Conditional generation of anime facial images with StyleGAN

Ivan Poliakov
Department of Advanced Computing Sciences
Maastricht University
Maastricht, The Netherlands

***Abstract*—The paper presents a research in the area of conditional generative adversarial networks (GANs) for Japanese animation character images generation. Existing literature provides several similar projects but they all use outdated techniques or pose slightly different goals so there is a lot of room for exploration. We use modern GAN architecture - StyleGAN that allows for stable training and generation of high quality images as well as output control via model's latent space. Together with conditionality we can achieve even better control over the output while maintaining general quality of the images. However, StyleGAN models are infamous for lengthy training times so we investigate possibilities for transfer learning. Conditionality of the model could be of great use here as separate classes of the same model could provide for a quicker start for different transfer learning models. We discover that using tag estimators we can achieve good class separability and using this conditional model as a basis for training other models we can save significant amount of training time.**

***Index Terms*—Condititonal StyleGAN, anime StyleGAN, StyleGAN transfer learning**

I. Introduction

As Japanese manga and animation industry started taking over the western audience during the 90's it has been confidently growing ever since[1]. Being animator or manga artist is a very demanding career in which employees are known for overworking and being underpaid[2]. At the same time, drawing is difficult to master and thousands of people who acquired the hobby lack the tools to express their creativity. These two factors create desperate demand for smart tools that would be able to assist with character design. Effective controlled image generation could help tremendously not only professional artists searching for inspiration, but also those who do not possess the skill of drawing seeking an alternative design approach. Fortunately, the recent advancements in the field of machine image generation have been nothing short of breathtaking with works such as DALL-E 2[3] and StyleGAN[4]. While OpenAI's DALL-E 2 is currently unavailable to public domain, StyleGAN architecture family of generative adversarial networks remains the most prominent deep learning solution. However, most of existing StyleGAN applications generate non-conditional images solely from random noise distributions and little research has been done regarding conditional generation. Therefore, the focus of this paper lies in the area of potential applications of conditional StyleGAN model.

This thesis was prepared in partial fulfilment of the requirements for the Degree of Bachelor of Science in Advanced Computing Sciences, Maastricht University. Supervisors: Gerasimos Spanakis, Alexia Briassouli

The existing literature provides several studies heavily related to the topic of the paper. The most notable attempt in conditional anime faces generation is MGM[5], light-weight DRAGAN model that allows for in-browser 256px image generation conditioned on 34 non-mutually exclusive tags. MGM showed that GANs are performing much better on color-based tags rather than complex ones such as *glasses* or *hat*. Zepeng Liu in 2020 published a paper on full-body anime character generation with StyleGAN[6], featuring impressive image quality results and latent space interpolation; however, the model was non-conditional. Unrelated to animation models include LoGAN[7] and LoGANv2[8], the latter features StyleGAN application for logo generation conditioned on categorical labels.

GANs are known for their lengthy training time so it is crucial to be able to reuse high-quality models. However, there is no scientific research on transfer learning for conditional GANs trained on anime faces. GANs also have a few significant disadvantages. They are performing worse when trained on monochrome images[9], which creates an obstacle for generating manga-styled output. They are also sensitive to dataset size and require a lot of diverse samples. Therefore, we would like to apply transfer learning in both of these complicated cases and build two models, both for monochrome (black and white manga) output and colored output based on a small dataset.

In summary, this paper research goals can be phrased as following:

- What output quality can we achieve with our models given our limited computational resources?
- How can we achieve a good class separation with a conditional model?
- We train two transfer learning models: one for monochrome output and one for colored output but with a small dataset. In both cases Conditional StyleGAN model is used as a baseline. How do the introduced complexities of monochrome dataset and small size dataset affect the models' output?
- Do we observe reduction of training time when using transfer learning in comparison to training the models from scratch?

II. Related work

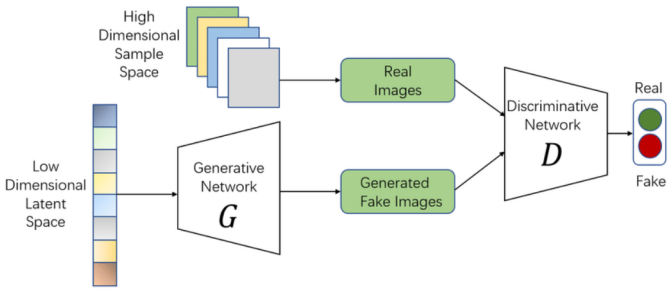

Fig. 1. General GAN architecture (source: Utilizing Amari-Alpha Divergence to Stabilize the Training of Generative Adversarial Networks[10])

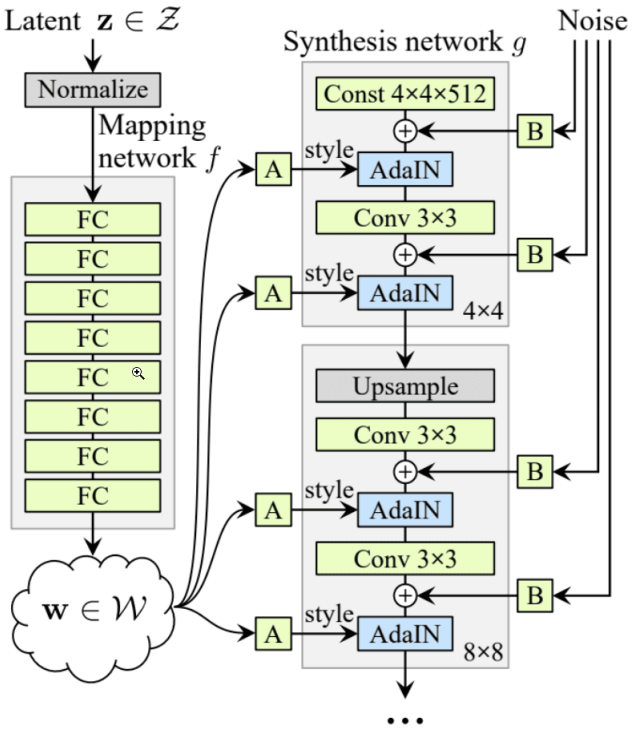

Fig. 2. StyleGAN architecture (source: A Style-Based Generator Architecture for Generative Adversarial Networks[4])

At the core of GAN is the idea of training two neural networks, generator and discriminator(Figure 1). While generator outputs images based on the random noise distribution, discriminator tries to separate generated images from original dataset. Training the networks in adversarial process leads to generator producing images of a distribution similar to the original dataset. First GAN was introduced by Ian Goodfellow in 2014[11]. Later vanilla architectures kept improving, leading to models such as DCGAN[12], WGAN[13] and WGAN-GP[14]. These architectures solved vanilla GAN training stabilization problems but could not produce high quality images. The next big breakthrough was BigGAN[15] model that demonstrated that GANs could scale to the whole ImageNet dataset[16] containing over a million images while also achieving 128px output resolution. Subsequent substantial improvement in both training time and training stability was made by another architecture named ProGAN[17]. However, the training time of ProGAN[17] was still excessively long and required weeks to finish. This led to the final breakthrough and appearance of StyleGAN[4] — the architecture that significantly reduced the training time while preserving generative capabilities of its predecessors.

A. StyleGAN architecture

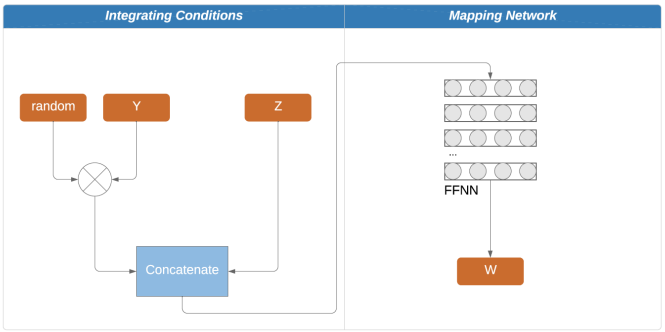

Fig. 3. Conditional StyleGAN input processing (source: LoGANv2: Conditional Style-Based Logo Generation with Generative Adversarial Networks[8])

StyleGAN is a progressive network(Figure 2)[4], meaning it is first trained on low-resolution images and higher resolution layers are added further. This idea is borrowed from ProGAN[17] and it plays an important role in stabilizing and speeding up the training. Progressive training also allows StyleGAN to use a unique way of using initial random noise vector. Instead of being passed directly to the generator, it is first mapped to low-dimensional intermediate latent space via feed-forward network[4]. Afterwards, the intermediate latent space representation is fed to each resolution stack of layers independently, allowing different types of details to be controlled by specific parts of the vector. Using adaptive instance normalization, yet another StyleGAN novelty, the network provides us with *style* control of both fine and coarse details via intermediate latent space vector, along with superior quality of feature disentanglement.

B. Conditional Stylegan

This paper utilizes StyleGAN Auxiliary Classifier(AC-StyleGAN) architecture which is implicitly supported by the official non-conditional StyleGAN implementation by Nvidia[18]. The model description does not mention it explicitly but StyleGAN can be configured for conditionality[19]. The most notable usage example is LoGANv2[8], which used Conditional StyleGAN to produce 32px logo images. MGM also utilizes auxiliary classifier GAN trained on animation data[5], however this model's architecture is different from StyleGAN. AC-StyleGAN has two major differences from default StyleGAN. First, label data must be passed together with random noise. LoGAN proposes to concatenate random noise vectors with label data in a form of one-hot

	Black hair	Blonde hair	Blue hair	Brown hair	Green hair	Pink hair	Purple hair	Red hair	Silver hair
Blue eyes	100%	100%	100%	100%	100%	100%	100%	100%	100%
Green eyes	100%	100%	100%	100%	100%	100%	100%	100%	100%
Red eyes	80%	100%	100%	100%	100%	80%	100%	100%	100%
Yellow eyes	100%	100%	100%	None	100%	100%	80%	100%	100%

TABLE I
DATASET TAG ESTIMATOR ACCURACY ON 5 SAMPLES

encoding[8](Figure 3). Afterwards, the concatenated vector is passed to the mapping network. Second, the adversarial loss of the StyleGAN, borrowed from WGAN-GP, has to be modified to account for labeled data[14][8]:

$$\nabla_{\theta D} = [f_{\theta D}(x|y) - f_{\theta D}(G(w|y))] + \lambda[(||\nabla_{\hat{x}} D(\hat{x}|y)||_2 - 1)^2] \quad (1)$$

It remains unclear what classification layers discriminator uses in this implementation. One-hot encodings imply that class labels must be mutually exclusive, which is a huge limitation as we will have to make each label encode multiple features and the model would not be able to learn those features separately. However, since LoGANv2 was able to build a model with good class separation even with the given limitation and there is no evidence that official StyleGAN implementation supports non-mutually exclusive class labels, we decided to embed classes as one-hot encodings as well.

C. Dataset

Obtaining a well-labelled anime dataset used to be a challenging task. MGM had to crop facial images from a gaming database website and then calculate crude embeddings with classification network for general illustrations(Illustration2Vec[20]) to extract approximate tags[5]. During the next several years publicly available datasets grew in both quality and quantity. Majority of them are sampled from terabyte-size dataset of crowd-tagged images collected from image board Danbooru[21]. This also allowed to train higher quality classification models, so that even non-tagged data could be used after processing with the classification model[22][23].

LoGANv2 introduced quite a creative approach of collecting categorical labels[8]: calculating image embedding with VGG[24] or ResNet[25] convolutional neural network and then applying k-means clustering to split the data in classes based on those embeddings. However, it is uncertain whether this approach could yield acceptable class separation on anime dataset since anime faces contain lots of plain spaces that create bias towards color-dominated separation. Moreover, major face features of every face are well distributed throughout the dataset, which means the clusters might not be well outlined with unrelated to color features.

III. DATASET COLLECTION

We use a large 512px resolution dataset of anime portraits from Kaggle as the dataset for our conditional model[26]. The dataset after cleaning contains over 300k of cropped faces. However, the data is not labelled, meaning that we have to use either image embeddings or image tag estimation to define the classes.

A. ResNet50 + clustering

The idea of calculating image embeddings with pretrained VGG/ResNet CNN and then applying on them k-means clustering was introduced in LoGANv2[8]. Potential advantage of this class extraction method is that the embeddings could capture meaningful features non-related to color. This in return means that those features could influence clustering and eventually lead to more complex classes. On the other hand, if the faces are evenly distributed then class separation by clustering is going to work poorly. For the sake of adequate computation time we sampled random 500 images from original 512px dataset and applied to it K-means with K=3,4,...,9. However, resulting clusters were not impressive. For the most part clusters were formed by the images of similar dominating color (e.g. hair color) and with low accuracy. A lot of images were misclassified by color and more complex tags did not seem to have any impact at all. Therefore a better solution has to be found.

B. Tag estimation

Compared to MGM that had to use a classification model for regular illustrations we have much more models at our disposal. One of them is a PyTorch ResNet classification model[23] pretrained on the Danbooru dataset which is the origin dataset of portraits we are using.

Ideally, this model should provide us with the same quality of labels as crowd-sourced tags that this model was trained on. Realistically, however, we are risking to introduce additional complexity to the problem. Features that are difficult to grasp for GAN can also be difficult for classificational CNNs and bad dataset labelling would further affect the output quality. With mutually exclusive classes we are also risking to significantly reduce the number of samples for some of the classes. Therefore, we decided to label the data with color-based classes.

Before we decide on the color tags we are going to use we first need to filter the dataset. The classifying model returns a vector with independent probabilities of each feature being present. We only consider tags with estimated probability ≥ 0.7. We filter the images without tags *solo*, *1girl* or *age_rating_s*. The first 2 serve as an additional filter to get images with strictly single faces, the latter is the tag indicating image board content safe for work. Afterwards, we exclude all instances that contain certain features: *eyes_visible_through_hair* or *third_eye*. Those tags indicate

Fig. 4. Labelled dataset samples, class *green hair + red eyes*

that the image is not quite suitable for our task. After that, we delete all the tags that are not related to hair and eye color. Those 2 types of tags are present in almost every picture so it would allow to keep dataset classes well represented. In order to reduce dimensionality of the model input we selected 9 hair color tags and 5 eye color tags. After filtering all the images with redundant tags we were left with 33065 images, number similar to what related research used[5][6].

Remaining samples are then clustered by tag pairs(*hair color + eye color*)(Figure 4). Each pair forms a class name which will be passed to a mapping network as a one-hot encoding together with random noise. In total we get 35 labels. We generated 5 samples of each class to check classification correctness(Table 1). 32 classes showed correctly labelled images and 3 remaining classes hit accuracy of 4/5. Obviously, images are distributed between the labels unevenly with 18 labels being assigned to less than 400 images each. However, those labels were deliberately not excluded so that we could investigate how quality of StyleGAN output depends on class size.

C. Manga dataset for transfer learning

We decided to use a custom manga dataset for working with monochrome images. This way not only we obtain images that are grayscale but also introduce an additional challenge. Manga drawings tend to be much more artistic than animation and have finer details. Combined with worse disentanglement of features based on color in black and white image it is unclear if StyleGAN could produce good results.

To built the dataset we scraped all chapters from popular manga series *Attack on Titan* by Hajime Isayama[27]. We used pretrained anime face detection model based on YOLOv3[28][29] and cropped out all the faces with sufficient resolution. This provided us with approximately 3700 images. After some manual cleaning this number was reduced to 1657.

D. Small colored dataset for transfer learning

To obtain a small colored dataset we took 55 images of a single character that were included in the baseline 512px dataset. The character is named Holo from a light novel *Spice and Wolf* by Isuna Hasekura and Jū Ayakura[30]. The dataset was then augmented to 400 samples. Such a small dataset size and limited number of sensible augmentation options(rotation, zoom and random crops) are a good representation of realistic circumstances. With this dataset it is probable that the model will be overfit so it is important to find correct training time to achieve a trade-off between baseline model potential output variety and inclusion of required character features.

IV. Metrics

To evaluate the models alongside making visual conclusions we will be using two metrics. The first is the Frenchet Inception Distance[31] which is a quantitative metric that most of the GAN research has been using to evaluate the similarity of original and generated image distributions. The second is the Truncation Trick[32] which is the name of a qualitative metric used to determine how diversified is intermediate latent space of the model.

A. Frechet Inception Distance

The Frenchet Inception Distance is defined by the following equation[31]:

$$FID = ||\mu_r - \mu_g||_2^2 + tr(\sum_r + \sum_g - 2 * (\sum_r \sum_g)^{\frac{1}{2}}) \quad (2)$$

The metric calculates the distance between two Gaussian distributions taking into account both their means and covariance. Here μ denotes mean and $\sum$ denotes covariance; r denotes real images and g generated images. The use of covariance is particularly the reason to use this metric; FID close to zero this means that generated images fully captured the original dataset diversity. To obtain the distributions we make use of a convolutional neural network Inception v3[33] trained on ImageNet[16]. We take the activations from the pool3 layer and estimate on them mean and covariance of each distribution.

B. Truncation Trick

Some regions of the model's intermediate latent space can produce worse results due to low representation in the original dataset[4]. Previous research have shown that truncating the latent space improves the output quality[32]. We calculate the mean of latent vectors generated by the mapping network and then use this mean to obtain modified latent vectors as follows[32]:

$$\hat{w} = E_z(FFNN(z)) \quad (3)$$

$$w' = \hat{w} + \psi(w - \hat{w}) \quad (4)$$

Fig. 5. *blond hair + blue eyes* sample evolution

Fig. 6. Final images, number of samples per class increases left to right: 20, 695, 1102, 3418, 7649. All images except the rightmost are misclassified.

	Black hair	Blonde hair	Blue hair	Brown hair	Green hair	Pink hair	Purple hair	Red hair	Silver hair
Blue eyes	100%	75%	75%	None	None	None	None	0%	100%
Green eyes	None	75%	None	25%	50%	None	None	None	None
Red eyes	75%	100%	75%	0%	0%	None	None	100%	None
Yellow eyes	None	50%	None	None	None	25%	None	None	None

TABLE II
STYLEGAN CLASSIFICATION ACCURACY ON 4 SAMPLES

V. EXPERIMENTS AND RESULTS

All models are trained with a single Nvidia V100 graphics card. We used network configuration for 256px resolution output with default batch size and learning rate.

While training, models' weights were consistently loaded into snapshot files, so we could evaluate how each model's output improved with time. This is particularly useful for evaluating transfer learning models. For example, if initially transfer learning exhibits faster conversion but then stagnates and reaches the same results as a model trained from scratch then transfer learning would still be beneficial because early convergence allows for more training transparency.

In total we do 3 experiments, one per each of the described datasets. We evaluate each model individually and then compare their FID. For each of the transfer learning experiments we also train a non-conditional model from scratch to measure if transfer learning provides us with faster resutls.

A. Conditional model

For the first experiment we trained Conditional StyleGAN model with 37 class labels and evaluated the results. The model was trained to 7000 kimgs(around 10 hours with V100 GPU). Its weights are then going to be used for transfer training other models. We can see how class separation evolved for the most represented class(*blonde hair + blue eyes*, over 7k samples out of 30k) in Figure 5. We observe that StyleGAN learns hair color especially well and exhibits early convergence. In case of facial images hair cover large areas and usually have distinct color which significantly facilitates StyleGAN learning. In case of eye color separation the model successfully learns eyes colors but not as fast as hair colors due to eyes features being smaller. This shows how convenient such models can be to work with since early label separation allows to test different labels without the need to train complete high quality model for several days or even weeks.

As for general features we can see that even though face, neck, hair shape and mouth are present in the image they all are not quite detailed. Particularly hair shape is fluid with hair branches below the neck taking very strange forms. The image contains a lot of artifacts with some of them resembling incorrectly drawn hair and others resulting from non-related backgrounds in the original dataset.

Afterwards, we explored final results for different labels(Figure 6) to assess the quality of class separation as well as find a threshold for minimum samples number per class. However, while investigating other classes we found that even those represented by more than 1000 samples are not perfectly classified. Majority of mistakes are occurring for eye colors. We can see in Table 2 that eye colors are learnt well only given a resembling hair color. Even some of the hair features are painted with slightly different colors than intended (red instead of brown, pink instead of red). Although worse than expected after investigating *blonde hair + blue eyes* output, class separation is still much better than random and we notice that color accuracy increases with number of samples per class

Fig. 7. *blonde hair* + *blue eyes* Truncation Trick ($\phi = 2, 1, 0.6, 0.04, -0.6, -1, -2$)

Fig. 8. Manga(monochrome) model output with $\phi = 0.6$

and reaches perfect accuracy for *blonde hair* + *blue eyes* tag. We also notice that 5 classes represented by less than 10 samples produce nonsensical output (although resembling faces in some cases), so we can ignore it. Even though as much as 50 samples can be enough to produce a distinct face, class separation and general image quality suffers severely up to 400 samples, consistently generating wrong eye color. We decided to set the threshold of 400 samples for further evaluation since lifting farther would eliminate many more classes.

As for general features and output quality we notice that the images are fairly diverse with variety of view angles and face positions. As we did not train a complete model we can expect that with further training will improve output diversity even further by learning to produce finer details. Moreover, after filtering out weakly represented classes we can see that, just like with *blonde hair* + *blue eyes* class, most of the images show distinct face, neck, hair, eyes and smile. On the other hand, majority of images suffer from hair shape fluidity and artifacts. A lot of images also show conflicts between closed mouth and open mouth features with StyleGAN trying to generate both at the same time.

Lastly, to evaluate latent space of the model we go back to *blonde hair* + *blue eyes* class and analyze truncation trick output(Figure 7). With truncation trick we observe that although our model cannot produce detailed images without artifacts, the images do not turn nonsensical with $\psi = 1$. Little quality difference between images generated with $\psi = 1$ and $\psi = 0.6$ is a solid sign that with more training we could effectively generate images with higher truncation value and, therefore, of higher diversity.

With such image quality it is evident that despite decent class separation the model has to be trained further in order to be used in practice.

B. Monochrome model

For the second experiment we used two models: a model which uses our conditional model as a basis(trained to 7500 kimgs, around 5 hours) and a non-conditional model trained from scratch on transfer learning dataset(again around 5 hours). It is only beneficial to use pretrained model basis if it provides results faster than a model trained from scratch. We could take a look at intermediate stages of baseline conditional model in order to approximate quality of the model trained from scratch, however, it can be that due to conditionality the model is progressing slower. Therefore we have to use a non-conditional model for comparison.

All images were assigned the label of black hair (without eyes color, this class was allocated earlier during conditional dataset labelling), we figured that such color choices would assist further transferring to black and white manga images. From the beginning of training it became evident that baseline conditional model was indeed beneficial for producing our target manga output(Figure 8). We samples 100 images from generator and around 10 images were heavy with artifacts but majority of other pictures were of good quality. A lot of output images resemble characters from manga based on which the dataset was constructed. Characters in output images show all different kinds of facial expressions. There are certain signs of overfitting becase of similar face positions but it can be due to limitations in view angles in the original dataset. We also cannot claim that the model is overfit before we train it to generate finer details with longer runs. However, within just 5 hours of training the model not only transfered to a new dataset but it also made significant improvements in comparison to the baseline model. Especially impressive on some images look hair, having much better outlined structure than with the conditional model.

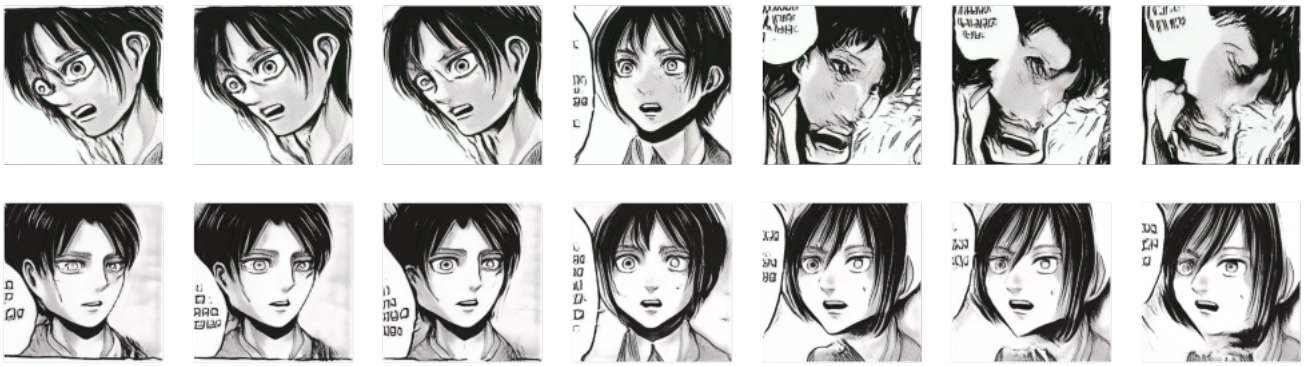

Fig. 9. Manga(monochrome) model Truncation Trick($\phi = 1, 0.8, 0.6, 0.04, -0.6, -0.8, -1$)

Fig. 10. Manga(monochrome) model interpolation($\phi = 0.6$)

Fig. 11. Holo model output with $\phi = 0.6$

To further analyze the images diversity we take a look at model input(random noise) interpolation(Figure 10) and Truncation Trick images(Figure 9). Given high quality first and last images for interpolation we observe that intermediate images are transforming into each other smoothly and are all of good quality. We used $\phi = 0.6$ for generating interpolated images and Truncation Trick shows that higher ϕ values would slightly lower the image quality.

C. Single character model

For the third experiment just like with monochrome transfer learning we train two models: a transfer learning model using the conditional model as a baseline and a non-conditional model trained from scratch. Both models are trained for approximately 5 hours. As a dataset we use our Holo colored images. We decided to use a class different from what would be correctly assigned to the character. This is due to Holo (the used character) being quite popular so initial facial dataset could already contain images of her and cause unnecessary bias. We mentioned earlier that initial 512px facial dataset had already contained images of Holo, however those images had very different filenames and were easily identified. There could be other images of our character that were distributed among samples with common filenames. Holo has brown hair and red eyes, so we decided to use $brownhair + greeneyes$ class instead. This is a very similar class due to hair color being a large feature. This class is well represented in the dataset and different eye color excludes unintentional images of our character in the samples of conditional dataset. Soon after starting the training we realised that convergence was very quick just like with monochrome images. General features were learnt during the first 30 minutes and more complicated features such as smiles were learnt over time. Overall resulting images quality is much better than with the model trained from scratch(Figure 11). From first observation it may seem as if the model is overfit since the output images are so similar to each other. However, after manually checking several output images with the samples in the dataset we found out that no output images can be linked directly to one sample in the dataset even though it certainly does resemble a group of samples. This is a good sign of feature diversity and model learning to combine knowledge about different images to generate unique results.

With interpolation(Figure 13) and Truncation Trick(Figure 12) we observe much better quality output than with the previous two models. Setting $\psi = 1$ does not reduce the output quality at all. However, later we find out that higher truncation value does not lower the model FID value so we decided to use $\psi = 0.6$ as a default truncation value for better stability.

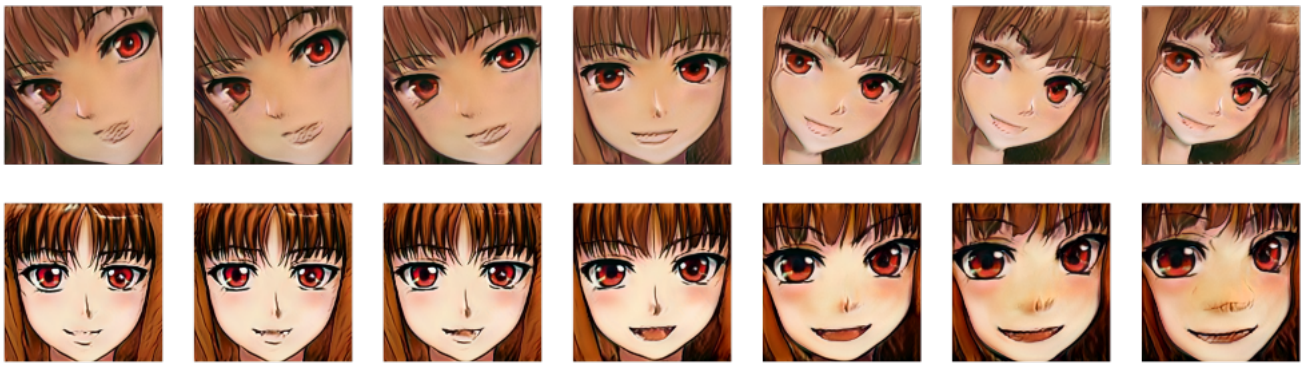

Fig. 12. Holo model Truncation Trick($\phi = 1, 0.8, 0.6, 0.04, -0.6, -0.8, -1$)

Fig. 13. Holo model interpolation($\phi = 0.6$)

D. Comparing FID

In order to estimate FID we use PyTorch port of official Tensorflow implementation[34]. As mentioned in the model description, due to differences in Tensorflow and PyTorch implementations the scores produced by them are slightly different. Majority of GAN papers use Tensorflow as a benchmark, however, since our models' FID is relatively high, small errors ≤ 1 do not matter as much and for convenience we use PyTorch port.

Calculating FID for each model we obtain the following numbers(Table 3): 64.128, 39.848 and 59.233 for conditional model, manga and holo models respectively. For reference, running FID between manga and Holo datasets yields FID score of 240. The numbers support our observations of significantly higher quality output with manga model than with conditional model. We can also see that both transfer learning models display better FID than the baseline conditional model. However, the score of the Holo model is much closer to the conditional model rather than the manga model. This is surprising since visually the Holo model produces the most detailed results with seemingly decent diversity. It might be that due to excessive data augmentation the model was not able to learn the original image distribution properly and indeed overfitted certain original image distribution regions.

	Frechet Inception Score
Experiment 1: Conditional Model	64.128
Experiment 2: Manga model(Monochrome images)	39.848
Experiment 3: Holo model(Single character)	59.233

TABLE III
FID SCORES FOR THE TRAINED MODELS GIVEN $\psi = 0.6$

VI. Conclusion

In this paper we explored performance of conditional StyleGAN model both as a complete model for generating conditional data and as a flexible basis for transfer learning as it provides many classes that could serve as a starting point for training. We found that ResNet50 embeddings combined with K-means clustering do not yield good class separation so we used tag estimators and clustered images based on hair and eyes colors together in order to gain sufficient number of labels while preserving the dataset size. After 10 hours of training the model exhibits good class separation on hair color but struggles with eye colors. Further training would be required for practical use of this model. With transfer learning we managed to drastically improve image quality and save crucial training time proving that conditional StyleGAN model could serve as a convenient baseline for further transfer learning. These two models, despite all their flaws thoroughly explored in the results section, are capable of generating visually pleasing results and can be useful for artistic purposes.

In the future we would like to address the models trained for longer time and to finer details as this time it was not possible due to low graphics card budget. We also would like to investigate possibilities of moving to multiple non-mutually exclusive classes. Ideally we would like to access crowd-sourced Danbooru tags instead of using a tag estimation model that introduces more error with complex tags. If successful, further advancements could make such models a useful tool for character design.

References

[1] Yamaguchi Yasuo. The evolution of the japanese anime industry, 2013.

[2] Jacqueline Ristola. Blood, sweat, ink, and tears: Exploitation of labour in the japanese animation industry. 2017.

[3] Gary Marcus, Ernest Davis, and Scott Aaronson. A very preliminary analysis of dall-e 2, 2022.

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2018.

[5] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the automatic anime characters creation with generative adversarial networks, 2017.

[6] Zepeng Liu. Generating full-body standing figures of anime characters and its style transfer by gan. 2020.

[7] Ajkel Mino and Gerasimos Spanakis. Logan: Generating logos with a generative adversarial neural network conditioned on color, 2018.

[8] Cedric Oeldorf and Gerasimos Spanakis. Loganv2: Conditional style-based logo generation with generative adversarial networks, 2019.

[9] Gwern Branwen. Making anime faces with stylegan, 2013.

[10] Likun Cai, Yanjie Chen, Ning Cai, Wei Cheng, and Hao Wang. Utilizing amari-alpha divergence to stabilize the training of generative adversarial networks. *Entropy*, 22(4), 2020.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[12] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015.

[13] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017.

[15] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2018.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2017.

[18] NVlabs. Stylegan — official tensorflow implementation. https://github.com/NVlabs/stylegan, 2019.

[19] Cedric Oeldorf. Conditionalstylegan. https://github.com/cedricoeldorf/ConditionalStyleGAN, June 2019. Accessed: DATE.

[20] Masaki Saito and Yusuke Matsui. Illustration2vec: a semantic vector representation of illustrations. *SIGGRAPH Asia 2015 Technical Briefs*, 2015.

[21] Anonymous, Danbooru community, and Gwern Branwen. Danbooru2021: A large-scale crowdsourced and tagged anime illustration dataset. https://www.gwern.net/Danbooru2021, January 2022. Accessed: DATE.

[22] Kichang Kim. Deepdanbooru, January 2020. Accessed: DATE.

[23] Matthew Baas. Danbooru2018 pretrained resnet models for pytorch. https://rf5.github.io, July 2019. Accessed: DATE.

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[26] Gwern Branwen, Anonymous, and Danbooru Community. Danbooru2019 portraits: A large-scale anime head illustration dataset. https://www.gwern.net/Cropsdanbooru2019-portraits, March 2019. Accessed: DATE.

[27] Hajime Isayama. *Attack on Titan*. Kodansha, 2009-2021.

[28] hysts. Anime face detector. https://github.com/hysts/anime-face-detector, 2021.

[29] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

[30] Isuna Hasekura. Jū Ayakura. *Spice and Wolf*. ASCII Media Works, 2006-present.

[31] Neal Jean. Fréchet inception distance. July 2018. Accessed: DATE.

[32] Marco Marchesi. Megapixel size image creation using generative adversarial networks, 2017.

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

[34] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.2.1.