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

20170264 Seungmin Park, 20190006 Sana Kang,
20190545 Seungjae Lim, 20200673 Hyunmin Choi

Abstract

Research before StarGAN had excellent results in image-to-image translation, but these approaches have limitations, which leads to independently building different models for every image domain pair, which requires processing three or more multi-domains. When doing so, there was a problem with limited scalability and robustness. To solve this problem, a stargan that can perform image-to-image translation for multiple domains using a single model was presented, and in this project, we want to reproduce and improve this stargan.

The unified model architecture of StarGAN enables simultaneous training of data sets from different domains within a single network, showing superior translation quality compared to the existing model and scalability and robustness by flexibly executing changes between domains. In this project, we reproduce and play StarGAN using pytorch. Then, after training using the CelebA dataset and RaFD, we would like to see the transformation result.

Based on this project, we found a section where vanilla StarGAN's performance decreased, analyzed the cause, and tried to come up with a solution. Our model is struggling for some specific class generation. Therefore, it can be seen that the current model has a low performance in finding geometric or structural patterns in faces. Therefore, we proposed self-attention as a solution to this.

1 Introduction

Image-to-Image translation, a topic we will cover in the paper we will reproduce, is the problem of changing one specific feature of a given image into another. This has been greatly advanced with the introduction of the Generative Adversarial Network (GAN). In particular, the problem solved in this paper is a multi-domain problem that converts two or more domains at the same time.
Although multi-domain translation was possible in previous studies, there were several problems. First, the model needs a k(k-1) generator to learn the mapping to the k domain, which is expensive to compute. Second, post-training models are inefficient because each generator does not share global features, so it cannot fully utilize all the training data. This is because, if the training data is not fully utilized, the translation quality will be lowered. Third, since each dataset is labeled separately, it is difficult to combine it into the learning domains of other datasets. Therefore, there were inefficiencies due to the lack of scalability for training data or for adding or changing features.
So, we build a novel generative adversarial network that learns mappings between multiple domains using a single generator and discriminator to reproduce the paper that solved these problems. Successfully learn multi-domain image transformations between multiple data sets by utilizing the mask vector method. In this reproduction, we plan to reproduce it using CelebA and RaFD. After training the reproduced model with CelebA and RaFD data, we plan to show that it is superior to the basic model through qualitative and quantitative comparison of the transformation results of other trained models.
Therefore, in this paper reproduction, it is meaningful to provide a model solution using one

generator for multiple domains, and to show simultaneous learning and scalability between multiple domains. This is also meant to secure the problems of previous papers and to understand why the limitations of the previous papers were overcome.

2 Theoretical Background

Since the concept of Generative Adversarial Network (GAN) was first introduced to academia, various variations of GAN have emerged. To figure out common limitations several GAN models had shared, we conducted a preliminary study on the following models: Pix2pix, CycleGAN, and WGAN.

1.1 Pix2pix

Pix2Pix refers to an image-to-image translation GAN that learns a mapping from an image X with a random noise Z to the output image Y. It was designed in the early stages of i2i translation right after the advent of cGAN. To learn this model, we need to prepare the paired data set that consists of two different domains – X (the edge information) and Y (the original data). This edge information X is considered as the condition itself as in the conditional GAN. Therefore, for the Pix2pix model, we need to make the edge information of each data on our own.

Especially, Pix2Pix uses both generator and discriminator with the module of the form convolution BatchNorm ReLu. In this network, the input goes through a series of multiple layers in two main directions: towards a bottleneck layer and backward from it. This mechanism is to deal with a great amount of low-level information of the input and output across the network during image translation. Details of the learning process of Pix2pix are as follows. Firstly, Pix2pix uses Generator G to create a fake image G(x) that meets condition x (edge information). After that, the fake image G(x) and condition x are put together in Discriminator D. Based on it, the model carries out the learning process to make D figure out generated image G(x) as a fake one. Conversely, for the G, the model tries to make it generate fake images which can deceive D as real ones. However, the limitation of Pix2pix is that, in reality, there exists a considerable amount of unpaired datasets, and it is also very costly to make paired data sets one by one.

1.2 CycleGAN

The Cycle Generative Adversarial Network, or CycleGAN, aims to train a deep convolutional neural network (CNN) for i2i translation tasks. In the learning process of this model, the image G(x) made by the generator becomes reconstructed back into the original image x. This whole process with the cycle-consistency loss enables the cycle GAN to change domain-related features effectively while preserving the content of the original image. For this purpose, cycleGAN uses two translators, G and F. Translator G is responsible for the conversion from X to Y; while Translator F is responsible for the conversion from Y to X in reverse.

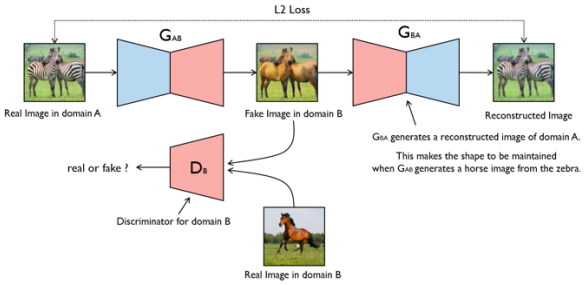

Figure 1: The process of CycleGAN.

However, there exist some cases where CycleGAN falls in a vicious loop – repeatedly presenting only one image corresponding to a particular domain regardless of input. For example, suppose you created a new zebra picture by adding a new attribute to the horse picture. In this case, for the corresponding fake image, the discriminator does not find the image as a fake one because it is very likely to exist. In this case, the generator eventually continues to generate that exact image that can deceive the discriminator over and over.

1.3 WGAN

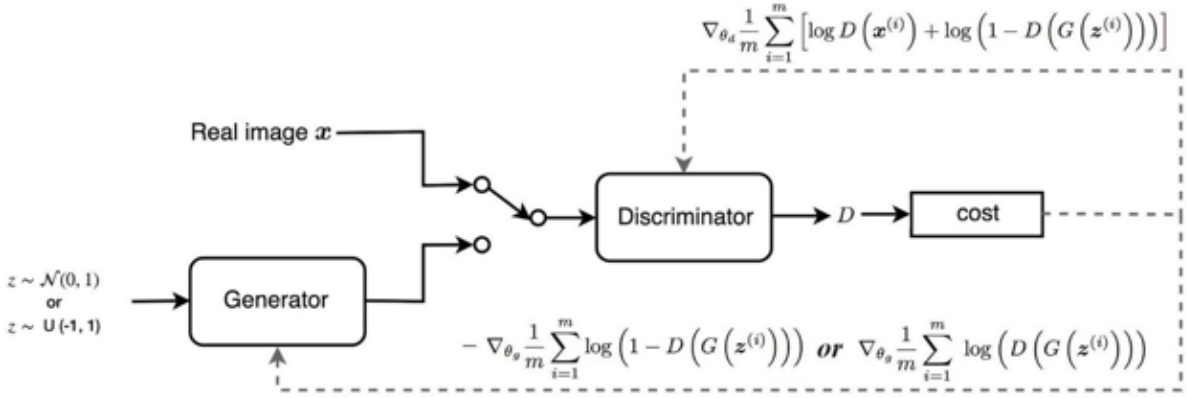

Figure 2: The process of WGAN.

WGAN (Wasserstein GAN) refers to a type of GAN that minimizes an approximation of the Earth-Mover's distance rather than the vanilla GAN with the Jensen-Shannon divergence. Likewise, it aims to solve the problems of the existing vanilla GAN. Those problems were [1] the difficulty of balancing between the discriminator and generator and [2] the occurrence of mode dropping even after learning is completed. The reason for these problems is that the vanilla GNN model was not able to learn itself to the optimum level because of the lack of ability of the discriminator. To solve it, WGAN uses the newly defined critic instead of the discriminator. While the discriminator uses a sigmoid function to determine whether the generated image is true or fake, the critic uses scalar values from the Earth Mover distance, the measure to calculate the distance between probability distributions. However, there exists a weight clipping issue in the WGAN, so WGAN-gp introduces the gradient penalty to solve it.

3 Challenges

After conducting the preliminary study, we summarized the following two issues as limitations that the previous GAN models had in common. The first limitation is that it is difficult to maintain the balance between the discriminator and generator when training a model. The second limitation is that no model has proposed an efficient method for efficient image-to-image translation in multiple domains.

4 StarGAN Design

So far, we describe about I2I-translation related works and limitations of them. From now, we will discuss about how StarGAN (Star Generative Adversarial Networks) architecture and loss functions are composed. Also, we will introduce how StarGAN manipulates multiple datasets containing different labels to perform image translation using them.

Like other image translation networks, StarGAN consists two modules, a discriminator D and a generator G. Main purpose of discriminator is to distinguish between real and fake image and classify the image which is judged to be real into its corresponding domain. Generator takes two inputs, image and target domain label, and generates a fake image.

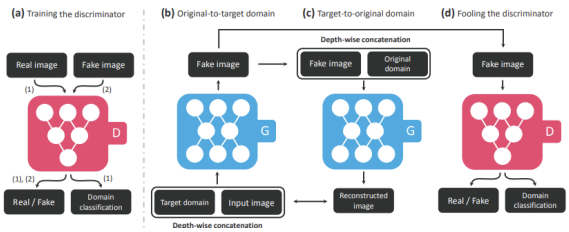

Figure 3: Overview of StarGAN training process.

4.1 Network architecture: Loss functions and Implementation

As mentioned before, we train G to translate an input image x into an output image y on target domain label c, $G(x, c) \rightarrow y$. In training phase, we will randomly generate domain label c so that G learns flexible translation. In case of discriminator D, it will produces probability distribution over

source images and domain labels, $D: x \rightarrow \{D_{src}(x), D_{cls}(x)\}$.

Adversarial Loss. To make the generated images indistinguishable from real images, we adopt an adversarial loss something like this. G tries to minimize this objective, while D tries to maximize it.

$$L_{adv} = E_x[\log D_{src}(x)] + E_{x,c}[\log(1 - D_{src}(G(x,c)))]$$

Domain Classification Loss. Given input x and target domain label c, starGAN should generate output image y which is properly classified into target domain. To achieve this, we need additional loss functions to optimize both D and G. In detail, domain classification loss for discriminator is defined below.

$$L_{cls}^{r} = E_{x,c'}[-\log D_{cls}(c'|x)]$$

By minimizing this term, D could learn to classify a real image x to its original domain c_prime. On the other hand, the loss function for generator is defined below.

$$L_{cls}^{f} = E_{x,c}[-\log D_{cls}(c|G(x,c))]$$

This term will help for distinguishing domain of fake images. By minimizing this term, G could learn to generate images that can be classified to target domain c.

Reconstruction Loss. To guarantee that translated images preserve the content of its original image while translation proceeds, we apply a cycle consistency loss to generator which is defined below.

$$L_{rec} = E_{x,c,c'}[|x - G(G(x,c),c')|_1]$$

We adopt the L1 norm as our reconstruction loss. Note that this is not for separate generators, but for single generator which translate in bi-direction.

Full Objective. Combining all loss functions we discuss about so far, the final object functions are defined as,

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

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

where two lambdas are hyper-parameters that control the relative importance of domain classification and reconstruction losses. We use $\lambda_{cls} = 1$, $\lambda_{rec} = 10$ here.

Improved GAN Training. For higher quality image generation, we modified adversarial loss function to Wasserstein GAN objective with gradient penalty, which is defined below.

$$L_{adv} = E_x[D_{src}(x)] - E_{x,c}[D_{src}(G(x,c))] - \lambda_{gp} E_{x'}[(|\nabla_{x'} D_{src}(x')|_2 - 1)^2]$$

Here, x' is sampled uniformly along a straight line between a pair of a real and a generated image. We use $\lambda_{gp} = 10$ for all experiments.

Network Architecture. Adapted from CycleGAN, StarGAN has generator network composed of two convolutional layers with stride size of two for down-sampling, six residual blocks, and two transposed convolutional layers with stride size of two for up-sampling. For generator, instance normalization is applied.

4.2 Training with Multiple Datasets

Strength of StarGAN is that it can incorporate various kind of datasets containing different types of labels, so it could control all the labels at the test phase. But, label information is only partially known to each dataset.

Mask Vector. To resolve this problem, we use a mask vector that allows StarGAN to ignore unspecified labels and focus on the explicitly known label.

Training Strategy. At training phase of StarGAN with multiple datasets, we use domain label concatenated with masked vector as input to the generator. By doing so, generator learns to ignore the unspecified labels, and focus on the explicitly given label. In discriminator's perspective, discriminator will try to minimize only the classification error associated to the known label.

Under multi-dataset training, discriminator will learn all of the discriminative features for all datasets.

5 Experimental Results

In this section, we first compare our replicated StarGAN against recent methods including original StarGAN on facial attribute transfer by conducting user-studies. Next, we prepare a new dataset named AffectNet, to check that our model works well on another dataset rather than RaFD and CelebA. Lastly, we demonstrate empirical results that StarGAN can learn image-to-image translation from multiple datasets.

5.1 Baseline Models

As our baseline models, we adopt DIAT and CycleGAN, both of which performs image-to-image translation between two different domains. We also adopt IcGAN as a baseline which can perform attribute transfer using a cGAN.

For comparison, we have to implement all of these models and train models multiple times for every pair of two different domains. However, due to lack of time to implement all of baseline models and long training time, the output of baseline models and original StarGAN is conducted from original paper.

DIAT uses an adversarial loss to learn the mapping from $x \in X$ to $y \in Y$, where x and y are face images in two different domains X and Y, respectively. This method has a regularization term on the mapping as $\|x - F(G(x))\|_1$ to preserve identity features of the source image, where F is a feature extractor pretrained on a face recognition task.

CycleGAN also uses an adversarial loss to learn to mapping between two different domains X and Y. This method regularizes the mapping via cycle consistency losses,
$\|x - G_{YX}(G_{XY}(x)\|_1$ and $\|y - G_{XY}(G_{YX}(y)\|_1$.
This method requires two generators and discriminators for each pair of two different domains.

IcGAN combines an encoder with a cGAN model. cGAN learns the mapping $G : \{z, c\} \rightarrow x$ that generates an image x conditioned on both the latent vector z and the conditional vector c. In addition, IcGAN introduces an encoder to learn the inverse mapping of cGAN, $E_z : x \rightarrow z$ and $E_c : x \rightarrow c$. This allows IcGAN to synthesis images by only changing the conditional vector and preserving the latent vector.

5.2 Datasets

CelebA. The CelebFaces Attributes (CelebA) dataset contains 202,599 face images of celebrities, each annotated with 40 binary attributes. We crop the initial (178, 218) size images to (178, 178) then resize them as (128, 128). We randomly select 2,000 images as test set and use all remaining images for training data. We construct seven domains using the following attributes: hair color (black, blond, brown), gender (male / female), and age (young / old).

RaFD. The Radboud Faces Database. This is the original dataset that paper used. However, our request to the RaFD website had been denied. Hence, we prepare another dataset that also contains large number of facial expression images.

AffectNet. is a large facial expression dataset with around 0.4 million images manually labeled for the presence of eight (neutral, happy, angry, sad, fear, surprise, disgust, contempt) facial expressions along with the intensity of valence and arousal. We resize and crop images to (256, 256).

5.3 Training

As we mentioned above, due to implementation time of baseline models and huge training time we only trained our replicated StarGAN model. Since we've trained with same parameter of original paper, comparing with the result of original paper and our model is reasonable enough. In AffectNet dataset case, we adopt same parameter of RaFD training of original paper. Each training takes about one day with KCloud VPN GPU server.

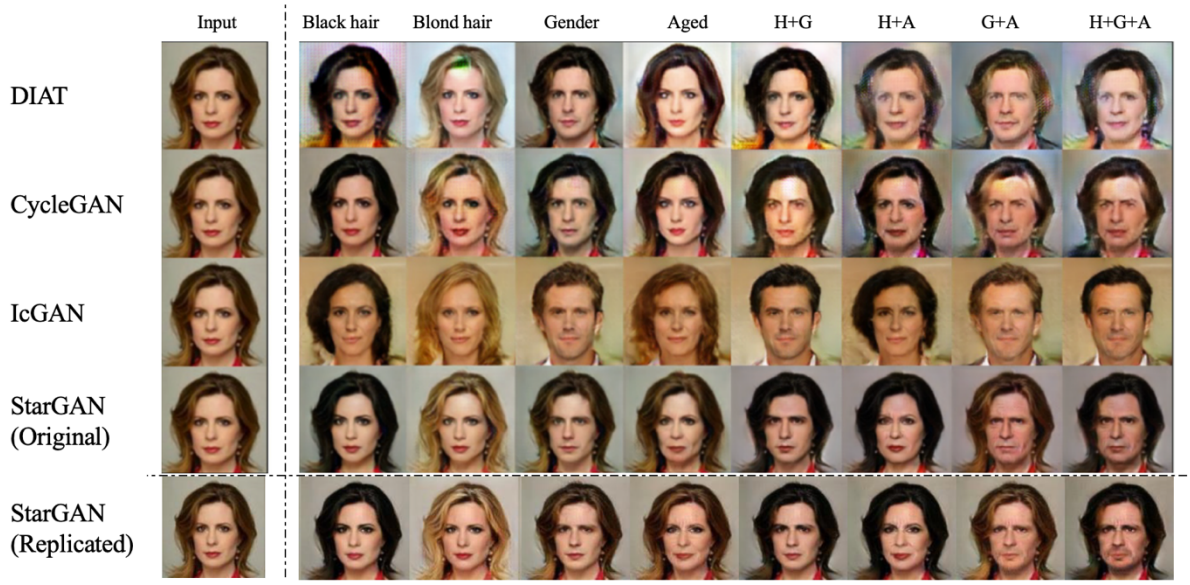

Figure 4: Facial attribute transfer results on the CelebA dataset. The first column shows the input images, next four columns show the single attribute transfer results, and rightmost columns show the multi-attribute transfer results. H: Hair color, G: Gender, A: Aged.

5.4 Experimental Results on CelebA

We first compare our proposed method to the baseline models on a single and multi-attribute transfer tasks. Since the implementation of the StarGAN in public github repository doesn't have multi-attribute transfer tasks, we implemented additional function that can support them.

As we mentioned above, all of result images are from original paper except our model.

Qualitative evaluation. Fig. 4 shows the facial attribute transfer results on CelebA. We observed that our model provides a higher visual quality of translation results on test data compared to the cross-domain models and similar quality compared to the original StarGAN model.

Quantitative evaluation protocol. For quantitative evaluations, the original paper used Amazon Mechanical Turk. Instead, for simplicity we performed two user studies in a survey format using google form to assess single and multiple attribute transfer tasks. Given an input image, each person were instructed to choose the best generated image based on perceptual realism, quality of transfer in attribute, and preservation of a figure's original identity.

First survey has a single attribute transfer in either hair color (black, brown, blond), gender, or age. In another study, the generated images involve a combination of attribute transfers. Each person asked 10 questions and total 98 KAIST students were participated.

Quantitative results. Table1 and 2 show the results of our google form survey on single and multi-attribute transfer tasks, respectively. Our replicated StarGAN or original StarGAN obtained the majority of votes for best transferring attributes in all cases. Even though the difference of DIAT and StarGAN (original) in 'Gender' case is 26.5% and 35.7% in Table 1, the difference of DIAT and StarGAN (original) in 'G+A' case in Table 2 becomes significant (8.2% and 46.9%). This result clearly showing the advantages of StarGAN in more complicated, multi-attribute transfer tasks. This is because unlike the other methods, StarGAN can handle image translation involving multiple attribute changes by randomly generating a target domain label in the training phase. Moreover, this result is from only one training of StarGAN and 20~ training of other models(only support two domains), we checked that StarGAN is more efficient than any other baseline models enough.

Method	Hair Color	Gender	Aged
DIAT	5.1%	26.5%	0%

CycleGAN	1%	4.1%	2%
IcGAN	4.1%	29.6%	5.1%
StarGAN (original)	27.6%	**35.7%**	**50%**
StarGAN (our)	**62.2%**	4.1%	42.9%

Table 1. Google form survey evaluation for ranking different models on a single attribute transfer task. Each column sums to 100%.

Method	H+G	H+A	G+A	H+G+A
DIAT	4.1%	1%	8.2%	3.1%
CycleGAN	2%	12.2%	3.1%	2%
IcGAN	23.5%	1%	29.6%	36.7%
StarGAN (original)	29.6%	38.8%	**46.9 %**	**44.9%**
StarGAN (our)	**40.8 %**	**46.9 %**	12.2%	13.3%

Table 2. Google form survey evaluation for ranking different models on a multi-attribute transfer task. H: Hair color; G: Gender; A: Aged.

5.5 Experimental Results on AffectNet

We next train out model on the AffectNet dataset to learn the task of synthesizing facial expressions. Since there are no any results about facial expression synthesis on the AffectNet of baseline models(due to lack of time to implement baseline model and training AffectNet dataset for all domain-domain pairs), we decide to only look at the transformed images' quality.

Qualitative evaluation.

As seen Fig. 5, StarGAN clearly generates the natural-looking facial expressions. However, our results were not as natural as the results of RaFD in original paper, which is presumed to be due to differences in datasets. Since all of the RaFD images are face-centered, looking straight ahead and same background color, it'll be easy to catch and extract features of each facial expression domain. However, most of AffectNet images are not face-centered and each person looking random direction. Moreover, their background color, chromaticity and brightness are very diverse. Therefore, it would have been difficult to extract common features in convolutional layer because the direction of the face was different. In addition, because each photo has a different color, the transformation result did not follow the overall color, and only the converted part showed unnaturally different colors. For this reason, a low-quality conversion seems to have appeared against using RaFD dataset.

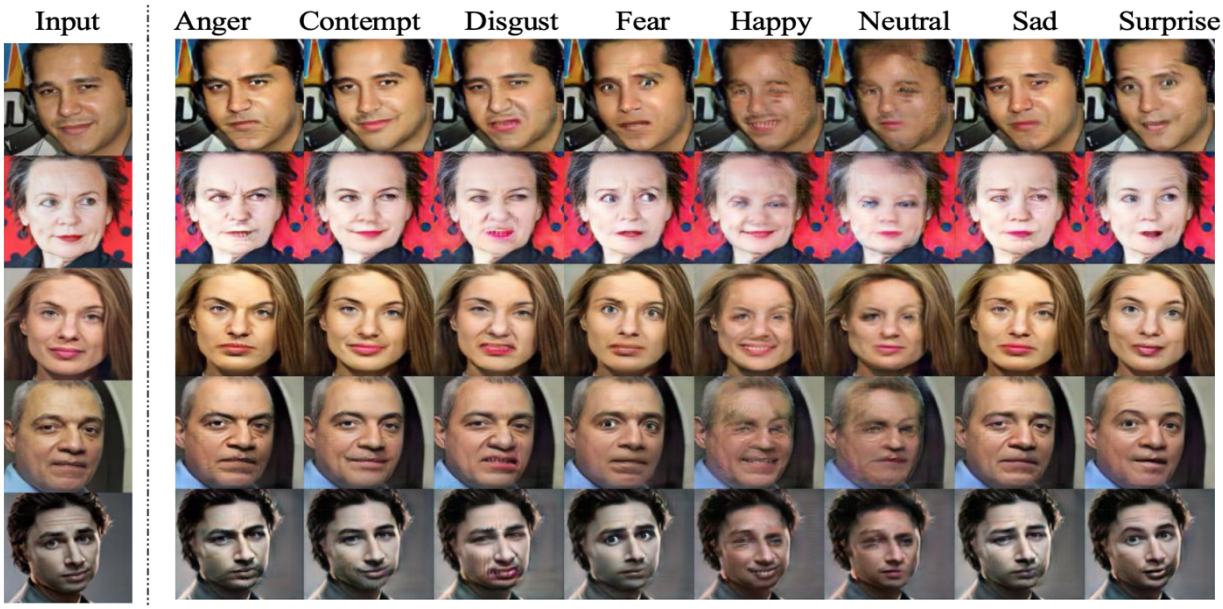

Fig 5. Facial expression synthesis results on the AffectNet dataset of our model.

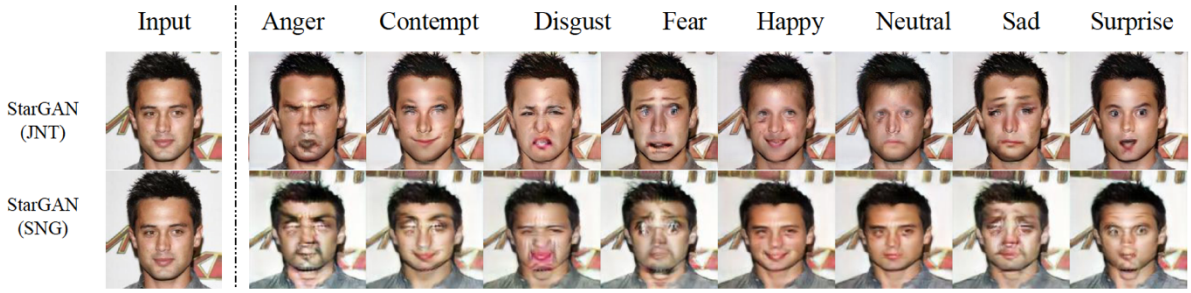

Fig 6. Facial expression synthesis results of StarGAN-SNG and StarGAN-JNT on CelebA dataset.

Fig 7. Learned role of the mask vector. All images are generated by StarGAN-JNT. The first row shows the result of applying the wrong mask vector, and the last row show the result of applying the proper mask vector.

5.6 Experimental Results on CelebA + AffectNet

Finally, we empirically demonstrate that our model can learn not only from multiple domains within a single dataset, but also from multiple datasets. We train our model jointly on the CelebA and AffectNet datasets using the mask vector. To distinguish between the model trained only on AffectNet and the model trained on both CelebA and AffectNet, we denote the former as StarGAN-SNG (single) and the latter as StarGAN-JNT (joint).

Effects of joint training. Fig. 6 shows qualitative comparisons between StarGAN-SNG and StarGAN-JNT, where the task is to synthesize facial expressions of images in CelebA. As the similar result of original paper, StarGAN-JNT exhibits emotional expressions with high visual quality than StarGAN-SNG. Both results contain some crushed image due to characteristics of AffectNet dataset mentioned above, but the amount of crush was greater at the StarGAN-SNG. The high visual quality of StarGAN-JNT is due to the fact that it learns to translate CelebA images during training but not StarGAN-SNG. In other words, StarGAN-JNT can leverage both datasets to improve shared low-level tasks such facial keypoint detection and segmentation. By utilizing both CelebA and AffectNet, StarGAN-JNT can improve these low-level tasks, which is beneficial to learning facial expression synthesis.

Learned role of mask vector. In this experiment, we gave a one-hot vector c by setting the dimension of a particular facial expression (available from the AffectNet dataset) to one. In this case, since the label associated with the second dataset is explicitly given, the proper mask vector would be [0,1]. Fig. 7 shows the case where this proper mask vector was given and the opposite case where a wrong mask vector of [1,0] was given. When the wrong mask vector was used, StarGAN-JNT fails to synthesize facial expressions, and it manipulates the age of the input image. This is because the model ignores the facial expression label as *unknown* and treats the facial attribute label as *valid* by the mask vector. From this behavior, we can confirm that StarGAN properly learned the intended role of a mask vector in image-to-image translations when involving all the labels from multiple datasets together.

6 Model Improvement Suggestions

6.1 Problems

Even though we derive acceptable results in vanilla StarGAN, there are several parts that can be improved. From generated images by StarGAN, we've found out that our model is struggling for some specific class generation. (Fig 8.) In specific, generated images for facial expression domain was unrealistic relatively than hair, age, gender domain. We could conclude that current model has low performance at recognizing geometric or structural patterns in face.

Fig 8. Result of image translations with same source, different target domain.

Most of GAN based models containing StarGAN use convolutional layers for image generation. Because convolution operator has local receptive field, convolution layer processes the information in local neighborhood. Long-range dependency in sample image will be managed only after several convolution layers. Therefore, using only convolution layers in GAN is inefficient for modeling long-rang dependency, which leads to lack of recognition of structural patterns. Simple solution for this problem is increasing kernel size of convolution layer. If size of kernel increases, representational capacity of network will be increase, and network can represent image more generally. But it can't take advantage of structural, computational efficiency coming from local convolution.

6.2 Our solution: Self-Attention Mechanism

To resolve this problem, we suggest to use attention mechanism in StarGAN. Attention mechanism will fully utilize the long-range dependency in internal representation of image. Specifically, self-attention mechanism which translates convolution feature maps to attention feature maps is represented below. (Fig 9.) At the beginning, Input x is fed into 3 different matrix function f, g, h which their wights pass through 1*1 convolution layer. Each of them has their own weights. To get attention map, matrix multiply result of f(x) transpose with result of g(x), then apply the softmax operation to result of multiplication.

$$\beta_{j,i} = \frac{\exp(s_{ij})}{\sum_{i=1}^{N} \exp\,(s_{ij})},$$

$$where\ \boldsymbol{s_{ij}} = \boldsymbol{f(x_i)^T g(x_j)}$$

Beta of above equation represents amount of relation between x_i and x_j. In other words, how much x_i is concerned in processing of x_j. After processing attention map, matrix multiply attention map with result of h(x), then fed it into v(x) which also passes its weight into 1*1 convolution layer to

get attention feature map o. Each element in output feature map is defined below.

$$o = v\left(\sum_{i=1}^{N} \beta_{j,i}\, h(x_i)\right), h(x_i) = W_h x_i, v(x_i) = W_v x_i$$

The final output is defined below. In final output, we add result of attention layer multiplied with learnable scale parameter with input feature maps. Introducing scale parameter γ, we can imply model to consider local neighborhood first, then gradually focus on non-local neighborhood information.

$$y_i = \gamma o_i + x_i$$

Finally, loss function of attention applied generator and discriminator is defined below. But we should combine them with original loss functions of StarGAN generator and discriminator properly to utilize the attention mechanism to StarGAN.

$$L_D = -E_{x,y\sim P_{data}}[\min(0, -1 + D(x,y))] - E_{z\sim P_z, y\sim P_{data}}[\min(0, -1 - D(G(z),y))]$$

$$L_G = -E_{z\sim P_z, y\sim P_{data}}[D(G(z),y)]$$

This unified model architecture of StarGAN allows us to train multiple datasets simultaneously with each different domain even within only one network. Even more than that, we additionally modified the adversarial loss function to the objective function of WGAN (Wasserstein GAN) along with the gradient penalty. We refer to our model as Vanilla GAN including all jobs until this point. Then, our second objective was to develop this Vanilla StarGAN model with the application of self-attention mechanism. This is our own original work, and it aims to increase the performance of StarGAN at recognizing geometric or structural patterns in face. We proposed the self-attention mechanism as a solution to this.

We also validated our theoretical approach with experiments. We used CelebA and AffectNet instead of RaFD. Due to excessively long implementation time of baseline models on the given GPU, we only trained our replicated StarGAN model as mentioned before. In the CelebA dataset, it was obvious that our StarGAN model outperforms the baseline model. On the other hand, we observed a low-quality conversion issue when training the AffectNet dataset, and we gave a logical reason for this phenomenon. We also proceeded with joint training with CelebA and AffectNet and further proposed the StarGAN-JNT.

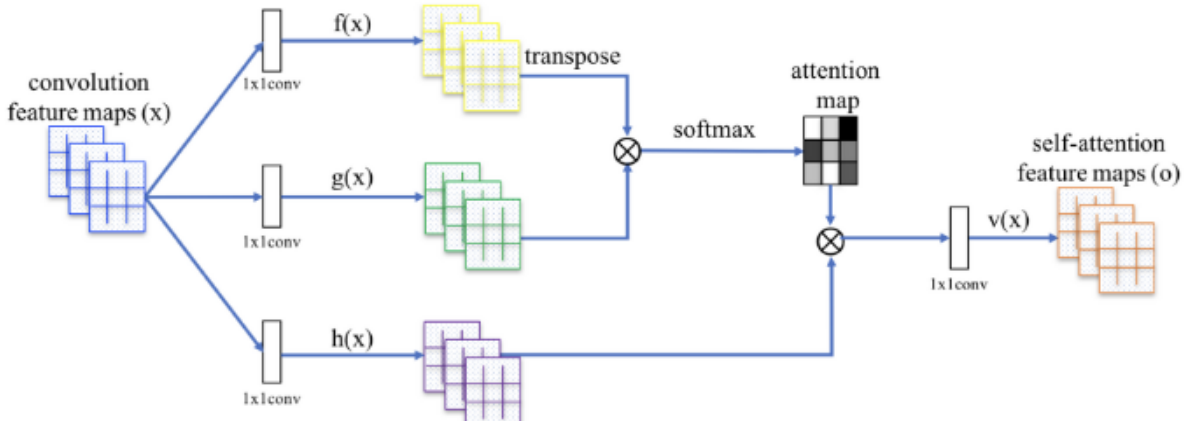

Fig 9. Overall structure of self-attention mechanism.

7 Discussion

In this paper, we proposed two specific research questions. The first objective was to solve the limited scalability and robustness issues of previous I2I models when handling the multi-domain. For this, we proposed the StarGAN model.

8 Conclusion

In this paper, we proposed the single-network multi-domain I2I translation model StarGAN and a possible improvement on this Vanilla StarGAN. The generated images from the StarGAN model

present higher quality compared to previous models. It's because the StarGAN possesses a high generalization capability with the multi-task learning setting. Furthermore, the mask vector of StarGAN enables it to easily manage multiple datasets with numerous domain labels sets. Also, we observed that this Vanilla StarGAN shows a poor performance when there exist geometric or structural patterns in faces of the dataset. We also gave a possible solution to this problem by applying the concept of self-attention mechanism. In the future investigation, we want to further study how to maintain the balance between the discriminator and generator when training a model in the starGAN, which is one of the issues that most I2I translation models need to deal with.

9 References

Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo. CVPR (2018, Oral). StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation https://arxiv.org/abs/1711.09020

Han Zhang Ian Goodfellow Dimitris Metaxa Augustus Odena (2018) Self-Attention Generative Adversarial Networks https://arxiv.org/abs/1805.08318

Ali Mollahosseini, Behzad Hasani, and Mohammad H. Mahoor, "AffectNet: A New Database for Facial Expression, Valence, and Arousal Computation in the Wild", IEEE Transactions on Affective Computing, 2017

M. Li, W. Zuo, and D. Zhang. Deep identity-aware transfer of facial attributes. arXiv preprint arXiv:1610.05586, 2016. 2, 5, 8

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In Proceedings of the IEEE Interna- tional Conference on Computer Vision (ICCV), 2015. 2, 4, 6

A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. arXiv preprint arXiv:1610.09585, 2016. 3, 5

G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. A´lvarez.Invertibleconditionalgansforimageediting. arXiv preprint arXiv:1611.06355, 2016. 5, 8

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image- to-image translation using cycle-consistent adversarial net- works. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017. 1, 2, 3, 4, 5, 8