for text-to-image synthesis that is not only efficient but also dramatically simpler. We propose a Text Auxiliary Regressor Generative Adversarial Network, namely TAR-GAN, that achieves state-of-the-art performance using a single generator and a single discriminator. By using a novel Auxiliary Regressor, that was designed specifically for text conditioning, TAR-GAN turns possible training text-to-image models directly at the target resolution. TAR-GAN is intended to close the performance gap between text-to-image generation and traditional class-conditional GANs [42, 74, 4].

Our experiments demonstrate that TAR-GAN favorably outperforms the previous state-of-the-art methods, despite being substantially simpler. We quantitatively evaluate the performance of TAR-GAN using the Inception Score (IS) [58] and the Fréchet Inception Distance (FID) [24]. Our method presents a 8% FID improvement in the CUB Dataset [64].

Finally, in this Chapter we present the following main contributions:

- We propose a novel method that presents state-of-the-art performance while being substantially simpler than previous approaches.
- We introduce an auxiliary regressor discriminator, that makes conditioning on text embedding more natural and consistent.
- We study how each conditioning approaches influences performance. By doing so, we bridge the performance gap between traditional class-conditional GANs and text-to-image generation.

5.2 Text Auxiliary Regressor GANs

Most previous work have followed a similar architectural designs when approaching text-to-image generation. In order to ease the learning of the complex relationship between image and text, most previous work [78, 71, 49, 81] apply a multi-stage training, *i.e.* training first to generate low resolution coarse images and then train to refine to higher resolution sharp images. This procedure is usually carried out using several networks, which makes training complex and computationally

expensive. In this work, we depart from this design altogether and rethink how to approach the complexity of text-to-image generation.

We propose a Text Auxiliary Regressor Generative Adversarial Network, that was designed to be a simple yet effective approach to text-to-image synthesis. TAR-GAN introduces a novel auxiliary regressor that makes GAN conditioning on text descriptions more accurate and natural. TAR-GAN also makes it possible to train text-to-image models directly at the target resolution, using a single generator and a single discriminator, making training procedure simpler and faster. The overall architecture of TAR-GAN is shown in Figure 5.1.

5.2.1 Text Encoder

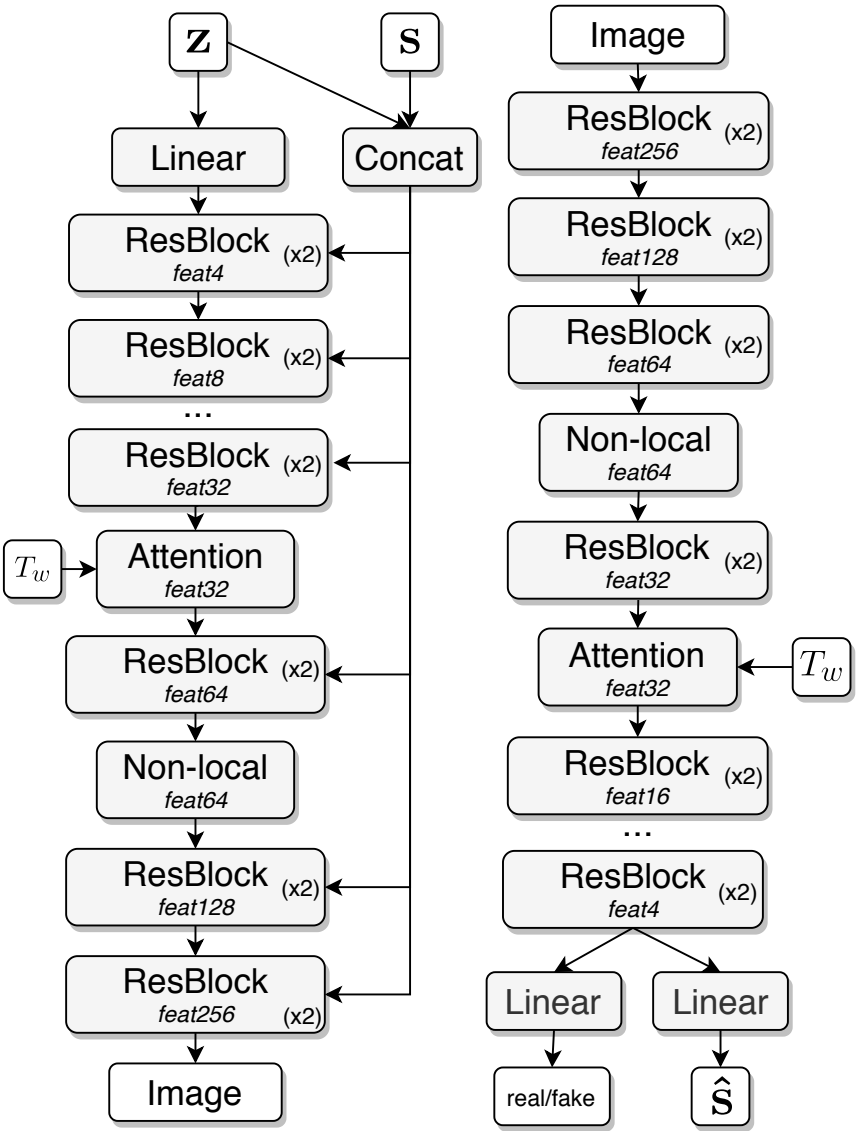
We encode text descriptions into a vector representations by using a pre-trained Deep Attentional Multimodal Similarity Model (DAMSM) [71]. The DAMSM module, similarly to approaches in [14, 69, 36], learns image and text encoding functions, namely $\varphi(I)$ and $\phi(S)$, that map images I and textual descriptions S into the same semantic multimodal space. Such a space is trained so that correlated image-caption pairs lie close to each other, while non-correlated pairs must present larger distance than the correlated ones. By optimizing that space, the learned text representation is forced to closely resemble the content from images, and therefore can be used as a conditioning vector $\boldsymbol{s} \in \mathbb{R}^{256}$ in our architecture.

Original image captions S are tokenized, and each token is represented by a specific vector $\mathbb{R}^{300}$. Those vectors feed a Bidirectional GRU network, which provides per-token hidden representations, as well as a final global vector. Hidden representations are used for learning fine-grained correlations with the spatial information of the images, while the global vector contains holistic high-level information of the caption. In this study, we use the global and word-level vectors as textual condition vectors, hence $\phi(S) = \boldsymbol{s}, T_w$.

5.2.2 Architecture

TAR-GAN is built upon the recent success of the BigGAN-Deep architecture. BigGAN-Deep (and BigGAN) were proposed by Brock *et al.* [4] to improve the performance of GANs in a large scale setting. BigGAN-Deep is based on residual blocks with bottleneck structure of He *et al.* [22], which makes deeper networks more computationally efficient and easier to train. Also, like SAGAN [74], BigGAN-Deep applies Spectral Normalization [42] and Non-local Blocks [66] in both generator and discriminator. Finally, BigGAN-Deep introduces conditioning information in the generator using Conditional Batch Normalization [13] and in the discriminator using the projection approach by Miyato *et al.* [43].

Specifically, in the BigGAN-Deep architecture, class labels are represented by *learnable class embeddings*. In the generator, class embeddings are concatenated with the $\boldsymbol{z}$ vector and



(a) Generator architecture. (b) Discriminator architecture.

Figure 5.1 – TAR-GAN architecture.

then linearly projected to form Batch Normalization [27] gains and biases. In the discriminator, class embeddings are used to perform the conditioning projection. The natural way to adapt the BigGAN-Deep architecture to handle text-to-image synthesis is by simply replacing its learnable class embeddings by sentence embeddings computed by the DAMSM module. In this setting, however, its performance fall below state-of-the-art. This is mainly because BigGAN-Deep, like other traditional GANs [42, 74], is designed to learn a class-conditional data distribution, not a text-conditional one. The nature of text is, notably, far more complex than a discrete set of class labels. To circumvent this issue, we propose a Text Auxiliary Regressor, that is designed specifically to learn a text-conditional distribution.

5.2.3 Text Auxiliary Regressor

Inspired by AC-GAN [45] and TAC-GAN [9], we design a novel auxiliary regressor that makes text conditioning more accurate and natural. Alongside with the standard output regarding the data source prediction (real/fake), we extend the discriminator to predict the sentence embedding as well. The goal is to make the discriminator learn the relationship between real images and their sentence embeddings so that the generator will be forced to make generated images closely related to sentence embeddings of real images. To do so, we train the discriminator to increase the cosine similarity between the embedding predicted for a real image $\boldsymbol{x}$ and its ground truth sentence embedding $\boldsymbol{s}$:

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{s}\sim q_{\text{data}}} \left[1 - \cos\left(D_{reg}(\boldsymbol{x}), \boldsymbol{s}\right)\right] \tag{5.1}$$

Since the discriminator learns the relationship between real data and their text embeddings, the generator objective becomes generating samples that make the discriminator predict text embeddings that resembles the real data:

$$\mathop{\mathbb{E}}_{\boldsymbol{s},T_w\sim q_{\text{data}},\boldsymbol{z}\sim p_{\boldsymbol{z}}} \left[1 - \cos\left(D_{reg}(G(\boldsymbol{z}, \boldsymbol{s}, T_w)), \boldsymbol{s}\right)\right], \tag{5.2}$$

where $G(\boldsymbol{z}, \boldsymbol{s}, T_w)$ is a fake image generated by the generator, $\boldsymbol{z} \in \mathbb{R}^{128}$ is a noise vector sampled from a normal distribution $\mathcal{N}(0, 1)$ and $T_w \in \mathbb{R}^{N\times 256}$ is matrix of word embeddings. Naturally, by producing adversarial examples, the generator may produce examples that do not necessarily correlate with the sentence embeddings of the real data and yet fool the discriminator. To counteract this issue we add the following term to the discriminator's loss:

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\tilde{\boldsymbol{s}}\sim q_{\text{data}}} \left[\cos\left(D_{reg}(\boldsymbol{x}), \tilde{\boldsymbol{s}}\right)\right], \tag{5.3}$$

where $\tilde{\boldsymbol{s}}$ is a random sentence embedding sampled from dataset. This restricts the discriminator to predict sentence embeddings correctly and reject adversarial examples.

5.2.4 Sentence Interpolation

In order to alleviate the problem of discontinuity in sentence embedding space, we employ the Sentence Interpolation strategy introduced in [59]. Formally, let I_i be the i^{th} image from the training dataset, and $S_{ij} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_n\}$ be the set of n correlated sentence embeddings that describe that particular image. We sample an n-sized vector of weights $\mathbf{m} \sim \mathcal{U}(0, 1)$, and further normalize it with a softmax function. Those normalized values are used to weight each one of the sentence vectors, so their sum consists in an interpolated representation of the original sentences.

Therefore, the vector $\dot{\boldsymbol{s}}$ that represents the interpolated textual embedding of a given image is calculated as follows:

$$\dot{\boldsymbol{s}} = \sum_{j=1}^{n} \left[S_j \times \left(\frac{e^{\mathbf{m}}}{\sum_{k=1}^{n} e^{\mathbf{m}_k}} \right)_j \right] \tag{5.4}$$

Such an approach makes a limited set of sentences to be represented by countless continuous points during the training process. The main implications of this technique are two-fold: (i) it makes the sentence embedding space to be more smooth; (ii) and also works as a data augmentation strategy, given that the same textual descriptions can assume different forms depending on the sampling of $\mathbf{m}$. In comparison to the Conditioning Augmentation (CA) module introduced by StackGAN [76], the sentence interpolation has the advantage of being deterministic. This is due to the fact that it is not used during the test phase. CA, on the other hand, introduces randomness when encoding sentence vectors during training and testing.

5.2.5 Objective Function

In the proposed Text Auxiliary Regressor Generative Adversarial Networks (TAR-GAN), the objective function plays a critical role in guiding the learning process of both the discriminator and the generator. To tailor the adversarial framework to the task of generating text-related data, we optimize a modified hinge version of the standard GAN loss for the adversarial components, along with an auxiliary regression loss to ensure the semantic consistency of the generated data with the provided text information.

The adversarial loss for the discriminator is formulated as a hinge loss, which has been shown to stabilize the training of GANs. Specifically, the discriminator loss $\mathcal{L}_{D_{adv}}$ is calculated as follows:

$$\begin{aligned} \mathcal{L}_{D_{adv}} = & \, \mathbb{E}_{\boldsymbol{x} \sim q_{\text{data}}(\boldsymbol{x})} \left[\max\left(0, 1 - D_{adv}(\boldsymbol{x})\right) \right] \\ & + \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})} \left[\max\left(0, 1 + D_{adv}\left(G(\boldsymbol{z}, \boldsymbol{s}, T_w)\right)\right) \right], \end{aligned} \tag{5.5}$$

where $\boldsymbol{x}$ represents real input data, $\boldsymbol{z}$ denotes a noise tensor sampled from a prior distribution $p(\boldsymbol{z})$, $\boldsymbol{s}$ stands for a sentence embedding, and T_w encapsulates a tensor of word embeddings corresponding to textual information. This loss encourages the discriminator to assign higher scores to real images and lower scores to fake ones generated by the generator.

In contrast, the adversarial generator loss $\mathcal{L}_{G_{adv}}$ is designed to deceive the discriminator by generating data that is indistinguishable from real data:

$$\mathcal{L}_{G_{adv}} = -\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[D_{adv}\left(G(\boldsymbol{z},\boldsymbol{s},T_w)\right)\right], \tag{5.6}$$

where the generator G aims to maximize the discriminator's mistake on the generated data.

The regression loss for the discriminator $\mathcal{L}_{D_{reg}}$ is employed to measure the similarity between the discriminator's outputs and the provided sentence embeddings, thus ensuring that the discriminator can accurately associate images with the respective textual descriptions. It is defined as:

$$\mathcal{L}_{D_{reg}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{s}\sim q_{\text{data}}}\left[1-\cos\left(D_{reg}(\boldsymbol{x}),\boldsymbol{s}\right)\right], \tag{5.7}$$

where $\cos$ denotes the cosine similarity and D_{reg} is the regression part of the discriminator that outputs a vector to be compared with the true sentence embedding $\boldsymbol{s}$.

Similarly, the regression loss for the generator $\mathcal{L}_{G_{reg}}$ ensures that generated images are semantically aligned with the provided text:

$$\mathcal{L}_{G_{reg}} = \mathbb{E}_{\boldsymbol{s}\sim q_{\text{data}}(\boldsymbol{s})}\left[1-\cos\left(D_{reg}(G(\boldsymbol{z},\boldsymbol{s},T_w)),\boldsymbol{s}\right)\right]. \tag{5.8}$$

In order to further enhance the discriminator's robustness against adversarial samples that might exploit the regression task, we introduce an additional loss term $\mathcal{L}_{D_{random}}$. This term penalizes the discriminator if it predicts high similarity for paired images and randomly chosen sentence embeddings $\tilde{\boldsymbol{s}}$, which are not genuinely corresponding to the images:

$$\mathcal{L}_{D_{random}} = \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{s}}\sim q_{\text{data}}}\left[\cos\left(D_{reg}(\boldsymbol{x}),\tilde{\boldsymbol{s}}\right)\right], \tag{5.9}$$

This component ensures that the discriminator does not get biased toward generating high similarity scores indiscriminately, thereby increasing its capacity to discern between semantically matched and unmatched image-text pairs.

Combining these components, the final objective function for the discriminator is given by the sum of the adversarial, regression, and robustness against random sentence embedding losses:

$$\mathcal{L}_D = \mathcal{L}_{D_{adv}} + \mathcal{L}_{D_{reg}} + \mathcal{L}_{D_{random}}, \tag{5.10}$$

The final objective for the generator harmonizes the adversarial goal with the semantic alignment of generated images and corresponding text:

$$\mathcal{L}_G = \mathcal{L}_{G_{adv}} + \mathcal{L}_{G_{reg}}. \tag{5.11}$$

By this, we set the stage for an adversarial training procedure where both the discriminator and the generator not only contest in the classic sense of a GAN but also collaborate to ensure that the generated images are semantically coherent with the textual descriptions, providing a synergy between the two tasks.

5.3 Experimental Results

5.3.1 Datasets

Caltech-UCSD Birds (CUB) [64]: The CUB Dataset is composed of 11,788 images of birds distributed among 200 class categories. The dataset is split in 8,855 images of 150 categories for training and 2,933 images of 50 categories for testing. Each image contains 10 text descriptions.

MS Comon Objects in Context (COCO) [37]: The COCO Dataset is composed of $80k$ images for training and $40k$ images for testing. Unlike the CUB Dataset, COCO images are composed of scenes containing multiple objects, which makes text-to-image generation particularly challenging. In the COCO Dataset, each image contains 5 text descriptions.

5.3.2 Evaluation

In order to quantitatively evaluate the performance of the proposed TAR-GAN, we employ two widely used metrics for evaluating generative models: the Inception Score (IS) [58] and the Fréchet Inception Distance (FID) [24].

5.3.3 Implementation Details

We use Adam optimizer [30] with a learning rate of 2×10^{-4}, $\beta_1 = 0$ and $\beta_2 = 0.999$ for both G and D. We train one D step per G step. We use synchronized implementation of BatchNorm, where statistics are aggregated across all devices. We keep an exponential moving average of the generator weights with a decay of 0.999 for sampling. Since BatchNorm statistics are not computed for averaged weights of the generator, we employ the “standing statistics" strategy of Brock *et al.*, where we run 100 forward passes to update BatchNorm statistics, making the generator invariant to batch sizes.

Figure 5.2 – Qualitative results in the CUB Dataset.

Table 5.1 – Quantitative results for TAR-GAN.

Method	# Networks		MS	IS↑		FID↓	
	D	G		CUB	COCO	CUB	COCO
GAN-INT-CLS [55]	1	1	No	2.88 ± .04	7.88± .07	-	-
GAWWN [56]	1	1	No	3.60 ± .07	-	-	-
StackGAN [76]	2	2	Yes	3.70 ± .04	8.45 ± .03	-	-
StackGAN++ [75]	3	3	Yes	4.04 ± .05	8.30 ± .10	15.30	81.59
HDGAN [78]	3	3	Yes	4.15 ± .05	11.86 ± .18	-	-
AttnGAN [71]	3	3	Yes	4.36 ± .03	25.89 ± .47	14.01	29.53
MirrorGAN [49]	1	3	Yes	4.53 ± .17	26.47 ± .41	-	-
DM-GAN* [81]	3	3	Yes	4.71 ± .06	32.43 ± .58	11.91	**24.24**
SD-GAN [73]	6	6	Yes	4.67 ± .09	**35.69 ± .50**	-	-
TAR-GAN	1	1	No	**4.75 ± .05**	-	**10.87**	-

* Updated according to pretrained models provided by the authors Github repository.

5.4 Comparison to state-of-the-art methods

In order to provide reassurance on the generative performance of our models, we compare their quantitative and qualitative results against current state-of-the-art methods [55, 56, 76, 75, 78, 71, 49, 81, 73]. Note that some of them have not reported FID results. Hence, we compare to the results publicly available.

Figure 5.3 – Qualitative results in the COCO Dataset.

5.4.1 Quantitative Analysis

Table 5.1 provides an extensive quantitative evaluation of the proposed Text Auxiliary Regressor Generative Adversarial Network (TAR-GAN) method, presenting empirical performance measurements in comparison to a curated set of state-of-the-art models. The entries in the table are arranged to demonstrate key metrics of generative model performance across various datasets. Each row corresponds to a different method, including ours, while the columns articulate the achieved scores for evaluation metrics such as the Fréchet Inception Distance (FID), alongside the architecture complexity indicated by the count of discriminators and generators used.

In an analytical review of the results, the data clearly evidences that our TARGAN method not only accomplishes but exceeds the benchmark figures in all the observed metrics, setting a new standard of state-of-the-art performance. Remarkably, this superior performance is attained with a notably lean architecture comprising only a singular discriminator and generator. This is a significant architectural simplification when contrasted against competing approaches, many of which utilize multiple discriminators or generator networks, yet do not achieve comparable results.

Focusing on dataset-specific performance, one observes that our method demonstrates its most pronounced improvement on the CUB dataset – a collection of bird images accompanied by textual descriptions. Compared to the strongest alternative with available public results, our approach provides a substantial relative boost, quantified as approximately an 8% enhancement in the FID score. This is a compelling argument in favor of our method's efficacy, particularly in scenarios that demand high-fidelity synthesis of detailed image features guided by text descriptions.

Despite the successes, our method showed diminished performance on the complex and diverse COCO dataset, which is well-known for its challenging, varied images and accompanying captions. The underwhelming results on COCO point out current limitations and suggest that our approach may require additional refinement to cope with the broad heterogeneity and intricacies contained within such a dataset. Future research endeavours should aim to bolster the model's robustness and its capacity to generalize across the wider spectrum of subject matter and linguistic nuances presented by image captions. This could potentially involve methodological enhancements, architectural changes, or even revisiting the training process to ensure stable learning when faced with high diversity in the input data.

5.4.2 Qualitative Analysis

In this section, we present a thorough qualitative analysis of the Text Auxiliary Regressor Generative Adversarial Networks (TARGANs), focusing on the visual improvements and semantic alignments of the generated images with respect to the input textual descriptions. Our evaluation considers models trained on two distinct datasets: the Caltech-UCSD Birds 200 (CUB) dataset and the Microsoft Common Objects in Context (COCO) dataset.

Figure 5.2 offers a comparative visualization of the generative capabilities of our proposed TARGAN model against established baseline models using the CUB dataset. Through the figures presented, several improvements introduced by our model become evident in comparison to the baselines. Firstly, the photorealism of the images produced by our model is notably enhanced, exhibiting a marked progression in visual appeal and realism. There is also a measurable enhancement in the semantic congruence between the generated images and the corresponding written descriptions, demonstrating the model's ability to capture and express the nuanced descriptions in visual form. Furthermore, the level of detail portrayed in the images—especially in the fine-grained textures and features within both the focal subjects (foreground elements) and the ambient setting (background elements)—is considerably more refined, lending to a more authentic and high-fidelity representation of the textual input.

Moving on to Figure 5.3, we depict results obtained from a TARGAN model fine-tuned on the COCO dataset. The graphical outputs demonstrate that our model has achieved the production of several images that exhibit a more coherent overall structure when contrasted with results from prior models. The images generated display an array of complex scenes and objects that are more structurally sound and visually pleasing. Despite these achievements, the model did encounter noteworthy limitations. A moderate degree of mode collapse was observed, indicating a restricted diversity in the generated images. Moreover, the model exhibited difficulty in learning to create imagery from various domains, a deficiency that was particularly visible when attempting to synthesize scenes not well-represented in the training dataset. Consequently, this shortcoming has had an adverse impact on the quantitative metrics that we used to evaluate our model's performance, thus highlighting the areas that require further investigation and improvement. Such enhancements are

imperative for our approach to be considered competitive and comparable with the state-of-the-art methods in the field.

In conclusion, while the TAR-GANmodel has demonstrated great progress in certain aspects of image generation, there are critical challenges that need to be addressed. Through targeted research efforts aimed at mitigating the issues of mode collapse, especially in multi-object datasets. It is noticeable that the full potential of the TAR-GANapproach can be realized, propelling it to the forefront of text-to-image synthesis technologies.

6. CROSS-LANGUAGE TEXT-TO-IMAGE SYNTHESIS

6.1 Introduction

Generating images from textual descriptions is a highly valuable task that can enhance numerous applications, ranging from support in computer-assisted drawing to creating text-based graphical content. While this field of text-to-image generation shows promise, it also poses a substantial level of complexity and challenge. Text descriptions inherently contain a degree of subjectivity, leading to a vast array of possible images that could align with a given description. Moreover, this task's multi-modal property—it involves both text and imagery—demands an integration of methodologies from the domains of computer vision as well as natural language processing.

Predominantly, cutting-edge frameworks for generating images from text are grounded on structures known as Generative Adversarial Networks (GANs) [18]. These networks have opened up new avenues for modeling generative processes over intricate data distributions conditioned on external information, such as generating image distributions based on textual descriptions. Diverging from GANs that hinge on conditioning with a fixed set of labels, like class identifiers, methods for synthesizing images from text must incorporate a different tactic for converting text descriptions into vector representations. Typically, text is transformed into a continuous vector feature space, which distinctively sets apart text-based image generation techniques from conventional conditional GAN approaches [41, 45, 43]. To navigate the unique challenges associated with text-to-image creation, researchers have introduced a variety of innovative strategies.

Most of the recent progress in text-to-image targeted the improvement of image generation, both in terms of quality and diversity. This includes making improvements in GAN architectures [51], training frameworks [42, 43] and even training scale [4]. Although this pursuit is valid and important, little attention has been paid to the textual part of the text-to-image framework. Moreover, due to the zero-shot characteristic of text-to-image (*i.e.* generating novel images for *any* text description), it is crucial that the textual representation is robust enough to generalize to all concepts.

Besides the lack of a study on the robustness on text encoding representation, so far, all text-to-image synthesis approaches have been limited to a single language. This is due to the fact that popular datasets like Caltech-UCSD Birds (CUB) [64] and Oxford-102 [44] only have texts descriptions available in english. It is widely acknowledged that acquiring data is laborious and expensive, but in this case, this limitation imposes a great toll to text-to-image research. This leaves a huge missed opportunity not only study text-to-image under multiple languages but also to bring text-to-image technology to non-english speakers.

In this work we propose the following study: i) First, we evaluate the most popular approach to encode text for text-to-image purposes: the Deep Attentional Multimodal Similarity Model method from AttnGAN [71]. We test the DAMSM encoder under different settings and measure how well it generalizes to unseen data. ii) We propose an extension to the current text-to-image

Figure 6.1 – Images generated by our XLANG-GAN. Blue text is English, pink text is French and green text is Portuguese.

framework in order to handle multiple languages. By doing so we allow a single model to generate images given text descriptions in three languages. We call this method Cross-language Generative Adversarial Network (XLANG-GAN). Figure 6.1 shows the results of XLANG-GAN.

Our experiments demonstrate that XLANG-GAN successfully work under three different languages while preserving the same performance of a single-language text-to-image model. We quantitatively evaluate the performance of XLANG-GAN using the Inception Score (IS) [58] and the Fréchet Inception Distance (FID) [24]. We qualitative show that XLANG-GAN is robust across languages (*i.e.* the same sentence in different languages yield similar images). Finally, we demonstrate that XLANG-GAN supports out of the box language mixing while preserving generation semantics and meaning.

6.2 Method

In this section we present in detail the proposed approach. Text-to-image synthesis methods have been neglecting the support to languages other than English. In a survey, Frolov *et al.* [16], is clear about the gap regarding studies that address text-to-image generation over the optics of different languages. Ideally, text-to-image system should be able to handle multiple languages si-

multaneously, so that the same system would be accessible by a great number of people. To address this gap, we propose an extension to text-to-image synthesis that allow for multiple languages at once.

6.2.1 Acquiring Language Data

In this study, we address the challenge of cross-lingual image captioning by leveraging the capabilities of Google Translate APIs [19] for translating the captions of our image dataset from English into Portuguese and French. This translation process involved several systematic steps to ensure accuracy and consistency across the dataset. Initially, the original English captions underwent a preprocessing phase to remove any non-standard elements, such as special characters, which could interfere with the translation quality. Subsequently, these cleaned captions were programmatically passed to the Google Translate API through a series of HTTP requests, where the target languages were specified as Portuguese and French. The API utilizes state-of-the-art machine learning models, to provide high-quality translations that consider contextual nuances. Each English caption was translated independently to maintain the integrity of the dataset. Upon receiving the translated captions, post-processing was implemented to rectify any potential API translation errors and ensure uniformity of sentence structure across languages. The translated captions were then paired with their corresponding images to form a multilingual dataset.

6.2.2 Text Encoder

For converting textual descriptions into continuous vector representations that can be effectively used within a generative adversarial framework, we employ pre-trained model known as the Deep Attentional Multimodal Similarity Model (DAMSM), as elaborated in Xu et al. [71]. This encoding approach is influenced by the foundational principles outlined in various studies [14, 69, 36], which detail the learning process of image and text encoding functions.

Specifically, DAMSM learns two distinct but related encoding functions: $\varphi(I)$ for images and $\phi(S)$ for text. The functions map their respective inputs, namely an image I and its textual description S, into a shared semantic multimodal space. This joint embedding space is engineered such that semantically related image-caption pairs are spatially closer to one another compared to unrelated pairs. The distance metric imposed in this space ensures that correlated pairs have a smaller intervening distance, whereas non-correlated pairs are comparatively further apart. Through the optimization of this embedded space, the text encoding function is guided to produce text representations that encapsulate image content with great fidelity. Consequently, this results in more accurate text-based conditional vectors $\boldsymbol{s} \in \mathbb{R}^{256}$ that serve as informative cues within our generative adversarial architecture.

Training the DAMSM on the original image captions S begins with tokenization, where each word token is mapped to a dense and unique high-dimensional vector $\mathbb{R}^{300}$. These vectors then serve as input to a bidirectional Gated Recurrent Unit (GRU) [8] network. The network's architecture provides two outputs: per-token hidden state vectors that capture the granular details of the corresponding image features, and a global sentence vector that encodes the overall meaning of the text caption.

Both levels of representation – the localized per-token hidden states and the summarized global sentence vector – offer a complementary perspective on the textual data. They collectively enable the learning of detailed, fine-grained correlations with spatial image features. For our method, we explicitly incorporate these two forms of textual representations as our textual condition vectors. Therefore, our text encoder function $\phi(S)$ computes a pair of outputs, specifically represented as $\boldsymbol{s}, T_w$, which denote the global and word-level vectors, respectively. These vectors are then used to condition the generative process in a nuanced and effective manner.

In this work, we introduce several enhancements to the DAMSM framework to create a more robust and versatile model capable of handling multilingual data. The core modification includes the integration of a multi-language byte pair encoding (BPE) tokenizer [23]. BPE is a subword tokenization method that represents words by iteratively merging the most frequent pair of bytes or characters into a single, new byte or character; this process continues until a set vocabulary size is reached or no more merges are possible. This approach cleverly captures the frequency distribution of character combinations in a given corpus, enabling the identification of common subword units across different words, such as morphemes or syllables, which are pivotal for understanding morphology in various languages.

One of the primary benefits of utilizing multi-language pre-trained BPE embeddings in the training of the DAMSM module is the out-of-the-box support for a diverse set of languages without the need for separate tokenization models for each language. This generalization allows the DAMSM to be naturally applied to cross-lingual tasks, where it can handle input data in various languages and still compute robust similarity measures between modalities. The DAMSM can thus become adept in the representation and matching not only within the same language but also across different languages, increasing its applicability in multilingual contexts.

The enhancement associated with the training process focus on the crucial role of the batch size within the context of the DAMSM, which employs a contrastive loss function known as softmax alignment loss. By adapting both the text and image encoders to utilize half-precision floating-point representation (float16), the memory consumption is effectively halved, enabling a substantial increase of the batch size. Specifically, the transition from the initial DAMSM's limitation of 16 images/caption pairs per batch is dramatically increased to 80 images/caption pairs per batch. This increase in the batch size significantly enriches the contrastive dynamics of the loss function, which in turn directly translates to an enhancement in the model's accuracy. Essentially, larger batch sizes makes a more discriminative learning process, where the model can better distinguish between different examples by contrasting more pairs within a single iteration.

6.2.3 Architecture

XLANG-GAN is built upon the TAR-GAN presented in Chapter 5. TAR-GAN employs a single generator and discriminator approach alongside a text auxiliary regressor designed specifically for text-to-image generation. We follow the same overall training procedure, with the same loss function and hyper-parameters.

6.2.4 Sentence Interpolation

In order to alleviate the problem of discontinuity in sentence embedding space, we employ the Sentence Interpolation strategy introduced in [59]. Formally, let I_i be the i^{th} image from the training dataset, and $S_{ij} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_n\}$ be the set of n correlated sentence embeddings that describe that particular image. We sample an n-sized vector of weights $\mathbf{m} \sim \mathcal{U}(0, 1)$, and further normalize it with a softmax function. Those normalized values are used to weight each one of the sentence vectors, so their sum consists in an interpolated representation of the original sentences. Therefore, the vector $\dot{\boldsymbol{s}}$ that represents the interpolated textual embedding of a given image is calculated as follows:

$$\dot{\boldsymbol{s}} = \sum_{j=1}^{n} \left[S_j \times \left(\frac{e^{\mathbf{m}}}{\sum_{k=1}^{n} e^{\mathbf{m}_k}} \right)_j \right] \tag{6.1}$$

Such an approach makes a limited set of sentences to be represented by countless continuous points during the training process. The main implications of this technique are two-fold: (i) it makes the sentence embedding space to be more smooth; (ii) and also works as a data augmentation strategy, given that the same textual descriptions can assume different forms depending on the sampling of $\mathbf{m}$. In comparison to the Conditioning Augmentation (CA) module introduced by StackGAN [76], the sentence interpolation has the advantage of being deterministic. This is due to the fact that it is not used during the test phase. CA, on the other hand, introduces randomness when encoding sentence vectors during training and testing. As a key differentiator, in this work we perform the sentence interpolation across all three languages: English, French and Portuguse. This richness in text captions, helps even further in learning a smooth embedding conditioning space.

6.3 Experimental Results

6.3.1 Datasets

Caltech-UCSD Birds (CUB) [64]: The CUB Dataset is composed of 11,788 images of birds distributed among 200 class categories. The dataset is split in 8,855 images of 150 categories for training and 2,933 images of 50 categories for testing. Each image contains 10 text descriptions.

Oxford-102 [44]: The Oxford-102 dataset is composed of 8189 images of flowers of 102 categories. The dataset is split in 7034 images for training and 1154 images for testing. Each image contains 10 text descriptions.

6.3.2 Evaluation

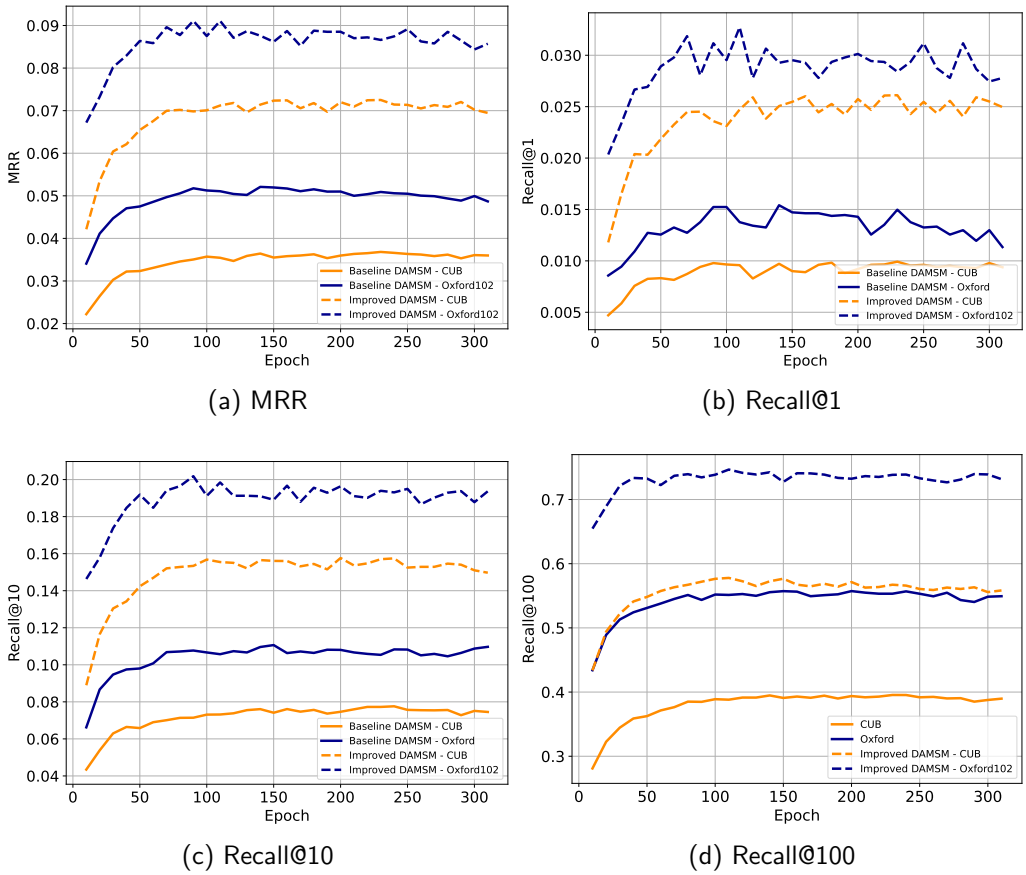
To assess the enhancements made to the DAMSM module, we adopt the standard metrics that are commonly utilized for gauging the performance of text-image alignment models. Specifically, we begin by calculating the embedding of each caption in our dataset. Subsequently, we measure the similarity between this caption embedding and the embeddings of all images within the dataset by ranking the images based on their proximity (or distance) to the caption embedding. Once we have this ranking in place, we proceed to evaluate the model's retrieval capabilities using two predominant metrics: the Mean Reciprocal Rank (MRR) and Recall@k. Here, "k" represents the number of top-ranked items we consider for the calculation of recall.

The MRR metric is computed by taking the average of the reciprocal ranks of the correct item (in this case, the corresponding image for a given caption) for each query across all queries in the test set. The reciprocal rank is the inverse of the rank at which the correct item is retrieved; if the correct image is ranked first, the reciprocal rank is 1, if it's the second, the reciprocal rank is $1/2$, and so on. Mathematically, MRR is given as:

$$MRR = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{rank_i}$$

where Q is the number of queries, and $rank_i$ is the position of the first relevant document for the i-th query.

Recall@k, on the other hand, measures the proportion of relevant items found in the top-k rankings. It is computed by evaluating each query to see if the relevant item (correct image) appears within the top-k positions in the ranked list of retrieved items. The Recall@k is then calculated as the number of queries for which the relevant item is within the top-k divided by the total number of queries Q. The formula for Recall@k can be expressed as:



(a) MRR (b) Recall@1

(c) Recall@10 (d) Recall@100

Figure 6.2 – Evaluation during the training process between our improved DAMSM *versus* the baseline DAMSM.

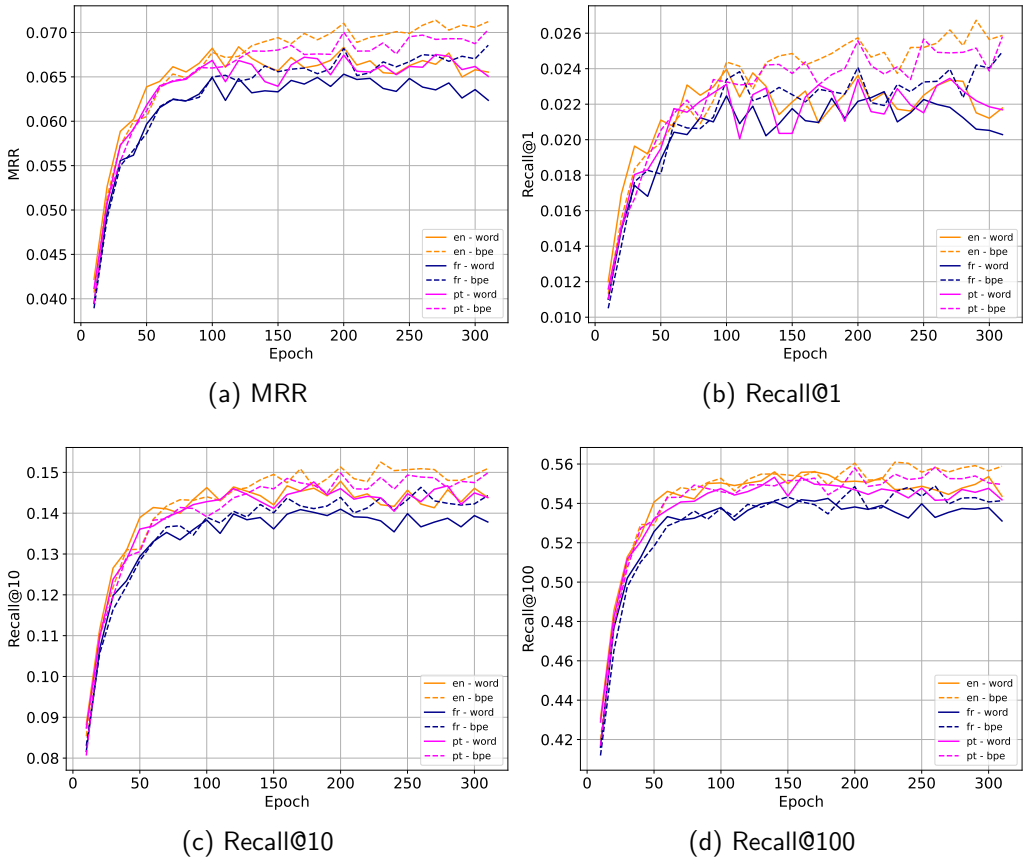
$$Recall@k = \frac{|\{q_i : rank(relevant\_item_i) \leq k\}|}{Q}$$

where $|\{q_i : rank(relevant\_item_i) \leq k\}|$ is the count of queries where the relevant item is ranked at or above the k-th position.

Together, MRR and Recall@k provide a comprehensive view of the performance of text-image alignment models, with MRR focusing on the average precision of the top-ranked retrieval and Recall@k indicating the model's ability to retrieve relevant items within the top-k positions.

6.3.3 Comparison to State-of-the-art

Since no previous work deeply assessed the performance of the DAMSM module. We show how the baseline DAMSM (as proposed in [71]) performs. Then, we present our improvements and demonstrate how they compare to the baseline DAMSM. Figure 6.2 show the performance of the baseline DAMSM versus our improved DAMSM during training. Our improved DAMSM performs significantly better in both CUB and Oxford-102 datasets and in all the four retrieval metrics: MRR, Recall@1, Recall@10 and Recall@100.



(a) MRR (b) Recall@1

(c) Recall@10 (d) Recall@100

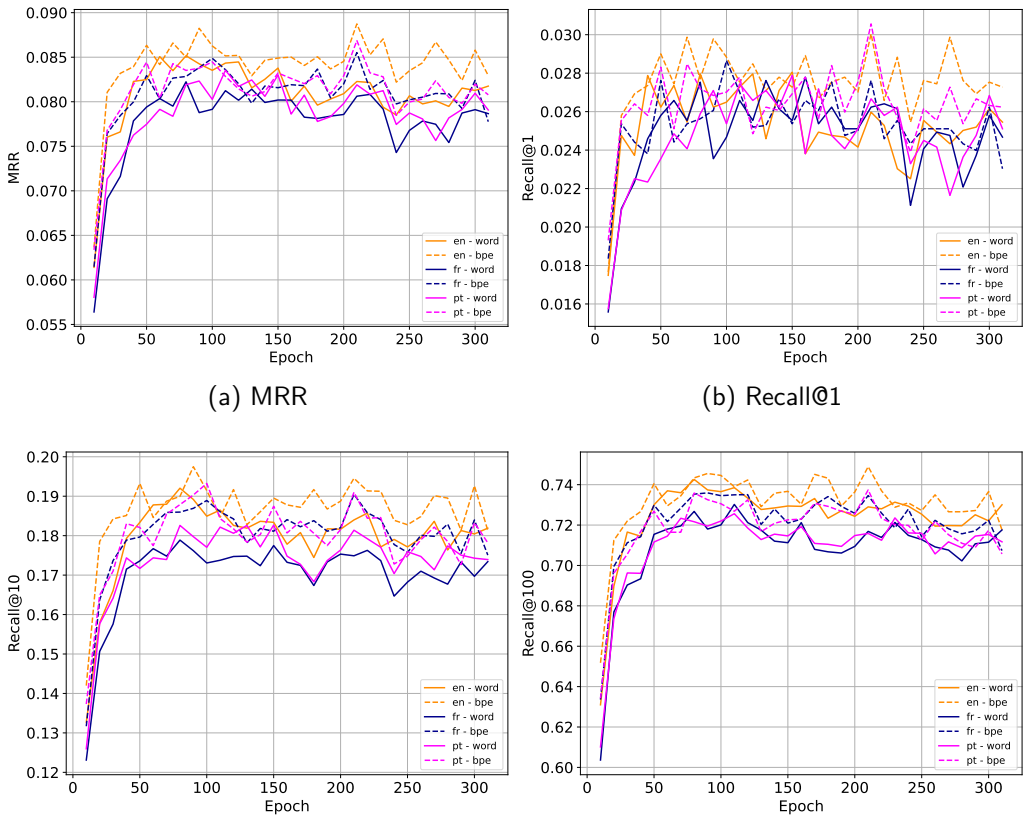
Figure 6.3 – Evaluation during the training process in the CUB dataset for all languages: English, Portuguese and French.

We also present language-wise performance improvements. Figure 6.3 shows the performance in the CUB dataset for English, Portuguese and French under two different tokenization strategies: word tokenization (the baseline), and the multi-language BPe (our improved version). English is the best performing language, followed by Porguese. French lags a little bit behind. Figure 6.4 present the similar results on the smaller Oxford-102 dataset.

6.3.4 Implementation Details

To train our DAMSM encoder, we use Adam optimizer [30] with a learning rate of 2×10^{-4}, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and a batch size of 80 pairs of image/caption. We employ the Sentence Interpolation (SI) introduced in Chapter 4 with the difference that we interpolate across all three languages. We train for 500 epochs and evaluate every 10 epochs to select the better model.

After we train and select the best DAMSM model, we compute the text vectors for all the captions in the datasets. The vectors, then, are used as condition to train the text-to-image model, which is a TAR-GAN. The configuration and hyper parameters are the same as presented in Chapter 5.



(a) MRR (b) Recall@1

(c) Recall@10 (d) Recall@100

Figure 6.4 – Evaluation during the training process in the Oxford-102 dataset for all languages: English, Portuguese and French.

7. DISCUSSION

This Chapter presents the discussion over the contributions of this thesis.

7.1 Summary of Contributions

This work presented three novel approaches to improve text-to-image generation, each setting new standards within their respective domain as measured by quantitative and qualitative metrics. In summary, we developed the following approaches:

7.1.1 Efficient Neural Architecture for Text-to-Image Synthesis

In this work, we shift the architectural paradigm currently used in text-to-image methods and show that an effective neural architecture can achieve state-of-the-art performance using a single stage training directly at the target resolution. By doing so, we not only introduce a simpler method for text-to-image synthesis but also point a new direction in text-to-image research, which has not experimented with novel neural architecture recently. This work was published on the International Joint Conference on Neural Networks (JCNN) in 2020 [59].

We introduce an adversarial training-based architecture that leverages full capacity of modern deep convolutional networks, alongside to an improved sentence embedding approach for generating photorealistic text-conditioned images. Both discriminator and generator networks draw inspiration from [4], though we provide important improvements on that architecture, allowing for the use of sentence embeddings rather than class labels as conditioning vectors. Results show that our models single-handedly outperform multi-stage state-of-the-art methods without heavy hyper-parameter optimization in two widely used benchmarks, namely CUB [64] and Oxford-102 [44] datasets, in terms of both Inception Score [58] and Fréchet Inception Distance [24].

7.1.2 Text-to-Image Generation with Text Auxiliary Regressor GANs

In this work we introduce a new approach for text-to-image synthesis that is not only efficient but also dramatically simpler. We follow the steps of our previous work [59] and employ a single generator/discriminator architecture. We propose a Text Auxiliary Regressor Generative Adversarial Network, namely TAR-GAN, that achieves state-of-the-art performance using a single generator and a single discriminator. By using a novel Auxiliary Regressor, that was designed specifically for text conditioning. TAR-GAN is intended to close the performance gap between text-to-image generation and traditional class-conditional GANs [42, 74, 4].

Our experiments demonstrate that TAR-GAN favorably outperforms the previous state-of-the-art methods. We quantitatively evaluate the performance of TAR-GAN using the Inception Score (IS) [58] and the Fréchet Inception Distance (FID) [24]. Our method presents a 8% FID improvement in the CUB Dataset [64].

7.1.3 Cross-language Text-to-Image Synthesis

In this work we propose a method to extend text-to-image generation models to handle multiple languages. To do so, we perform the following study: i) First, we evaluate the most popular approach to encode text for text-to-image purposes: the Deep Attentional Multimodal Similarity Model method from AttnGAN [71]. We test the DAMSM encoder under different settings and measure how well it generalizes to unseen data. ii) We propose an extension to the current text-to-image framework in order to handle multiple languages. By doing so we allow a single model to generate images given text descriptions in three languages. We call this method Cross-language Generative Adversarial Network (XLANG-GAN).

Our experiments demonstrate that XLANG-GAN successfully work under three different languages while preserving the same performance of a single-language text-to-image model. We qualitative show that XLANG-GAN is robust across languages (*i.e.* the same sentence in different languages yield similar images). Finally, we demonstrate that XLANG-GAN supports out of the box language mixing while preserving generation semantics and meaning.

7.2 Impact

The work proposed in this thesis has made a significant contribution to the field of text-to-image generation. At the heart of this impact is the proposed Efficient Neural Architecture for Text-to-Image Generation [59], a pivotal paper emerging from the thesis work, which has garnered considerable attention as evidenced by citations from several relevant and contemporaneous studies. The proposed architecture has achieved a harmonious balance between computational efficiency and the ability to generate high-resolution, contextually accurate images from textual descriptions, effectively pushing the boundaries of what is possible in creative AI applications.

The other methods proposed in this work, despite not being published yet, offer novel components for text-to-image generation research. As pointed out by Frolov *et al.* [16], the multi-linguistic component of text-to-image generation has been severely neglected by current research. The ability to handle multiple languages is crucially important as it makes technology more accessible to a huge part of the population with is not familiar with the English language. Moreover, cross-linguistic approaches may open the field to new ideias and methodologies that can bring several advances, including the ones that can be benefit text-to-image in its foundation, like improving image quality and diversity.

7.3 Comparative Analysis of Methods

In text-to-image research, model performance is measured by automated quantitative metrics. Since the challenge of evaluating *quality* and *variability* of generated images is immense, each metric has its drawbacks. Below we discuss the caveats of quantitative metrics used in this work.

7.3.1 Evaluation Metrics

The first automated proposed metric for evaluation of generative models for images is the Inception Score (IS) [58]. IS is intended to measure both image quality and diversity. The metric is computed over class probabilities computed by an image-classification Inception Network [61] – you can view more details in Section 3.2.3. Even though IS adoption as comparison measure is widespread in the research field, this metric present some flaws. First, since the metric is computed of class probabilities, if by any chance, we produce a collapsed model that can generate *only one and the same* image of each class, then we would maximize IS. For instance, if we train a GAN on Imagenet and our model is capable of generating only 1000 good and unique images, one corresponding to each class from the dataset, then IS will be maximized. Therefore, it's implementation by default is faulty in regards to capturing variability. The second drawback is regarded to the use of the Inception Network. In order for the measurements to be comparable between different works, everybody needs to use the same Inception Network implementation with the same weights. To this day, the 6 year old Inception weights are still being used to compute IS. Using this old implementation is not only difficult technically because it uses old frameworks and implementations but also because it is unreliable, different measurements doesn't always produce the exact the same results.

To circumvent some of the aforementioned problems, Heusel *et al.*proposed the Fréchet Inception Distance (FID). The FID differs from the IS in the sense that it does not use class probabilities. Instead, FID compute the intermediate features of both images from the training set and generated images and then measure the Fréchet distance between the two sets of features. The intuition is that, if the model produces sample with the same statistical properties as the training data, then it is a good model. Since FID calculates a *distance*, the lower the measurement, the better. If we were to compute the FID between the training set and itself, the result would be zero. For more details on how FID is computed, see Section 3.2.3. Even though FID presents some advantages of IS, it also built upon heavy assumptions. First, FID assumes that the set of image features are a perfectly multidimensional normal distribution, which may not be case in practice. Secondly, it uses the same 6 year old Inception Network as the IS, which brings several drawbacks.

For text-to-image generation specifically, some metrics were proposed to address the use case more precisely; however, they are not as widespread as IS and FID. Other evaluation metrics for text-to-image generation include qualitative human judgment and quantitative measures like the Structural Similarity Index Measure (SSIM) [67], the Learned Perceptual Image Patch Similarity

(LPIPS) [77], and the R-precision metric. The SSIM is a metric that assesses the quality of an image based on an initial uncompressed or distortion-free image as the reference. SSIM considers changes in texture, luminance, and contrast when comparing the quality of images, making it a good fit for capturing structural information. In text-to-image generation, SSIM can be used to compare the structural similarity between a generated image and a ground-truth image, if one is available. LPIPS, meanwhile, uses deep learning features to compute the similarity between images. It was developed to better reflect human judgment by considering perceptual differences between images. When applied to text-to-image generation, LPIPS can indicate how perceptually similar a generated image is to a set of real images, given the same input text. R-precision is a retrieval-based metric that quantifies how well a generated image matches a given text description relative to other distracting images. During evaluation, a text prompt is used to generate an image, which is then matched with a reference set containing the correct corresponding image among other decoys. The R-precision score is the fraction of cases where the generated image is closer to the corresponding real image than to any of the decoys, reflecting the model's ability to create relevant and specific images based on text descriptions.

7.4 Limitations

The contributions of this thesis helped improving text-to-image synthesis in different ways: a simplified neural architecture, a better performing loss function, and a novel cross-language method. However, each of those approaches have their own limitations. Despite presenting good results on single-object datasets like CUB and Oxford-102, our methods failed to excel in multi-object datasets like MS COCO. The high variation and complexity of dataset like MS COCO has been a challenge for text-to-image methods to address. We theorize that this difficult of caused by the small amount of data compared to its variation, which causes the model to fail to learn a function that generalizes well. Another limitation that must be acknowledged is the fact that we used machine translation to create language data for our third method. This means that our model is limited to the capability of the translation model. Ideally, to train truly multi-lingual models, language data must come from native speakers who can capture all the nuances in language and therefore produce high quality data points.

These limitations must be acknowledged as they may impair application in certain domains. It also helps us to emphasize the areas where future research could focus on enhancing the robustness and effectiveness of text-to-image synthesis models. Addressing these issues may involve using additional data for training, possibly applying some pretraining strategy and/or transfer learning strategy. This way, we a foundational knowledge about image synthesis, the model could better be adjusted to the complexity of multi-object datasets.

8. RELATED WORK

This chapter provides an overview of the foundational and recent advancements in the machine learning domains essential to our study. We focus primarily on Generative Adversarial Networks (GANs), and the evolution of methods in the text-to-image synthesis field.

8.1 Generative Adversarial Networks (GANs)

Introduced by Goodfellow et al. [18], Generative Adversarial Networks (GANs) have revolutionized the generative modeling landscape. A GAN comprises two competing neural network models: a Generator G, which creates images aiming to be indistinguishable from real images, and a Discriminator D, which aims to distinguish between the generator's output and genuine images. During training, G and D engage in a minimax game, with G learning to produce increasingly realistic images and D improving its ability to detect artificial ones.

The field has seen significant improvements tackling issues like training stability and enhancing the quality of generated images. Milestones include novel training strategies and loss functions [1, 39, 20], as well as architecture innovations [29, 42, 74, 40, 4]. Additionally, the utility of GANs has been demonstrated across various applications, including image-to-image translation [28, 7, 65, 80], image inpainting [48], image editing [79], and image super-resolution [35].

8.1.1 GANs for Text-to-Image Synthesis

Reed et al. [55] pioneered the integration of textual conditioning into GANs to drive the generation of corresponding images. This approach involves first encoding text descriptions into vector form, then inputting these vectors into a Conditional GAN [41] to steer image synthesis. Further, Reed and colleagues [56] expanded the model's abilities to account for spatial relationships described in text inputs.

A breakthrough came with the proposition of StackGAN by Zhang et al. [76], which introduced a multi-stage generation process beginning with low-resolution images that are subsequently refined. Their innovative Conditioning Augmentation (CA) technique contributed to enhanced training stability by projecting text embeddings into a well-behaved distribution. StackGAN++ [75] built upon this by incorporating multiple sets of generators and discriminators for successive resolution enhancements. Likewise, HDGAN [78] employed a multi-stage approach, incorporating a patch-wise adversarial loss to achieve high-quality images.

Emerging from the limitation of using only global sentence embeddings, AttnGAN [71] leveraged attention mechanisms to incorporate fine-grained word-level details into image generation. Complementary methods like MirrorGAN [49] involved iterative processes of image creation and

redescription, while DM-GAN [81] focused dynamic memory modules on critical text elements during image refinement. SD-GAN [73] implemented a siamese network structure to ensure consistency in images corresponding to variant textual descriptions.

Our research departs from the usual text-to-image architecture. We simplify the framework, replacing multi-stage processes with a single pair of Generator/Discriminator architecture that can be trained directly at the target resolution. Integrating a novel sentence interpolation technique, we present a model trained in a smoother conditional space, which enhances generative quality and enables more natural image manipulation through latent space arithmetic. Quantitative and qualitative analysis present the improves our methods over the previous state-of-the-art.

8.2 Cross-language Text-to-image Synthesis

Investigations into text-to-image synthesis have historically been restricted to single-language generation, mainly because most popular datasets contain English captions only. Frolov *et al.* [16] pointed this open challenge regarding text-to-image research. To this end, we propose the method presented in Chapter 6. We translated the most used datasets, CUB and Oxford-102, to two new languages, Portuguese and French. Finally, we propose a new method that allow text-to-image generative models to support multiple language simultaneously.

9. CONCLUSION AND FUTURE WORK

In this thesis, three novel methodologies designed to enhance the capability of small-scale text-to-image synthesis have been introduced and evaluated. Each methodology aims at overcoming common challenges in the synthesis process, such as computational requirements, the complexity of frameworks, and the capability of multilingual support. The improvements made through these methods have contributed to the evolution of state-of-the-art, as evidenced by improved performance metrics and enhanced visual aspects. Collectively, this research provides insights for future exploration within the area of generative adversarial networks and multimodal learning.

Chapter 4 introduced a novel method that simplifies the text-to-image synthesis framework. This approach was presented at the International Joint Conference on Neural Networks (IJCNN) with the paper titled "Efficient Neural Architecture for Text-to-Image Synthesis" [59]. It proposes an architecture that is not only more straightforward but also demonstrates superior performance compared to its predecessors. This method represents a shift in the design of text-to-image neural networks, allowing for high-quality image generation through an end-to-end, single-stage training process at the target resolution. This improvement not only offers an easier solution for text-to-image synthesis but also points to new directions for research, departing from the usual go-to architectural choice.

The work mentioned in Chapter 5 builds upon the Efficient GAN with a novel loss function, enhancing the treatment of text conditioning in the model. We point out that the conventional textual embedding might not be the most suitable for conditioning GANs, which were designed for discrete data types like class labels. Responding to this, we introduced a new text-specific conditioning methodology that incorporates a specialized Auxiliary Regressor, allowing for direct model training at the target resolution. This contribution, referred to as TAR-GAN, narrows the gap in performance between text-to-image synthesis and conventional class-conditional GANs [42, 74, 4].

Lastly, Chapter 6 expands the text-to-image synthesis to add multilingual capabilities. This study was motivated by two main factors: the lack of robustness studies on text encoding representations and the limited language availability, predominantly for English, in current models. To this end, the study focussed on: i) an evaluation of the Deep Attentional Multimodal Similarity Model (DAMSM) method from AttnGAN [71] to assess its generalization across varying scenarios, and ii) the proposition of a new framework extension designed to handle image generation from text descriptions in multiple languages. The resultant method, named Cross-language Generative Adversarial Network (XLANG-GAN).

While the contributions of this thesis helps to push the boundaries of text-to-image synthesis, there are several promising directions for future research:

- **Scalability to Large-Scale Datasets**: The advances presented were validated on commonly-used benchmarks such as CUB and Oxford-102 datasets. Scaling these methods to more diverse and larger datasets could showcase the utility of the approaches in real-world scenarios.

- **Extending Language Support**: While XLANG-GAN proved effective in handling three languages, incorporating additional languages and exploring the impact of language complexity on generation quality would be the next step in global applicability.
- **Semantic Consistency Across Translations**: Further research could focus on ensuring semantic consistency when descriptions have multiple valid translations, possibly through advanced cross-linguistic embedding methods.
- **Improvements in Disentanglement**: Investigating methods to better disentangle and control individual attributes within generated images based on nuances in textual input may provide better quality and diversity.
- **Integration with Other Modalities**: Future work could involve integrating other modalities such as sound or video cues into the text-to-image generation process, thus increasing the expressiveness and applicability of the generated images.
- **Ethical Considerations and Bias Reduction**: As generative models become more potent, their susceptibility to embedding societal biases becomes a growing concern. Future work must prioritize the development of methods to identify and mitigate biases within generative models.

The aforementioned future research directions highlight the challenges in the domain of text-to-image synthesis, that, despite having great breakthroughs, is in its very early stages. Continuing research in this field holds the promise of unlocking further capabilities of GANs and other generative models, opening the horizons for better textual understanding, multimodal interactions, and creative computer vision applications.

REFERENCES

[1] Arjovsky, M.; Chintala, S.; Bottou, L. "Wasserstein gan", *arXiv preprint*, vol. 1701.07875, Dec 2017, pp. 1–32.

[2] Bengio, Y.; Lamblin, P.; Popovici, D.; Larochelle, H. "Greedy layer-wise training of deep networks". In: Proceedings of the 20th Advances in Neural Information Processing Systems Conference, 2007, pp. 153–160.

[3] Betker, J.; Goh, G.; Jing, L.; Brooks, T.; Wang, J.; Li, L.; Ouyang, L.; Zhuang, J.; Lee, J.; Guo, Y.; et al.. "Improving image generation with better captions", *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, vol. 2–3, 2023, pp. 8.

[4] Brock, A.; Donahue, J.; Simonyan, K. "Large scale gan training for high fidelity natural image synthesis", *arXiv preprint*, vol. 1809.11096, Feb 2019, pp. 1–35.

[5] Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al.. "Language models are few-shot learners", *Advances in neural information processing systems*, vol. 33, 2020, pp. 1877–1901.

[6] Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. "Learning phrase representations using rnn encoder-decoder for statistical machine translation", *arXiv preprint*, vol. 1406.1078, Sep 2014, pp. 1–15.

[7] Choi, Y.; Choi, M.; Kim, M.; Ha, J.-W.; Kim, S.; Choo, J. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation", *arXiv preprint*, vol. 1711.09020, Sep 2017, pp. 1–15.

[8] Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. "Empirical evaluation of gated recurrent neural networks on sequence modeling", *arXiv preprint*, vol. 1412.3555, 2014, pp. 1–9.

[9] Dash, A.; Gamboa, J. C. B.; Ahmed, S.; Liwicki, M.; Afzal, M. Z. "Tac-gan-text conditioned auxiliary classifier generative adversarial network", *arXiv preprint*, vol. 1703.06412, Mar 2017, pp. 1–9.

[10] De Vries, H.; Strub, F.; Mary, J.; Larochelle, H.; Pietquin, O.; Courville, A. C. "Modulating early visual processing by language". In: Proceedings of the 30th Advances in Neural Information Processing Systems Conference, 2017, pp. 6594–6604.

[11] Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. "Imagenet: A large-scale hierarchical image database". In: Proceedings of the 22nd Computer Vision and Pattern Recognition Conference, 2009, pp. 248–255.

[12] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al.. "An image is worth 16x16 words: Transformers for image recognition at scale", *arXiv preprint*, vol. 2010.11929, 2020, pp. 1–22.

[13] Dumoulin, V.; Shlens, J.; Kudlur, M. "A learned representation for artistic style", *arXiv preprint*, vol. 1610.07629, Feb 2017, pp. 1–26.

[14] Faghri, F.; Fleet, D. J.; Kiros, J. R.; Fidler, S. "Vse++: Improving visual-semantic embeddings with hard negatives", *arXiv preprint*, vol. 1707.05612, Jul 2017, pp. 1–14.

[15] Fréchet, M. "Sur la distance de deux lois de probabilité", *Comptes Rendus Hebdomadaires des Seances de L Academie des Sciences*, vol. 244–6, Jan 1957, pp. 689–692.

[16] Frolov, S.; Hinz, T.; Raue, F.; Hees, J.; Dengel, A. "Adversarial text-to-image synthesis: A review", *Neural Networks*, vol. 144, 2021, pp. 187–209.

[17] Goodfellow, I. "Nips 2016 tutorial: Generative adversarial networks", *arXiv preprint*, vol. 1701.00160, Apr 2017, pp. 1–57.

[18] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. "Generative adversarial nets". In: Proceedings of the 27th Advances in Neural Information Processing Systems Conference, 2014, pp. 2672–2680.

[19] Google LLC. "Google translate api". Source: https://cloud.google.com/translate/?hl=en, June 2022.

[20] Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. "Improved training of wasserstein gans", *arXiv preprint*, vol. 1704.00028, Dec 2017, pp. 1–20.

[21] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. "Mask r-cnn". In: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

[22] He, K.; Zhang, X.; Ren, S.; Sun, J. "Deep residual learning for image recognition". In: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition, 2016, pp. 770–778.

[23] Heinzerling, B.; Strube, M. "BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages". In: Proceedings of the 11th International Conference on Language Resources and Evaluation, 2018, pp. 5.

[24] Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: Proceedings of the 30th Advances in Neural Information Processing Systems Conference, 2017, pp. 6629–6640.

[25] Ho, J.; Jain, A.; Abbeel, P. "Denoising diffusion probabilistic models", *Advances in neural information processing systems*, vol. 33, 2020, pp. 6840–6851.

[26] Hochreiter, S.; Schmidhuber, J. "Long short-term memory", *Neural computation*, vol. 9–8, 1997, pp. 1735–1780.

[27] Ioffe, S.; Szegedy, C. "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *arXiv preprint*, vol. 1502.03167, Mar 2015, pp. 1–11.

[28] Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A. A. "Image-to-image translation with conditional adversarial networks", *arXiv preprint*, vol. 1611.07004, Nov 2018, pp. 1–17.

[29] Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. "Progressive growing of gans for improved quality, stability, and variation", *arXiv preprint*, vol. 1710.10196, Feb 2018, pp. 1–26.

[30] Kingma, D.; Ba, J. "Adam: A method for stochastic optimization", *arXiv preprint*, vol. 1412.6980, Jan 2017, pp. 1–15.

[31] Kingma, D. P.; Welling, M. "Auto-encoding variational bayes", *arXiv preprint*, vol. 1312.6114, Dec 2013, pp. 1–14.

[32] Krizhevsky, A.; Hinton, G. "Learning multiple layers of features from tiny images", *Master's Thesis. University of Toronto*, vol. 1–1, Apr 2009, pp. 1–60.

[33] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. "Imagenet classification with deep convolutional neural networks". In: Proceedings of the 25th Advances in Neural Information Processing Systems Conference, 2012, pp. 1097–1105.

[34] LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86–11, 1998, pp. 2278–2324.

[35] Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al.. "Photo-realistic single image super-resolution using a generative adversarial network", *arXiv preprint*, vol. 1609.04802, May 2017, pp. 1–19.

[36] Lee, K.-H.; Chen, X.; Hua, G.; Hu, H.; He, X. "Stacked cross attention for image-text matching", *arXiv preprint*, vol. 1803.08024, Jul 2018, pp. 1–25.

[37] Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L. "Microsoft coco: Common objects in context". In: Proceedings of the 13th European Conference on Computer Vision, 2014, pp. 740–755.

[38] Maaten, L. v. d.; Hinton, G. "Visualizing data using t-sne", *Journal of Machine Learning Research*, vol. 9–Nov, 2008, pp. 2579–2605.

[39] Mao, X.; Li, Q.; Xie, H.; Lau, R. Y.; Wang, Z.; Smolley, S. P. "Least squares generative adversarial networks". In: Proceedings of the 12th International Conference on Computer Vision, 2017, pp. 2813–2821.

[40] Mescheder, L.; Geiger, A.; Nowozin, S. "Which training methods for gans do actually converge?", *arXiv preprint*, vol. 1801.04406, Jul 2018, pp. 1–39.

[41] Mirza, M.; Osindero, S. "Conditional generative adversarial nets", *arXiv preprint*, vol. 1411.1784, Nov 2014, pp. 1–7.

[42] Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. "Spectral normalization for generative adversarial networks", *arXiv preprint*, vol. 1802.05957, Feb 2018, pp. 1–26.

[43] Miyato, T.; Koyama, M. "cgans with projection discriminator", *arXiv preprint*, vol. 1802.05637, Aug 2018, pp. 1–21.

[44] Nilsback, M.-E.; Zisserman, A. "Automated flower classification over a large number of classes". In: Proceedings of the 6th Indian Conference on Computer Vision, Graphics & Image Processing, 2008, pp. 722–729.

[45] Odena, A.; Olah, C.; Shlens, J. "Conditional image synthesis with auxiliary classifier gans". In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 2642–2651.

[46] Odena, A.; Olah, C.; Shlens, J. "Conditional image synthesis with auxiliary classifier gans", *arXiv preprint*, vol. 1610.09585, Jul 2017, pp. 1–14.

[47] Oord, A. v. d.; Vinyals, O.; Kavukcuoglu, K. "Neural discrete representation learning", *arXiv preprint*, vol. 1711.00937, Nov 2017, pp. 1–11.

[48] Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A. A. "Context encoders: Feature learning by inpainting". In: Proceedings of the 29th Computer Vision and Pattern Recognition Conference, 2016, pp. 2536–2544.

[49] Qiao, T.; Zhang, J.; Xu, D.; Tao, D. "Mirrorgan: Learning text-to-image generation by redescription". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1505–1514.

[50] Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al.. "Learning transferable visual models from natural language supervision". In: Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 8748–8763.

[51] Radford, A.; Metz, L.; Chintala, S. "Unsupervised representation learning with deep convolutional generative adversarial networks", *arXiv preprint*, vol. 1511.06434, Jan 2016, pp. 1–16.

[52] Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. "Hierarchical text-conditional image generation with clip latents", *arXiv preprint*, vol. 2204.06125, Apr 2022, pp. 1–27.

[53] Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. "Zero-shot text-to-image generation". In: Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 8821–8831.

[54] Razavi, A.; van den Oord, A.; Vinyals, O. "Generating diverse high-fidelity images with vq-vae-2". In: Proceedings of the 32nd Advances in Neural Information Processing Systems, 2019, pp. 11.

[55] Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; Lee, H. "Generative adversarial text to image synthesis", *arXiv preprint*, vol. 1605.05396, Jun 2016, pp. 1–10.

[56] Reed, S. E.; Akata, Z.; Mohan, S.; Tenka, S.; Schiele, B.; Lee, H. "Learning what and where to draw". In: Proceedings of the 29th Advances in Neural Information Processing Systems Conference, 2016, pp. 217–225.

[57] Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. "High-resolution image synthesis with latent diffusion models". In: Proceedings of the 2022 Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.

[58] Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. "Improved techniques for training gans". In: Proceedings of the 29th Advances in Neural Information Processing Systems Conference, 2016, pp. 2234–2242.

[59] Souza, D. M.; Wehrmann, J.; Ruiz, D. D. "Efficient neural architecture for text-to-image synthesis", *arXiv preprint*, vol. 2004.11437, Apr 2020, pp. 1–8.

[60] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. "Going deeper with convolutions". In: Proceedings of the 2015 Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[61] Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. "Rethinking the inception architecture for computer vision". In: Proceedings of the 29th Computer Vision and Pattern Recognition Conference, 2016, pp. 2818–2826.

[62] Vahdat, A.; Kreis, K. "Improving diffusion models as an alternative to gans, part 1". Source: https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/, April 2022.

[63] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; Polosukhin, I. "Attention is all you need". In: Proceedings of the 30th Advances in Neural Information Processing Systems, 2017, pp. 11.

[64] Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. "The Caltech-UCSD Birds-200-2011 Dataset", Technical Report CNS-TR-2011-001, California Institute of Technology, 2011, pp. 8.

[65] Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Tao, A.; Kautz, J.; Catanzaro, B. "High-resolution image synthesis and semantic manipulation with conditional gans", *arXiv preprint*, vol. 1711.11585, Aug 2018, pp. 1–14.

[66] Wang, X.; Girshick, R.; Gupta, A.; He, K. "Non-local neural networks". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7794–7803.

[67] Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P. "Image quality assessment: from error visibility to structural similarity", *IEEE transactions on image processing*, vol. 13–4, 2004, pp. 600–612.

[68] Wasserstein, L. N. "Markov processes over denumerable products of spaces describing large systems of automata", *Problems of Information Transmission*, vol. 5–3, 1969, pp. 47–52.

[69] Wehrmann, J.; Lopes, M. A.; Souza, D.; Barros, R. "Language-agnostic visual-semantic embeddings". In: Proceedings of the 2019 International Conference on Computer Vision (ICCV), 2019, pp. 5803–5812.

[70] Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. "Show, attend and tell: Neural image caption generation with visual attention". In: Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 2048–2057.

[71] Xu, T.; Zhang, P.; Huang, Q.; Zhang, H.; Gan, Z.; Huang, X.; He, X. "Attngan: Fine-grained text to image generation with attentional generative adversarial networks". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1316–1324.

[72] Yang, J.; Li, C.; Dai, X.; Gao, J. "Focal modulation networks", *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 4203–4217.

[73] Yin, G.; Liu, B.; Sheng, L.; Yu, N.; Wang, X.; Shao, J. "Semantics disentangling for text-to-image generation". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2327–2336.

[74] Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. "Self-attention generative adversarial networks", *arXiv preprint*, vol. 1805.08318, Jun 2019, pp. 1–10.

[75] Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D. "Stackgan++: Realistic image synthesis with stacked generative adversarial networks", *arXiv preprint*, vol. 1710.10916, Jun 2018, pp. 1–16.

[76] Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D. N. "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks". In: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5907–5915.

[77] Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; Wang, O. "The unreasonable effectiveness of deep features as a perceptual metric". In: Proceedings of the 2018 Conference on Computer Vision and Pattern Recognition, 2018, pp. 586–595.

[78] Zhang, Z.; Xie, Y.; Yang, L. "Photographic text-to-image synthesis with a hierarchically-nested adversarial network". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6199–6208.

[79] Zhu, J.-Y.; Krähenbühl, P.; Shechtman, E.; Efros, A. A. "Generative visual manipulation on the natural image manifold". In: Proceedings of 14th European Conference on Computer Vision, 2016, pp. 597–613.

[80] Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A. A. "Unpaired image-to-image translation using cycle-consistent adversarial networks", *arXiv preprint*, vol. 1703.10593, Nov 2018, pp. 1–18.

[81] Zhu, M.; Pan, P.; Chen, W.; Yang, Y. "Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5802–5810.

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Pesquisa e Pós-Graduação
Av. Ipiranga, 6681 – Prédio 1 – Térreo
Porto Alegre – RS – Brasil
Fone: (51) 3320-3513
E-mail: propesq@pucrs.br
Site: www.pucrs.br