Controlled Modification of Generated (Style)GAN Latent Vectors

Yifei Liu

June 28, 2019

Abstract

StyleGAN is a neural network architecture that is able to generate photo-realistic images. The diversity of generated images are ensured by latent vectors. These latent vectors encodes important features of generated images. They provide us insight-full information about properties of image generation in StyleGAN, which may also occur similarly in other neural network architectures. Using pre-trained StyleGAN models, we have conducted several experiments to show properties on StyleGAN based on results of style modification. The experiments have shown the influence of noise and styles in StyleGAN and how latent vectors can be manipulated in generated images.

1 Introduction

Generative Adversarial Network [7] (GAN) is a deep neural net architecture which is able to generate new data. It has been used in areas such as image generation [16, 2, 3], audio generation [4], language processing [13] and image-inpainting [14]. Recently, GAN architectures [16, 11, 12] have shown that they are able to produce photo-realistic images. However, the inner-workings of GANs are still not well understood. It is unclear where and how high-level attributes in images are represented. While most GAN can generate images, they do not enable controlled image modifications. For example, given a GAN that generate images of human faces, we can not control high-level attributes such as the skin colour in generated images.

StyleGAN has shown that it can produce photo-realistic images for human faces and different furniture objects. A StyleGAN has two inputs: noise vectors and a randomized latent vector. The noise is generated from normal distribution. It creates stochastic variations such as exact placement of hair or frickles within a face image. The randomized vector is sampled from normal distribution. It determines more global features in images such the shape of the head in face images. The latent vector and the noise ensures the variety in generated images.

There are currently three literature towards the research of StyleGAN. The original StyleGAN paper [12] has proposed and explained the architecture of StyleGAN. Image2StyleGAN [1] can convert colour images to StyleGAN vectors. StyleGAN-Encoder [15] has experimented to swap binary high-level attributes gender(male,female) and smile in face images by modifying the intermediate vectors in StyleGAN.

This work has three contributions:

- We identify effects of noise and style interpolation in StyleGAN networks. We also find the difference between Z and W in StyleGAN (Section 3).

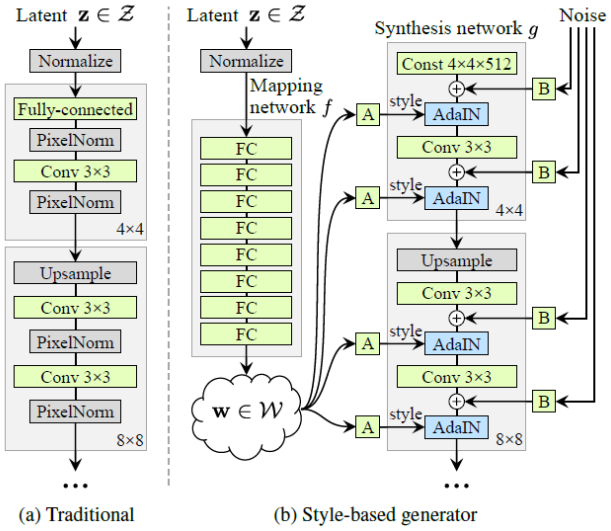

Figure 1: Figure retrieved from original StyleGAN paper [12]. Figure a shows the generator architecture of ProGAN [11]. A ProGAN has vector z as input and image as output. Figure b shows the generator architecture of StyleGAN. Compared to traditional generators, StyleGAN generators are divided into synthesis and mapping network. The output of the mapping network f outputs an intermediate w vector, this vector is used by the synthesis network g to generate new images. Other components in the figure are explained in Section 2

- We propose a simple method to retrieve encodings of high-level attributes such as gender and emotion in StyleGAN latent vectors (Section 4).
- We propose a method to modify high-level attributes such as gender and emotion in existing face images using extrapolation on StyleGAN latent vectors (Section 5).

In Section 6 the conclusion are given. First the necessary background and related works are explained in Section 2.

2 Related Works

GAN

GANs are first introduced by Goodfellow et al. in 2014 [7]. A traditional GAN consists of two separate networks: a generator and a discriminator. The generator generates samples similar to the training data, while the discriminator distinguishes training data from generated data. Using appropriate training methods and tricks, the generator will be able to generated 'fake' data similar to training data. In the following subsections, will will only concern with the generator.

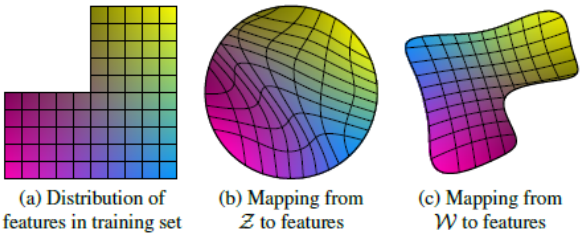

Figure 2: Illustrative example with two factors of variation (image attributes, e.g., masculinity and hair length). (a) An example training set where some combination (e.g., long haired males) is missing. (b) This forces the mapping from Z to image features to become curved so that the forbidden combination disappears in Z to prevent the sampling of invalid combinations. (c) The learned mapping from Z to W is able to 'undo' much of the warping. Retrieved from original StyleGAN paper [12].

StyleGAN Generator

StyleGAN is introduced by *Keras et al* from Nvidia Research in 2018 [12]. The generator is able to produce photo-realistic images for human faces, bedrooms, cars and cats. The authors of StyleGAN paper have published five pre-trained networks. The model with human face dataset have obtained highest FID score [12]. This model is intended to generate 1024×1024 human face images. It was trained on Flickr-Faces-HQ (FFHQ) dataset, which consists of 70,000 high-quality images at 1024^2 resolution.

The StyleGAN architecture is inspired by ProGAN [11]. The architecure of ProGAN generator is shown in Figure 1*a*. The generator has a vector z has input. The z vector will be propagated through the networks followed by series of operations such as convolution and normalization. We will not discuss this architecture in details since they are uninteresting for our experiments.

The architecture of StyleGAN is shown in Figure 1*b*. Compared to Progressive GAN generators, StyleGAN generators can be divide into two parts: a mapping network f and a synthesis network g.

The mapping network f is a Multilayer Perceptron with a latent code $\mathbf{z} \in Z$ as input. $\mathbf{z}$ is a 512×1 vector sampled from a normal distribution $N(0, 1)$. The mapping network outputs a style vector $w \in W$ which also has the shape 512×1. The purpose of the mapping network f is to let w learn the distribution of training dataset. For example, given a dataset where long haired male are missing, the mapping networks forces style vector w not to contain the combination of long hair and male. This constraint is harder to reinforce on the z vectors since they are sampled randomly from a Gaussian. Figure 2 shows an illustrative example of mappings from Z vectors and w vectors to binary features.

The synthesis network g creates new images using noise and style vector w. The synthesis network can be divided in blocks. The synthesis network g consists of 7-9 blocks depending on training image resolution ($256^2 - 1024^2$). Each block contains 6 operations: upscaling (1x), noise addition (2x), AdaIN (2x), and 3x3 convolution (1x). All blocks perform the same operations except the first block, where a constant tensor is initialized. In each block, vector w and noises are used as inputs. The learned affine transformation (shown as 'A' in Figure 1) converts vector w into 2 scalars: the style mean and the style variance. The

learned per-channel scaling (shown as 'B' in Figure 1) fits the noise shape to the propagating tensor and sum them. Each affine transform 'A' and scaling 'B' uses different weights. A final RGB convolution will be applied to the output which transforms the tensor into an RGB image.

AdaIN

In StyleGAN, the adapted style is injected into the synthesis network f through adaptive instance normalization (AdaIN). AdaIN is a type of normalisation proposed by Huang [8]. The AdaIN operation is defined as:

$$AdaIN(x_i, y) = y_{s,i} \cdot \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i} \tag{1}$$

Here $y_{s,i}$ stands for standard deviation of the adapted style, $y_{b,i}$ stands for mean of the adapted style, x_i stands for the feature tensor passed from the previous operations. μ stands for the mean and σ stands for the variance. The purpose of AdaIN is to pass the properties of the adapted style to the next style. The adapted styles are obtained from affine transformation of w vectors. These affine weights are trained.

Logistic Regression

Logistic regression is used for the classification of categorical problems. A Logistic regressor outputs a probability for a category based on the given data. The formula is defined as:

$$P(Y = 1|x) = \frac{1}{1 + e^{-\left(\alpha + x^T \beta\right)}} \tag{2}$$

Here x stands for the data vector, β stands for the weight vector and α stands for the bias of the regressor. Since that the regressor outputs the probability of a class, the output is bounded by [0,1].

Regularization is often used during the training to regressors and classifiers. Regularization discourages a high complexity of models and prevent overfitting on the training set. The objective is to have an accurate prediction of the data, while also having a relatively simple weights.

L1 regularization [17] is a regularization method and it penalizes the absolute value of weights. The advantage of L1 regularization is its' interpretability. Given a strong regularization parameter, the coefficients for unimportant features are shrunk to 0, and important features are distinguished from unimportant features. The regressor reduces the dimensionality of the dataset and yield to a more interpretable model compared to other regularization methods [9].

StyleGAN-Encoder

StyleGAN-Encoder architecture [15] utilizes StyleGAN to modify high-level attributes in existing face images. Coefficients of logistic classifiers are used to modify these attributes. The program can successfully modify human smiles and gender. An example is shown in 3.

StyleGAN-Encoder performs in the following steps:

- A logistic classifier will be trained with w as the classifier input. The weights of the classifiers are retrieved as direction vectors of the binary attribute.

Figure 3: An example of how a styleGAN varies the similing attibutes in face images. These images are generated with StyleGAN-Encoder [15]. Middle column shows the original image. The leftmost image shows the generated image without smiling. Rightmost image shows the generated image with smiling. The modifications are performed in w with regressor coefficients.

- Given an existing image, it is possible to retrieve $\tilde{w}$ latent of the image. Perceptual loss [10] are used to optimize the $\tilde{w}$ latent.
- Given an existing vector w, a new w' can be generated with $w' = w \pm \lambda \cdot d$. d stands for the classifier weights retrieved from the classifier, while λ stands for the strength parameter towards a class direction. During the generation of new images, both w and w' are used. Lower blocks in the synthesis network $(4^2 - 64^2)$ uses w' as the input style vector, high blocks in $(128^2 - 1024^2 2)$ uses w as the input style vector. The new image will show a preference towards a certain attribute, while maintaining other attributes from the original image. The new generated images are shown in 3.

The author of StyleGAN-Encoder has also published his training data, the data contains 20307 Z,W latent vectors encoding human faces and their corresponding labels. The labels are generated with Microsoft Face API [1].

3 StyleGAN Properties

3.1 Noise

Figure 9 shows the effect of noise inputs at different layers of the generator. The w vector was kept the same, different noises were applied to the same image. In the example we can see that noise do not affect the overall composition and the high-level identity in the image. Bed-shapes, skin colour are displayed consistently in all generated images. Coarse Noise (noises added at layer $2^2 - 32^2$) determines the exact placement of smaller object in images, while the fine noise (noises added at layer $62^2 - 256^2$) determines finer details such as textures and shadow boundaries in generated images.

[1] https://azure.microsoft.com/en-us/services/cognitive-services/face/

AUC	w **vectors**	z **vectors**
gender	0.907026	0.783424
young/old	0.894913	0.790319
glasses	0.927020	0.793470
smile	0.877382	0.713348

balanced accuracy	w **vectors**	w **vectors**
gender	0.907759	0.783334
young/old	0.899178	0.788602
glasses	0.940497	0.792567
smile	0.878751	0.713761

Table 1: Left table shows AUC score [5] of classifiers on binary attributes with Q or W as classifier input. Right table shows the balanced accuracy score on same classifiers. Scores are taken from 10-fold cross validation using L1 logistic classifier with $c = 0.1$.

Visually, noise in bedroom- and cat-StyleGAN has larger effect to the final image compared to FFHQ-StyleGAN. Face noise seems to effect local attributes and finer attributes more compared to generated bedroom- and cat images. For example, the eyebrow location is not influenced by noise in generated images. We hypothesize that effects of noise are also dependent on network parameters and training data: the original bedroom dataset contains low resolution images, while FFHQ dataset contains high resolution face images, so noise in bedroom-StyleGAN encode more diversities. Furthermore, the fixed shape of the objects in training set also contributes the consistency in the generated image. The cat dataset contains more varied poses and zoom levels compared to other dataset, so noises can effect larger attributes such as shapes in generated images.

3.2 Linearity of W

In StyleGAN, the mapping network f is used to transform Z that is drawn from normal distribution $N(0, 1)$ to some representation in W, such that w can be used to encode high-end attributes. It is hoped that attributes, such as gender, age are easily distinguished in this representation. This property can be tested with logistic regression. Logistic regressors linearly combines a set of characteristics in data to perform classification tasks. Simple logistic classifier should have better performance based on w instead of z. Table 1 shows the classification scores with W and Z as classifier inputs. For all four binary attributes, regressors based on W inputs performs significantly better compared to regressors based on Z inputs.

3.3 Style Interpolation

Style interpolation uses two (or more) existing latent vectors to generate new images. In style interpolation, vectors w_1 and w_2 are used to create a new style vector w by $w = \lambda \cdot w_1 + (1 - \lambda) \cdot w_2$. New image can be generated using w subsequently. This interpolation is also possible for z vectors.

Figure 8 shows the transition of style interpolation on W and Z. The figure shows that the interpolation can generate high-quality images across all StyleGAN types. Interestingly, both latents have produced smooth transitions during interpolation. However, it is noticeable that Z interpolation can generate extra features in transition images (baldness in row 1, cat shape in row 3), while W interpolation seems to be consistent across all transitions. We conjecture that extra feature do not appear in W latents due to the more linear nature of W.

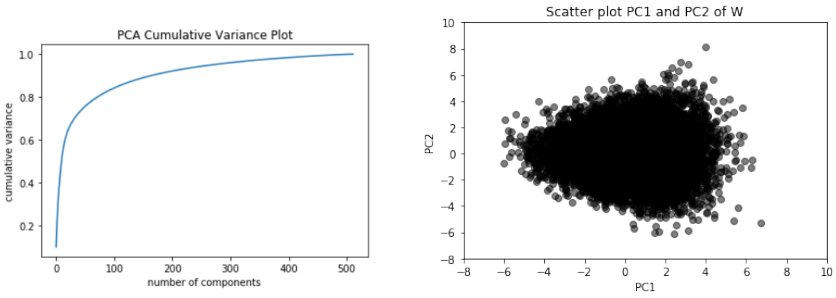

Figure 4: PCA on stylegan-encoder dataset.

Figure 5: The first column shows five images with smallest PC1 values from StyleGAN-Encoder dataset. The second column shows five images with largest PC1 values.

4 Interpretable attributes in W

In StyleGAN, W latent vectors encode high-level image attributes. It would be interesting to understand how these attributes are correlated with the latents. We use PCA [6] and logistic regression [9] to investigate this problem.

4.1 Principal Component Analysis

In StyleGAN, the vector w has 512 dimensions. We are interested if all dimensions are encoding attributes during the generation of the images. Principal Component Analysis [6] is a dimension reduction tool that can be used to reduce a large set of dimension to a smaller set while holding most information in dataset. The first principal component accounts for as much of the variability in the data as possible, each succeeding principal component account for as much of the remaining variability as possible.

Figure 4*a* shows the cumulative sum of the PCA explained variance of StyleGAN-Encoder dataset. The plot shows that first 100 principal components cover 80 % of all variances in the data. Figure 4*b* project the data vectors onto the first and second principal components. We can see that latents are clustered into 1 cluster. Figure 5 shows images with highest and lowest PC1 scores from the data set. The distinction between two groups are clear: images with low PC1 score contains more blue pixels, while images with high PC1 score contains more red pixels. The colour is represented instead of the shapes in the principal component.

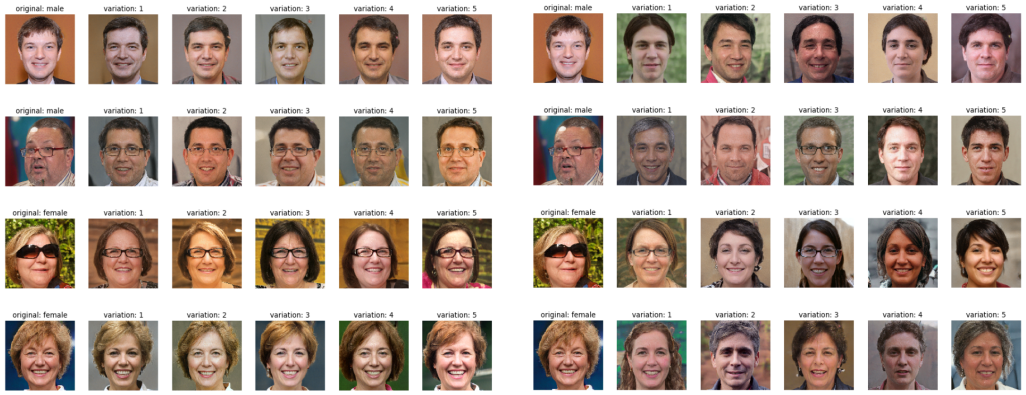

Figure 6: The figures show the effect of fixing values encoding gender in w. In both figures, the leftmost column shows the original image with labeled gender. Other columns show generated images with new generated w, where values encoding gender was fixed. In the left figure, 219 of 512 values in w are fixed. All images are photo-realistic while the gender are consistent across all images. In the right figure, only 54 of 512 values in w are kept the same, the gender is less consistent (4e image in row 1, 3rd image in row 4).

4.2 Direction Vectors

High-level attributes such as gender and emotions are individual factors hidden in W. It should be possible to find values in w that modifies these attributes. StyleGAN-Encoder shows the possibility to extract binary attributes with classifier weights, but it has flaws. First, StyleGAN-Encoder mixes extrapolated vector w' and original w to generate images, therefore it is hard determine which vector is truly responsible for changes. Furthermore, StyleGAN have shown that attributes are linearly separable [12]. This property is not utilized by the encoder.

Based on current state of StyleGAN-Encoder, The following changes were made:

- During the generation of new images, only extrapolated vector w' are used, so styles are not mixed during the generation of images.
- Since that w is linear separable on binary attributes, L1 regularization is added to the regressor. L1 regularization enforces a split between relevant and irrelevant values in W, thus direction vector should encode relevant directions exclusively.

Figure 6 shows different generated images by fixing values encoding gender in w. The figures show that the gender can be determined by 50 values in the latent vector. By fixing the values, we are still able to generate photo-realistic images, but the attribute (gender in this case) are consistent across all generated images.

5 Extrapolation of W vectors

Similar to StyleGAN-Encoder, we can also use direction vectors (Section 4.2) to modify existing images. We conducted vector extrapolation based on the high-level attributes.

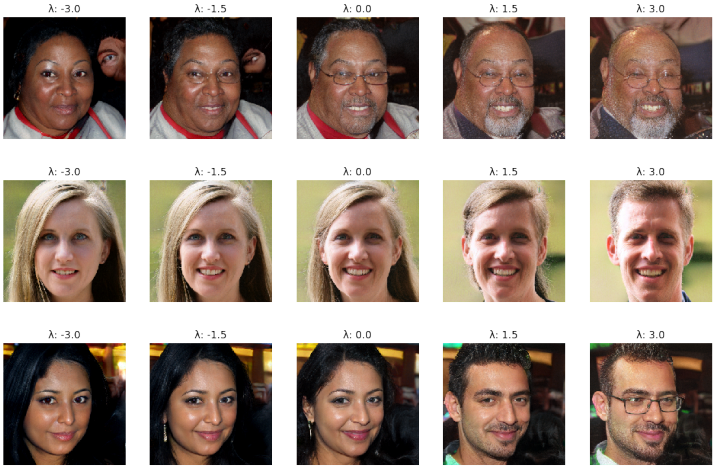

Figure 7: Results of extrapolation based on gender direction vectors. Middle column shows the original image. First and second columns show generated images with modified latents towards direction 'female'. Last two columns show generated images toward direction 'male'.

New images with direction vector and modified w' latents are created. The classifiers are trained with logistic regressors and L1 regularization.

Figure 7 has shown that we are able to create photo realistic with extrapolation on w vectors. The generated images contains the same background and skin colour, but the gender change in the images are clearly visible. Figure 10 shows generated images with different regularization parameter on classifiers. Lower regularization parameter implies that stronger regularization are applied, so less values in the vector are modified. In the last row, only 1 value in latent vector w is modified. The example shows that we are able to flip binary attributes in new generated images. The example also shows that the regularization parameter effects the resemblance with the original image. Generated images with weak regularization are more similar to the original image, while stronger regularization are less similar. Furthermore, the modified style vectors can also be combined using interpolation. The result of combined latents are shown in Figure 11.

6 Discussion

The results of Section 3 have shown that StyleGAN inputs enforce the diversity of generated images. The noise creates stochastic variations in images, while w vectors defines more visible attributes. The tests are performed on face-, cat- and bedroom generator. The generated images clearly shows changed attributes in images. We have also shown that high-level attributes are distinguished easier from w vectors rather than Q vectors. This is trained on 4 binary attributes of face latent vectors. However, it would be interesting if the distinguishable property also hold for continuous- or multi-categorical attributes, such as mouth size or hair colour.

We also performed PCA on values in w vectors. An interesting observation is that first 300 (of 512) principal components covers almost all variance in w. This indicates that w vectors can be reduced to a smaller size. Moreover, the results show that first two principal

components do not necessarily contain any high-level attributes of human faces images. PC1 only tells us about the colouring in the image. This raises an interesting question: how are the face attributes encoded within the principal components?

The extrapolation on w latent vectors have shown promising results. We are able to swap one or multiple high-level attributes given an existing w vector. Modified w vectors are still able output photo-realistic images (Shown in Figure 11). Unfortunately, the experiments are evaluated on a small scale, when labels are available the experiments can be extended to include more quantitative evaluations.

Prior methods such as [1] and [15] have brought us some insights into the inner-workings of StyleGAN. In this work, we have taken a small step towards understanding and manipulating the architecture of StyleGAN. There are still questions that we cannot yet answer. For example: can we distinguish any attributes in intermediate images? How likely can extrapolated images be generated using z and the mapping network? Further work will be needed to understand these properties of StyleGAN.

7 Conclusions

We have studied multiple aspects of StyleGAN properties such as linearity and style-interpolation. We performed experiments to validate these properties. Furthermore, we propose a simple method to modify high-level attributes in images using logistic regression. The method can classify on high-level binary attribute, such as gender and smile from intermediate vectors. Furthermore, the method has also shown the possibility to modify these vectors such that the high-level attributes in images are swapped.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? *CoRR*, abs/1904.03189, 2019.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018.

[4] Chris Donahue, Julian McAuley, and Miller Puckette. Synthesizing audio with generative adversarial networks. *CoRR*, abs/1802.04208, 2018.

[5] Tom Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, June 2006.

[6] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[7] Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017.

[8] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *CoRR*, abs/1703.06868, 2017.

[9] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[10] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

[11] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017.

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[13] Matt J. Kusner and José Miguel Hernández-Lobato. GANS for sequences of discrete elements with the gumbel-softmax distribution. *CoRR*, abs/1611.04051, 2016.

[14] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. *CoRR*, abs/1804.07723, 2018.

[15] Dmitry Nikitko. stylegan-encoder. https://github.com/Puzer/stylegan-encoder, 2018.

[16] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015. cite arxiv:1511.06434Comment: Under review as a conference paper at ICLR 2016.

[17] Fadil Santosa and William W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM J. Sci. Stat. Comput.*, 7(4):1307–1330, October 1986.

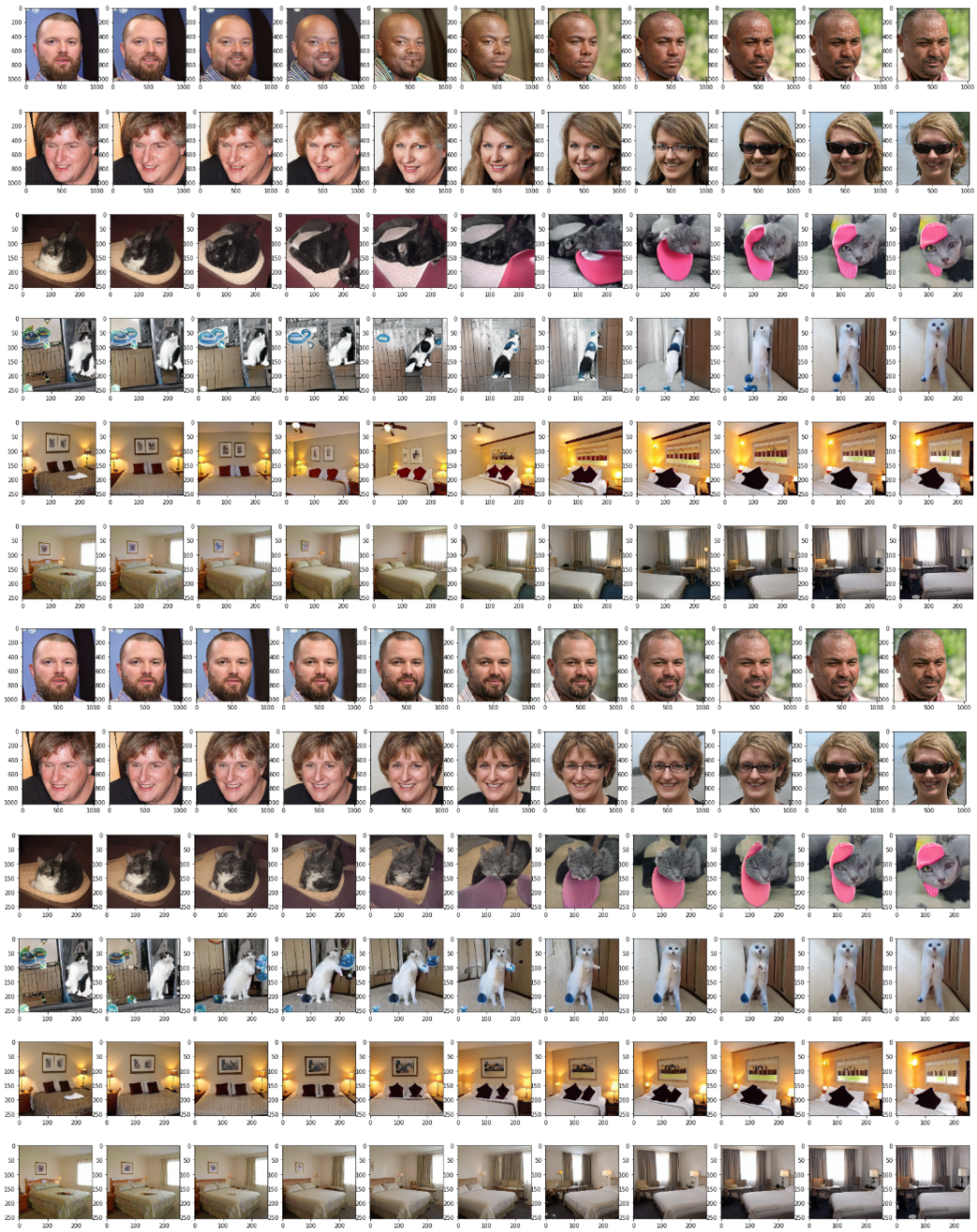

Figure 8: Output images of interpolation between two latents (left-most and right-most image). First 6 rows shows linear interpolation on Q latents. Last 6 rows shows linear interpolation on W latents. ffhq-StyleGAN, bedroom-StyleGAN and cat-StyleGAN are used to generate following images.

Figure 9: Effect of noise inputs at different layers of our generator. The first column: the original image without any noise. Second and Third columns: generated images with coarse noise only. Fourth and Fifth columns: generated images with fine noise only.

Figure 10: Extrapolated images with different direction vector. First row: 12 values in w encoding gender were modified. Second row: 7 values were modified. Third row: only 1 value was modified. Hyperparameter λ is fitted according to regularization.

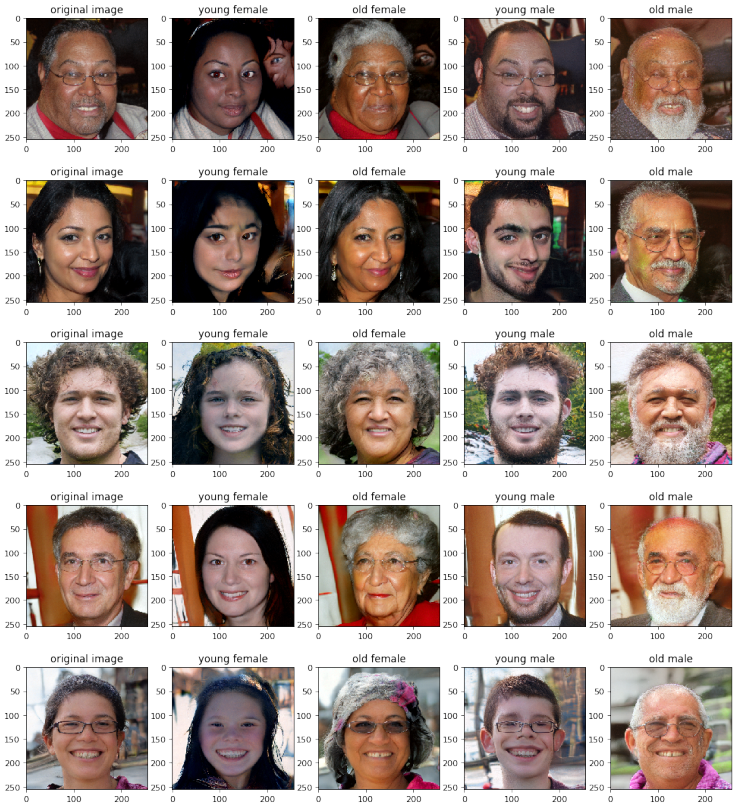

Figure 11: Results combining two extrapolated w latents with style interpolation.