Synthesizing Comics via Conditional Generative Adversarial Networks

Darwin Burkard Morris
Delft University of Technology
Supervised by Prof. Lydia Chen, Dr. Zilong Zhao

***Abstract*—The creation of comic illustrations is a complex artistic process resulting in a wide variety of styles, each unique to the artist. Conditional image synthesis refers to the generation of *de novo* images based on certain preconditions. Applying machine learning to conditionally generate novel comics proves an intriguing yet difficult task. This paper aims to answer whether Generative Adversarial Networks (GANs) can be used for conditional comic synthesis. Recent advancements in Generative Adversarial Networks have increased the capability of image synthesis to hyper-realistic levels. Despite this, the performance of GAN models is almost always assessed on photo-realistic images. To extend experimental knowledge of unconditional GAN performance into the domain of comics, an empirical analysis was performed on the unconditioned generative performance of three cutting edge GAN architectures: Deep Convolutional GAN (DCGAN), Wasserstein GAN (WGAN), and Stability GAN (SGAN). This paper showed that the SGAN implementation far outperforms both the DCGAN and WGAN architectures on a dataset of *Dilbert* comics, achieving an FID score of 89.1. Due to their relative simplicity, comics provide an intriguing candidate for conditional generation. A comic panel can likely be described using a few specific labels (eg. background and characters). Two conditional networks were created, using the SGAN architecture as a baseline. Multi Class SGAN (MC-SGAN) used a traditional multi-class conditional approach while the Multi Label SGAN (ML-SGAN) utilized a multi-label auxiliary classification approach. Multiple experiments were performed between these two networks resulting in hundreds of hours of training. While performance between the networks was quite similar on simple conditional tasks, on more complex tasks MC-SGAN outperformed ML-SGAN. MC-SGAN was able to conditionally generate comics based on character and color, with desired conditions distinguishable in almost all outputs. Issues with traditional methods of auxiliary classifier training in the MC-SGAN implementation are additionally identified and discussed.**

I. Introduction

Comics represent a highly expressive form of visual storytelling. The illustrations cover a myriad of artistic styles, each extremely unique to the artist. What if one could guide the extension of their favorite series, the characters, and colors being in their control while the artistic style is matched automatically to the illustrator? The generation of novel comics using machine learning techniques proves an interesting yet difficult task. Recently, there have been marked advancements in the quality and success of image synthesis techniques, although most of the existing research pertains to photo-realistic images [1]. Extending the field of conditional image synthesis, the controlled generation of novel images, to comics serves not only to assess the boundaries of current technology but extend the state-of-the-art into a completely new domain.

Generative Adversarial Networks (GANs) excel at learning to reproduce real-world data distributions [2]. Novel image generation is one of the foremost applications of the GAN [3]. All GAN implementations build off the framework presented by Goodfellow et al. in which two networks: the generator and discriminator, compete in a zero-sum game [4]. In this game the generator acts as a counterfeiter, trying to create data that matches the original distribution, while the discriminator network tries to distinguish between artificial and real data.

Due to GANs notoriously unstable nature, numerous studies have sought to identify the architecture and loss functions that are most conducive to convergence. The first step towards a generalizable GAN was the introduction of the Deep Convolutional GAN (DCGAN) [5]. This model used deep convolutional networks in its architecture, resulting in the ability to synthesized high-resolution images in a wide array of applications. Despite its general success, the DCGAN had issues with vanishing gradients, a problem defined by a lack of gradient presented by the discriminator network. This was addressed by the Wasserstein GAN (WGAN), which replaced the cross entropy loss of DCGAN with Wasserstein distance, a loss function with superior theoretical properties for convergence [6]. The WGAN left the underlying architecture of the DCGAN unchanged. A gradient penalty was later introduced that further improved stability, this network was named WGAN-GP [7].

Mescheder et al. later published a paper on the convergence of GANs, presenting an improved network that they claimed had better convergence properties than WGAN [2]. Mescheders network, Stability GAN (SGAN), utilized ResNet architecture alongside their R_1 regularizer to achieve more stable training and superior image quality. DCGAN, WGAN, and SGAN represent the state-of-the-art of image generating GANs and provide a basis for the unconditional generations of comics.

While the default GAN frameworks produce *de novo* images at random, it would be advantageous to influence

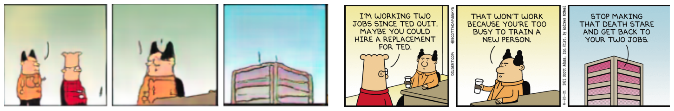

I'M WORKING TWO JOBS SINCE TED QUIT. MAYBE YOU COULD HIRE A REPLACEMENT FOR TED.
@SCOTTADAMSSAYS
DILBERT.COM
THAT WON'T WORK BECAUSE YOU'RE TOO BUSY TO TRAIN A NEW PERSON.
6-26-21 ©2021 Scott Adams, Inc./Dist. by Andrews McMeel
STOP MAKING THAT DEATH STARE AND GET BACK TO YOUR TWO JOBS.

Fig. 1: The image on the left is a *Dilbert* styled comic strip composed of images generated using SGAN. The comic strip on the left is an example of an original *Dilbert* comic.

generation based on a set of conditions. Conditional GANs allow for output to be influenced by class labels. The most naive implementation of GAN conditioning involves adding class-specific embeddings to the input of the generator, which has been shown to produce favorable results in simple classification problems such as MNIST [8]. When data and classes become more numerous and complex it is useful to add an auxiliary classifier to the network [9]. Auxiliary classification is the act of expanding the task of the discriminator to not only predict the source of the image but also the class to which it belongs.

Comic generation presents a compelling application of conditional generative networks for multiple reasons. The difference in terms of structure and simplicity between comics and photo images is vast; it has yet to be determined if the performance of established GAN models is comparable between the two. Additionally, due to the simplicity of comics, a multi-label case can describe its contents in relative entirety. Even given just the two conditions of characters and background, one could create an output that, along with dialogue, could tell a story. This creates a research opportunity to further experimental understanding while providing a practical application.

The aim of this paper is to answer the following question: Can Generative Adversarial Networks be used for conditional comic synthesis? In order to answer the research question the paper must answer the following sub questions: (i) Can DCGAN, WGAN-GP, or SGAN synthesise images of a quality conducive to conditional generation (ii) Can a multi-label and multi-class version of the best performing architecture conditionally synthesize comics? (iii) Is there an advantage to one conditional architecture over the other?

The paper is structured as follows with the contributions highlighted: Section II discuses the related work. Section III provides an overview of an empirical analysis study aimed at comparing results of the previously mentioned architectures on unconditional comic synthesis (**contribution I**). Section IV describes the methodology of the creation and analysis of a conditional GAN for comic synthesis (**contribution II**). Section V discusses the experimental setup, while the results and discussion can be found in section VI (**contribution III**). Section VI reflects on the ethical implications, reproduciblity and integrity of the research. Section VIII provides a conclusion of the research as well as suggestions for future work.

Network:	Architecture:	Loss:	Regularization:
DCGAN	DCNN	BCE	None
WGAN-GP	DCNN	Wasserstein Loss	Gradient Penalty
SGAN	ResNet	BCE	$R1$

TABLE I: The differing architectures, loss functions, and regularization techniques used in the DCAN, WGAN-GP, and SGAN implementations.

II. Related Work

The following related works sections will compare the defining differences of the Deep Convolutional GAN, Wasserstein GAN, and Stability GAN as they relate to unconditional generation. Techniques of conditional generation such Auxiliary Conditional-GAN will also be discussed.

A. Non-conditional GAN

The original GAN implementation used an RNN architecture for both the generator and discriminator, allowing for impressive results on simple data sets such as MNIST. Despite its success on such tasks, the original GAN implementation was not largely scalable [4].

1) DCGAN: The Deep Convolutional Generative Adversarial Network replaced the previously fully connected layers of the original RNN with deep convolutional layers. The use of strided convolutions, batchnorm, and specified activation functions for the generator and discriminator resulted in the first truly stable GAN using a deep convolutional network. The modifications allowed for the successful generation of images from a much more complex distribution. DCGAN remains a benchmark for GAN performance on the task of image synthesis [5].

GAN training involves a minmax objective between the generator and the discriminator networks. In the case of the DCGAN the game is as follows [10]:

$$\underset{G}{min}\underset{D}{max}\, V(D,G) = E_{x\sim Pdata(x)}[logD(x)] + \\ E_{z\sim p_z}[log(1-D(G(z)))] \tag{1}$$

In this equation the discriminator (D) is trying to maximize the log of its output when classifying data from the original distribution and minimize its output when classifying data that is produced by the generator (G) (this is represented as the term $log(1 - D(G(z)))$. Network G on the other hand is trying to minimize the same equation. $Pdata(x)$ Represents the distribution of real data while p_z represents the distribution of data from the latent space.

2) WGAN-GP: GANs are notoriously unstable when training. In the case of DCGAN it often occurs that the Discriminator performs too well at the task of identifying whether an image is from the original distribution or not, especially early in training [11]. This results in a problem called vanishing gradients, in which the generator no longer has relevant gradients on which to propagate. Wasserstein-GAN (WGAN) attempts to solve this problem by using Wasserstein Loss [6]. Wasserstein loss is defined as follows:

$$\underset{G}{min}\underset{D\epsilon\beta}{max}\, V(D,G) = E_{x\sim Pdata(x)}[D(x)] + E_{z\sim p_z}[D(G(z))] \tag{2}$$

In Wasserstein Loss β is a set of 1-Lipschitz continuous functions. Rather than classifying output as real or fake, D minimizes the Wasserstein Distance, $W(q,p)$, with respect to the generator parameters $W(Pdata, p_z)$. The most stable version of this technique uses gradient penalty to assure Lipschitz Continuity, this network is named Wasserstein GAN with Gradient Penalty or WGAN-GP. WGAN-GP provides a training environment that has been shown to converge in most cases [7].

3) Stability-GAN: More recently Mescheder et al. stated in a paper on the convergence of GANs that WGAN-GP does not always converge. In order to assure convergence they introduced a new $R1$ regularization technique along with a ResNet architecture for both the generator and discriminator. They called their findings Stability-GAN. Stability-GAN was able to produce impressive results on Imagenet and CelebA-HQ datasets. [2].

DCGAN, WGAN-GP, and SGAN represent a relatively comprehensive set of state-of-the-art GANs for image generation. Previous research has has noted that the results of image synthesis given a specific GAN architecture are highly dependant on the training data [12]. In short, there is no way to predict how these networks will perform on comic data without an analytic comparison of their results. The emperical analysis of DCGAN, WGAN-GP, and Stability-GAN is presented in section III.

B. Conditional GAN

There are many methods of conditioning GANs. The simplest form of network conditioning involves attaching a class conditional to the latent vector for use as input to the generator. This usually takes the form of concatenating an embedding of a one-hot-vector representing the class to the latent input [8]. While this works in simple cases, its efficacy drops significantly during more complex conditional generation tasks [13].

Auxiliary Conditional GAN or ACGAN uses the help of auxiliary classification in order to improve results and stabilize training during conditional synthesis. In ACGAN the discriminator not only gives a probability distribution over the source of the image (X_{Real} or X_{Fake}) but also over the corresponding class of the image (c). This necessitates a two part loss function:

$$\begin{aligned} L_S = E[logP(S \\ = real|X_{real})] + E[logP(S \\ = fake|X_{fake})] \end{aligned} \tag{3}$$

$$\begin{aligned} L_C = E[logP(C \\ = c|X_{real})] + E[logP(C \\ = c|X_{fake})] \end{aligned} \tag{4}$$

Where D is attempting to maximize $L_S + L_C$ and G is trying to minimize $L_C - L_S$.

Existing conditional methods focus mainly on generating images within a multi-class problem. The use of these techniques in multi-label conditional generation remains to be assessed. Section IV and VI explore this question.

III. Empirical Analysis of Non-Conditional GANs

In this section, the non-conditional generation of comics is compared across DCGAN, WGAN-GP, and SGAN architectures. This allows for a comparison of performance between state-of-the-art image generating networks on comics. This empirical analysis is necessary to provide a basis for the choice of network architecture used in the conditional network. Each network varies in its architecture, loss, and regularization method, these differences are presented in TABLE I. Full descriptions of architecture and training methods are available in section V.

The analysis focuses on the ability of the selected networks to produce images that are conducive to conditional generation. Background color and character presence were selected as necessary observable conditions, due to their pertinence to further experiments. Additional insight into these choices is presented in section IV. Three categories of output quality were defined for assessment: no conditions distinguishable, conditions are distinguishable but not identifiable, conditions are both distinguishable and identifiable. Networks were also qualitatively ranked based on image quality. The final results of this analysis are presented in TABLE II. This analysis is additionally intended to determine what size of image is necessary for the conditional network. Comparisons were determined qualitatively based on the level of identifiability of conditions within the results.

A. DCGAN

The Deep Convolutional GAN was trained on the *Dilbert* data in both 64 x 64 and 128 x 128 configurations. The documented issues with training stability in DCGAN were immediately apparent. Multiple training attempts were necessary in order to avoid the effects of vanishing gradients in the 64 x 64 configuration. On the trials where convergence was reached, conditional presence in the output was assessed. Final results showed clearly evident background color, individual characters were distinguishable but not easily identifiable.

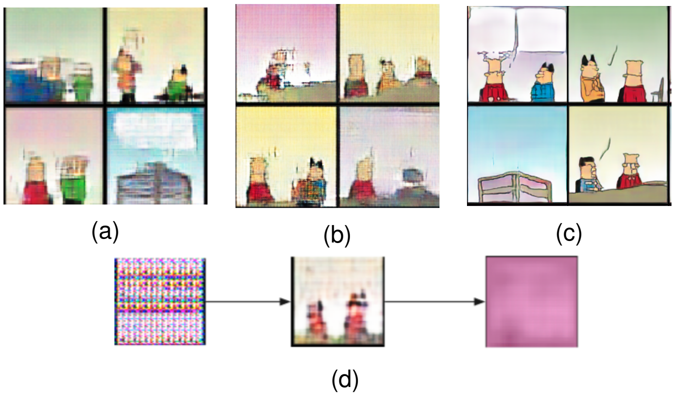


Fig. 2: 2a, 2b, and 2c represent the output from WGAN 64 x 64, WGAN 128 x 128 and SGAN 128 x 128 trials respectively. 2a depicts the output of DCGAN 128 x 128 at 5, 20, and 60 epochs. The results of vanishing gradients can clearly be seen.

The previously identified issues with vanishing gradients became an increasing issue in the 128 x 128 image size. The model continually failed to converge even after multiple trials. The resulting images of DCGAN in its 128 x 128 configuration were extremely noisy, leaving conditions neither distinguishable nor identifiable. Examples of vanishing gradients during DCGAN training can be seen in *Fig* 2d.

B. WGAN-GP

The findings of previous literature stating that WGAN-GP fixes many of the instabilities in DCGAN training were confirmed. WGAN-GP had no issues with vanishing gradients in both trials. The 64 x 64 implementation resulted in clearly defined comics that accurately represented the *Dilbert* distribution. Results from this trial can be seen in *Fig* 2a Conditions were both distinguishable and identifiable. Despite this, the size of the image limited the clarity of the characters, meaning only certain identifying characteristics were present. At this point is was determined that 128 x 128 resolution was necessary for conditional synthesis. Due to this SGAN trials were only run in the larger image size.
The 128 x 128 configuration of WGAN-GP had limited improvement in terms of image quality over the 64 x 64 implementation. It appeared that the training was stable, although the model converged to sub-optimal results. Limited clarity of characters was achieved. These results can be seen side by side in *Fig.* 2.

C. SGAN

SGAN had extremely impressive results when generating 128 x 128 images. Training was extremely stable, no issues of vanishing gradients were experienced. The network generated images where conditions were clearly distinguishable and identifiable, this can be seen in *Fig.* 2b. Qualitative analysis of results from SGAN determined that it produced images of far superior quality to both DCGAN and WGAN-GP.

Network	64 x 64	128 x 128	Rank
DCGAN	🟨	🟥	3rd
WGAN-GP	🟩	🟨	2nd
SGAN	⬜	🟩	1st

TABLE II: Results from empirical analysis of DCGAN, WGAN-GP, and SGAN. Red: no conditions distinguishable. Yellow: conditions are distinguishable but not identifiable. Green: conditions are both distinguishable and identifiable. Grey: generation not conducted

D. Analysis

The non-conditional experiments run on DCGAN, WGAN-GP, and SGAN architectures demonstrated the importance of stability in GAN training. It also highlighted the superiority of SGAN's $R1$ regularizer and ResNet architecture in the task of comic generation. It was evident from the results that the SGAN architecture would form a good basis for the creation of a conditional GAN for comic generation. It was also determined that the 128 x 128 size would be necessary in order to provide good results in a conditional implementation, as clarity is limited by pixel count in the 64 x 64 image size.

IV. Methodology

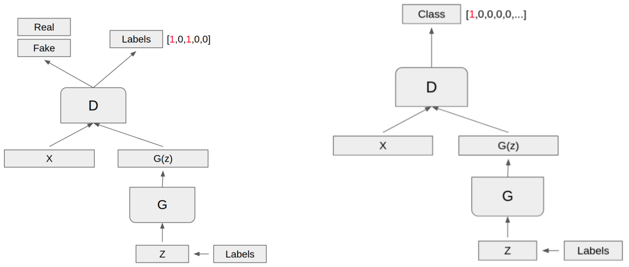
This section will cover the method by which the conditional generation of comics from the *Dilbert* dataset using GANs was performed and assessed. It will cover the chosen conditions, created data-sets, and architectures as well as an in-depth analysis of why they were chosen. A brief description of the experiments is also included.

Conditional image synthesis is an extension of image synthesis. An empirical analysis was performed in section III in order to determine the architecture that created the highest quality images without the input of conditions. This analysis found that the SGAN implementation synthesized superior images to both DCGAN and WGAN-GP in the domain of comics. This resulted in the determination that the ResNet based SGAN architecture would be used as a basis for the architecture of the proposed conditional GANs.

A. Conditional Networks

The majority of existing c-GAN and ACGAN architectures aim to solve multi-class classification problems, each produced image aims to fall into the category of one of many classes. Despite this, conditional comic generation fits better into the category of multi-label classification. In multi-label classification problems, the network tries to learn the distribution over a set of independent labels [14]. In the case of character conditioning, each label would correspond to the presence of a certain character in the panel.

Multi-label problems can be converted to multi-class problems by means of label powerset (LP) transformation [15]. LP transformation attributes each possible grouping of labels to a



(a) ML-SGAN

(b) MC-SGAN

Fig. 3: In 3a and 3b D represents the discriminator network and G represents the generator network while Z represents the latent space vector input to the generator. An embedding of input labels is joined to C in both models. In ML-SGAN D performs multi-label classification on source and label while in MC-SGAN D outputs the probability over class labels.

class, for example the labels A and B might be converted to the classes represented by [0 0], [1 0], [0 1], and [1,1]. The LP transformation implicitly accounts for label dependence. Limitations to this method exist due to an exponentially increasing number of classes necessary to represent a given set of labels. This method allowed us to use existing multi-class conditional GAN architectures on an inherently multi-label problem.

B. Multi-Class Stability-GAN (MC-SGAN)

The multi-class implementation of SGAN modifies the discriminator by adding a single fully connected output layer with one node for each desired class. The output of the final layer is passed through a sigmoid function to ensure output is between zero and one for each node while maintaining independence between classes.

In order to condition the generator network, the class conditional is one-hot encoded and embedded in a 128-bit input vector. This 128-bit vector is then concatenated onto the 128-bit latent noise vector for input to the network.

The loss for the generator and discriminator are calculated by performing binary-cross-entropy (BCE) loss on the output node that corresponds to the desired class. The labels used for performing BCE loss are either one for real or zero for fake. The discriminator network tries to minimize the loss on the correct source. The generator network tries to minimize the loss that its output is classified as real.

This conditioning strategy was employed due to its success on the 1000 class ImageNet dataset in previous studies [2]. A depiction of the architecture can be referenced in *Fig.* 3.

C. Multi-Label Auxiliary SGAN (ML-SGAN)

Due to the lack of scalability provided by the LP transformation method, we created a GAN architecture that could be conditioned purely on multi-label data. The multi-class implementation took advantage of the fact that a single node of the output layer could be used to calculate loss based on source, this was no longer possible with multi-label data. An auxiliary classifier was added to the network to make the necessary separation between loss based on source and loss based on label.

The Multi-Label Auxiliary SGAN (ML-SGAN) preserves the underlying ResNet architecture of MC-SGAN while adding a multi-label auxiliary classifier. The multi-label model allows for multi-label conditional generation without the requirement of performing an LP transformation. The lack of LP transformation in this model greatly reduces the complexity of the network on the same set of labels when compared to the MC-SGAN.

In MLA-SGAN the loss is determined using an auxiliary classification technique [9]. Two fully connected output layers are added to the ResNet discriminator architecture found in S-GAN. One of the output layers performs the task of binary classification of the source, either from the original distribution or a generated image. The second output layer has a node for each corresponding label. The sigmoid function is applied to both output layers and loss is calculated by summing the binary cross entropy across the layers for the generator and discriminator. The discriminator and generator follow the loss proposed in Eq. 3 to determine loss based on source and the proposed loss in Eq. 4 to determine loss base on label. The method of conditional embedding to the generator remains unchanged from MC-SGAN. The general architecture can be referenced in *Fig.* 3.

D. Conditions

Comics provide a compelling application for conditional synthesis as their simplicity allows for a finite number of labels to provide a relatively complete description of their content. It was determined that the most discernible and indicative labels of comics were background and character presence. With these two conditions it becomes possible to generate panels that contribute to a coherent story. Datasets were created for the conditions of both background color and character presence.

The original idea for this paper was to use the dataset of *phdcomics*. While there were sufficiently many examples of *phdcomic* panels, the style varied greatly from comic strip to comic strip and from year to year, additionally the panels were in many different shapes and sizes making generation impossible. After further researcher it was determined that *Dilbert* provided ideal attributes to use in this study. The *Dilbert* comics remain stylistically consistent, they are available on the web, and they are all of the same size. 5000 *Dilbert* comics were scraped from the web for use unconditionally. An automated process was created in order to remove text from the panels in order to reduce noise.

Pre-processing was performed on this set of comics in order to create datasets for conditional generation.

E. Experiments

A set of experiments were designed in order to assess the capability of both ML-SGAN and MC-SGAN to conditionally synthesise images based on the previously mentioned conditions. Experiments were to include a simple multi-class experiment, as well as a series of multi-label experiments to provide a comparison between ML-SGAN and MC-SGAN. Experiments were designed in order to allow for comparison at varying levels of conditional complexity. The experiments were designed in a manner of incrementally increasing complexity in order to allow comparison when a point of failure was reached.

1) Color as a Condition: Due to its easy distinguishability within panels, color was thought to represent a relatively simple classification task. Experimental analysis of color as a condition would provide insight into the performance of ML-SGAN on a simple single label classification task. It would additionally provide a simple multi-class problem for MC-SGAN in order to assess architectural functionality. It was hypothesized that the results between both networks on this task would be very similar.

2) Multi-Label Experiments: Two multi-label experiments of varying complexity were designed. The first experiment involved the conditional generation of comics based on the presence of The Boss, Dilbert, both, or neither. This created a 4 class task for MC-SGAN and a 2 label task for ML-SGAN. The distribution of data between classes was relatively equal in this experiment. Results from this study would determine if conditional generation based on character presence was possible in the comics domain.

The second multi-label experiment involved the four most common characters in the *Dilbert* dataset: Dilbert, The Boss, Wolly, and Alice. These four characters comprised the maximum set of characters for which at least one example existed in each class of the LP transformation. 14 total classes were created for input to MC-SGAN (the set including all and no characters was excluded). The resulting distribution of examples through these classes varied greatly with sets having as few as 21 and as many as 200 examples. This experiment provided a rigorous comparison between MC-SGAN and ML-SGAN on a complex conditional generation task.

V. Experimental Setup

This section will focus on the networks, environment, datasets and evaluation metrics used in experiments throughout this paper. This section is additionally applicable to the work conducted in section III.

A. Environment

All the networks were trained on Google Cloud VM instances. Training was performed on the NVIDIA Tesla K80 GPU. NVIDIA's CUDA parallel computing platform was used during model training. Training time for the networks ranged from 24-96 hours and 150 cloud credits were used over the course of all experiments. The open source PyTorch v1.9 library was used in the implementation of all networks.

B. Networks

1) DCGAN and WGAN-GP: The DCGAN and WGAN-GP implementation are based off the respective papers by Gulrajani et al. and Radford et al. [7][5]. Both papers provided an architecture for generation of 64 x 64 images. The hyper parameters, training setup, and network architectures were followed exactly as they are laid out in the papers.

For generation of 128 x 128 images an additional convolutional layer was added to the discriminator with a stride of 2, padding of 1, and kernal size of 4. The generator was additional modified by adding a deconvolutional layer with the same parameters.

The details of the differences in loss functions between the networks is presented in section II.

2) SGAN, MC-SGAN, ML-SGAN: SGAN, MC-SGAN, and ML-SGAN share the same base ResNet architecture and hyperparameters. Training, hyperparameters, and architecture remains consistent with the setup described by Meschder et al. in their experiment on the CelebA dataset [2]. Implementation details of the conditioning technique used in MC-SGAN and ML-SGAN are presented in section IV.

C. Datasets

1) Unconditional Generation: That dataset used in unconditional generation included 5000 dilbert panels scraped from *Dilbert.com*. Text was cleared from all panels.

2) Color as Condition: The original set of 5000 *Dilbert* images was processed in order to identify all panels that contained solid background colors. Panels were automatically labeled with background color in the CIELAB color space. CIELAB expresses colors in a 3d space where distance is equivalent to the difference in human perception [16]. K-Means clustering was used in order to identify classifications of colors based on their perceived similarity [17]. The panels were labeled with the most commonly found background colors in the original dataset: green, yellow, and purple. There were 623, 740, and 637 examples respectively.

3) Character as Condition: The characters in a set of 2000 *Dilbert* Dilbert comics were manually labeled. Panels with non-recurring characters or with more than 3 characters present were removed in order to simplify the data. This resulted in a dataset of 1444 comics labelled with

Network	Color	Two-Character	Four-Character
MC-SGAN	100%	96%	84%
ML-SGAN	100%	92%	63.1%

TABLE III: This table highlights the accuracy of MC-SGAN and ML-SGAN across color, two-character, and four-character conditions. Accuracy represents the visibility of conditions across all samples.

corresponding characters. Datasets were further filtered and class labels were attached for both the two-character and four-character experiments.

D. *Evaluation Metrics*

Quantitative evaluation of results was determined using the Frechet Inception Distance (FID) [18]. FID represents the Wasserstein distance between two multidimensional Gaussian distributions, in this case, the generated and real images. This method improved over the earlier technique of the Inception Score (IS), which only evaluated the quality of generated images [19]. Using FID over IS is especially significant in this study as the generated comics are not meant to be photorealistic. Conditional generation will be evaluated based on the appearance of the specified condition in network output. 25 samples will be generated for each class and label presence on those samples will determine accuracy.

$$FID(r,g) = ||\mu_r - \mu_g||_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}) \quad (5)$$

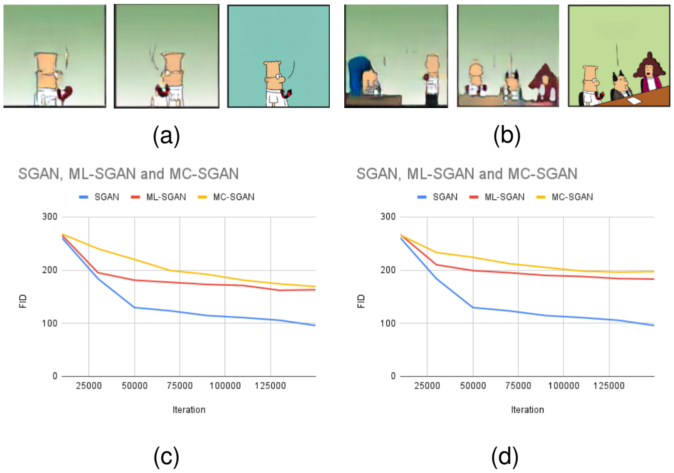
The loss of the generator and discriminator were recorded during training to gain insight into the training stability. The FID was also recorded over time in order to gauge the networks convergence.

VI. Results and Discussion

This section will present the results of the experiments designed to assess ML-SGAN and MC-SGAN on conditional generation of Dilbert comics. It will also provide discussion on the results. All networks were run for 125,000 iterations for the following experiments. Accuracy Results for all experiments can be seen in Table TABLE III.

A. *Color as Condition*

Conditional generation of comics based on color proved to be a simple classification task. Both ML-SGAN and MC-SGAN performed perfectly in this task with the correctly conditioned color being present in all generated images. Additionally, when images were generated using a latent vector that was identical, except for the conditional embedding, the produced images only varied in background color. This showed a great ability for both ML-SGAN and MC-SGAN to conditionally generate comics based on color. This result was expected and provided a reference for increasingly more complex conditional generation tasks.



(a) (b)

(c) (d)

Fig. 4: 4a depicts the results from the two-character experiment. From left to right: the output of ML-SGAN with Dilbert label, the output of MC-SGAN with Dilbert label, example of Dilbert from the original dataset. 4b depicts results from the four character experiment when the networks were conditioned to produce images containing Dilbert, The Boss and Alice. Results are presented in the same order. In this case you can see mis-generated image from ML-SGAN as it only contains The Boss and Dilbert. 4c and 4d depict FID over iteration for both character experiments, the FID of SGAN on unconditional data is provided for reference.

B. *Two-Character*

Both MC-SGAN and ML-SGAN performed exceedingly well at the task of generating images conditioned on the presence of the Boss and Dilbert. The Boss was easily identifiable by his distinctive pointy hair. Differentiation between Wolly and Dilbert in generated images was slightly more difficult as they both wear white shirts and have similar skin tones. Both networks seemed to pick up the difference in tie color between the two, allowing for distinction.

The accuracy of character presence between networks was similarly high as shown in TABLE III. The majority of incorrect outputs occurred due to Dilbert being interchanged for Wolly. This is likely due to their previously stated similarities. The FID scores were also similar between the networks, although both networks performed worse than the unconditional SGAN. The discrepancy in performance can almost certainly be explained by the difference in volume of training data. SGAN was trained on 5000 images while the training set for ML-SGAN and MC-SGAN contained only 1444 examples.

C. *Four-Character*

Greater discrepancies were seen between the networks in the four character classification task. MC-SGAN performed exceptionally well in this task with 84% classification accuracy across all 14 classes. ML-SGAN performed significantly worse in this task with 63.1% classification accuracy across all classes. This is thought to be due to

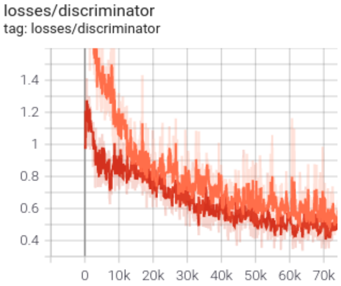


Fig. 5: Discriminator loss of MC-SGAN and ML-SGAN on the four-character experiment. ML-SGAN is in orange and MC-SGAN is in red. The collapse of the auxiliary classifier can be seen around 20k iterations.

a collapse in the auxiliary classifier which is discussed in the following sub-section. Despite research stating that increasing the number of conditions on given data is likely to increase image quality, the opposite effect was actually noted [9]. The FID score for both networks was slightly lower than in the two character trial, this is presented in *Fig.* 2. This could also be attributed to changes in training data.

These results showed the impressive ability of LP transformation to convert a multi-label problem into a multi-class problem. Even on classes with relatively few examples the generator was able to learn to produce images that accurately matched the conditional output. The MC-SGAN architecture proved extremely effective in conditional comic synthesis.

D. *Auxiliary Classifier Collapse*

During the four-character experiment it became apparent that the multi-label auxiliary classifier in ML-SGAN stopped providing meaningful loss to the generator network during training. This was identifiable by analysis of discriminator loss during training. This is visible in *Fig.* 5. After around 20k iterations the loss for the auxiliary classifier on both real and fake data was nearly zero, while the loss from the source classifier reduced much more gradually. This meant that after 20k iterations the generator backpropagation was being performed only on loss from the binary source classifier and not the multi-label auxiliary classifier. This resulted in a stagnation of conditional learning after around 20k iterations in the ML-SGAN.

The cause of this collapse can be attributed to the training setup proposed by Odena et al. in the original paper on ACGANs [9]. The proposed training steps dictate that the weights of the auxiliary classifier be updated by the loss generated both on the fake and real data. Updating the weights on the loss from the real data allows the network to learn to identify the distribution of labels in the real data, eg. it can identify when Dilbert is in an image. This is an advantageous feature of training as it provides meaningful loss on the basis of labels to the generator. Backpropagating loss based on the classification of fake data results in the auxiliary classifier learning a bi-modal distribution for each label, one mode being the real representation and the other being the generators representation at that point in training. This learned bi-modal distribution results in a lack of meaningful loss from the auxiliary classifier during the generator training step.

VII. Responsible Research

The quality and value of research are dictated in large part by their integrity and reproducibility. The results in this paper represent a high level of integrity and transparency. The information in this paper also provides sufficient background and information to make the results fully reproducible.

A. *Integrity*

Integrity denotes the absence of data fabrication, falsification, data trimming, and conflict of interest. Due to the nature of GANs, their resulting output contains a great deal of randomness. When looking at individual results from the network, it becomes easy to skew the picture of its overall quality. Thus, evaluation metrics have been calculated based on a large volume of outputs, creating a holistic picture of performance across the entire distribution. When data was filtered for classification tasks the results were also clearly dictated. It is important to note that the produced comics are for research purposes only and therefore hold no commercial incentive. Under the copyright of *dilbert.com* the use of material for non-commercial purposes is permitted.

B. *Reproducibility*

The difficulties associated with reproducibility within Artificial Intelligence require thorough attention. The first step towards reproducibility in this study is assuring clarity around the data used in training. While it is often possible to use pre-existing data sets, this study required that unique data be gathered and processed. The data that was used is thoroughly documented and can be recreated based on the method description. Next, the architecture and hyper-parameters of the networks were provided. Open source solutions were additionally always referenced. The training cycle was also clearly stated. The results are therefore able to be reproduced as they appear in the paper.

VIII. Conclusion

The purpose of this report was to examine the efficacy of conditionally generating comics using Generative Adversarial Networks. DCGAN, WGAN-GP, and SGAN were assessed on the task of unconditional comic generation in order determine which architecture had superior performance. The result of this empirical analysis guided the creation of both a multi-label and multi-class network for conditional comic synthesis. These networks were then compared across a set of experiments using varying input conditions.

The SGAN created higher quality images than both the DCGAN and WGAN-GP implementations. The results from

SGAN were extremely impressive, clearly representing the style of *Dibert* comics. The SGAN based ML-SGAN and MC-SGAN networks were both successful in conditionally generating comics. They both performed equally well in the generation of comics based on the condition of color as well as the simple two-character generation task. The MC-SGAN implementation outperformed the ML-SGAN during the more complex four-character task, achieving 84% accuracy on condition presence. The discrepancy in performance was likely the result of the auxiliary classifier providing insufficient loss to the generator during training. Further discussion on the collapse of the auxiliary classifier can be found in section VI. It can be concluded from this work that Generative Adversarial Networks provide a viable method for conditional comic synthesis.

In future work it would be advantageous to resolve the training problems of the auxiliary classifier in MC-SGAN. Due to the exponential growth of the label powerset transformation, multi-label classification proves the only viable solution when a large number of labels is needed in generation. This could be fixed by implementing an auxiliary classifier that was pre-trained on real data [9]. In experimentation, transfer learning using VGG16 performed exceptionally on the task of label classification of characters in *Dilbert* comic panels [20]. Using a pre-trained auxiliary classifier likely has the ability to create a more representative loss for the generator, thus vastly improving results.

References

[1] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[2] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for gans do actually converge?" in *International conference on machine learning*. PMLR, 2018, pp. 3481–3490.

[3] P. Shamsolmoali, M. Zareapoor, E. Granger, H. Zhou, R. Wang, M. E. Celebi, and J. Yang, "Image synthesis with adversarial networks: A comprehensive survey and case studies," *Information Fusion*, 2021.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[5] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[6] J. Adler and S. Lunz, "Banach wasserstein gan," *arXiv preprint arXiv:1806.06621*, 2018.

[7] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[8] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[9] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *International conference on machine learning*. PMLR, 2017, pp. 2642–2651.

[10] S. Liu, X. Li, Y. Zhai, C. You, Z. Zhu, C. Fernandez-Granda, and Q. Qu, "Convolutional normalization: Improving deep convolutional network robustness and training," *arXiv preprint arXiv:2103.00673*, 2021.

[11] L. Weng, "From gan to wgan," *arXiv preprint arXiv:1904.08994*, 2019.

[12] X. Cao, S. Dulloor, and M. Prasetio, "Face generation with conditional generative adversarial networks."

[13] M. Górriz, M. Mrak, A. F. Smeaton, and N. E. O'Connor, "End-to-end conditional gan-based architectures for image colourisation," *arXiv preprint arXiv:1908.09873*, 2019.

[14] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

[15] N. Spolaôr, E. A. Cherman, M. C. Monard, and H. D. Lee, "A comparison of multi-label feature selection methods using the problem transformation approach," *Electronic Notes in Theoretical Computer Science*, vol. 292, pp. 135–151, 2013.

[16] C. Connolly and T. Fleiss, "A study of efficiency and accuracy in the transformation from rgb to cielab color space," *IEEE transactions on image processing*, vol. 6, no. 7, pp. 1046–1048, 1997.

[17] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[18] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019.

[19] A. Mathiasen and F. Hvilshøj, "Fast fr\'echet inception distance," *arXiv preprint arXiv:2009.14075*, 2020.

[20] Y. Wu, X. Qin, Y. Pan, and C. Yuan, "Convolution neural network based transfer learning for classification of flowers," in *2018 IEEE 3rd international conference on signal and image processing (ICSIP)*. IEEE, 2018, pp. 562–566.

Appendix

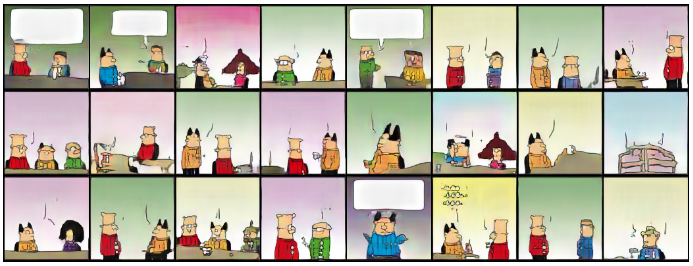


Fig. 6: A selection of generated images from SGAN

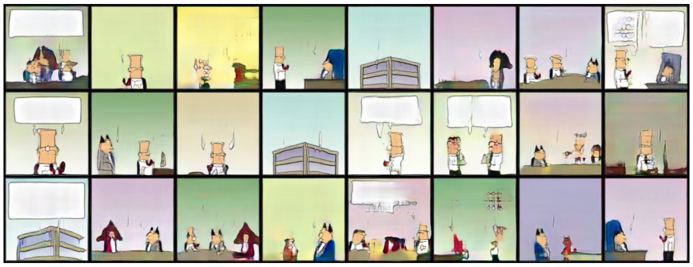


Fig. 7: A selection of generated images from MC-SGAN

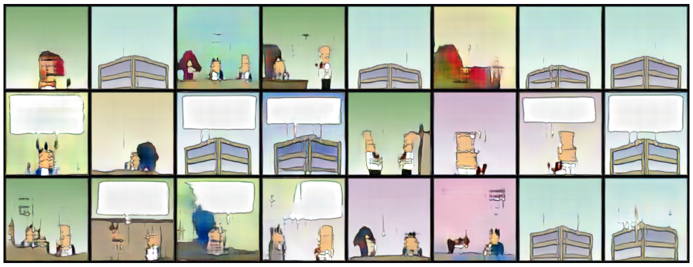


Fig. 8: A selection of generated images from ML-SGAN

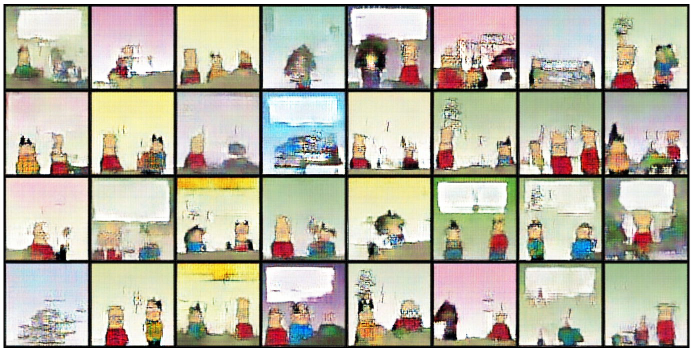


Fig. 9: Results from WGAN-GP in 128 x 128 configuration

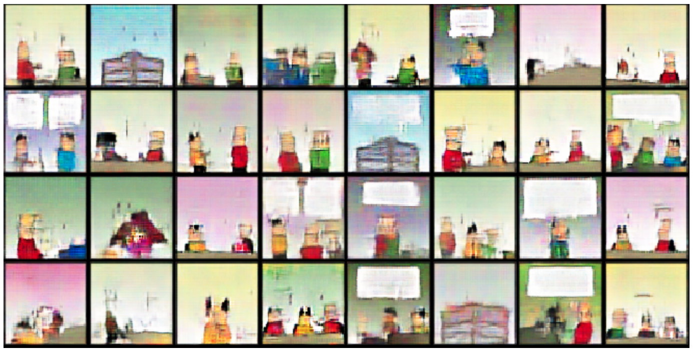


Fig. 10: Results from WGAN-GP in 64 x 64 configuration