Text-to-Comic Generative Adversarial Network

Benjamin Provan-Bessell
under supervision of
Lydia Chen, Zilong Zhao
Delft University Of Technology
bprovanbessell@student.tudelft.nl, y.chen-10@tudelft.nl, z.zhao-8@tudelft.nl

Abstract*—Drawing and annotating comic illustrations is a complex and difficult process. No existing machine learning algorithms have been developed to create comic illustrations based on descriptions of illustrations, or the dialogue in comics. Moreover, it is not known if a generative adversarial network (GAN) can generate original comics that correspond to the dialogue and/or descriptions. GANs are successful in producing photo-realistic images, but this technology does not necessarily translate to generation of flawless comics. What is more, comic evaluation is a prominent challenge as common metrics such as Inception Score will not perform comparably, as they are designed to work on photos. In this paper, we: 1. Extend state of the art GANs to enable a comics-focused GAN architecture; 2) We implement DescriptionGAN, a novel text-to-comic pipeline based on a text-to-image GAN that synthesizes comics according to text descriptions. 3. We describe an in-depth empirical study of the technical difficulties of comic generation using GAN's. DescriptionGAN has two novel features: *(i)* text description creation from labels via permutation and augmentation, and *(ii)* custom image encoding with Convolutional Neural Networks. We extensively evaluate the proposed DescriptionGAN in two scenarios, namely image generation from descriptions, and image generation from dialogue. Our results on 1000 Dilbert comic panels and 6000 descriptions show synthetic comic panels from text inputs resemble original *Dilbert* panels. Novel methods for text description creation and custom image encoding brought improvements to Frechet Inception Distance, detail, and overall image quality over baseline algorithms. Generating illustrations from descriptions provided clear comics including characters and colours that were specified in the descriptions.

I. Introduction

Comics are amusing, fun, and bring a humorous aspect to our everyday lives. They join text and illustrations together in a unique and complex way. Here we explore the possibility of using machine learning processes to generate comic illustrations from the dialogue present in comics, and from descriptions of illustrations. Generative Adversarial Networks (GAN) [1] are successful at creating new data that emulates real data, and have been applied to image production [2]. Automatic image synthesis from text using GANs has been researched for realistic images taken by cameras but has not been applied to the domain of comic illustrations. We show that the success of GANs with photos does not translate well to generating high quality comics, and standard algorithms that generate superb images, do not generate superb comics (See Fig. 1). Generation of cartoons using deep convolutional GAN's has been attempted, but comparisons of the best models has not been documented. Text-to-comic synthesis has not before been attempted. Comics are different in their makeup compared to regular photos, which brings specific challenges using GANs for comic generation, and to the evaluation of the success of this process. Therefore, we seek to further the area of GAN research within the realm of comics, to generate entertaining, engaging, and high quality comics.

(a) Original panel **(b)** Text-to-image **(c)** Deep Convolutional

Fig. 1: Example comic panels generated by standard 'out-of-the-box' versions of text-to-image and deep convolutional GAN respectively versus an original Dilbert comic panel from 2008/04/24 by Scott Adams. Unrecognisable, blurry characters and low image resolution should be improved upon.

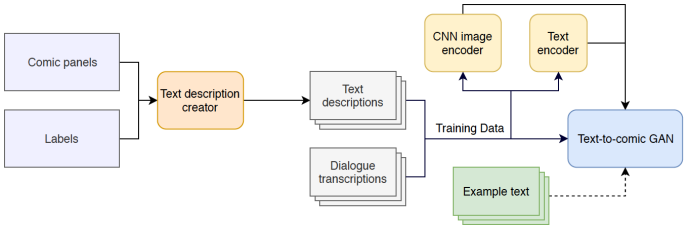


Fig. 2: Text-to-comic model pipeline.

The main goal of this research was to produce a model for text-to-comic generation. We apply our comic-generation methodology to *Dilbert* [3]. The model should produce a comic based on a text input (see Fig. 2). To achieve this task, different research questions were investigated:

1) What is unique about visual and textual features of comics that bring specific technical challenges to generate and evaluate comics?
2) How to modify state-of-the-art text-to-image models to generate comics from text, including automatic text description creation and computer vision techniques for image encoding?
3) What is the best Deep Convolutional GAN architecture

to generate comics?

Many challenges were evident from the start of this project. *Text-to-image* models such as AttnGAN [4, 5] learn semantic representation of the words within the image, e.g. that the word *'red'* corresponds to the colour red within the image. The initial proposed use case of this model was to generate comics from the dialogue; however the connection between this text and the image is very abstract, it proved challenging to learn the text-to-image model relation. Therefore, descriptions with a direct connection to the images are required. Automatic processes to create multiple varied descriptions per image are necessary to satisfy the need for training data. Furthermore, methods in text-to-image algorithms are designed to work on photo-realistic images and do not necessarily translate optimally to work on comics. To overcome this issue, replacement modules designed specifically for comics were investigated to improve the quality of the generated panels.

A second major challenge involved evaluating the performance of the comic-generation pipeline. Comic generation from text is a task that has not yet been researched. There are no baseline models to compare to, and datasets with text descriptions for comics do not exist. Inception Score (IS), a metric commonly used to evaluate image quality is not applicable to evaluating comics, as it is trained for the purpose of photo realistic images. Text-to-image evaluation of how the image corresponds to the text is difficult, with metrics to measure this not commonly available to be applied to this project. Therefore extensive quantitative evaluation, and partial evaluation with Frechet Inception Distance (FID) [6] was necessary to judge results of the comics. The GAN models experimented with were very computationally expensive to train which limited the number of experiments in the project's timeframe.

The contributions of this research: *(i)* The application of a *text-to-image* model (AttnGAN) to create DescriptionGAN: a new model to generate comic illustrations from text. This included implementation and investigation of: methods for generation of text descriptions of comics from labels and; custom visual feature extraction image encoder models. *(ii)* The creation of comic generation models via the use of deep convolutional GAN's. *(iii)* An in-depth study of technical challenges specific to generating comics such as: text correlation between comic and dialogue; extraction and uniqueness of visual features of comics and; evaluation of generated comics. Our results show that synthetic comic panels remarkably resemble original comics and achieve a high FID quantitatively. DescriptionGAN can further produce comics that match a text description.

This paper is structured as follows: Section II covers related work. Section III analyses GAN architectures for optimal comic generation. Methodology for the text-to-comic pipeline, DescriptionGAN, appears in Section IV. Section V discusses why comics bring unique challenges to generation with GAN technology. Experiment results and evaluation of all models and pipeline elements used to create comics can be found in Section VI. Section VII reflects on reproducability, and ethical implications of this research. Section VIII discusses conclusions of the research, and promotes future work.

II. Related work

Artificial image synthesis refers to the task of creating an image, usually based on some constraint. This is an active area of research, and while some techniques using recurrent neural networks [7] have been successful, GAN's are the most prominent and effective machine learning model to generate new images [8, 9]. We summarize the relevant related studies into sections regarding technologies of: Deep Convolutional GAN; Text-to-image GAN; Comic feature extraction with Convolutional Neural Networks (CNN).

Most state-of-the-art image generators build upon the Deep-Convolutional GAN (DC GAN) [10]. This method provided a framework for creating images using de-convolutional layers in the generator. These methods could be used as a basic first step to synthesising new comics, emulating techniques as in Simpsons cartoon generation and GANfield [11, 12].

There are successful text-to-image GAN's that can generate a detailed image based on a text description of the image. The original text-to-image GAN [2] is based on a model with a single generator and single discriminator. The input text description is encoded, and a resulting image that corresponds to the semantic meaning of the sentence is generated. This correspondence is checked by the discriminator model. Models such as MirrorGAN [5], AttnGAN [4], and StackGAN[13], extend [2] by creating additional generators that add detail at the level of each word in the description. Additionally, multiple discriminator networks are added to check that the image being generated is representative of the sentence, and the individual words in the sentence.

Encoding and performing feature extraction on the output image is an important sub-task. Both high-level features, and a final vector representation of the image are used in the attentional generation of the image, and for verifying that the image corresponds to the semantics of the text input. The image encoder used in AttnGAN which is based on the Inception v3 CNN [14], pre-trained on ImageNet dataset [15], should be substituted for a CNN more suited to encoding comics. Object detection for manga [16, 17, 18] have used CNN based techniques to extract features from the comics. Convolutional architectures used for feature extraction and classification such as VGG and Inception v3 [14, 19] are very powerful, but also are quite complex.

Generating images with GANs both with and without text are domain specific tasks, and out of the box solutions do not generate high quality comics. Moreover, specialized techniques are often needed to capture the distinct features and characteristics of comics. Modifications and improvements will need to be made to GAN and CNN models to improve their performance on comics.

Network	Architecture	Loss	Regularization
DCGAN	DCNN	BCE	None
WGAN-GP	DCNN	Wasserstein	Gradient Penalty
Stability GAN	ResNet	BCE	R1

TABLE I: DCGAN Architectures. Each model required implementation in PyTorch, specific set up, and parameter tuning before training, which took place on Nvidia P1000 and K80 GPU's for periods of up to 96 hours.

Network:	Resolution			Qualitative ranking
	64x64	128x128	256x256	
DCGAN				3
WGAN-GP				2
Stability GAN				1

TABLE II: DCGAN Results.

III. Empirical Analysis of DCGAN comic generation

To generate comics that are like the originals without text input, different versions of the Deep-Convolutional GAN (DCGAN) were implemented and experimented with. Each GAN architecture was modified to produce comics, and was trained on a dataset of *Dilbert* panels. Tables I and II summarize GAN architectures and results. GANS were trained on 5000 *Dilbert* comic panels for varying numbers of epochs, as different models took different amounts of time and iterations to converge.

The following acronyms in table I represent:

- DCNN: Deep convolutional discriminator and generator architecture as in [10].
- ResNet: ResNet CNN architecture for both discrinimator and generator as in [20].
- BCE: Binary Cross Entropy loss.
- R1: *R1* regularization as described in [21].

The fill colours in table II represent different levels of condition quality and recognisability in the generated image. Grey represents that this architecture was not experimented with at the according resolution. Please see Fig. 3, and refer to appendix A for generated images.

- Red: No distinguishable characters and blurry background.
- Yellow: Characters and background colour are starting to become recognisable.
- Green: Characters and background colour are clear, and recognisable compared to the original comic.

DCGAN [10] was implemented to generate 64x64 pixel images. Higher resolution (128x128, 256x256) generator and discriminator models were achieved by adding additional convolutional and de-convolutional blocks. Stride lengths and filters in the convolutional blocks were modified to better suit the higher image resolution. Vanishing gradient and modal collapse [22] were problems encountered, especially when generating higher resolution images.

(a) DCGAN 64x64 **(b)** WGAN 64x64 **(c)** Stability GAN 128x128

Fig. 3: Example comic panels generated the three DCGAN. Improvement and progression in image resolution and character and background colour quality and clarity is visible over the models.

Loss functions such as Wasserstein loss [23], and Wasserstein loss with gradient penalty [24] were implemented and integrated. The DCGAN models with this loss function had improved training as it eliminated the vanishing gradient problem, generating better images at both lower and higher resolutions.

Final implementation of comic generation via DCGAN's required the use of R1 regularization (see [21] for details). This regularization, and the ResNet generator and discriminator architecture brought better results for generating comics at higher resolutions, as generator and discriminator models continued to improve at later epochs, as images continued to to have better FID scores.

IV. Comics illustration synthesis

This section will discuss the methods used: to extract and create text descriptions that correspond to comics; to augment and modify AttnGAN to create comics from text; and build a CNN image encoder for comics. Implementation details on each will be provided in respective sections.

A. Text-to-Image GAN

Generating images from text took three steps: (1), the GAN model used as a baseline was researched, understood and re-implemented. (2), a suitable training dataset of images with corresponding text captions was procured. (3), the GAN model was trained and fine-tuned.

1) AttnGAN reproduction: AttnGAN [4] is a state-of-the-art GAN-based model that generates images from text, using images with multiple corresponding text captions as training data. The model works as follows: First, a Deep Attentional Multimodal Similarity Model (DAMSM) is trained on the images and captions. This produces a *text-encoder* and an *image-encoder* which encode an input caption and each word in that caption into feature vectors, and the input image into 17x17 vector regions representing high-level image features and a feature vector representing the entire meaning of the image. A generator first creates a low resolution (64x64 pixels) image from a text encoding of the entire input caption. Two subsequent generators upscale the image to (128x128) and (256x256) by using the output of the previous generator stage

as input. Attention mechanisms add detail by enhancing sub-regions of the image according to the relevant input words. The DAMSM model compares the encoding of the generated image to the encoding of the text caption to ensure that the image generated is semantically representative of the text. The loss from the DAMSM model and three discriminators is used to train the generator models. Finally, when the model is trained text can be used as input to generate an image.

This model was reproduced and tested on original text-image datasets [25]. This model was used as a baseline for experimentation of generation of comics, and was fine tuned and augmented to improve results.

2) Text description formulation: A text dataset corresponding to the comic panels was created. The steps and mechanisms used to label comics and subsequently create the text descriptions are described in this section.

1) Comic panels were manually labelled with characters, and the background colour. Recurring characters had a unique number, while non-recurring characters were grouped together under one number. Background colour was chosen as the most dominant colour present.
2) From these labels, descriptions were generated. To create a basic description, character numbers were replaced with the text names, and separated by *"and"*. To complete the description, the background colour was added to the text in the form of *"with colour background"*.
3) To increase the number of descriptions per panel, augmentation was performed on the descriptions. The order of the characters could be permuted, providing $nfactorial$ unique descriptions, where n is the number of characters present. Descriptions were multiplied further by placing the background colour at the start, or end of the character portion of the description, $doubling$ the number of descriptions.

Image synthesis from comics dialogue required the parsing of the comic dialogue. A resource for transcriptions of Dilbert comics [26] was downloaded, and a script was created to parse the dialogue into a dataset where each comic panel was associated with the text that appeared in it.

3) DescriptionGAN training and fine-tuning: The AttnGAN model was trained to generate new comics from text. Scripts to format *Dilbert* data in a way such that it could be trained on comics dataset were implemented. This required the reverse engineering of processes used to create the metadata used for the datasets used for original training on AttnGAN. DescriptionGAN was then configured and trained. Parameters and hyperparameters were tuned to suit comic datasets.

4) Multi-Label Conditional GAN: A multi-label conditional GAN could be created from a modified version of AttnGAN. While AttnGAN can produce high quality images from input text, it requires a large volume of detailed descriptions corresponding to images as training data. Multi-labeled image datasets are more commonly available and are easier to create by virtue of being simpler. AttnGAN uses a text-encoder (Recurrent Neural Network) to encode the input text as a feature vector. By simplifying the architecture of AttnGAN, label vectors could be passed in as input instead of text, foregoing the need for a text-encoder. Furthermore, the image-encoder could be simplified to a multi-label classifier, by reducing the output dimension of the CNN to that of the number of input classes with a linear layer. The DAMSM would compare the input label vector to the label vector predicted for the generated image. The 17x17 high-level regional features produced by the image-encoder would also have to be reshaped to the number of number of classes, and then compared to single encodings of the classes present in description. This modification of the model would remove the need to create the text-encoder, and remove any chance of misrepresentation or misinterpretation of the contents of the image that could be caused by the text encoder. This could improve image quality in two ways: images would have better correspondence to the input labels; images could be of better general quality. These improvements would occur because the DAMSM component could learn the labels and features on the images better, and therefore provide a more accurate loss to better train the generators. This modification could not be created due to time limitations of this research, but is proposed as future work.

B. Comic feature extractor CNN

A multi-label classifier was trained on comics to classify the labels from which the comic descriptions were generated: the colour of the background and the characters in the comic. As it was time intensive and expensive to judge the results of image encoders while integrated as part of AttnGAN (due to the long training times), feature extraction capability was judged based on accuracy of the classifier. A higher accuracy means that the appropriate features are being extracted by the model.

The image encoder built into AttnGAN is based on the Inception v3 [14] CNN architecture. It is used to extract 17x17 regional features, and a final image encoding. To better capture these encodings of the features, a CNN created specifically to encode comics was designed and implemented.

The comics CNN is constructed as follows: The input (image of size 299x299 pixels with 3 channels of red, green and blue) is put through two convolutional blocks, consisting of two 2-d convolutional layers and a max pooling layer, which reduce the spatial size from 299 to 35, while increasing the depth to 128. Three inception blocks inspired by [14] were applied, further increasing depth and reducing spatial size to 17. These blocks output the 17x17 feature regions. This output is then processed through two more inception blocks, before final processing by average pooling, dropout, and flattening layers. A final fully connected layer outputs the final number of classes. As in inception v3, the comics CNN uses the feature regions while training to improve the loss of the model. The 17x17 features

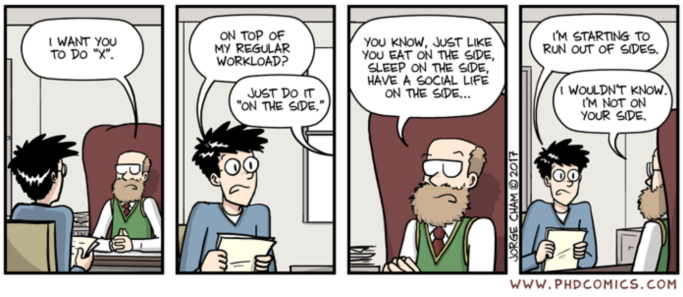
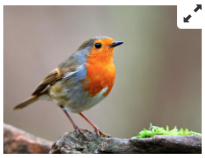


Fig. 4: Example of *PhD Comics*, author: Jorge Cham. The dialogue is not a clear description of what is going on in the image.



(a) Real life photos generate accurate descriptions.

(b) Comic strips do not generate meaningful descriptions.

(c) Text removal dose not improve results.

Fig. 5: Automatically generated descriptions for comics and real life photos using IBM's image caption generator [27]. Comics are not recognised properly.

are used to predict the final class output as well. The final loss which is back-propagated through the model is: $loss + 0.4 \times featureloss$ Where $loss$ is the binary cross entropy loss between the final layer prediction and the actual output, and $featureloss$ is the binary cross entropy loss between the feature layer prediction and the actual output. The addition of the fractional loss $featureloss$ ensures that the regional features capture representative characteristics of the image.

V. Technical Difficulties of Comic Synthesis with GAN Technology

While GANs are competent at creating photo-realistic images, successfully applying this technology to generate comics has specific challenges and difficulties. This section describes an in-depth study of why GAN methods that work on images does not translate to comic-generation.

A. Text correlation between comic and dialogue

Initially it was thought that the comics dialogue could be used as input text for generating comics. However, problems with this approach were evident immediately. Text-to-image models are trained on images and text. Each image has corresponding textual descriptions which describe the subject/object within the image. There is a direct semantic connection between the text and the image. This is not at all what the comics dialogue and subtext is. There is no description of the characters or objects within the comic (see Fig. 4). The connection between the dialogue of comics and the illustration is abstract at best.

Automatic generation of descriptions was infeasible without the training of a custom ML model (see Fig. 5). This is due to the fact that most description generators are trained on real life photographs, as opposed to illustrations. Therefore a study on correlation between comic and text had to be carried out. Images cannot be generated from text without a text dataset. Text descriptions had to be manually created. This led to questions: What type annotations should be created? What should they contain? How should they be composed such that the text-to-image models can learn the representation of the text that is present in the images?

A direct semantic connection between the text and contents of each image is needed for the text-to-image model to learn to create images that represent text. Initial intuition led to emulation of the descriptions in other text-image datasets, which contained a description of the main subject of the image, using adjectives such as colours, shape, and size. This led to complicated descriptions, for which it was infeasible to create enough data to learn these descriptions (see Section VI-B1 for results). Final text datasets were create from characters and background colour present in the comic (See section IV-A2).

B. Comic image feature extraction and uniqueness

Comics are different in their makeup in comparison to real images. Computer vision applications trained specifically on photo realistic images are not necessarily effective when applied to comics. This was observed when descriptions generators did not work appropriately on comics, but did on real life images (see section V-A). Features of comics are different to those in images. Generally they are simpler and bolder: Black lines distinguish all characters and objects, characters each have unique hair, facial features and clothes. Backgrounds are often a single block colour, and do not contain the same level of variation and detail.

C. Evaluation of generated comics

GANs are notoriously hard to evaluate. They have no objective loss function, and results such as illustrations are subjective [27]. Due to the differences between photo images and comic illustrations, common evaluation metrics (e.g., IS) are not applicable for evaluating generated comics, as they are suitable only for photo-realistic images. Furthermore, Comic illustration synthesis from text has not been previously studied. Baseline algorithms are not available to provide comparisons on novel models created in this research.

Absence of comic datasets with text descriptions proved to be a major problem. Text descriptions had to be manually created for each comic panel, which is time consuming and infeasible to carry out for the number of comic panels needed for a training set. Methods to create descriptions from character and

colour labels helped partially automate text dataset creations, but it did not solve the problem. A set of comics, labelled with traits suitable to create descriptions was still needed. Such datasets do not exist. Due to time limitations of this project, only 2500 *Dilbert* comics were labelled. Filtering of unusable comics, this reduced the dataset size to 1000 panels. Moreover, data-augmentation transformation applicable to images (e.g., brightness, rotation, cropping) are not applicable to comics, as they should have specific colours and orientations. Ideally, evaluation of GAN generation methods with more data, and multiple different comic datasets (*Garfield, PhD comics*) is desired to form a more complete judgement. Furthermore, more descriptive text, rather than text derived from labels could improve generated image quality, and would serve as a good comparison to what text-to-image models can learn.

Training GANs is time and computationally intensive. Limited time and resources restricted the experiments that could be run. As a result, not all experiments were able to be evaluated with FID to the extent that was desired.

Specific solutions should be engineered and implemented to improve comics generation. The comic CNN multi-label image encoder (see Section IV-B) implemented in this research is an example of technology innovations needed to capture features of comics better. However, further development is need to optimise comic generation.

VI. Evaluation

Over 15 experiments regarding different generative models were conducted. Results of the best DCGAN model (see Section III) will be compared to the best versions of DescriptionGAN. DescriptionGAN will then be extensively evaluated.

Image Datasets. *Dilbert* was the main dataset used to evaluate the models created. Models were trained on single comic panels. Further image processing such as removing the text from the comics was done to simplify the dataset. Filtering of comics was performed based on labels of comics such as removal of uncommon and non recurring characters.

Text Datasets. Dialogue datasets could be extracted automatically, for each comic that had a transcription. Descriptions were created manually, or translated from descriptions as described in Section IV-A2.

Text datasets created:

- Dialogue of the comic. The text that is inherent in the comics, spoken by the characters.
- Detailed character description. Text that describes the character. E.g. *Dilbert is wearing a white shirt and a red tie and is sitting down by his computer.*
- General description of the comic. Text telling what characters are in the comic and what the background colour is. E.g. *Dilbert and Alice with a green background.*

Evaluation Metrics.

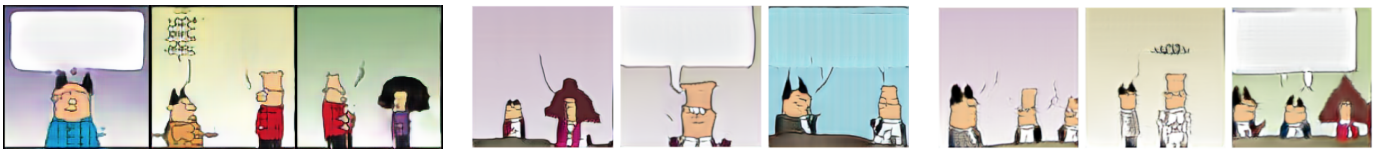
Fréchet inception distance (FID) [6] was used to quantitatively evaluate the quality of images generated by GAN's. FID compares the generated images to the images from which the GAN is created. Inception Score (IS) [28] is another metric used to assess the quality of GAN generated images. FID improves over IS as it compares the features to the distribution of generated images $N(\mu_w, \Sigma_w)$ to the distribution of features of ground truth training images $N(\mu, \Sigma)$ (see equation 1), as opposed to IS which just evaluates the generated distribution. This is relevant when evaluating comics, as inception score is designed to judge photo-realistic images, not illustrations.

$$FID = |\mu - \mu_w|^2 + tr(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{\frac{1}{2}}) \quad (1)$$

Generator and discriminator loss over epochs was used to evaluate the performance and stability of models, which can diagnose the vanishing gradient problem, and can give information as to whether the model has converged or will improve with more training.

Qualitative evaluation was also performed on models. Intermediate results at different checkpoints provide insight into in image generation. As images were being generated from text, the correspondence between the meaning of the text and the generated image was checked. This correspondence could be verified for text descriptions manually.

Multi-label classifiers were evaluated with accuracy, and F1 score. Accuracy is defined as the number of images for which the class predictions were correct, over the total number of predictions. In the context of multi-label classification, accuracy marks predictions that are partly correct as incorrect. Therefore F1 score defined as $2 \times \frac{precision \times recall}{precision + recall}$ was also used to give an indication of the proportion of classes in general predicted soundly.



(a) Stability GAN examples **(b)** Examples generated from descriptions. **(c)** Examples generated from dialogue.

Fig. 6: Generated comic panels. Models were trained on a Google-cloud virtual machine with a Nvidia K-80 GPU. PyTorch was the main implementation tool for models. Training for each model varied between 24-96 hour time periods.

Model	DCGAN	DescriptionGAN	DialogueGAN
FID	89.233	150.953	143.803
Epochs	1200	350	400

TABLE III: Model FID and epochs trained

Fig. 7: Images generated for captions of Dilbert and dog, and dog respectively. As can be seen in the centre attention map, the model was not able to learn the mapping of the word 'Dilbert' to its character.

A. Overall Results Comparison

This section will provide an overall comparison of results from the best generative models: DCGAN (StabilityGAN); Text-to-image GAN trained on descriptions (DescriptionGAN); Text-to-image GAN trained on dialogue (DialogueGAN).

Fig. 6 shows generation results for the three GAN models. The comics generated show a good variation in background colour, and they also capture characters recognisable as Dilbert, the pointy-haired boss, Alice and Wolly. The DCGAN achieved the best FID score (see table III), with the two text-to-image models having comparable scores. The better FID scores correlated with longer training time and more data. While the DCGAN had the best FID score, and comics generated were more consistent, DescriptionGAN generates higher resolution images (256x256 as opposed to 128x128), and you can control the characters that appear in the image, as well as the background colour.

A common downfall of all models was their inability to create characters such as Catbert and Dogbert. Their small size and infrequent appearances most likely contributing to the model's failure to reproduce them. This is a standard problem in machine learning when there is insufficient data.

B. Text-to-Comic analysis

In this section, text-to-image models based on AttnGAN are evaluated. FID and generator and discriminator losses are compared, as well as extensive qualitative evaluation. Overall 6 models were trained and evaluated. Meaningful experiments are grouped into 3 stages: (1) Initial models using AttnGAN to create comics from text. (2) DescriptionGAN trained on comic descriptions. (3) The DescriptionGAN created from training data of comic dialogue only.

1) Initial experiments: A text-to-image model trained on a small set of (75) *Dilbert* images and corresponding descriptions was first created. This model was not trained for long enough, or on enough data, as results did not resemble any comic. A further text-to-image model trained on a dataset of 300 images of only Dilbert and/or Dogbert was created. Text descriptions were also limited to the form of *Dilbert* or *dog* or *dilbert and dog*.

Model	Epoch 100	Epoch 200	Epoch 350
FID	200.258	182.163	150.953

TABLE IV: Model FID score taken from epoch checkpoints

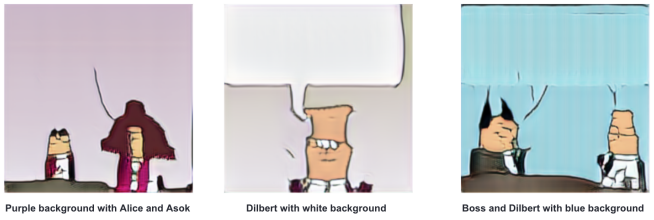


Fig. 8: Generated examples comics. Text descriptions used to generate the illustrations appear below each respective comic.

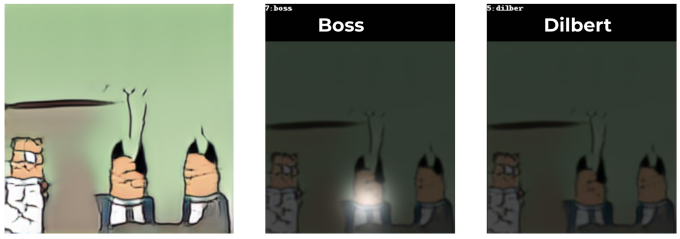


Fig. 9: Attention mapping for words of *Boss* and *Dilbert*. As can be seen, the model learnt where the where the boss should be detailed, but not Dilbert.

While the results of this second model left much to be desired (See Fig. 7), the improvement from the first model indicated that a larger dataset, with more descriptions per image would be necessary to generate clearer comics. Qualitative evaluation on a multiple generated images provided insight that the model had started to learn to draw characters and speech. These generator models were not possible to evaluate with FID, due to a minimum requirement of 2048 images in both training and generated images set.

2) Character and colour analysis: A text-to-image model was trained on a dataset of descriptions of text and images created by methods described in IV-A2. This formed a dataset of 1000 images, each with 6 corresponding character and colour descriptions, for 6000 text descriptions in total. Analysis for models saved at different points in training time was carried out, which was used to generate images for 2048 sample text descriptions.

Further training time improved the quality on the images. Characters became clearer, and the general quality of images got better. This is supported by the FID score results (see table IV). Discriminator loss was generally much lower, with

Model	Epoch 100	Epoch 200	Epoch 300	Epoch 400
FID	231.912	217.680	170.554	143.803

TABLE V: Model FID score taken from epoch checkpoints, trained on dialogue descriptions

(a) Dilbert comic from 2015-01-26, author: Scott Adams.

(b) Generated comic strips from dialogue from above comic

(c) Attention map of words from the second panel. As can be seen by the greyed out image, not word was recognised to give detail in the image. This was the case for all other attention maps.

Fig. 10: Ground truth comic, generated comic and attentional map for the DialogueGAN

vanishing gradient starting to become a problem towards the end of training, which led to the conclusion that for the current data, the model was close to convergence.

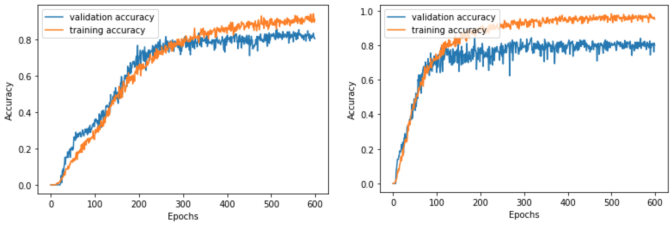
Based on qualitative evaluation, this model learned the main features of the comics, and the text. Coloured backgrounds matched the text description, and the colour gradient was almost identical to that seen on the real *Dilbert* comics. Almost all characters present in the training set could be recognised (see Fig. 9). Dilbert's unique face shape and glasses, The Pointy haired boss's hair, Alice's hair, Asok's face shape and complexion, and Wolly's stature (to some extent) were all recognisable in the generated comics. This attempt had some issues. For example, Wolly was often generated as a character that looked like Dilbert. While the generated images did capture most of the characters, they did not always correspond directly to the description.

Improvements could be brought by creating a more balanced dataset, with more evenly distributed numbers of background colours, and characters. This would teach the GAN to learn to create these characters and colour as opposed to ignore them.

3) Image generation from dialogue: Generation of comics from dialogue only was experimented with. An extracted

CNN	Comics CNN	Inception pre-trained	Inception
Accuracy (%)	76.9	80.77	71.1
F1-score	.936	.945	.917
Training time (minutes)	37	52	156

TABLE VI: CNN accuracy and F1 scores on test set, and training time to converge in minutes.



(a) Inception v3 trained from scratch **(b)** Comics CNN trained from scratch

Fig. 11: Training and validation accuracy over epochs for CNN multi-label classifiers.

dataset of comic dialogue transcriptions was available [26]. This text was then paired to its respective comic panel. This formed a text-annotated image dataset. 2250 comic panels along with 2250 corresponding text dialogues were used to train the AttnGAN model. Images were generated on text used in the training set, and text not in the training set.

The GAN did learn to create comics that resembled the original comics (see Fig. 10a, 10b) . However the images generated had no correspondence to the dialogue text. This result can be seen in Fig. 10c, where the attention maps are not able to link input words to details in the image. This result is as expected. The model was able to learn to create comics with recognizable characters, but they lacked detail, and connection with the input text.

C. Multi-label Feature Extraction Analysis

Three distinct experiments regarding CNN models for comic feature extraction were conducted. Training and validation accuracy over epochs, as well as accuracy and F1 score on a test set was used to analyse the performance of the multi-label classifier models.

Three different CNN architectures were experimented with:

- Custom comics CNN architecture.
- Inception v3 pre-trained on ImageNet, final classification layer trained on comics.
- Inception v3 trained from scratch on comics.

The inception v3 architecture pre-trained on imagenet performed best (see Section VI, although it was prone to over-fitting on the training set. Inception v3 architecture trained from scratch took far longer to train and did not reach the same accuracy compared to its pre-trained version. The comics CNN outperformed the scratch version of Inception, showing a 6% improvement in accuracy on the test set, and taking far less time and epochs to converge (see Fig. 11). While

the comics CNN was not able to exceed accuracy of the pre-trained version of inception, the improvement over the scratch version of inception indicates that the comics CNN is more suited to extracting features of comics. Therefore, a version of comics CNN pre-trained on a large dataset of comics should outperform the pre-trained version of inception.

VII. Responsible research

A. Integrity

Data manipulation could impact the results severely. GANS need large volumes of data to train on to be effective, and changing this data can have profound effects on the resulting models. All datasets, and processing of each dataset is explained in Section VI, along with motivation for the used transformations. While generated images have been chosen to illuminate specific examples of success and failure of the models, other metrics (FID) that evaluate the whole distribution of generated data were used.

Plagiarism was a considered issue, as comics in already existing styles are being replicated. However, these generated comics were created for research purposes only, and will not be used commercially or for profit. Terms of use for the official comics had to be agreed to, and permission had to be acquired where applicable. As stated in the Dilbert terms Comics must also be shown without edits, and the author must be credited. Educational use of Dilbert comics requires no specific access request as declared in licensing permissions, as long as the aforementioned rules are followed.

B. Reproducibility

All results should be largely reproducible. I will publish the code used to achieve all results on GitHub. This repository will include all scripts used to pre-process comic dataset images, annotate images, convert annotations into text descriptions, and create metadata and configuration scripts necessary to train the GAN model. Furthermore, the code for the modified version of AttnGAN used to train all models will also be available on GitHub [1]. However, comic image dataset will not be made public according to terms and conditions. Instructions for how to set up experiments, and reproduce results will be provided in clear documentation. All theory and motivation behind design choices will be explained in this report. As deep learning has some stochastic processes, results will likely not be exactly the same when reproduced with the same data and similar environment [29]. Moreover, data generated by GANs takes as part of the input a vector of noise randomly sampled from a distribution. These random vectors used to generate data were not saved, therefore it is highly unlikely that specific images will be directly reproducible. However, large samples will be representative of the findings of the paper.

[1] Code available here

A possible ramification of providing reproducible work is that someone could use the posted source code to generate new comics, and then use those comics in a way that violates the terms of use of those comics, such as commercialisation.

VIII. Conclusions and Future work

The main purpose of this research was to create a text-to-image GAN to create comics, and to apply the pipeline (using text input) to comics in the style of Dilbert. We have developed a text-to-image pipeline for comics synthesis based on existing methodology. A novel text-to-image GAN, DesriptionGAN was created by using automatic text description generation methods and custom comic feature extraction using CNN's. DCGAN techniques for comic generation were investigated, implemented and compared. The differences between comics and photo-realistic images was studied, and the ramifications of such differences on generation of comics with GAN technologies was described.

Synthesis of comics illustrations was shown to be successful and representative of *Dilbert* comics. StabilityGAN and DescriptionGAN models brought significant improvements in FID score and image quality to comic generation over baseline models. A pipeline to create comic descriptions corresponding to comic panels to create text annotated comic datasets was designed and implemented. This enabled the training of text-to-image synthesis models, which produce comic panels characteristic of their description. High quality comics can be created which include the characters and colour in the description; however characters were misinterpreted occasionally. The comics CNN designed for specific comics feature extraction outperformed inception v3, with data suggesting that a large comics database would further improve results with pre-training. Differences between comic illustrations and photos were highlighted, and possible solutions to improve GAN generation results were proposed.

Future work and improvements include: adapting StoryGAN [30] to the task of generating an entire comic strip, composed of multiple related panels that follow one after another. Further evaluation should include detailed analysis of various comic datasets, with multiple detailed text descriptions per image. Further modifications to AttnGAN could be made such as simplifying the input to a set of labels, and changing the DAMSM loss component accordingly to create a novel multi-label conditional GAN.

References

[1] Ian J Goodfellow et al. "Generative adversarial networks". In: *arXiv preprint arXiv:1406.2661* (2014).

[2] Scott Reed et al. "Generative adversarial text to image synthesis". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1060–1069.

[3] Scott Adams. *Dilbert*. May 2021. URL: https://dilbert.com/.

[4] T. Xu et al. “AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1316–1324. DOI: 10.1109/CVPR.2018.00143.

[5] Tingting Qiao et al. “Mirrorgan: Learning text-to-image generation by redescription”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1505–1514. DOI: 10 . 1109 / CVPR.2019.00160.

[6] Martin Heusel et al. “Gans trained by a two time-scale update rule converge to a local nash equilibrium”. In: *arXiv preprint arXiv:1706.08500* (2017).

[7] Karol Gregor et al. “Draw: A recurrent neural network for image generation”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1462–1471.

[8] Shirin Nasr Esfahani and Shahram Latifi. “Image Generation with Gans-based Techniques: A Survey”. In: *International Journal of Computer Science and Information Technology* 11 (Oct. 2019), pp. 33–50. DOI: 10.5121/ijcsit.2019.11503.

[9] Ruben Tolosana et al. “Deepfakes and beyond: A survey of face manipulation and fake detection”. In: *Information Fusion* 64 (2020), pp. 131–148.

[10] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *arXiv preprint arXiv:1511.06434* (2015).

[11] Rob Beschizza. *GANfield: ai-generated Garfield*. Feb. 2020. URL: https://boingboing.net/2020/02/05/ganfield-ai-generated-garfiel.html.

[12] Greg Surma. *Image Generator - Drawing Cartoons with Generative Adversarial Networks*. Apr. 2020. URL: https://gsurma.medium.com/image-generator-drawing-cartoons - with - generative - adversarial - networks - 45e814ca9b6b.

[13] Han Zhang et al. “Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5907–5915.

[14] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. arXiv: 1512.00567 [cs.CV].

[15] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[16] Hideaki Yanagisawa, Takuro Yamashita, and Hiroshi Watanabe. “A study on object detection method from manga images using CNN”. In: *2018 International Workshop on Advanced Image Technology (IWAIT)*. 2018, pp. 1–4. DOI: 10.1109/IWAIT.2018.8369633.

[17] Toru Ogawa et al. “Object Detection for Comics using Manga109 Annotations”. In: *CoRR* abs/1803.08670 (2018). arXiv: 1803.08670. URL: http://arxiv.org/abs/1803.08670.

[18] Xiaoran Qin et al. “A faster R-CNN based method for comic characters face detection”. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 1. IEEE. 2017, pp. 1074–1080.

[19] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[20] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[21] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. “Which Training Methods for GANs do actually Converge?” In: *International Conference on Machine Learning (ICML)*. 2018.

[22] *Common Problems — Generative Adversarial Networks*. Feb. 2020. URL: https://developers.google.com/machine-learning/gan/problems.

[23] Martin Arjovsky, Soumith Chintala, and Léon Bottou. “Wasserstein generative adversarial networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[24] Ishaan Gulrajani et al. “Improved training of wasserstein gans”. In: *arXiv preprint arXiv:1704.00028* (2017).

[25] P. Welinder et al. *Caltech-UCSD Birds 200*. Tech. rep. CNS-TR-2010-001. California Institute of Technology, 2010.

[26] Alfred Arnold. *Dilbert transcription*. June 2021. URL: http://john.ccac.rwth-aachen.de:8000/ftp/dilbert/dilbert.txt.

[27] IBM Developer Staff. *Image Caption Generator*. Mar. 2018. URL: https://developer.ibm.com/technologies/artificial - intelligence / models / max - image - caption - generator/.

[28] Tim Salimans et al. “Improved techniques for training gans”. In: *arXiv preprint arXiv:1606.03498* (2016).

[29] Chao Liu et al. “On the Replicability and Reproducibility of Deep Learning in Software Engineering”. In: *arXiv preprint arXiv:2006.14244* (2020).

[30] Yitong Li et al. “Storygan: A sequential conditional gan for story visualization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6329–6338.

APPENDIX

Fig. 12: DCGAN 64x64

Fig. 13: DCGAN 256x256

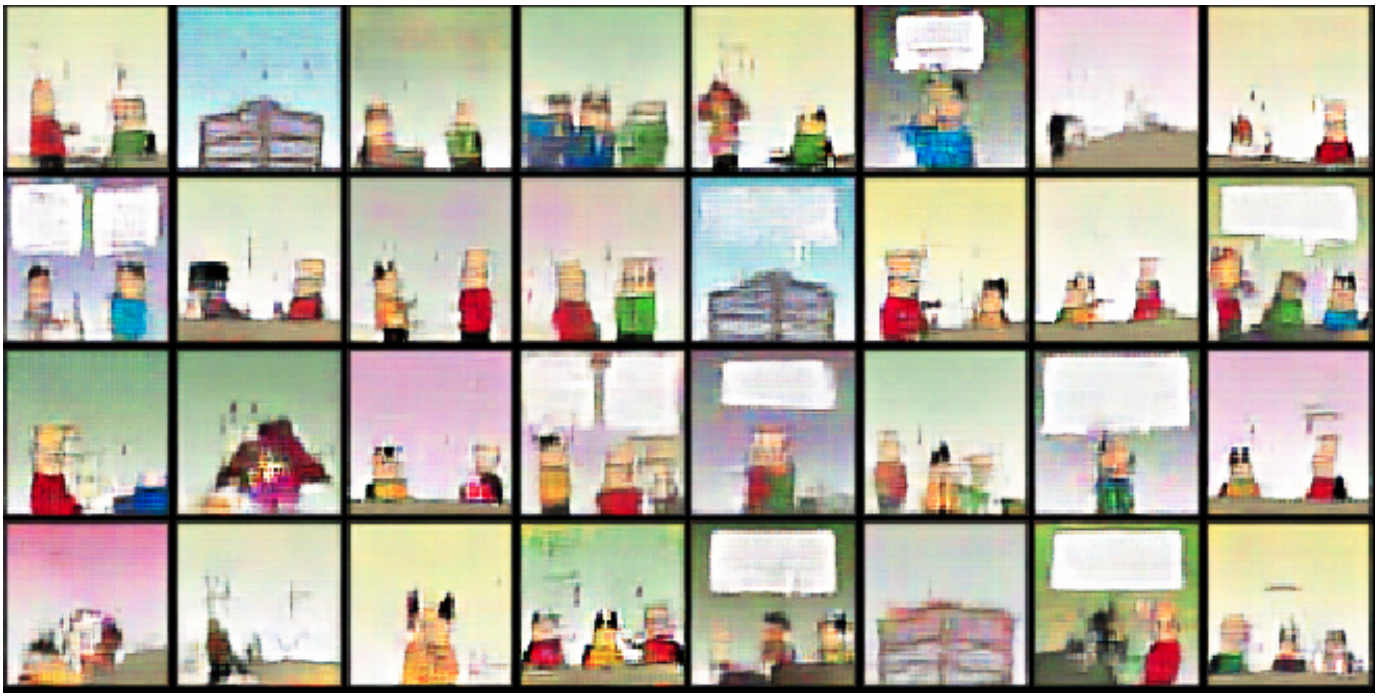


Fig. 14: WGAN 64x64

Fig. 15: WGAN 128x128

Fig. 16: Stability GAN 128x128

(a) Original panel

(b) Text-to-image

(c) Deep Convolutional

Fig. 17: Example comic panels generated by DescriptionGAN and StabilityGAN versus an original Dilbert comic panel from 2008/04/24 by Scott Adams.

Fig. 18: AttnGAN 256x256

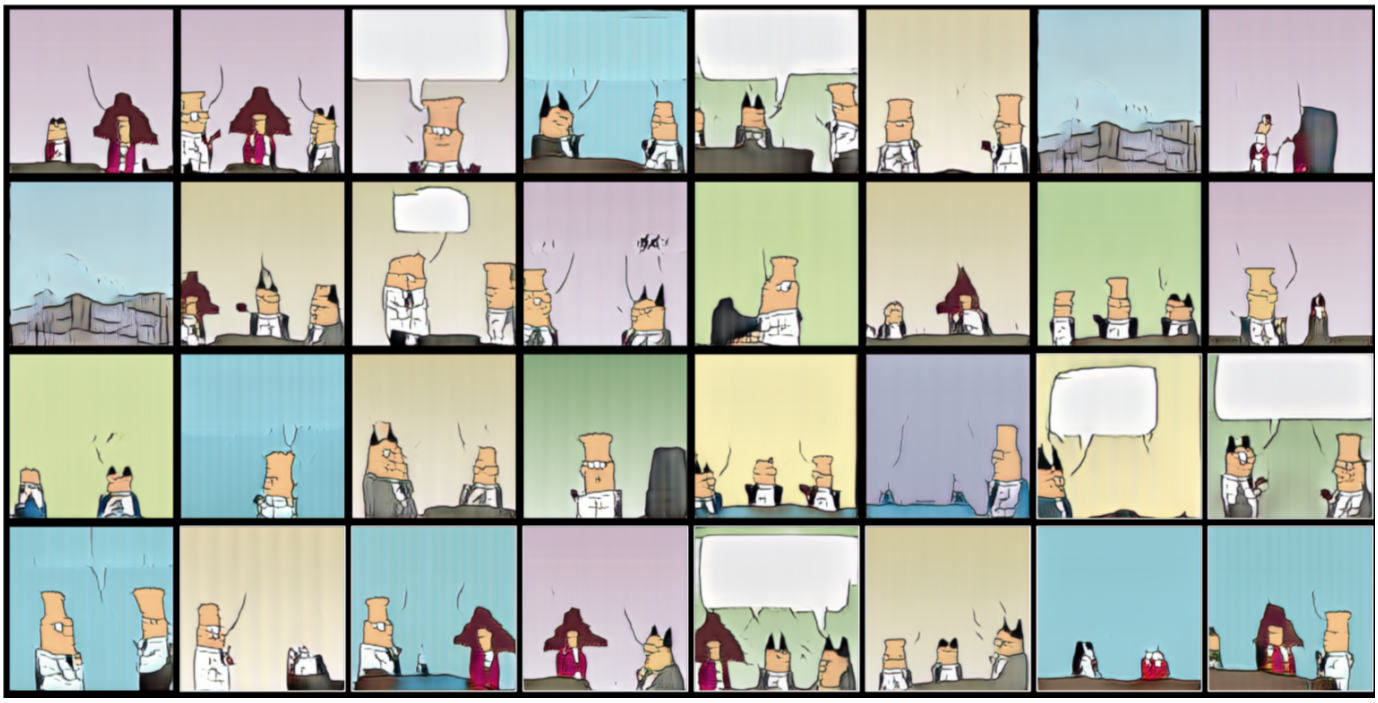


Fig. 19: DescriptionGAN 256x256

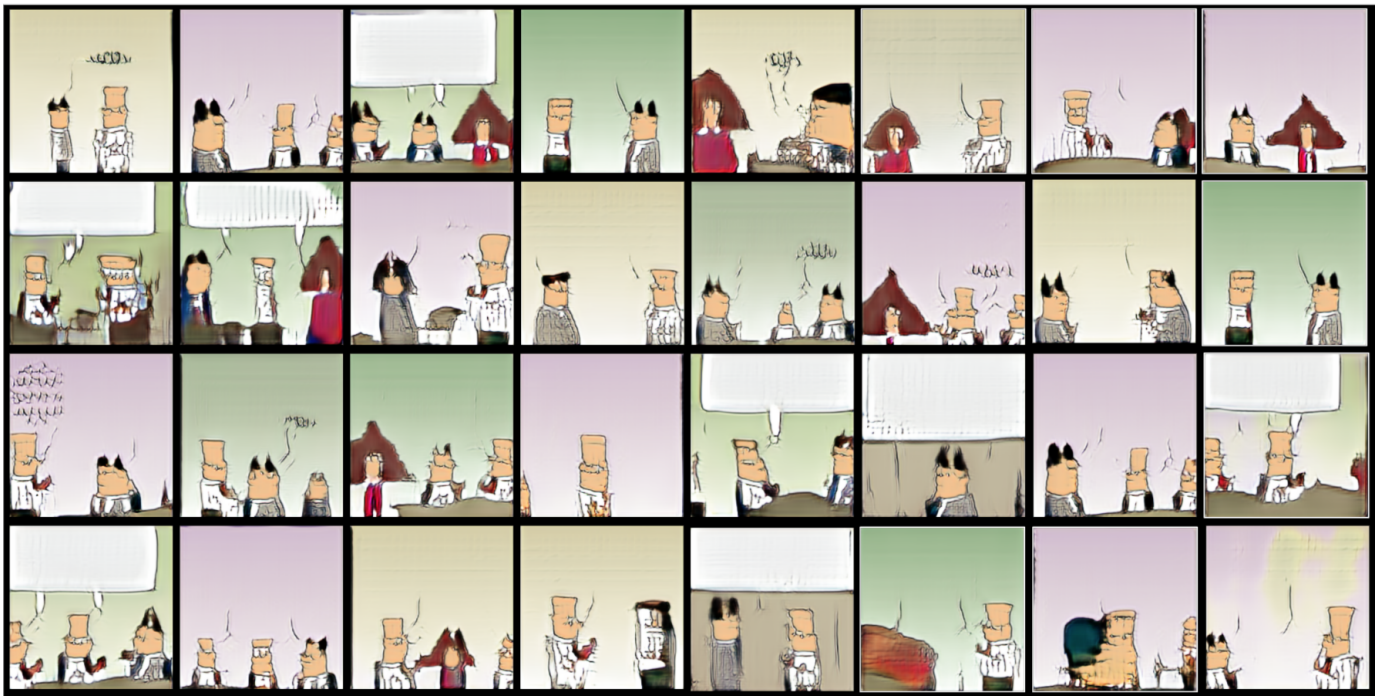


Fig. 20: DialogueGAN trained on dialogue 256x256