Improved Multi-Domain Image-to-Image Translation GAN

Jeongik Cho

This work was supported by "University Innovation Grant" from the Ministry of Education and National Research Foundation of Korea

Abstract

StarGAN, which uses three important loss (adversarial loss, classification loss, reconstruction loss), has shown impressing performance in image-to-image translation. However, StarGAN has three important hyperparameters in loss (adversarial weight, classification weight, reconstruction weight), so it takes too much time to search for optimal hyperparameters. I propose an attribute loss, improved version of conditional GAN loss, which is like having multiple GANs, and simplified reconstruction loss, which uses the generator only once, to reduce hyperparameters and improve training speed. I also suggest image framing, not to distort the background, and bi-directional progressive-growing architecture, to improve training speed.

1. Introduction

StarGAN [1] uses an adversarial loss of WGAN-GP [2], reconstruct on the loss of CycleGAN [3] and classification loss of conditional GAN [4] to transfer the image to target domain without much distortion.

$$L_D = -L_{adv} + \lambda_{cls} L_{cls}^r$$

$$L_{cls}^r = E_{x,att\sim P_r(x,att)}[-\log(D_{cls}(\mathrm{att}|\mathrm{x}))]$$

$$L_G = L_{adv} + \lambda_{cls} L_{cls}^g$$

$$L_{cls}^g = E_{x',att'\sim P_g(x',att')}\left[-\log\left(D_{cls}(att'|x')\right)\right]$$

In $x, att \sim P_r(x, att)$, x means real data, and att is the binary vector that expresses the attribute of real data. In $x', att' \sim P_g(x', att')$, x' means generated data and att' is the target binary vector to make x' .

In the conditional GAN, adversarial loss trains model well because there are well known the loss such as LSGAN [5] or WGAN-GP that can produce meaningful gradients even if real data distribution and generated data distribution are far from each other. However, classification loss of conditional GAN, which is using cross-entropy is hard to produce meaningful gradients because of cross-entropy measures only the KL-divergence.

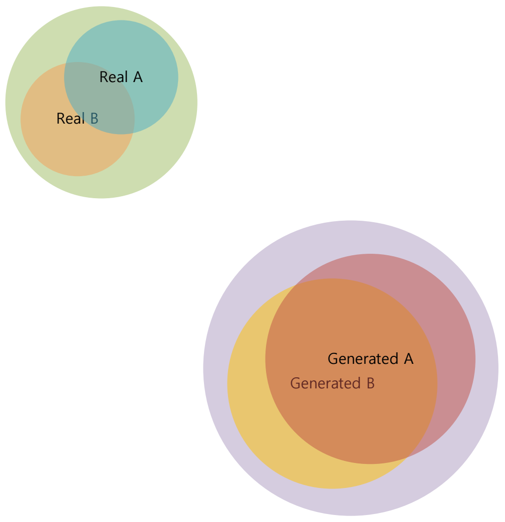


Fig1. Data distribution at the beginning of training

In the above figure, the circle containing Real A and Real B is the distribution of the real data, and the circle containing Generated A and Generated B is the distribution of the generated data. Real A is real data with attribute A and Generated A is data generated by the generator with condition A. In the early stage of learning, the classification loss does not produce meaningful gradients because the distance between the real data distribution and the generated data distribution is far. Only adversarial loss produces meaningful gradients. Intuitively, when GAN generates only noise-like images at the beginning of the training, reducing classification loss will not help GAN to train.

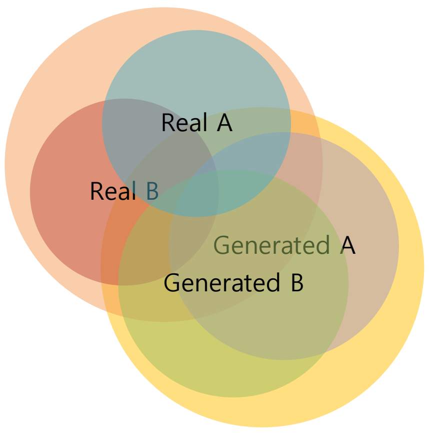


Fig2. After long training

As the learning progresses to some extent, the actual data distribution and the generated data distribution are somewhat similar, and classification loss starts to produce meaningful gradients when each conditional data distribution overlap (Real A with Generated A, Real B with Generated B).

To solve the problem that classification loss does not have meaning at the beginning of learning, I propose attribute loss, which is similar to having many GANs that each GAN learns only one attribute. Each Generator only generates data with each attribute. Each Discriminator determines that it is true only for the real data with each attribute and that it is a fake for the data that the generator generates for each attribute. Attribute loss can replace adversarial loss and classification loss of StarGAN.

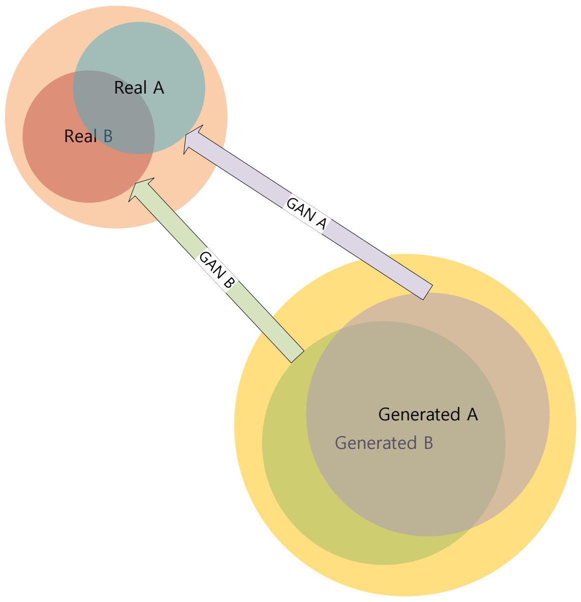


Fig3. Multi-GAN loss

Attribute loss is the sum of each GAN loss. Each GAN has its adversarial loss. Therefore, using LSGAN loss or WGAN-GP loss for each GAN can generate meaningful gradients at the beginning of learning. Since each discriminator shares all layers except the output layer, and each generator shares all layers except the input layer, the training time does not increase significantly.

Also, attribute loss can replace adversarial loss and classification loss. Original StarGAN needs three important hyperparameters (adversarial weight, classification weight, reconstruction weight). Reducing one hyperparameter by replacing adversarial loss and classification loss with attribute loss can dramatically reduce the time to search for optimal hyperparameters (only need to search for attribute weight and reconstruction weight).

In the original StarGAN paper uses reconstruction loss from CycleGAN.

$$L_{rec} = E_{x,att\sim P_r(x,att)}[||G(G(x,\ att'),att) - x||_1]$$

In $x, att\sim P_r(x, att)$, att is the original attribute of real image x, and att' is the target attribute. Generally, use a random binary vector for training.

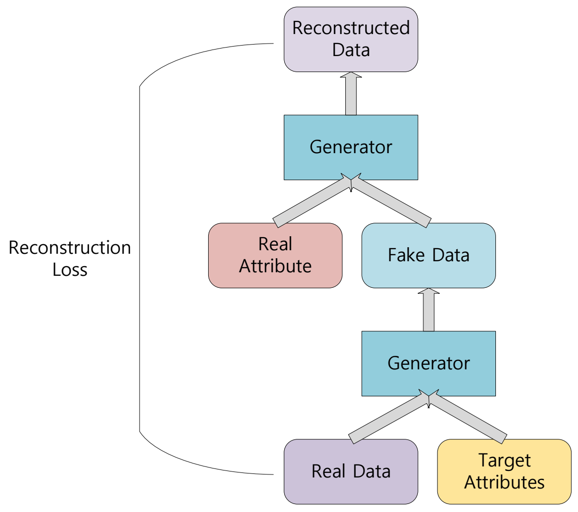


Fig4. Original reconstruction loss of StarGAN

To ensure that the generated data is not too different from the original data, StarGAN uses the Reconstruction loss of CycleGAN. Reconstructed data should be similar to original data.

However, since the original image and the generated image need only be somewhat similar, the real data does not have to go through the generator twice.

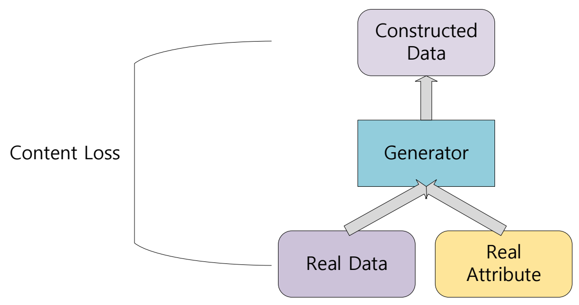


Fig5. Simplified content loss

The amount of computation can be reduced by simplifying the reconstruction loss as fig5.

2. Improved Star GAN

First, it is assumed that attribute information is matched with real data.

2.1 Loss

Overall Loss is as follows.

$$L_D = L_{att}^D$$

$$L_G = L_{att}^G + \gamma_{cnt} L_{cnt}$$

Attribute Loss

Attribute loss is as follows.

$$L_{att}^D = \sum_c^{att} L_c^D$$

$$L_{att}^G = \sum_c^{att} L_c^G$$

$$L_c^D = E_{x,c\sim P_r(x,c)}[(D_c(x)-1)^2] + E_{x'\sim P_{G_c}(x',1)}[D_c(x')^2]$$

$$L_c^G = E_{x\sim P_r(x)}[(D_c(G_c(x,1))-1)^2]$$

c means one specific attribute among several attributes. L_c^D and L_c^G are the losses of one discriminator and one generator that discriminate against a particular attribute c. L_{att}^D is the sum of the attribute losses of all discriminators and L_{att}^G is the sum of the attribute losses of all generators.

G_c is a generator that converts an image x to have an attribute c when the image x and binary value 1 are received as inputs. G_c tries to trick D_c only if binary value 1 is entered with x, and does not care if 0 is entered (not learn).

D_c determines only about attribute c. D_c discriminates real only for real data with attribute c and doesn't care about real data without attribute c and determines fake when received the fake image from G_c that receives real image x and 1.

L_{att}^D is the sum of each discriminator. Each discriminator shares all layers with other discriminators except the output layer. By considering a set of discriminators as one discriminator, the loss can be changed like below.

$$L_{att}^D = E_{x,att\sim P_r(x,att)}\left[(D(x)-1)^2 \cdot att\right]$$

$$+E_{x',att'\sim P_g(x',att')}[D(x')^2 \cdot att']$$

In $x, att \sim P_r(x, att)$, x is the real image, and att is attribute binary vector. '$\cdot$' means inner product.

In $x', att' \sim P_g(x', att')$, x' is generated image and att' is a binary attribute vector input generator to make x'.

Since each generator also shares all layers except the input layer, L_{att}^{G} can be written as the following by considering a set of generators as one.

$$L_{att}^{G} = E_{x\sim P_r(x)}[(D(G(x, att')) - 1)^2 \cdot att']$$

att' is a binary vector representing the attribute you want to change in the real image x. Use random binary vectors for training.

This is an example of using the least square loss as an adversarial loss. Wasserstein-GP or other adversarial loss can be used for attribute loss.

Incidentally, $G_c(x, 0)$ does not convert x to x' that doesn't have attribute c but simply disables G_c. Therefore, if you want to remove attribute c from image x, you need to add the attribute 'not c' while training.

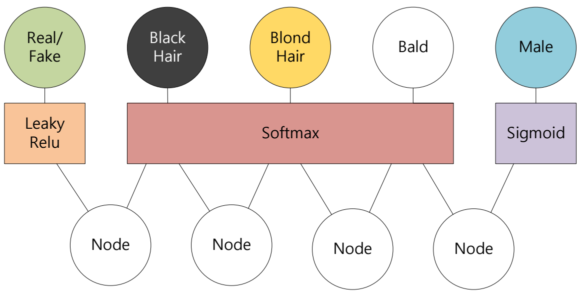


Fig6. Discriminator output of StarGAN example

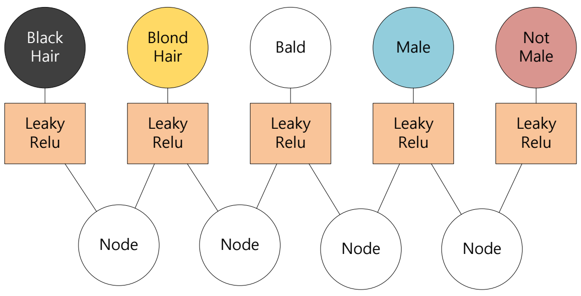


Fig7. Discriminator output with attribute loss example

(Assume P(Black Hair) + P(Blond Hair) + P(Bald) = 1, P(Male) + P(Not Male) = 1)

Content Loss

The original reconstruction loss of StarGAN is as follows.

$$L_{rec} = E_{x\sim P_r(x)}[||G(G(x,\ att'), D(x)) - x||_1]$$

In $x \sim P_r(x)$, x is the real image. att' is a random binary attribute vector.

To calculate original reconstruction loss of StarGAN, original data should pass generator two times. I used simplified content loss, which is a simplified version of reconstruction loss to reduce calculation.

$$L_{cnt} = E_{x,att\sim P_r(x,att)}[||(G(x,\ att)) - x||_1]$$

In $P_r(x, att)$, x is real image. att is binary attribute vector of real image x.

Image Framing

As image completion imply [10], while training, framing the generated image with the original image makes generated image match to the background of the original image.

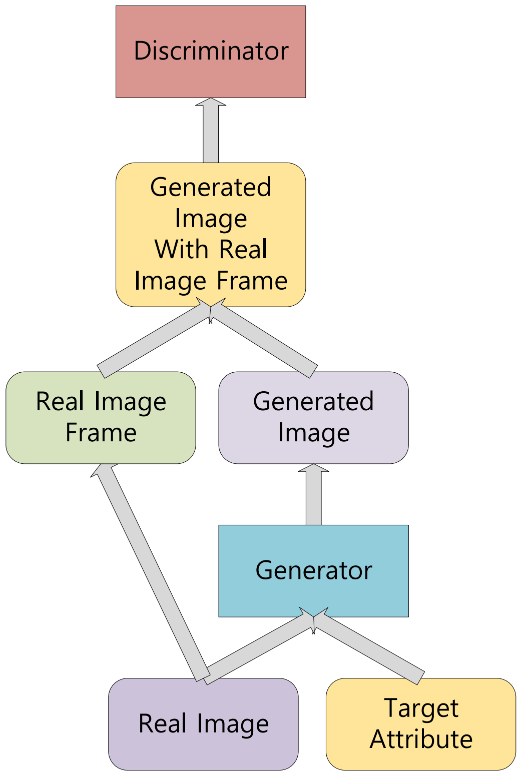


Fig8. Image Framing

2.2 Architecture

Generator

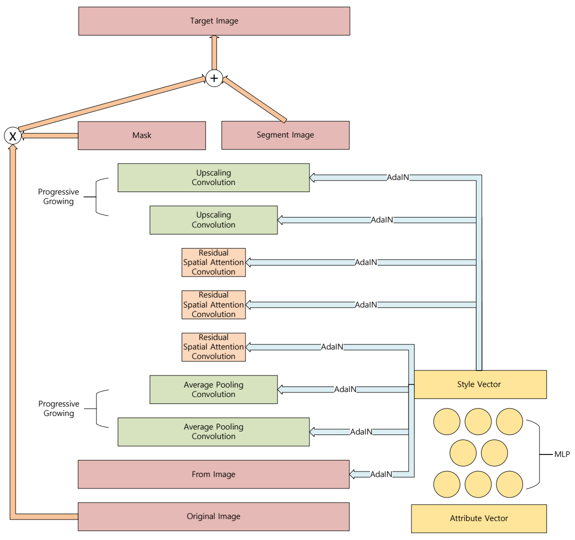


Fig9. Generator Architecture

In generator architecture, AdaIN module and embedder of Style-based generator [6], mask of CAGAN [7], and convolution block attention module of CBAM [8] was used. There is no batch normalization in generator.

To improve the learning speed, I suggest a bi-directional progressive growing generator, which grows in both input and output directions, not just in one direction.

Discriminator

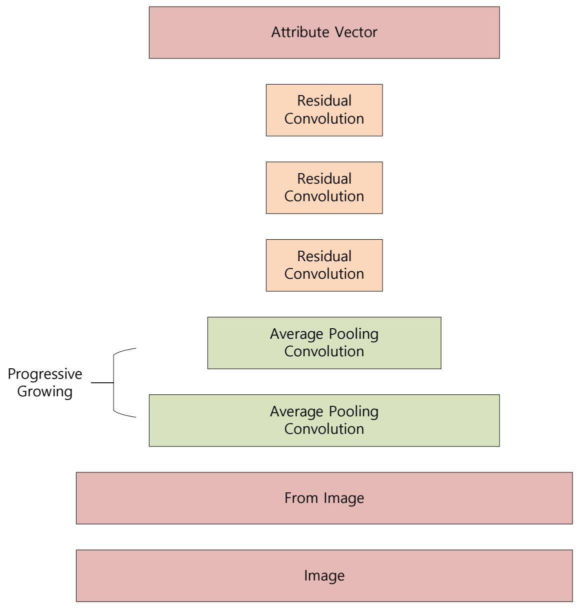


Fig10. Discriminator Architecture

Discriminator has attribute outputs that each output discriminates whether real image with each attribute or generated image with each attribute. Batch normalization was applied between each layer.

Mixed data training

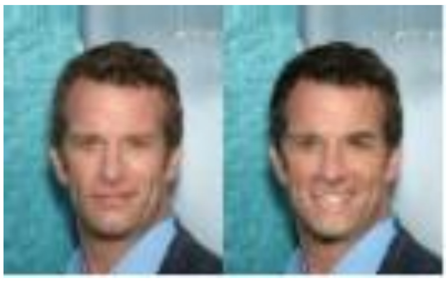
Recently, using batch normalization has become orthodox. However, using batch normalization in discriminator could be dangerous because the attribute distribution of generated data and real data could be different. Suppose the ratio of attribute A and attribute not A in real data is 3:7. If the ratio of attribute A and not-A is 5:5 in the target attribute for the generator, the generator will be trained to generate the ratio of attribute A and not A become 3:7 if there is batch normalization in the discriminator. This means some generated data with target attribute A input could have attribute not-A. To avoid this problem, I suggest using a mixed batch of real data and fake data entered into the discriminator.

3. Experiments

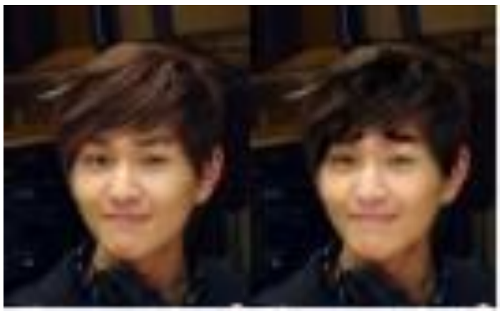
Used celeb_a [9] training dataset (162,770 pictures with attribute label) while training. Used celeb_a test dataset (19,962 pictures with attribute label) for test.

Model was trained almost four hours on two rtx2080ti. Resolution of all image is 72 by 88.

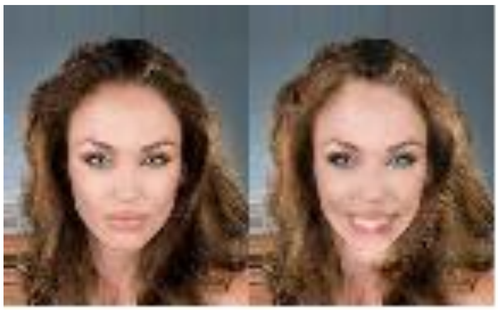
All left pictures are original pictures and right pictures are generated pictures



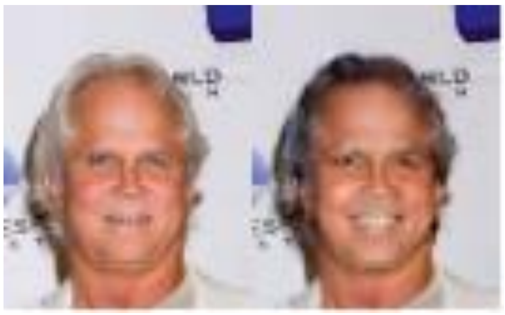
Target Attribute: black hair, mouth slightly open, smiling.



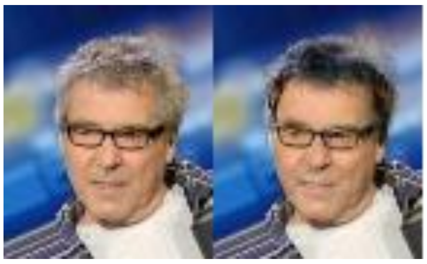
Target attribute: black hair, not mouth slightly open, smiling



Target attribute: blond hair, mouth slightly open, smiling



Target attribute: black hair, mouth slightly open, smiling



Target attribute: black hair, not mouth slightly open, smiling

References

[1] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo

StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation

https://arxiv.org/abs/1711.09020

[2] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville

Improved Training of Wasserstein GANs

https://arxiv.org/abs/1704.00028

[3] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

https://arxiv.org/abs/1703.10593

[4] Mehdi Mirza, Simon Osindero

Conditional Generative Adversarial Nets

https://arxiv.org/abs/1411.1784

[5] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, Stephen Paul Smolley

Least Squares Generative Adversarial Networks

https://arxiv.org/abs/1611.04076

[6] Tero Karras, Samuli Laine, Timo Aila

A Style-Based Generator Architecture for Generative Adversarial Networks

https://arxiv.org/abs/1812.04948

[7] Nikolay Jetchev, Urs Bergmann

The Conditional Analogy GAN: Swapping Fashion Articles on People Images

http://openaccess.thecvf.com/content_ICCV_2017_workshops/papers/w32/Jetchev_The_Conditional_Analogy_ICCV_2017_paper.pdf

[8] Sanghyun Woo, Jongchan Park, Joon-Young Lee, In So Kweon

CBAM: Convolutional Block Attention Module

https://arxiv.org/abs/1807.06521

[9] Ziwei Liu Ping Luo Xiaogang Wang Xiaoou Tang

Large-scale CelebFaces Attributes (CelebA) Dataset

http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[10] Satoshi Iizuka, Edgar Simo-Serra, Hiroshi Ishikawa

Globally and locally consistent image completion

https://dl.acm.org/citation.cfm?id=3073659