Korea university Biomedical Engineering

SNUH RADIOLOGY

CCIDS 의료영상데이터사이언스센터 Center for Clinical Imaging Data Science

2020. 03. 15

StarGANv2: Diverse Image Synthesis for Multiple Domains

Sang Wook Kim
Korea university
Department of Biomedical Engineering

Yunjey Choi, et al., “StarGANv2: Diverse Image Synthesis for Multiple Domains”, CVPR, 2020

contents

Introduction

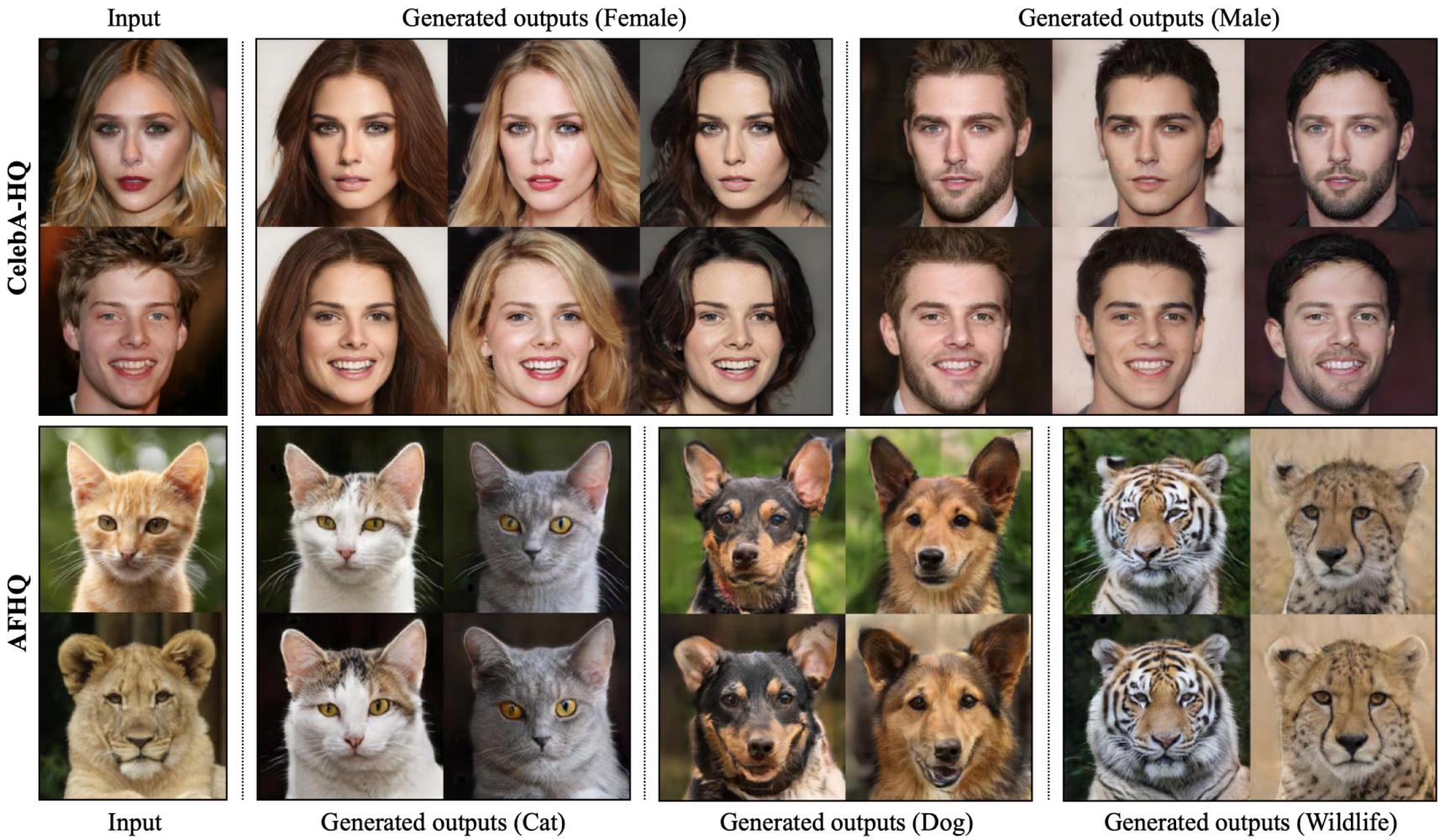


Figure 1. Diverse image synthesis results on the CelebA-HQ dataset and the newly collected animal faces (AFHQ) dataset. The first column shows input images while the remaining columns are images synthesized by StarGAN v2.

Introduction

1. Img2Img translation aims to learn a mapping btw/ diff. visual domains

1. Ideal img2img
    1. SHOULD be able to **synthesize images considering div. images in each domain**
    2. HOWEVER, the number of style is too big to train such models

2. StarGAN learns mapping btw/ all available domains

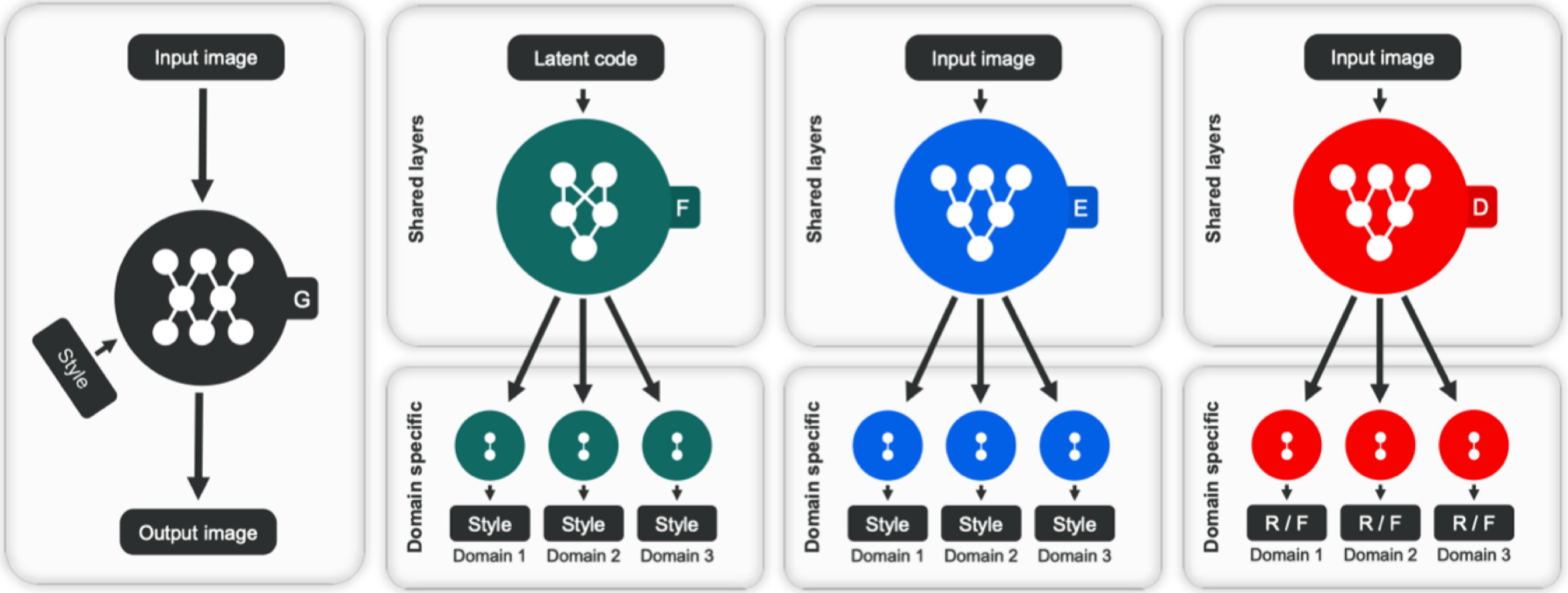
1. It avoids one-to-one mapping btw/ domains by training
2. HOWEVER, it still learns a deterministic mapping / each domain
    ==> Limitation , Each domain is indicated by a pre-determined label
3. SO, it cannot avoid limited variability in generated images

3. StarGAN v2

1. Generate diverse images across multiple domains
    1. REPLACE domain label w/ newly proposed domain-specific style code (diverse styles of a specific domain)
2. Newly designed module
    1. mapping network : transform gaussian latent vector into style code
    2. style encoder : extract style code from a given reference img

Architecture

Generator (G)



(a) Generator (b) Mapping network (c) Style encoder (d) Discriminator

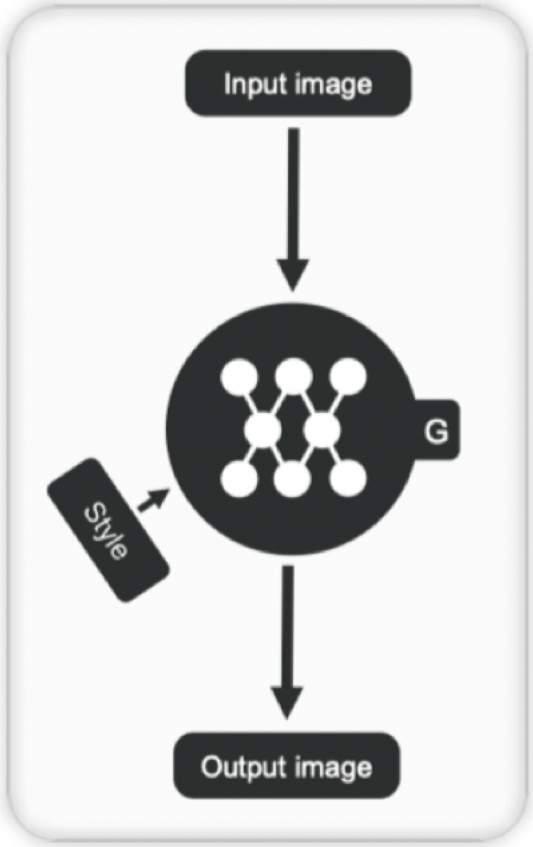
Figure 2. Overview of StarGAN v2, consisting of four modules. **(a)** The generator translates an input image into an output image reflecting the domain-specific style code. **(b)** The mapping network transforms a latent code into style codes for multiple domains, one of which is randomly selected during training. **(c)** The style encoder extracts the style code of an image, allowing the generator to perform reference-guided image synthesis. **(d)** The discriminator distinguishes between real and fake images from multiple domains.

Architecture

Generator (G)

1. input image x 를 input 으로 받고, mapping network 혹은 style encoder 에서 만들어진 style vector 를 또 다른 input 으로 받아 두 개의 input 을 합성하여 새로운 image 를 생성해낸다.

2. 이미 style code 가 특정 Domain 에서 나왔기에 domain 에 대한 정보를 generator 에 같이 input 으로 넣어줄 필요가 없다



(a) Generator

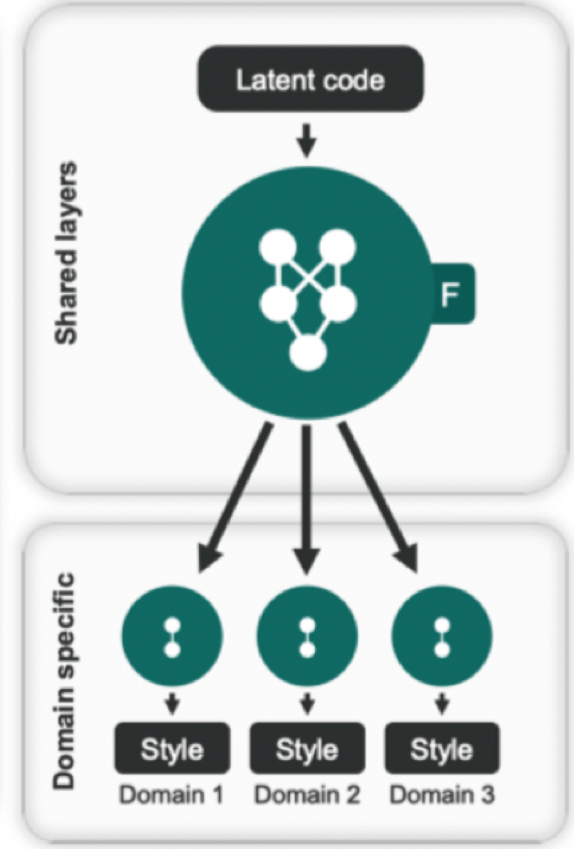
Architecture

Mapping Network (F)

1. Domain y 를 represent 하는 latent code z 를 style code s 로 mapping 해준다

1. Making various style codes by sampling latent vector z (~ Z) and the domain y (~ Y) randomly
2. CAN provide style codes for all available domains

2. Mapping network consists of simple MLP



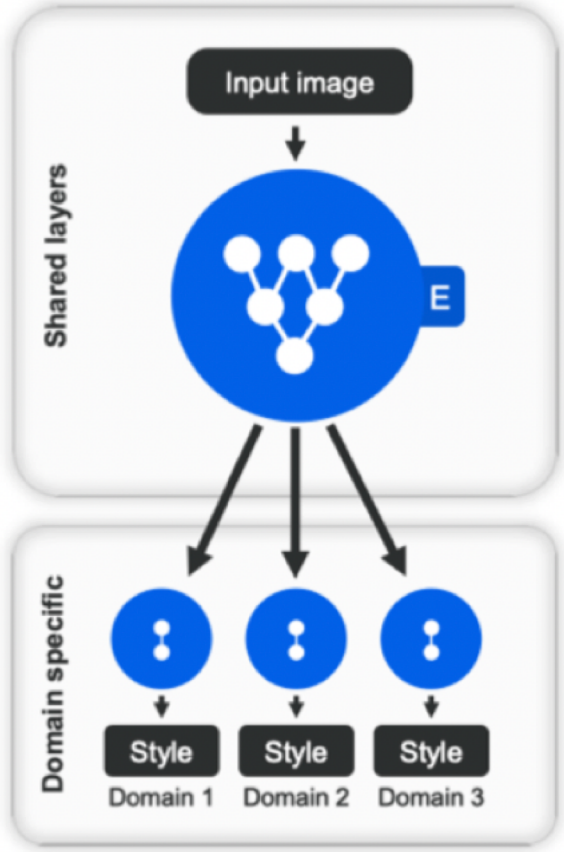
(b) Mapping network

Architecture

Style Encoder (E)

1. CAN produce diverse style codes using different ref. images

2. Allows Generator to synthesize an output image reflecting the style s of a reference image x



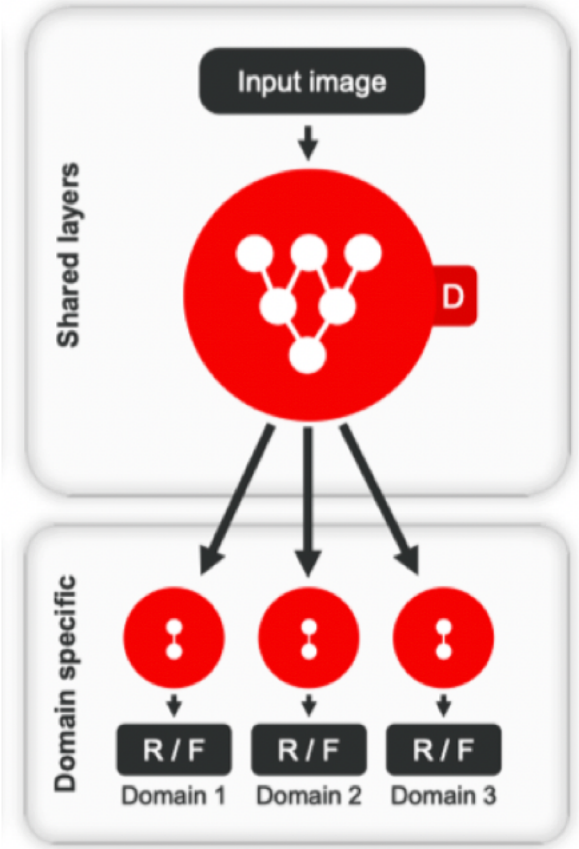
(c) Style encoder

Architecture

Discriminator (D)

1. Multitask discriminator - multiple output branches

1. Each branch learns a binary classification determining whether an image x is a real image (of its domain) or a fake image G(x, s) produced by G



(d) Discriminator

Architecture

Adversarial Loss

$$L_{adv} = E_{x,y}[logD_y(x)] + E_{x,\tilde{y},z}[log(1 - D_{\tilde{y}}(G(x, \tilde{s})))]$$

1. Training , sample latent code z and target domain y randomly

2. Generate a target style code s~ (using mapping network)

1. Mapping Network learns to prov. s~ that is likely in the tgt. domain

3. Generator G takes an image x and s~ as inputs & learns to gen. output image G(x, s~) via adv. loss

1. Generator learns to util. s~, gen image indistinguishable from real images of the domain y

Architecture

Style Reconstruction Loss

$$L_{sty} = E_{x,\tilde{y},z}[|\tilde{s} - E_{\tilde{y}}(G(x,\tilde{s}))|_1]$$

1. Enforcing G to util. style code s~ => Use style recon. loss

2. Employ multiple encoders to learn a mapping (image -> latent code)

1. Train just a single E to encourage diverse outputs for multiple domains.

Architecture

Style Diversification Loss (Regularization)

$$s_{\tilde{i}} = F_{\tilde{y}}(z_i), i \in 1, 2$$

$$L_{ds} = E_{x,\tilde{y},z_1,z_2}[|G(x, \tilde{s_1}) - G(x, \tilde{s_2})|_1]$$

1. Enabling G to produce diverse images => Regularize G w/ diversity sensitive loss

1. Forcing G to explore image space and discover meaningful style features

2. X optimal point => Linearly decay weight of loss to zero

Architecture

Style Diversification Loss (Regularization)

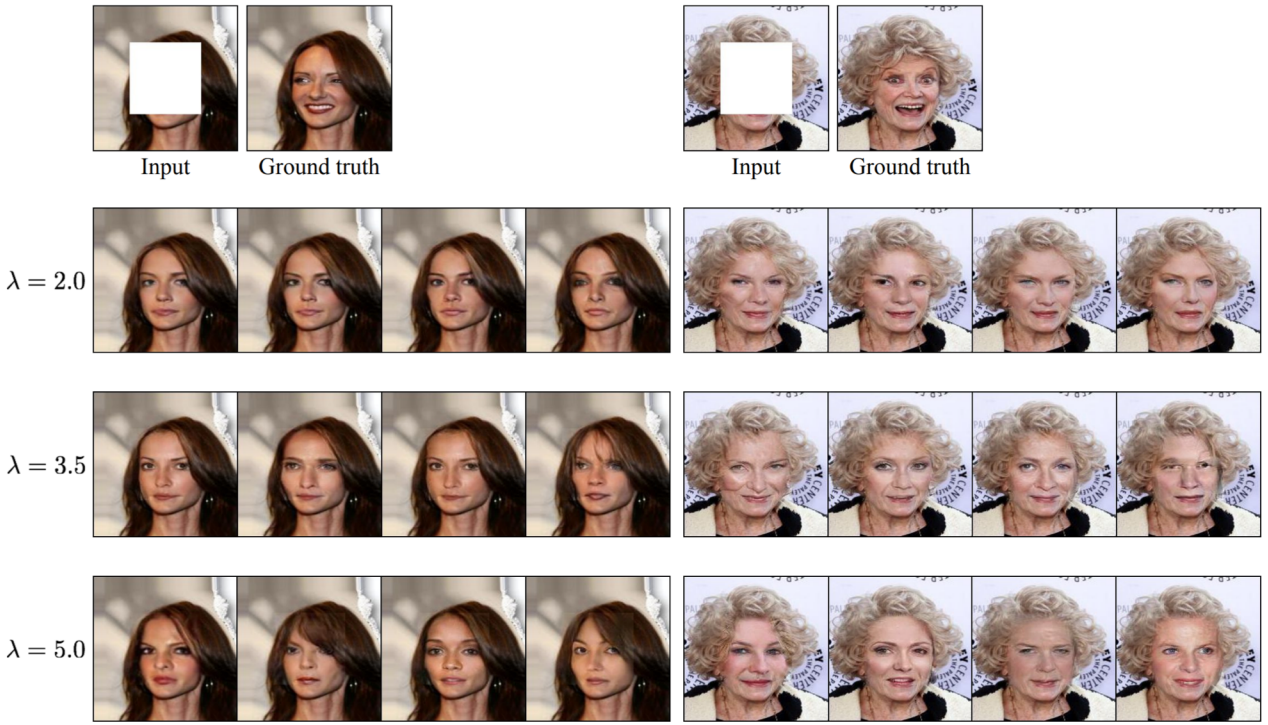


Figure F: Image inpainting results with different λ. We observe more diversity emerges from the genrator outputs as we increase the weights for our regularization.

Dingdong Yang, et al., “DIVERSITY-SENSITIVE CONDITIONAL GENERATIVE ADVERSARIAL NETWORKS”, ICLR, 2019

Architecture

Preserve original source characteristic Loss
(a.k.a cycle consistency loss)

$$L_{cyc} = E_{x,y,\tilde{y},z}[|x - G(G(x, \tilde{s}), \hat{s})|_1]$$

$$\hat{s} = E_y(x)$$

1. To guarantee Gen. images properly preserve domain-invariant characteristics

1. domain 마다 갖고 있는 고유한 특성을 적절히 반영하기 위한 loss

Architecture

Total Objective

$$L_D = -L_{adv}$$

$$L_{F,G,E} = L_{adv} + \lambda_{sty} * L_{sty} - \lambda_{ds} * L_{ds} + \lambda_{cyc} * L_{cyc}$$

Experiments

Baselines

1. MUNIT, DRIT, MSGAN
2. Multi-modal mappings btw/ two domains Train every pair of image domains StarGAN

Datasets

1. CelebA-HQ - Separate into two domains (male / female) Not using facial attr.
2. AFHQ - Three domains (cat, dog, wildlife) Not using facial attr.
3. Resized into 256 x 256

Evaluation Metrics

1. FID : Visual Quality + Diversity of gen. images
2. LPIPS : Learned perceptual image patch similarity
3. Compute two metrics for every pair of image domains => report their avg. values

Experiments

Analysis of individual components - Components added to StarGAN baseline model

1. (B) Replacing AC-GAN discriminator => multi-task discriminator
=> Allows the Gen. to transform the glob. structure of input image
2. (C) R1 regularization, switching depth-wise concatenation into AdaIN
3. (D)
    1. latent code from Gauss. dist. -> X provide meaningful styles and fails to provide as much div. as expected
    2. Finds it diff. to change the overall structure
4. (E) Style mapping structure added -> providing style codes for a specific domain
    1. Have diff. style codes per domain -> injected to the generator
5. (F) Diversity Regularization
    1. Reflect diverse styles of given ref. images wo/ hurting source characteristics

Experiments

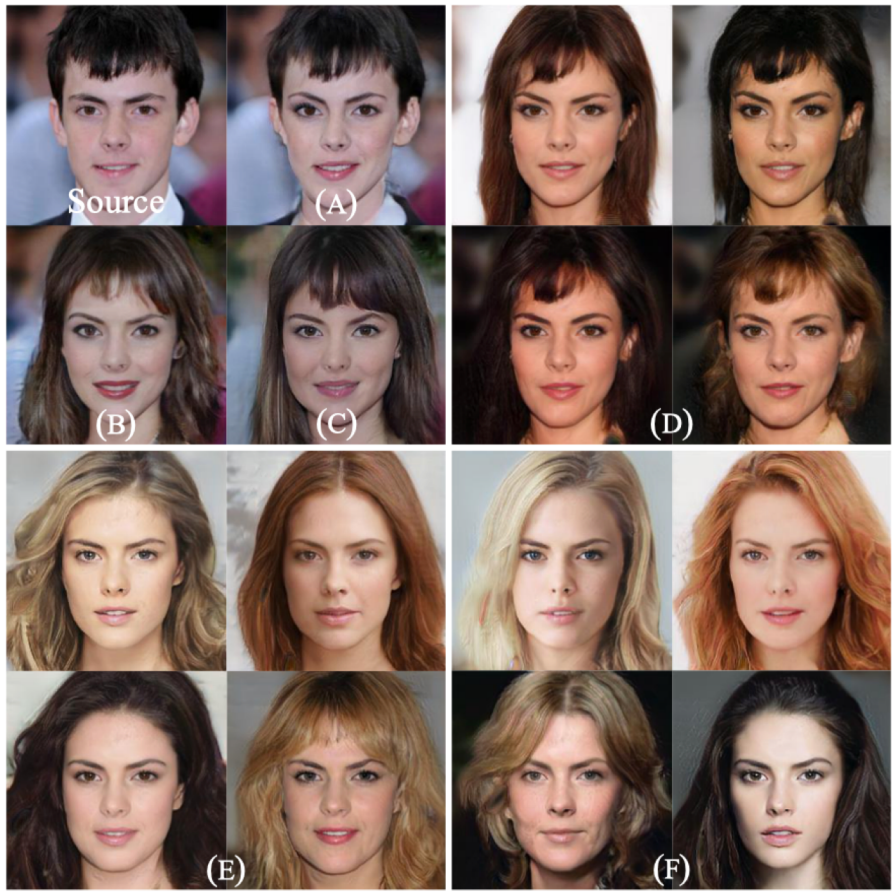


Figure 3. Visual comparison of generated images using each configuration in Table 1. Note that given a source image, the configurations (A) - (C) provide a single output, while (D) - (F) generate multiple output images.

Experiments

Comparison on diverse image synthesis | Latent guided synthesis

1. Comparison of the images generated from latent vector z (~ Gauss. Dist.)
    1. CelebA-HQ : starGANv2 is the only model that can successfully change the entire hair style
    2. AFHQ :
        1. starGAN v1 not good HOWEVER, starGAN v2 still good
2. 2x performance improvements compared to previous leading method in FID
    1. LPIPS is also the highest - produces the most diverse results given a single input image

Experiments

Comparison on diverse image synthesis | Latent guided synthesis

(a) Latent-guided synthesis on CelebA-HQ (b) Latent-guided synthesis on AFHQ

Figure 5. Qualitative comparison of latent-guided image synthesis results on the CelebA-HQ and AFHQ datasets. Each method translates the source images (left-most column) to target domains using randomly sampled latent codes. **(a)** The top three rows correspond to the results of converting male to female and vice versa in the bottom three rows. **(b)** Every two rows from the top show the synthesized images in the following order: cat-to-dog, dog-to-wildlife, and wildlife-to-cat.

Method	CelebA-HQ		AFHQ	
	FID	LPIPS	FID	LPIPS
MUNIT [16]	31.4	0.363	41.5	0.511
DRIT [27]	52.1	0.178	95.6	0.326
MSGAN [33]	33.1	0.389	61.4	0.517
StarGAN v2	**18.0**	**0.428**	**24.4**	**0.524**
Real images	15.1	-	13.1	-

Table 2. Quantitative comparison on latent-guided synthesis. The FIDs of real images are computed between the training and test sets. Note that they may not be optimal values since the number of test images is insufficient, but we report them for reference.

Experiments

Comparison on diverse image synthesis | Reference guided synthesis

Obtaining style code from a reference image => Sample test images from a target domain and feed them to the encoder network

1. CelebA-HQ

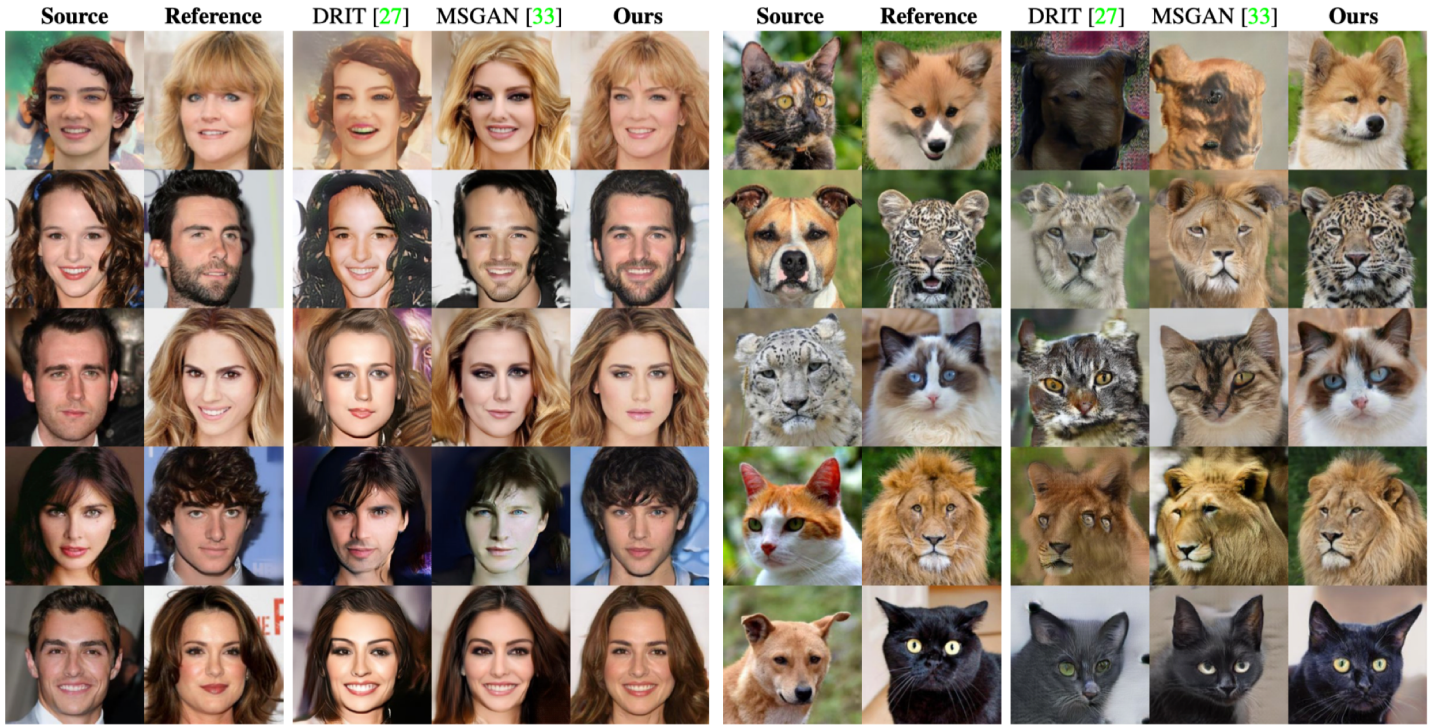
1. **Do resonably well**

2. AFHQ

1. **baseline models suffer from domain shift - only match the domain**

Experiments

Comparison on diverse image synthesis | Latent guided synthesis



(a) Reference-guided synthesis on CelebA-HQ (b) Reference-guided synthesis on AFHQ

Figure 6. Qualitative comparison of reference-guided image synthesis results on the CelebA-HQ and AFHQ datasets. Each method translates the source images into target domains, reflecting the styles of the reference images.

Method	CelebA-HQ		AFHQ	
	FID	LPIPS	FID	LPIPS
MUNIT [16]	107.1	0.176	223.9	0.199
DRIT [27]	53.3	0.311	114.8	0.156
MSGAN [33]	39.6	0.312	69.8	0.375
StarGAN v2	**20.2**	**0.397**	**19.7**	**0.503**
Real images	15.1	-	13.1	-

Table 3. Quantitative comparison on reference-guided synthesis. We sample ten reference images to synthesize diverse images.

Experiments

Human evaluation

1. Amazon Mechanical Turk (AMT) - To compare the user preferences of our method
    1. 4 images were randomly given from each methods

2. 100 questions / Answered by 10 workers

    1. Unworthy workers are excluded.
    2. Only 76 workers left.

Method	CelebA-HQ		AFHQ	
	Quality	Style	Quality	Style
MUNIT [16]	6.2	7.4	1.6	0.2
DRIT [27]	11.4	7.6	4.1	2.8
MSGAN [33]	13.5	10.1	6.2	4.9
StarGAN v2	**68.9**	**74.8**	**88.1**	**92.1**

Table 4. Votes from AMT workers for the most preferred method regarding visual quality and style reflection (%). StarGAN v2 outperforms the baselines with remarkable margins in all aspects.

Conclusion / Discussion

1. Following the insight of StyleGAN, style space is produced by non-linear transformation from a Gauss. dist.
    1. Provides more flexibility to starGANv2

2. Style codes are separately generated per each domain by multi-branch encoder

    1. By using the style code from specific domain, generator can focus only on the style codes from specific domain

3. StarGANv2 benefit from fully exploiting training data from multiple domains

    1. Shared part of each module should learn domain-invariant features
    2. Encouraging better generalization to unseen samples

Reference

1. http://kozistr.tech/deep-learning/2020/02/10/StarGANv2.html