Journal of Critical Reviews

ISSN- 2394-5125 Vol 7, Issue 8, 2020

GENERATIVE ADVERSARIAL TRAINING AND ITS UTILIZATION FOR TEXT TO IMAGE GENERATION: A SURVEY AND ANALYSIS

Pranjal Jain[1], Tanmay Jayaswal[2]

[1,2]**Department of Computer Science and Technology,**
SRM Institute of Science and Technology, Chennai, Tamil Nadu, India
E-mail: [1]jainpranjal125@gmail.com, [2]jayaswalt@gmail.com

Received: 20.05.2020 ***Revised: 17.06.2020*** ***Accepted: 06.07.2020***

Abstract
Generative models are being utilized to synthesize realistic data in a highly versatile manner. Training neural networks to generate the image with specific conditions enables us to make an image we require just by giving input of natural language. This paper aims to understand how generative models based on adversarial training works, how images are generated using the various techniques, and how the process has evolved. The architecture utilized is called Generative Adversarial Network that uses game theory to train a neural network to generate images that are indistinguishable from the real images. The paper will also analyse and compare the various techniques and highlight the advantages and drawbacks of each.

Keywords--- Generative adversarial networks, GANs, Text-to-Image generation

DOI: http://dx.doi.org/10.31838/jcr.07.08.290

INTRODUCTION

Generating Images from Descriptive text is a popular problem that can have a variety of applications in almost all industries. Recent innovations in Generative Adversarial Networks has made it a very popular methodology for generating realistic data. Ian Goodfellow's Generative Adversarial Networks [1] is a higher form of neural networks that employs multiple neural networks and contests them against each other to give a rendered images with remarkably high accuracy and output which is very close to being indistinguishable. By applying game theory we can explain their working. There is a neural network called a generator that makes images and a discriminator that classifies the image as being authentic or not. When contested with each other, initially the generator will get it wrong and generates random images, but eventually, after training for several epochs we can see it starts to learn what the target actually should look like, with each successive iterations, the generator will learn to get better and better till the point the discriminator won't be able to distinguish the authenticity and at this moment we have achieved convergence, where we render authentic images reliably.

This architecture can be modified to generate images based on a condition [7], such as encoded textual data. The use of descriptive text to generate images has wide use cases from art generation, architectural designs in CAD applications, which eases the work of the developer by simply giving instructions of what we want to render. There are various types of neural networks that can be utilized. With the versatility of Deep Learning networks and the remarkably high accuracy of GANs, we can implement "Text to Image" architecture. Architectures like Stack-GANS [4] [5] AttnGANs [6] are specifically designed to render high-resolution images from text-based conditional input.

Though training the model to generate images from descriptive words is ventured upon and implemented, we still haven't perfected it. Various architectures have surfaced over a short period, each with impressive results, but still many don't work with perfection. Each new paper improves upon the old techniques and solves the inadequacies using creative solutions. We see the architecture become better and better with each paper.

SURVEY

Generative Adversarial Networks [1]

This paper discusses the original concept of Generative Adversarial Networks and how using game theory we can train two neural networks against each other to get significantly high accuracy for outputs that are specifically non-generalized. One of the neural networks is called a generator whose task is to generate the target and another network called Discriminator that classifies the image as fake or real. The generator network renders an image that the discriminator classifies as authentic or inauthentic. Initially, the generator has very low accuracy, but after successive epochs of training, the generator becomes proficient in rendering the target. The paper discusses the architecture and loss functions to help train the model efficiently and further discusses the various advantages of the system like the high individuality of the outputs. Various implementations are elucidated in the paper and how GANs have excelled in them as compared to the traditional methods. The disadvantages of the architecture are also presented like the highly unstable nature of training or various types of recognition problems where the architecture cannot decipher the attributes of the features and their frequency and spatial features. These errors are Loss of dimension where the images rendering the target are flattened; the Loss of Frequency: where the images generated the same features multiple times or less than the usual, for example, a dog image with 6 legs instead of 4. The solution to all these inconsistencies is the extensive training of the discriminator network and using different algorithms. Various other problems are also presented with solutions like the mode collapse and non-convergence, where the generator never understands how to generate the target.

This architecture is very powerful as the generator is never trained on the training data so there is less chance of overfitting. Since the generator network is only exposed to the decisions of the discriminator, the generator eventually understands how to generate the image and how the various features of the data should be generated. This makes the work of this particular generative model highly unique. The generator also properly generates the various features of target output which might be lost in the Variable Auto Encoderor mean squared cost functions

as described in [1]. That is what makes GANs a very suitable tool for generative tasks.

Conditional Generative Adversarial Nets [7]
The original Generative adversarial networks define a highly versatile deep learning mechanism that utilizes game theory to get accurate output, but since it requires a noise input which causes its output to be always random; in use cases where we require a specific result it becomes useless. This paper focuses on this problem and the implementation of a new system that solves it. This system is the modified version of the original GAN paper's architecture [1] that uses a condition as an input to generate the output. The conditional GANs are made so that we can construct data based on some input and hence develop target output that is not entirely random. This paper showcases the implementation of a generation of handwritten digits that look highly realistic from a single input of a digit corresponding to it. The dataset used is the MNIST dataset [11] for Handwritten Digits with class labels, on a 28X28 pixel resolution and gray scale compositions. The architecture used is two fully connected neural networks that then contest it against each other via a min-max function, once there is a convergence we can get target images generated. With the help of the input, we can generate images of things we want to generate rather than a random image. The input in the MNIST example is that of a single-digit input and the output of the corresponding image of the digit in a 28x28 pixel.

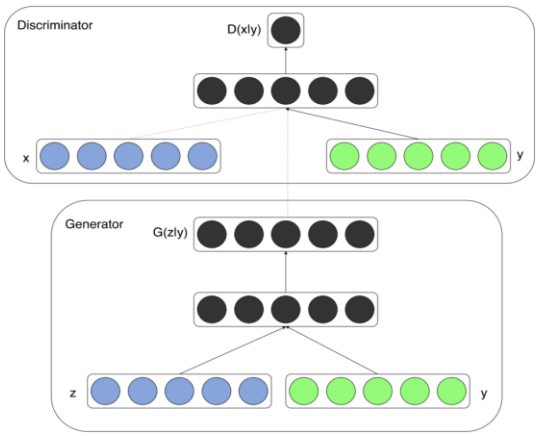

Figure 1. This is the architecture diagram for the conditional GANs,source[7] that depicts the inner working of how it generates the image. The Generator and discriminator are similar to the original GAN architecture but the exception of using a definite user given input alongside the noise vector. The architecture is trained along with the input and a solution is acquired.

The paper is a very crucial method as it is the first step in using the generative models to build a specific output. Though the methodology can be adapted for a variety of use cases, the problem with the network is that it still uses a primitive network and primitive inputs only, and generates a very primitive image with limited resolution. The architecture has been modified and improved upon to approach the text to image synthesis problem.

Progressive Growing of GANs for Improved Quality, Stability, and Variation [2]
This paper elucidates a newer novel method to generate high resolution and a highly detailed image. This was previously not possible due to hardware technology constraints and large training time. The newer method uses a different approach to train the model, rather than training the entire network at once, the network is initially trained on a low resolution of 4x4 dataset (shrunk down from the initial one) and then train the next layer to generate images of higher resolution i.e. 8x8. The network is then progressively grown to generate higher resolution images with high fidelity. The network is trained individually at each layer to render the high resolution until it finally renders a high-quality image. The resolution of the image at each stage is doubled giving a logarithmic increase in resolution. The discriminator and the generator are grown together for higher resolution and helps in training more stably. This is a very efficient method as it is faster than previous ways and yields an authentic image of much higher resolution.

The architecture uses a Reverse Convolutional Neural Network and outputs an image of the desired size. To increase the size of the image we have to grow the system as well and train it on the dataset accordingly.

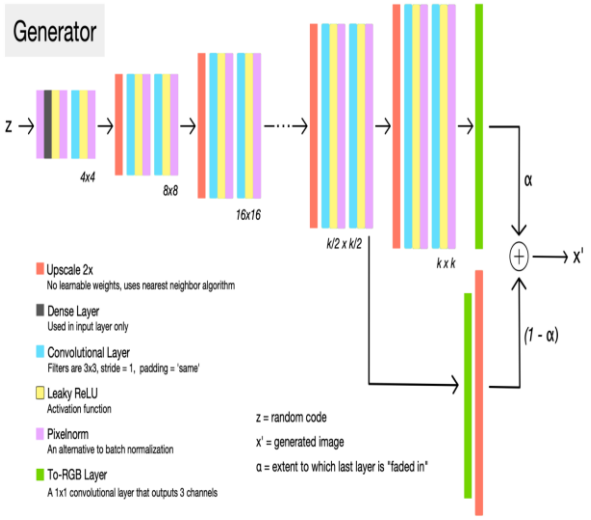

Figure 2. This is the architectural schematic diagram of the Progressive GAN, source[2]. We can see that at each stage the image size is doubled and the image is made more detailed and feature full at each stage. In the end, the output of the last layer is converted into an image of (kxk) size *with 3 channels for Colour RGB*

Due to this, the GAN takes less time to train and becomes more accurate than previous methods of generating images. The architecture also doesn't require repetitive training but rather a more consistent approach for training as we have to train at each stage. The technique still demands a heavy utilization of GPU and computing resources and takes days or weeks to train on. The dataset used is a CIFAR10 dataset [12] and CELEBA dataset[13].

Learning What and Where to Draw [8]
The GAWWN or "*Generative adversarial What and Where Network*" is a generative network that is utilized for text to Image purposes. It is used to generate 128x128 images using the Caltech UCSD Birds Dataset [14]. The system uses the text descriptions and converts it into an image with captions visualized. The system takes a conditional approach inspired by the conditional GANs[2] and implements a bounding box strategy where it takes the caption and converts it into a word embedding which is subsequently replicated to form a feature map. This is then subjected to convolution functions to form features. The generator then branches into two processes, a global and a local. The local stage processes the image subsection inside the bounding box and the global works on the entire image. The final stage is to combine the results of both using depth concatenation.

The Generator and discriminator are trained via the image datasets: Caltech UCSD Birds Dataset [14]and MPII Human Pose[15]. The generator generates the image and the

discriminator identifies whether it is fake or real, using this the generator changes the functioning and then tries to become better till it successfully deceives the discriminator and renders a very accurate image. This is an innovative approach as the generator doesn't directly render the image, but does two separate processes and combines them. This helps in breaking down the task and gives highly detailed and overall coherent features. This helps in rendering multiple elements rather than one, like in conditional GANs. The use of embedding instead of direct text input also makes the system more reliable as the generator doesn't have to understand Natural Language processing and also forgo any problems caused by it like semantic context understanding.

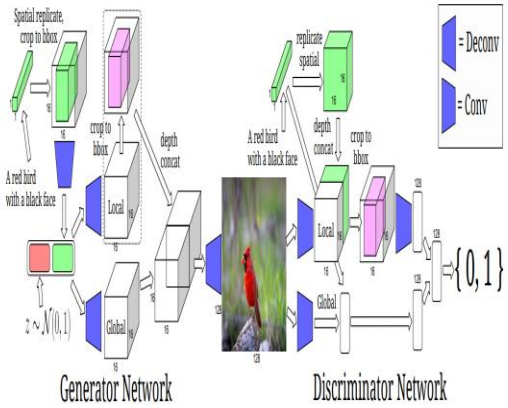

Figure 3. This image depicts the architecture for GAWWN, source[8]. The system uses natural language as input and then converts them into word embedding. It is then given as input to the generator that converts it into a local and global context feature vector. The local form is converted to match the global form and combined. The image is then generated. The GAN is trained using this form

The system is very effective in achieving text to image with proper feature generations and ease of training but still has many shortcomings like structural defects and the limited resolution. The subject of focus is rendered properly but its surroundings are blurred or generic.

Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space [16]

The Plug and Play Generative Adversarial Networks or PPGNs is a modification on Conditional GANs which can generate all 100 categories of the image Net dataset [17] with a high resolution of 227x227. This is done by using a generator trained on the large dataset for generating 100 categories and the use of a separate conditional network that augments and instructs the generator in rendering the image. This system increases the output capability by generating not one but many scenes and categories and also a higher resolution than 32x32 with structural coherence. The system also renders images with more diversity in the features with more detailing on words depicted features. This is done by using a probability frame for energy-based models. This helps as we can use multiple conditions as priors for generating an image. We can "plug" in the energy sums and hence generate images with high accuracy.

TAC-GAN - Text Conditioned Auxiliary Classifier Generative Adversarial Network[3]

Text Conditioned Auxiliary Classifier Generative adversarial Network or TAC-GAN is a GAN that is the modified version of AC-GAN [10], where it is conditioned to generate images that visualize the description of the caption. The novel methodology of TAC-GAN was inspired by the idea to use class formation and variegate the sample as it resulted in increased coherence of the image and the various elements generated were much more detailed. The architecture works by converting the caption into a text feature embedding and which is then fed into the generator. The generator takes this embedding and a noise vector to render images of variety. The Generator and discriminator are then trained via adversarial training on the oxford 102 dataset of flowers [19], and once convergence is achieved we can render high-quality 128x128 images.

The TAC-GAN is one of the initial papers to combat the structural coherence problem of text to an image problem and improves upon the existing technologies. It still does not successfully eliminate the defect and we can see improvement in the technologies inspired by this method like the Stack GAN [4][5]. The TAC-GAN also suffers from a variety of problems. It can only generate one or few items successfully but when exposed to render multiple items it fails to do so due to a more concentrated focus on rendering single objects in the entire frame.

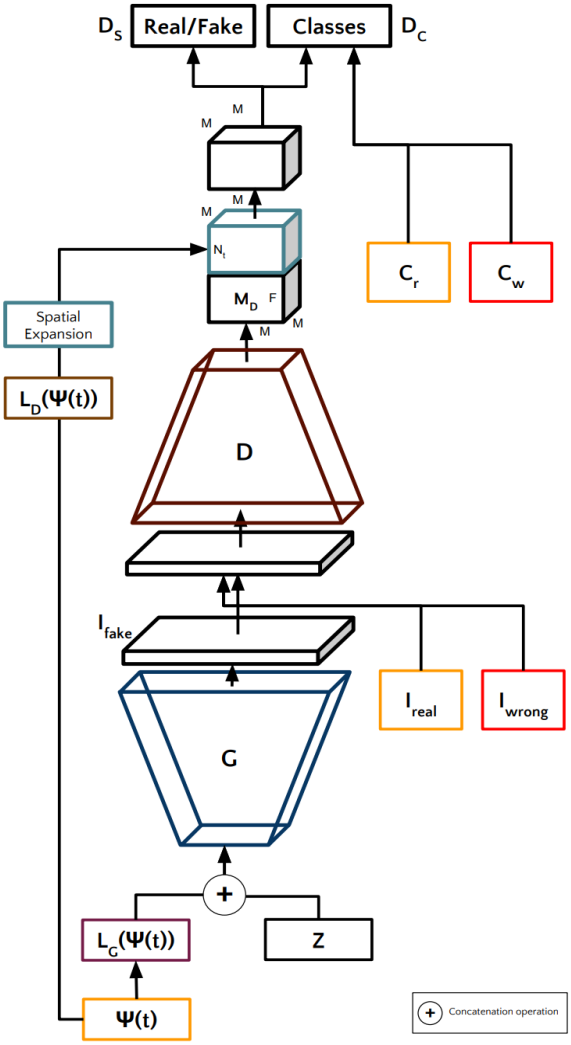

Figure 4. The architecture diagram depicts the workings of TAC-GAN, source[3]. The input is a Natural Language Caption that is then converted into a text embedding and fed to the generator, which then renders an image. This image is then passed to the discriminator that then determines if it is fake or real

Stack GAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks [4]

To implement Descriptive text to Image, we utilize a new and innovative technology of Stack GAN that converts input text to image in two stages consisting of two GANs in a succession to one another. The caption-description is first converted into a word embedding which is then passed via a conditional augmentation. Then the first stage is the initial GAN network that converts this embedding into the corresponding image, the rendered image is a 64X64 pixel. Here the quality is extremely low and the image is a low resolution of 64x64 pixels. The stage 1 network is trained to develop images with features, where detailing is not necessary but rather the feature generation is focused upon. Even though the image is of low quality, in the results we can notice there are lots of features, especially the ones that are required as per the description. The second stage takes the stage 1 GAN output and converts it into a high quality and detailed image of a large resolution via up sampling locks. This is done by another GAN network, where the generator takes in input image of low-quality images and outputs high-quality images with detailed features and a resolution of 256x256 pixels. The discriminator is trained in a way to ensure the information is not lost or mistranslated into an incorrect feature. The System was trained on multiple datasets: MS COCO [18], Caltech UCSD Birds Dataset [14], and LSUN [20].

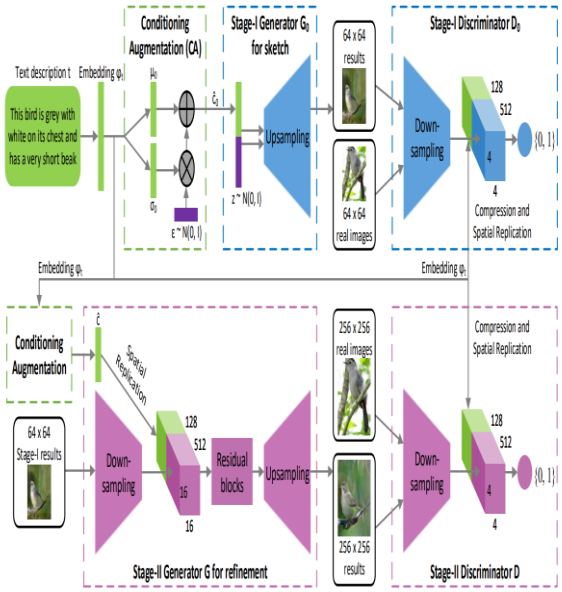

Figure 5. The architectural diagram for Stack-GAN source [4].

This architecture is much easier to train as we have divided the work of rendering the image with two generators, this helps reduce the time retraining and eases hyper-parameter tuning. Even with this modification, the system has several drawbacks relating to structural coherence. The images generated still had various extra features generated like multiple beaks of the birds or like generated windows in the ceiling (these examples are trivial and there could be multiple defects that exist in the image). These structural defects exist due to a lack of understanding of elements the word is describing and how to render them amongst other objects with full structural coherence.

Stack GAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks[5]

The Stack GAN Version 2 or Stack Gan++ [5] is a modification over the first version of the Stack GAN [4], improving upon it and increasing the inception score with higher structural coherence. The new version discards the multi-stage network, which was not able to inter-communicate between the stages, with a higher-order network that utilizes all the previous layer's output to build its output. The StackGAN-v2 has the option of adding pairs of Generators and Discriminators to generate higher resolution images as seen in the Progressive GAN methodology [2]. The idea of using local and global divergence helps in building higher-order relationships among the various objects that improve the rendering of the image sub-portions. The architecture works by taking the text description and converting it into word embedding. Then the word embedding is passed via the various Generator and Discriminator pairs. Each pair up-samples the image by 4 times and adds details and features. The previous pairs also pass information to the higher pairs. The training of such a complex system is done by training all the pairs of Generators and discriminators together in each step. The image then gives an output of the text description with high accuracy.

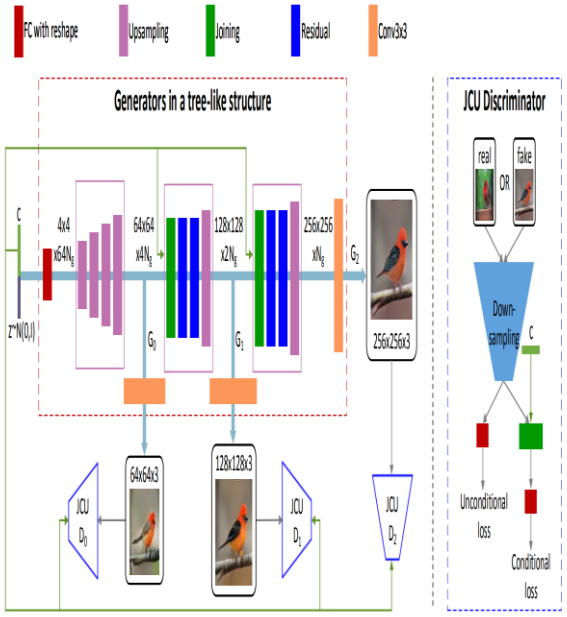

Figure 6. The following is the architecture for Stack-GAN version 2, source [5]. We can see that it utilizes a JCU discriminator and a tree-like structure for generators by fully connecting the various stages of the GANs into one large system that intercommunicates.

The architecture is very versatile where we observe a higher level of accuracy than the previous version. The structural coherence is increased but the defects are noteliminated. A significant increase in the inception score is also seen. The architecture is trained on the Caltech UCSD Birds Dataset [14], MS COCO [18], Oxford 102 [19], LSUN [20], and ImageNet[17] giving high accuracy for all.

Chat Painter: Improving Text to Image Generation using Dialogue [21]

With previous Text to Image conversion strategies, the GAN was provided with an input caption and an image was given as output depicting the caption. This method was straightforward but didn't yield the best results most of the time as the words were not properly interpreted into an image. This paper focuses on a new strategy where it takes a caption and a dialogue to depict the image. The idea was inspired by a Police sketch artist that converses with the witness to depict the scene via dialogue and asking questions to give a solution. This is an innovative method as we can provide the generator with additional information

when it is not able to properly render the scene and that too with increased structural coherence.

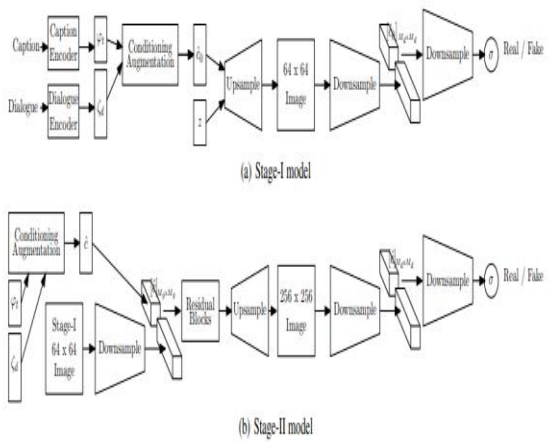

Figure 7.The following is the architecture for Chat Painter, source [21]. The use of the two-stage GAN setup is to get higher resolution. Stage 1 generates a 64x64 feature-rich image and stage 2 generates a 256x256 image conditioned on the stage 1 output.

The paper discusses the dialogue system that it uses called VisDial Dialogues along with a captioning dataset (MS COCO)[18]. The main GAN architecture is a modification of the Stack-GAN architecture and we see that it gives much better images with increased coherence.

Attn GAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks[6]

This paper bases on the Stack-GAN architecture [4] [5] and modifies it by adding Attention Network to the generators of the system. This converts the architecture from encoder-decoder pairs to a more versatile sequence to sequence conversion of text to image. With the help of the attentional network, the generator can use the word and sentence formations more accurately by focusing on the generation of image's sub-sections according to the words and what it is describing in the image. This helps in rendering an image that is more accurate and coherent than the Stack-GAN [4] implementation with better detailing.

The architecture of the Attn-GAN starts by taking the input description of text that needs to be converted to an image. This is then passed via a bidirectional text encoder with a concatenation of hidden states for both forward and backward directions. Using the encoder sentence feature matrix and word feature vectors are generated. Next, the architecture is mixed with noise which is used for creating a unique output for the generator, the generator then outputs a context feature vector.

The feature is then up-sampled by a factor of 2 without the use of a word-level feature vector, giving a 64x64 image. The resultant context is then sent to the attention network which combines the previous context feature with the word feature vector.

This is done by bringing the word feature to a common space and generating a word–context vector region which associates image subsections to the words and develops features more accurately. The subsequent generators are used to render images by upscaling the images by 2 times via a Res-block and word feature vector. The architecture also supports the idea of using "n" number of generators (with more generators we can achieve a higher resolution) by adding generators one after the other.

The subsequent generators and attention networks are all similar to the first ones. The paper though only uses two sets due to TPU restraints. The generators are trained using the discriminators via the adversarial training. The dataset used is the Caltech UCSD Birds Dataset [14] and the Coco Dataset [18]which both contain the image and its corresponding captions that describe the image.

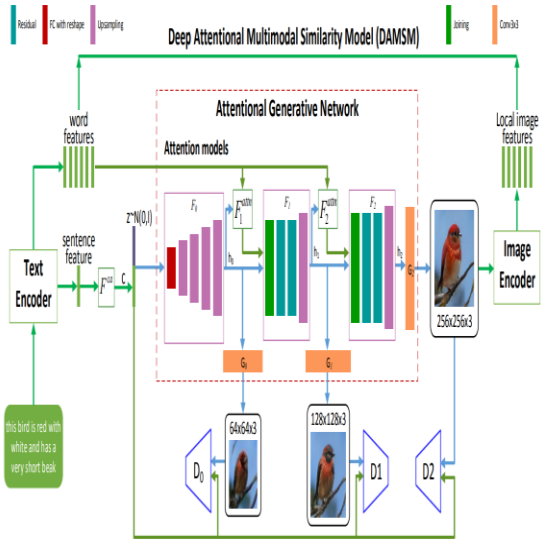

Figure 8. The following depicts the architecture of Attn-GAN, source [6]. The architecture is a modification of the Stack-GAN-V2 and uses an additional DAMSM metric system to determine the consistencies of the image objects and caption's intended depiction. The use of a word and sentence feature to generate an image is very useful as we get high accuracy from it.

The Image generated is evaluated using a separate system called the Deep Attentional Multi-modal Similarity model (or DAMSM) where the image and its descriptions are used to find the posterior probability and a loss. The evaluation is done via an Inception score (for detailing) and an R-Precision (for structure coherence) to find the overall success of the system.

The system is very powerful and gives a better image output with more coherence and less failure than the previous systems. The utilization of the attention network helps in accurately creating sub-portions of the image using the words and maintains the overall coherence. The system utilizes the TPU and GPU heavily and takes lots of time to train on.

DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image Synthesis[22]

This paper focuses on the generation of images from Natural Language via a multi-stage refinement process using a dynamic memory module. In the previous methodologies, the final image is based on the quality of the initial image, if this initial image is not properly rendered i.e. the features are not generated satisfactorily or the frequency of the features is incorrect or the placement of features is wrong, the refinement process can hardly make any difference to refine it to make high quality and accurate image.

The word representation also becomes ineffective in a multistage system as each word has a different level of information depiction, the words like "a", "the" and "and" don't have the same significance such as "bird", "red" and "grass". Previous models use all words at the same level information depicting the scene.

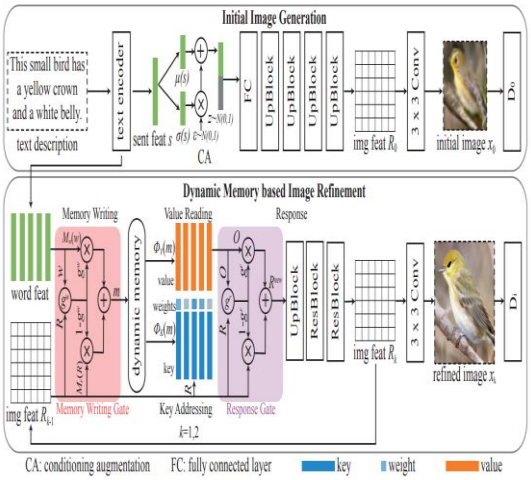

Figure 9. The following depicts the architecture of DM-GAN, source [22]. It uses the dynamic memory module to manipulate and increase the feature quality as the initial image is not able to render it properly. This memory writing is used to add value to the features

To solve the problems in multistage GAN systems, this paper focuses on the utilization of a memory module that can be used to enhance the incorrectly generated initial images that hinder the refinement process. The fuzzy features (the features we wish to enhance to make them more suitable for refinement) are read as queries from the memory module and are used by the refinement stages of the system as separate input for rendering much better features as a replacement.

The architecture also includes a memory writing gate to use words that are more important and depict more information in the caption to generate features properly. This helps in generating images that are much better and have more structural coherence. The system helps in making efficient multistage systems that can have more of an impact on accuracy and coherence.

Semantic Object Accuracy for Generative Text-to-Image Synthesis

Semantic Object Accuracy Generative Text to Image synthesis is a novel method that approaches the idea of using a much more reliable metric called the Semantic Object Accuracy instead of the traditional approach of using inception score and Fréchet Inception Distance (FID) [23]. The reason for using a newer metric is that the traditional metric relies upon a pre-trained Convolutional neural network to classify the image's detailing and its structural coherence.

This is not a very powerful or accurate metric as it won't properly evaluate the objects and the image. The inception score measures two factors: firstly, if the object is rendered and has all the features that make it recognizable and second, whether the model can generate an object of many types and classes. This is done using the network's output results. The FID which uses activations of the layer's output and the real image. If the difference is small: the image is considered of high quality. Though these metrics help find the accuracy of an image's detailing, it doesn't take into account the description of the image, hence there is no understanding whether the image is actually what we want it to be and whether the objects we intend to render are properly rendered. The metric just verifies if the image has elements that can properly signify visual features then the metric is high even if it doesn't have the required object in the caption.

The R-precision in attn GAN paper [6] comes farther in expressing the similarity of the caption and what it is depicting with the generated image but still fails in a plethora of ways. The R precision calculates if the caption is properly implemented, the individual objects, however, are not subjected to quality constraints. The metric uses many captions to identify foreground objects and if they do not properly identify then the metric goes down irrespective of the object.

The SOA metric improves upon all the previous ones by utilizing an object detector (YOLO v3) [24] to verify if the objects in the caption are rendered properly or not. They classify all the objects in the dataset and then calculate how the specified object was generated using the detector and evaluate a metric to it.

This was used in the experiment and yielded results that were previously not seen with other systems, this helped in making the system more thorough and giving a better result. The idea of using a metric that evaluates the subjects of the image helps in giving a more accurate image and overall the generators can properly amend their activations for faster training.

Realistic Image Generation using Region-phrase Attention[25]

Using Generative Adversarial Networks and the attention systems have been useful in increasing the structural coherence and the overall quality of the image. The attention is applied over a regular grid-based system and renders elements of the subject in focus described by the natural language caption. But due to the usage of homogenous grids over the image, the elements in the foreground don't get the needed attention and loose many of the desired features, this is also due to the complexity of the natural language semantics and the grid attention layer.

This paper proposes a novel solution to this problem by utilization of the "true grids" or grids that are based on the location extracted from the word phrases and where they indicate the element of the features to be generated. This modification was done on the Attn-GAN [6] system by adjusting the DAMSM loss metric with the true-grid strategy and yielded a better foreground renders of elements when generated.

The subjects in the foreground were having more coherence and more clarity. With increased complexity, the images were depicting the scene better than the Attn-GAN with proper distinguishable features.

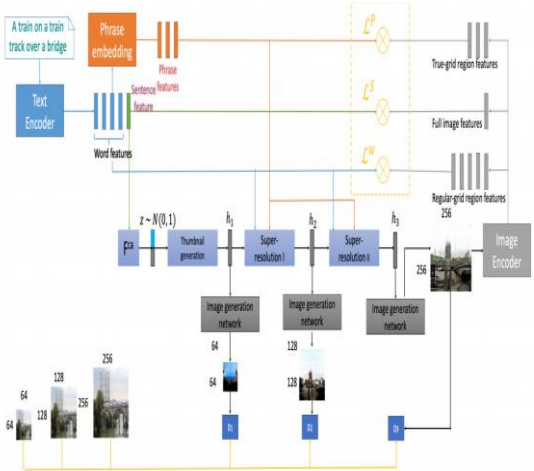

Figure 10. The following depicts the architecture, source [25]. It is based on Attn-GAN and has a grid adjusting mechanism.

Controllable Text-to-Image Generation [26]
This paper proposes a novel architecture for Text to an image called control-GAN which can be used to generate images and "control" parts of the image via the captions. This uses the caption's word-level context and attention driven mechanisms to render the features of the objects by disentangling the different object features and focuses on using the attention layer to render the various image elements.

This helps by making the system very stable as compared to other GANs that are very unstable and have no control over the generation of features. The use of this attention driven mechanism is very efficient and specific to our needs. In previous versions, if the user changes one word of the caption, the resultant image would be drastically different and this instability is very undesirable as the system becomes random and the use cases reduce for the developer.

The image's elements are controllable in this methodology as it allows users to direct the system via Natural Language to modify features like colour and textures.

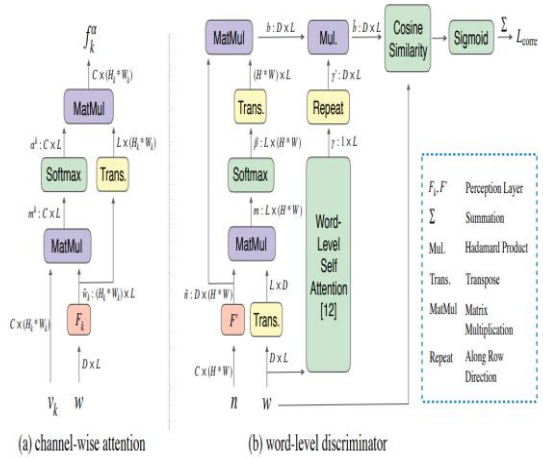

Figure 11. The following depicts the architecture of controllable-GAN, source [26]

Object-driven Text-to-Image Synthesis via Adversarial Training [27]
Object-Driven Attention Generative Adversarial Networks or Obj-GAN is the modification on the two-stage generation process similar to Stack GAN, but add a novel object driven attentive module for image generation that can be used to generate objects in focus with great accuracy.

This is done by giving attention to the most important words in the caption and the initially generated layout in the first stage. In previous methods, the system used to take the entire sentence vector which is converted from Natural Language. Since Natural Language is ambiguous and complex, the systems miss the important information that could be useful to generate the image with structural coherence.

Though Attn-GAN [6] uses attention for object generation it only proves efficient for simple datasets like for birds and flowers. They fail to map and synthesize the complex relation of the multi-Object depiction which is seen in the COCO dataset. The system fails due to the inability to understand the complex relationship between the scenes and objects and hence results in a blurred background.

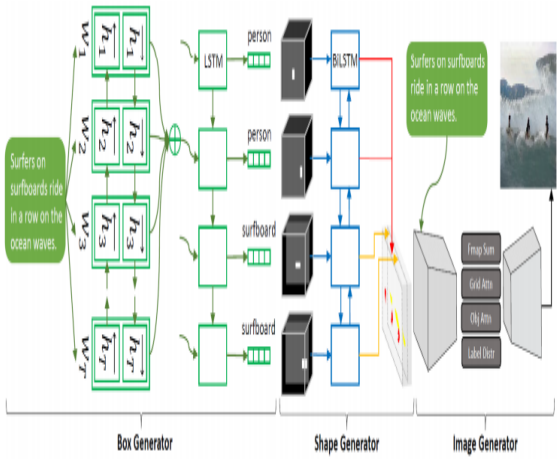

Figure 12. The following depicts the architecture of Obj-GAN, source [27].

The architecture proposed for Obj-GAN solves the problems by capturing the fine-grained word/object level depiction of information and utilizing the features efficiently to properly determine the scene we require as the target. The system takes the input text and a pre-generated layout via a step by step process. The image is generated via a bounding box method that focuses on one object inside the box via an attention network, utilizing the context of the sentence and the words to generate an image.

Each stage performs on the previous layer's output and generates a very high-quality output until we get our target quality or resolution. The system also uses a "Fast reverse Convolutional Neural network" for each bounding box to discriminate the image's quality.

COMPARISON

Model	**IS↑** [28]	**FID↓** [23]	**R-Precision↑** *(k =1)*[6]	**SOA-C↑** [9]
Original Image	34.88 ± 0.01	6.09 ± 0.05	68.58 ± 0.08	74.97
Obj-GAN	24.09 ± 0.28	36.52 ± 0.13	87.84 ± 0.08	27.14
OP-GAN	28.57 ± 0.17	**26.65 ± 0.09**	87.90 ± 0.26	33.11
DM-GAN	**32.32 ± 0.23**	27.34 ± 0.11	**91.87 ± 0.28**	**33.44**
Attn-GAN	23.61 ± 0.21	33.10 ± 0.11	83.80	25.88
Stack-GAN	8.45 ± .03	74.05	N/A	N/A
Stack-GAN++	8.30 ± .10	81.59	N/A	N/A
GAWWN*	3.62 ± .07	N/A	N/A	N/A
PPGN*	9.58 ± 0.21	N/A	N/A	N/A
ChatPainter*	9.74	N/A	N/A	N/A
[26]*	24.06	N/A	N/A	N/A

Table 1: This table elucidates the scoring of various generative models discussed in this paper based on the four most popular metrics utilized to find the efficiency of the image they generate. The metrics used are Inception Score [28] (The higher the better or ↑), FID [23] (The lower the better or ↓), R-Precision [6] (The

higher the better or ↑), SOA-C [9] (The higher the better or ↑). These are the following models that were evaluated on the MS-COCO [18] to make it standardized and relative. This was done so that they are tested on the same data and all its properties, though some models work better on different datasets like Caltech UCSD Birds Dataset [14] for Stack-GAN [4][5].

Another reason for it is that the MS-COCO dataset [18] has many classes of different objects, unlike the others that have only birds or flowers. This was done to test the various systems and their likeliness on generating multifarious objects and retain context. The scores are compared to the original image's metrics as well to evaluate and to signify the disparity in the model's proficiency and the actual image that it should mimic, which in this case is taken from the dataset itself.

The data illustrates that the DM-GAN [22] comes to the closes of all to generate the best image as possible when the caption is provided. The high IS score signifies that it is rendering a very realistic image. The high R-precision and SOA score illustrate the high structural coherence and use-case of the words of the caption to make the best image. It does lack in the FID score, but not a lot from the next best which is the OBJ-GAN

INFERENCE

The survey has elucidated us on the various methodologies and techniques to incorporate Generative Adversarial Networks in the problem of Text to Image generation. All the implementations have their merits and demerits and improve upon the previous implementation. The initial conditional GAN was only able to take single-digit inputs and render the digit. This was improved upon by TAC-GAN implementation that could take text description as input. Even though it could generate only single objects with blurred backgrounds the architecture was better than Conditional GAN's single-digit 28x28 image, TAC-GAN properly understood the various features and how words could be used to render those features and how it can be structurally coherent, GAWWN took it one step further by improving the structural coherence in TAC-GAN which used to fail in multiple subject scenarios. GAWWN implemented the idea of focusing on image sub-portions via the bounding box strategy and generate multiple objects based on words in the caption. The structural coherence was improved but not eliminated, also the images were still low resolution and with inadequate detailing.

Stack-GAN and its updated version Stack-GAN V2 adopted a strategy similar to Progressive GANs, where it generated a low-resolution image and using multiple stages of GANs up-sampling the image with increased detailing and quality. The images were more detailed with very high resolution, but we still had many coherence defects that caused the captions to be inadequately visualized or at times misinterpreted. This was focused on the makings of the newer networks like Attn-GAN and SOA-GAN. Till now all the generative models utilized the Inception Score as the metric for classification of the network's proficiency, this did not properly evaluate the system as the score doesn't take into account the caption and its relation to the image rendered. In Attn-GAN, the metric used was R-precision. This helped reliably relate the caption to the image and scored it to assess the proficiency of the network. SOA-GAN improves by implementing a new evaluating metric called the Semantic Object Accuracy, which uses a convolutional neural network to verify each object of the image.

The use of an attention network or the utilisation of bounding box strategies has been very effective in making objects and elements of the image more precise with more structural coherence as seen in [6][25][26][27].

By this survey, we understand that the metric is as important as the neural network architecture utilized in solving this problem. The newer networks had an increase in the accuracy once the correct metric was implemented. The architecture relatively remained the same with a few minor updates, but since the GAN has a more accurate strategy to evaluate the image, we see training it is much easier than before.

REFERENCES

1. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672–2680.
2. T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in International Conference on Learning Representations, 2018.
3. Dash, Ayushman&Gamboa, John & Ahmed, Sheraz& Afzal, Muhammad Zeshan&Liwicki, Marcus. (2017). TAC-GAN - Text Conditioned Auxiliary Classifier
4. H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in Proceedings of the IEEE International Conference on Coputer Vision, 2017, pp. 5907– 5915.
5. Zhang, Han & Xu, Tao & Li, Hongsheng& Zhang, Shaoting& Wang, Xiaogang& Huang, Xiaolei& Metaxas, Dimitris. (2017). StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 10.1109/TPAMI.2018.2856256.
6. T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1316–1324.
7. M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784, 2014
8. Reed, S. E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., & Lee, H. (2016). Learning what and where to draw. In Advances in neural information processing systems (pp. 217-225).
9. Hinz, Tobias, Stefan Heinrich, and Stefan Wermter. "Semantic Object Accuracy for Generative Text-to-Image Synthesis." arXiv preprint arXiv:1910.13321 (2019).
10. X. Xia, R. Togneri, F. Sohel and D. Huang, "Auxiliary Classifier Generative Adversarial Network With Soft Labels in Imbalanced Acoustic Event Detection," in IEEE Transactions on Multimedia, vol. 21, no. 6, pp. 1359-1371, June 2019, doi: 10.1109/TMM.2018.2879750.
11. LeCun, Y. & Cortes, C. (2010), 'MNIST handwritten digit database', http://yann.lecun.com/exdb/mnist/
12. Alex Krizhevsky and Vinod Nair and Geoffrey Hinton. "-10 (Canadian Institute for Advanced Research" http://www.cs.toronto.edu/~kriz/cifar.html
13. Liu, Z., Luo, P., Wang, X., & Tang, X. (2018). Large-scale celebfaces attributes (celeba) dataset. Retrieved August, 15, 2018.) http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
14. Wah, C., Branson, S., Welinder, P., Perona, P., &Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset. http://www.vision.caltech.edu/visipedia/CUB-200.html
15. Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014). 2d human pose estimation: New benchmark and state of the art analysis. In Proceedings of the IEEE Conference on computer Vision and Pattern Recognition (pp. 3686-3693).
16. Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A., &Yosinski, J. (2017). Plug & play generative networks: Conditional iterative generation of images in latent space. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4467-4477).
17. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., &Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.
18. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ...&Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.
19. Nilsback, M-E. and Zisserman, A., 2008 Oxford 102 Flowers
20. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., & Xiao, J. (2015). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365.
21. Sharma, S., Suhubdy, D., Michalski, V., Kahou, S. E., &Bengio, Y. (2018). Chatpainter: Improving text to image generation using dialogue. arXiv preprint arXiv:1802.08216.
22. Zhu, M., Pan, P., Chen, W., & Yang, Y. (2019). Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5802-5810).
23. Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S., &Klambauer, G. (2018). FréchetChemblNet Distance: A metric for generative models for molecules. arXiv preprint arXiv:1803.09518.
24. Redmon, J., &Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
25. Huang, W., Xu, Y., &Oppermann, I. (2019). Realistic image generation using region-phrase attention. arXiv preprint arXiv:1902.05395.
26. Li, B., Qi, X., Lukasiewicz, T., &Torr, P. (2019). Controllable text-to-image generation. In Advances in Neural Information Processing Systems (pp. 2063-2073).
27. Li, W., Zhang, P., Zhang, L., Huang, Q., He, X., Lyu, S., &Gao, J. (2019). Object-driven text-to-image synthesis via adversarial training. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 12174-12182).
28. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In Advances in neural information processing systems (pp. 2234-2242).