GROUP4
| Kaderka Christine (2021318796)
| Ko Kyounghyeon (2017313272)

StarGAN v2

Diverse Image Synthesis for Multiple Domains

Paper Review & Reproduction

Outline

I. Introduction

- Problem
- Motivation

II. Methods

- Framework
- Experiments

III. Evaluation

- Dataset
- Metrics
- Result

IV. Code Demo

V. Challenges

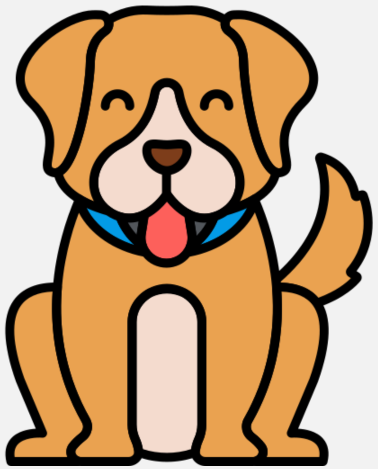
VI. Conclusion

- Summary
- Limitations

I. INTRODUCTION

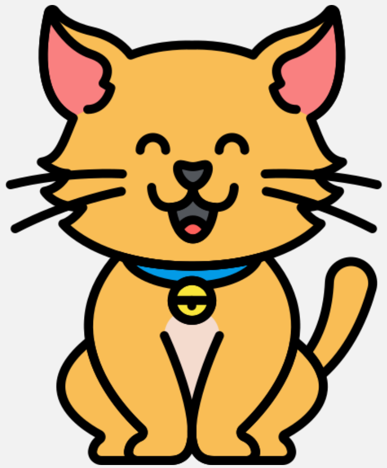
Image-to-Image Translation

Introduction



Source domain : *Dog*
Style : *ears down, smile face*

GAN MODEL



Target domain : *Cat*
Style : *ears up, smile face*

Image-to-Image Translation

Good Image-to-Image Translation?

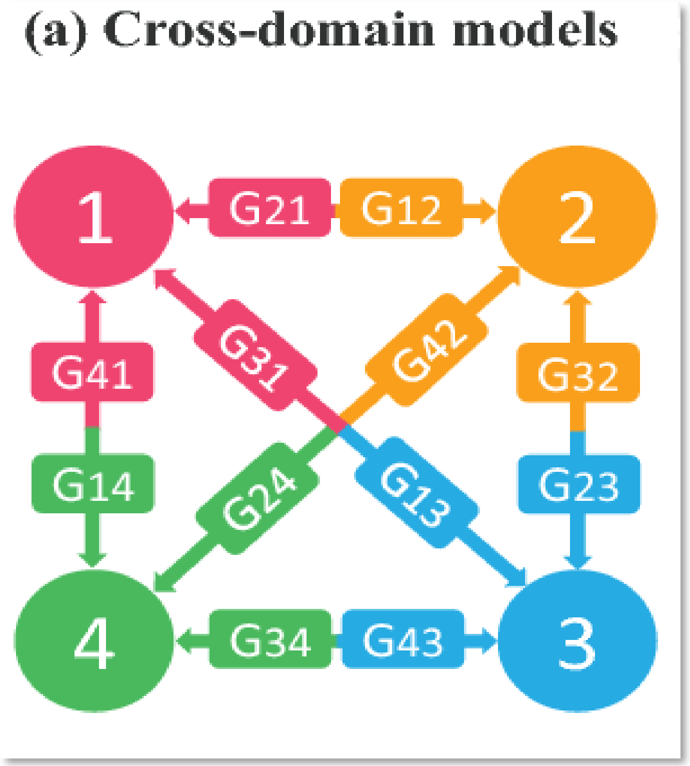
① **Diversity** of generated images

→ Diverse images across an increasing number of domains

② **Scalability** over multiple domains

→ Generating difference images per each domain

But.. previous existing models



- Diversity of generated images

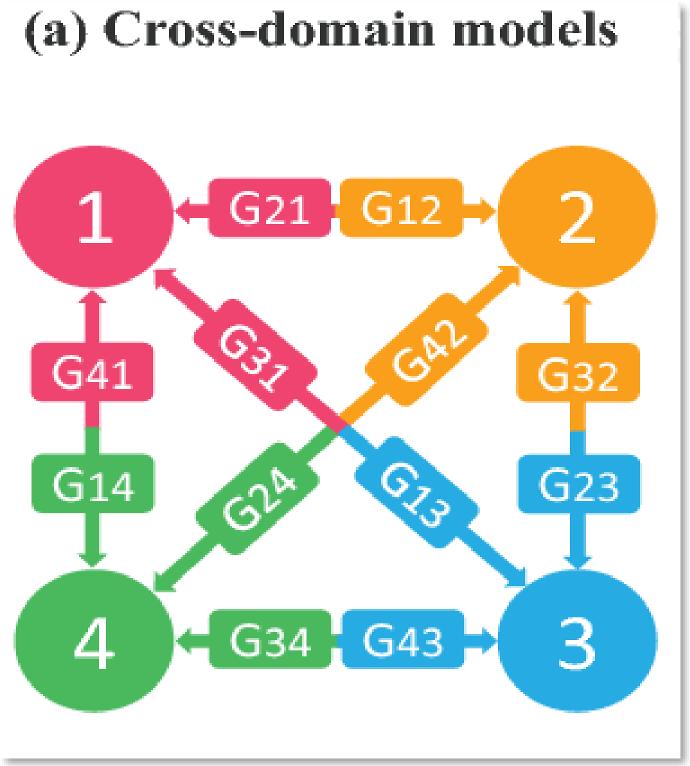
Good diversity

Mapping only two domains

→ Not scalable to increasing number of domains

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

But.. previous existing models



- Scalability over multiple domain

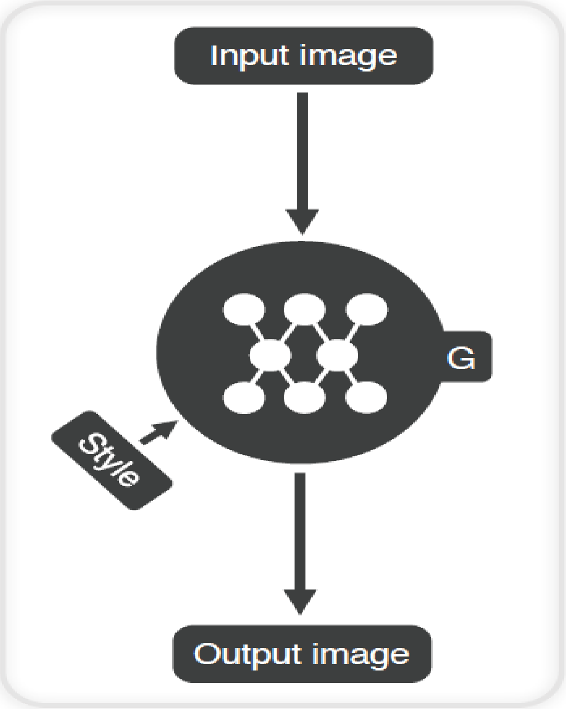
+ Mappings between all available domains

− Deterministic mapping per each domain

→ Generates the same image domain

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

StarGAN v2 (2020)



(a) Generator

Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

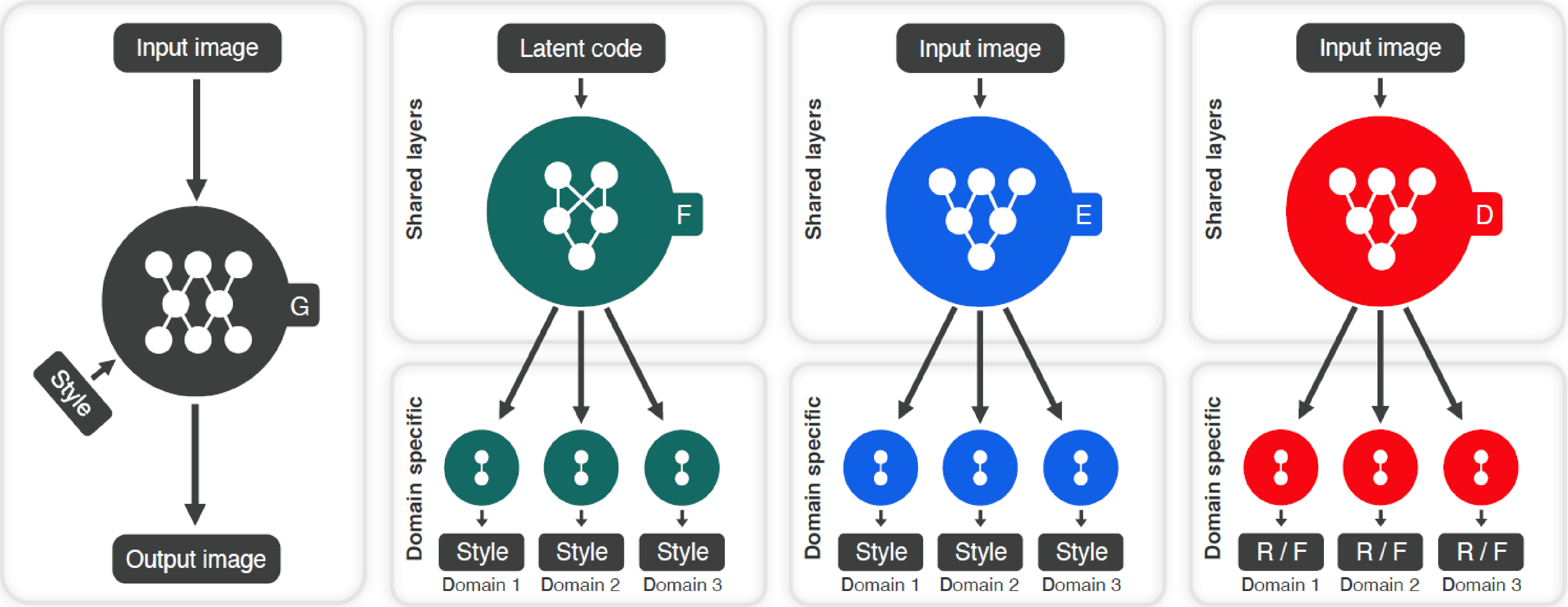
- A breakthrough in the field of CV
- First model that combines two properties.

StarGAN v2 : Motivation

- Most **recent state-of-the-art** image-to-image translation model
- **Solves the two major challenges** in image-to-image translation
- Produces most **realistic** images
- Works well even with **large domain** differences
- Performs well on **unseen data**

II. METHODS

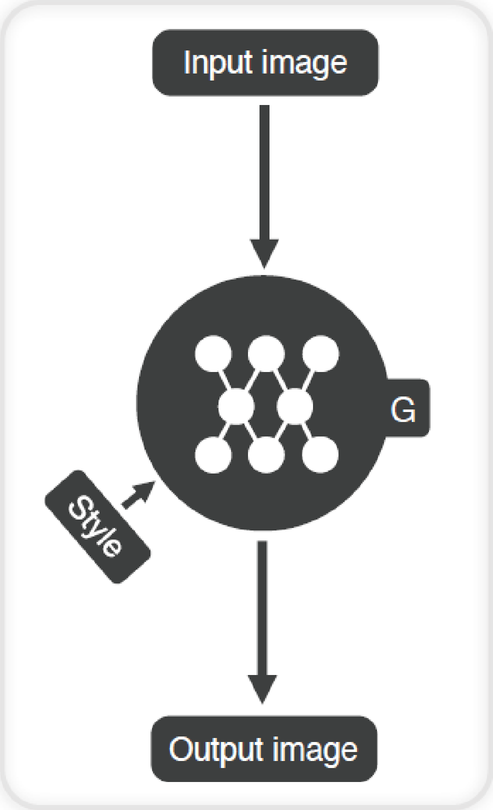
Four Modules



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

Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

Generator (G)



(a) Generator

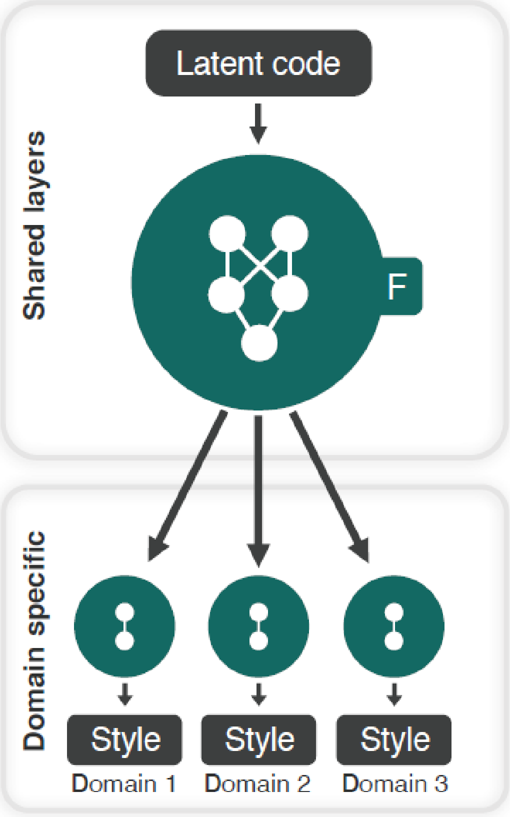
[Concept]

- Translates an input image into an output image
- Style code from F/E is injected

[Technique]

- Adaptive instance normalization (AdaIN)

Mapping Network (F)



(b) Mapping network

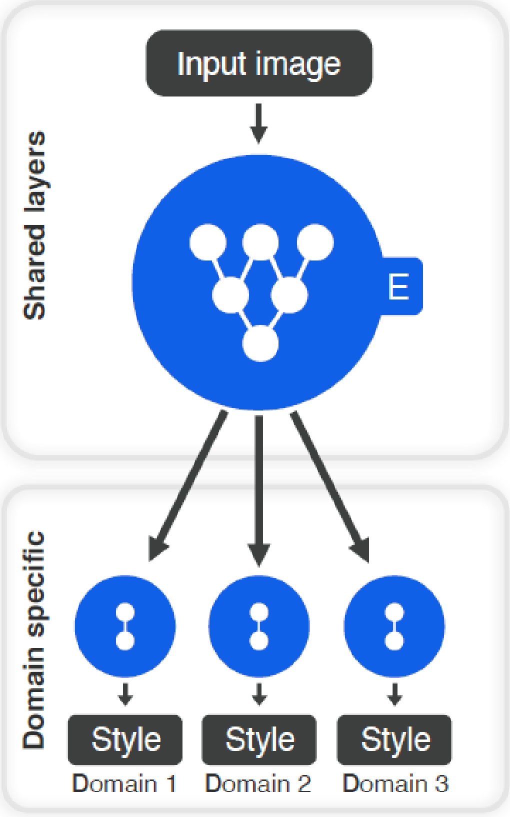
[Concept]

- From latent vector, generates diverse style code
- Multiple output branches

[Technique]

- Linear layer
- Domain-shared layers and -unshared layers

Style Encoder (E)



(c) Style encoder

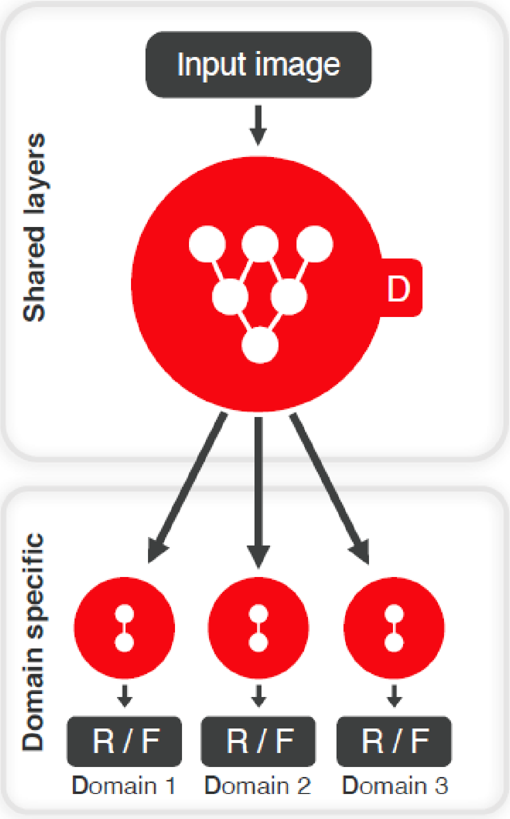
[Concept]

- Given an input image, extracts the style code
- Like F, style code will be injected to the generator

[Technique]

- Domain-shared blocks and -unshared layers

Discriminator (D)



(d) Discriminator

[Concept]

- Multi-task discriminator
- Output branches perform a binary classification on whether an image is fake or real on its domain

[Technique]

- No normalization methods (BN, IN)
- Similar architecture with E

Loss functions

$$\min_{G,F,E} \max_{D} \quad \mathcal{L}_{adv} + \lambda_{sty}\,\mathcal{L}_{sty} - \lambda_{ds}\,\mathcal{L}_{ds} + \lambda_{cyc}\,\mathcal{L}_{cyc}$$

Loss functions

$$\min_{G,F,E}\max_{D}\quad \mathcal{L}_{adv} + \lambda_{sty}\,\mathcal{L}_{sty} - \lambda_{ds}\,\mathcal{L}_{ds} + \lambda_{cyc}\,\mathcal{L}_{cyc}$$

Adversarial loss

$$\mathcal{L}_{adv} = \mathbb{E}_{\mathbf{x},y}\left[\log D_y(\mathbf{x})\right] + \mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}}\left[\log\left(1 - D_{\widetilde{y}}(G(\mathbf{x},\widetilde{\mathbf{s}}))\right)\right]$$

- G → Creates an image which is close to reality
- D → Determines whether image is real or fake

Loss functions

$$\min_{G,F,E} \max_{D} \quad \mathcal{L}_{adv} + \lambda_{sty}\,\mathcal{L}_{sty} - \lambda_{ds}\,\mathcal{L}_{ds} + \lambda_{cyc}\,\mathcal{L}_{cyc}$$

Style reconstruction loss

$$\mathcal{L}_{sty} = \mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}}\left[\| \widetilde{\mathbf{s}} - E_{\widetilde{y}}(G(\mathbf{x}, \widetilde{\mathbf{s}})) \|_1 \right]$$

- E → Outputs multiple style codes
- G → Transforms the image better with the style of reference

Loss functions

$$\min_{G,F,E} \max_{D} \quad \mathcal{L}_{adv} + \lambda_{sty}\,\mathcal{L}_{sty} - \lambda_{ds}\,\underline{\mathcal{L}_{ds}} + \lambda_{cyc}\,\mathcal{L}_{cyc}$$

<u>Diversity sensitive loss</u>

$$\mathcal{L}_{ds} = \mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}_1,\mathbf{z}_2}\left[\|G(\mathbf{x},\widetilde{\mathbf{s}}_1) - G(\mathbf{x},\widetilde{\mathbf{s}}_2)\|_1\right]$$

- F → Outputs multiple target style codes
- G → Is forced to explore the image space more and therefore finds more style information, more diverse images.

Loss functions

$$\min_{G,F,E} \max_{D} \quad \mathcal{L}_{adv} + \lambda_{sty}\,\mathcal{L}_{sty} - \lambda_{ds}\,\mathcal{L}_{ds} + \lambda_{cyc}\,\mathcal{L}_{cyc}$$

Cycle consistency loss

$$\mathcal{L}_{cyc} = \mathbb{E}_{\mathbf{x},y,\widetilde{y},\mathbf{z}}\left[||\mathbf{x} - G(G(\mathbf{x},\widetilde{\mathbf{s}}),\hat{\mathbf{s}})||_1\right]$$

- Preserves the characteristics of the input image

Loss functions

$$\min_{G,F,E} \max_{D} \quad \mathcal{L}_{adv} + \lambda_{sty}\mathcal{L}_{sty} - \lambda_{ds}\mathcal{L}_{ds} + \lambda_{cyc}\mathcal{L}_{cyc}$$

Adversarial loss

$$\mathcal{L}_{adv} = \mathbb{E}_{\mathbf{x},y}\left[\log D_y(\mathbf{x})\right] + \mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}}\left[\log\left(1 - D_{\widetilde{y}}(G(\mathbf{x},\widetilde{\mathbf{s}}))\right)\right]$$

Style reconstruction loss

$$\mathcal{L}_{sty} = \mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}}\left[||\widetilde{\mathbf{s}} - E_{\widetilde{y}}(G(\mathbf{x},\widetilde{\mathbf{s}}))||_1\right]$$

Diversity sensitive loss

$$\mathcal{L}_{ds} = \mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}_1,\mathbf{z}_2}\left[||G(\mathbf{x},\widetilde{\mathbf{s}}_1) - G(\mathbf{x},\widetilde{\mathbf{s}}_2)||_1\right]$$

Cycle consistency loss

$$\mathcal{L}_{cyc} = \mathbb{E}_{\mathbf{x},y,\widetilde{y},\mathbf{z}}\left[||\mathbf{x} - G(G(\mathbf{x},\widetilde{\mathbf{s}}),\hat{\mathbf{s}})||_1\right]$$

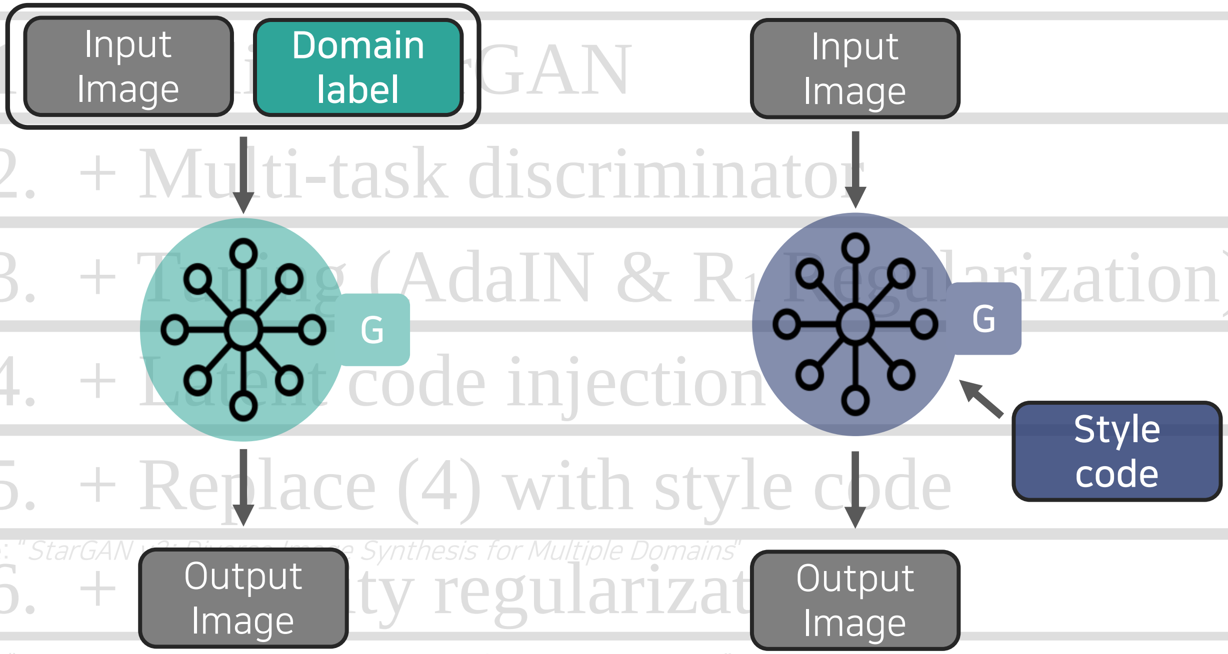
Model building process

Method
1. Baseline StarGAN
2. + Multi-task discriminator
3. + Tuning (AdaIN & R_1 Regularization)
4. + Latent code injection
5. + Replace (4) with style code
6. + Diversity regularization

Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

Model building process

StarGAN (2018)	vs	StarGAN v2 (2020)



Source: "StarGAN v2: Diverse Image Synthesis for Multiple Domains"

Model building process

Method
1. Baseline StarGAN
2. + Multi-task discriminator
3. + Tuning (AdaIN & R_1 Regularization)
4. + Latent code injection
5. + Replace (4) with style code
6. + Diversity regularization

Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

Model building process

Method
1. Baseline StarGAN
2. + Multi-task discriminator
3. + Tuning (AdaIN & R_1 Regularization)
4. + Latent code injection
5. + Replace (4) with style code
6. + Diversity regularization

Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

Model building process

Method
1. Baseline StarGAN
2. + Multi-task discriminator
3. + Tuning (AdaIN & R_1 Regularization)
4. + Latent code injection
5. + Replace (4) with style code
6. + Diversity regularization

Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

Model building process

Method
1. Baseline StarGAN
2. + Multi-task discriminator
3. + Tuning (AdaIN & R_1 Regularization)
4. ~~+ Latent code injection~~
5. + Replace (4) with style code
6. + Diversity regularization

Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

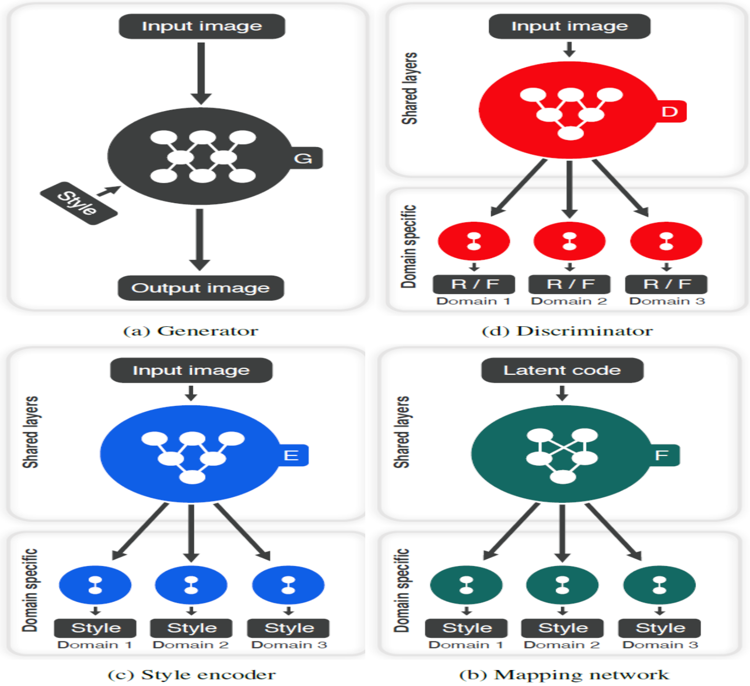
Complete Model

Method
1. Baseline StarGAN
2. + Multi-task discriminator
3. + Tuning (AdaIN & R_1 Regularization)
4. ~~+ Latent code injection~~
5. + Replace (4) with style code
6. + Diversity regularization

Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

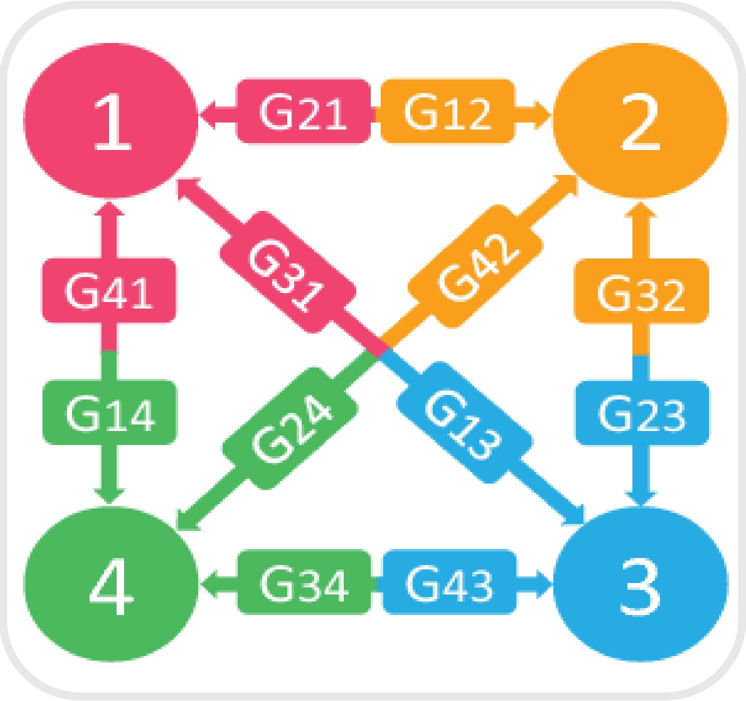
III. EVALUATION

Baseline



Single Generator
StarGAN v2

VS



Multiple generators
MUNIT, DRIT, MSGAN

Evaluation

Two Datasets

CelebA-HQ

Source: "*Large-scale CelebFaces Attributes (CelebA) Dataset*"

AFHQ

Source: "*clovaai/stargan-v2: StarGAN v2 - Official PyTorch Implementation (CVPR 2020)*"

Two Datasets

CelebA-HQ

Source: "*Large-scale CelebFaces Attributes (CelebA) Dataset*"

- HQ dataset with 30K **celeb faces**
- 1,024 × 1,024 resolution
- 40 available domains
- **Female and Male**

Two Datasets

AFHQ

- HQ dataset with 15K **animal faces**
- 512 × 512 resolution
- 3 available domains
- **Dogs, Cats, and Wildlife**

Source: "*clovaai/stargan-v2: StarGAN v2 - Official PyTorch Implementation (CVPR 2020)*"

Two Datasets

CelebA-HQ

AFHQ

- Only domain labels were taken
- Add generalization to model

Source: "*Large-scale CelebFaces Attributes (CelebA) Dataset*"

Source: "*clovaai/stargan-v2: StarGAN v2 - Official PyTorch Implementation (CVPR 2020)*"

Transformation

```
crop = transforms.RandomResizedCrop(
    img_size, scale=[0.8, 1.0], ratio=[0.9, 1.1])
rand_crop = transforms.Lambda(
    lambda x: crop(x) if random.random() < prob else x)

transform = transforms.Compose([
    rand_crop,
    transforms.Resize([img_size, img_size]),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5, 0.5, 0.5],
                         std=[0.5, 0.5, 0.5]),
])
```

Source: "*clovaai/stargan-v2: StarGAN v2 - Official PyTorch Implementation (CVPR 2020)*"

- Resized to **256 × 256**
- Randomly cropped & flipped
- Normalized

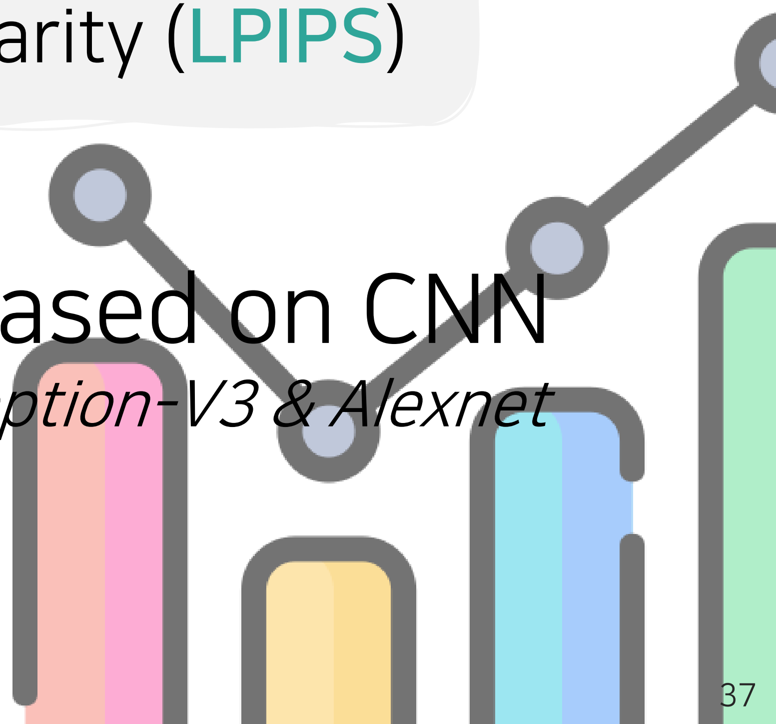
Two Standardized Metrics

#1 Fréchet Inception Distance (FID)

#2 Learned Perceptual Image Patch Similarity (LPIPS)

Use **feature extraction** methods based on CNN

Pretrained Inception-V3 & Alexnet



Evaluation

FID (2017) — LPIPS (2018)

Quality of generated images

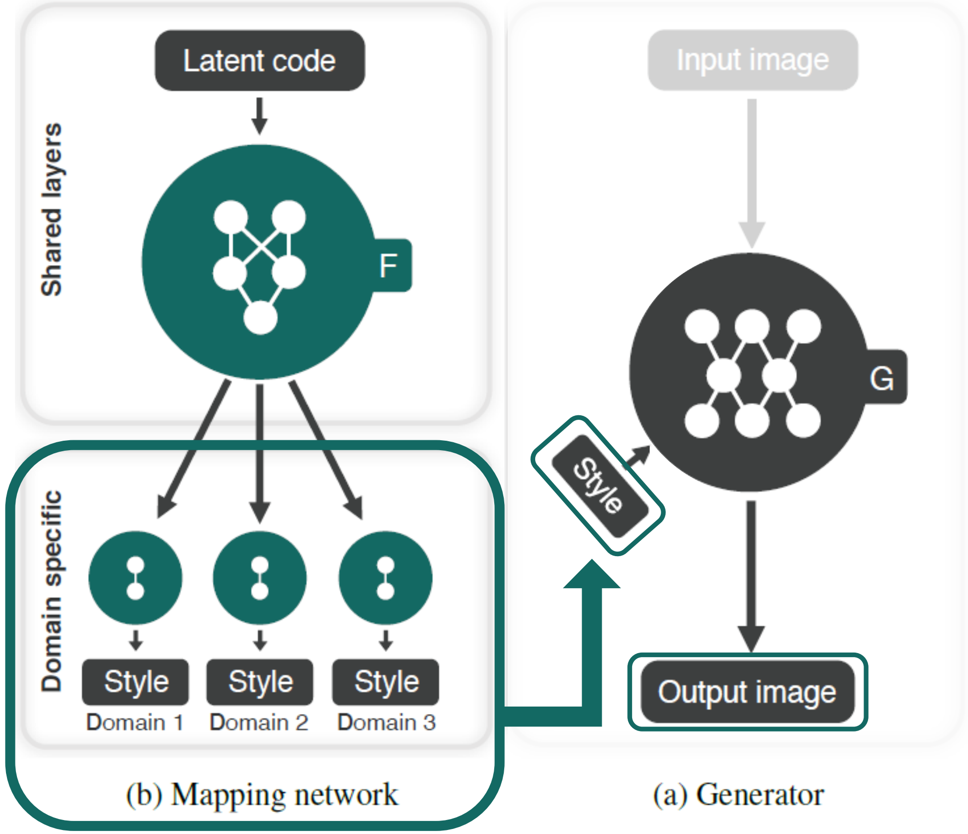
FID (2017)

- Measures the **discrepancy** between two sets of images
- Compares **generated** image with images in **target domain**
- **Lower** is better

LPIPS (2018)

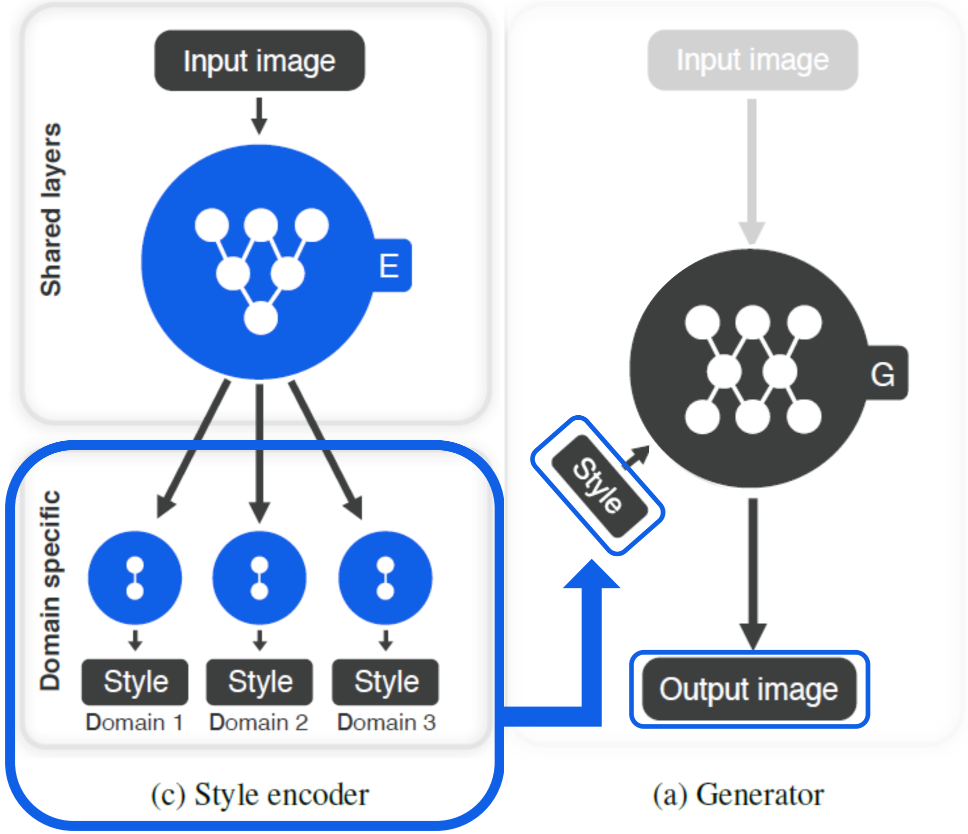
- Determines the **similarity** of two images
- Compares **generated** image and **input** images
- **Higher** is better

Latent-guided images



Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

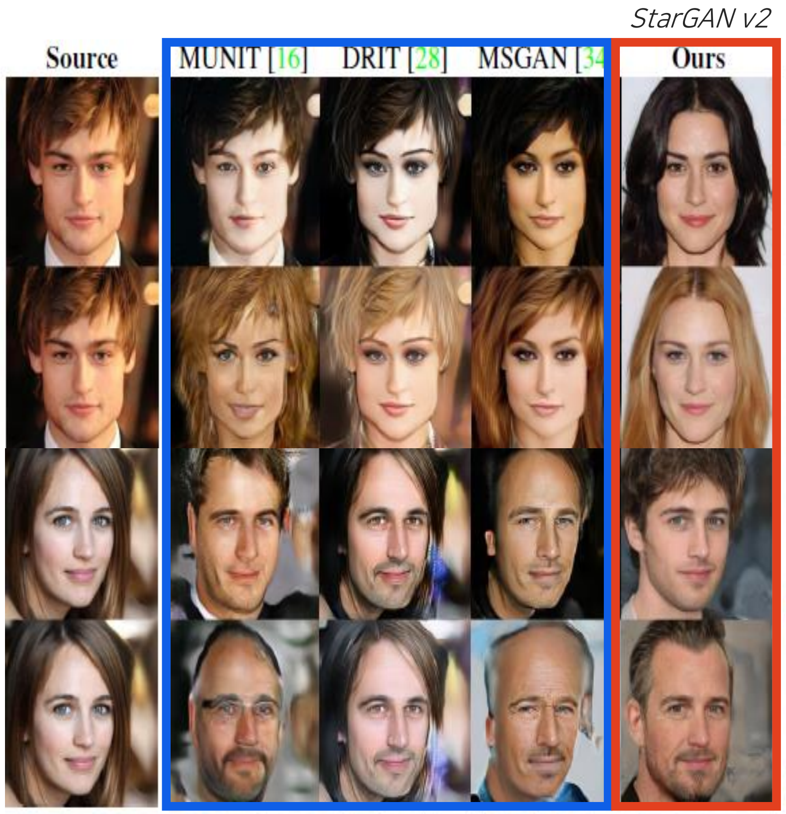
Reference-guided images



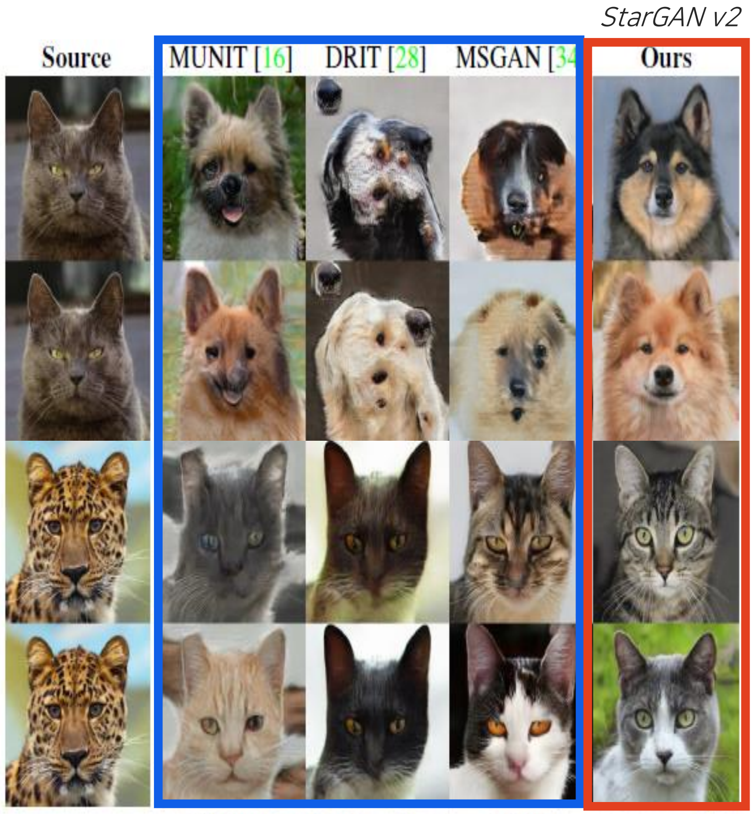
Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

Evaluation

Latent-guided images



(a) Latent-guided synthesis on CelebA-HQ



(b) Latent-guided synthesis on AFHQ

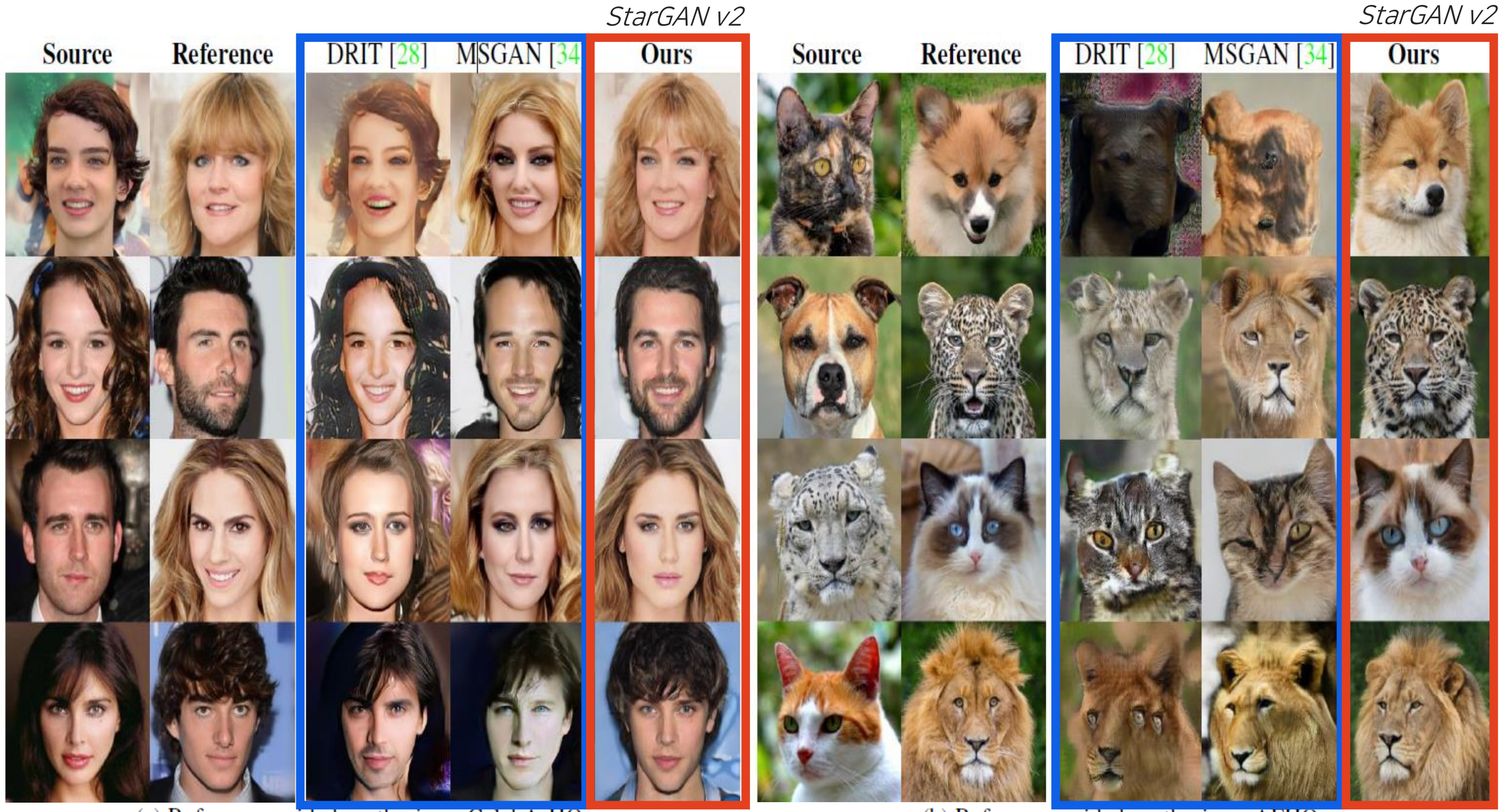
Method	CelebA-HQ		AFHQ	
	FID	LPIPS	FID	LPIPS
MUNIT [16]	31.4	0.363	41.5	0.511
DRIT [28]	52.1	0.178	95.6	0.326
MSGAN [34]	33.1	0.389	61.4	**0.517**
StarGAN v2	**13.7**	**0.452**	**16.2**	0.450
Real images	14.8	-	12.9	-

StarGAN v2 is outstanding on both datasets

Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

Evaluation

Reference-guided images



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

Method	CelebA-HQ		AFHQ	
	FID	**LPIPS**	**FID**	**LPIPS**
MUNIT [16]	107.1	0.176	223.9	0.199
DRIT [28]	53.3	0.311	114.8	0.156
MSGAN [34]	39.6	0.312	69.8	0.375
StarGAN v2	**23.8**	**0.388**	**19.8**	**0.432**
Real images	14.8	-	12.9	-

StarGAN v2 is outstanding on both datasets

Source: "*StarGAN v2: Diverse Image Synthesis for Multiple Domains*"

IV. CODE DEMO

V. CHALLENGES

Challenges

Google Colab disconnected often

Limited runtime

Paid off Colab Pro

VI. CONCLUSION

StarGAN v2

- **First GAN model** to solve the **two main problems** in I2I translation
  1. **Diverse images** across an increasing number of domains
  2. **Generating different images** per each domain
- Multiple domains with **single model**
- Usage of **style code** to generalize
- Great **multi-task discriminator**

StarGAN v2 : Requirements

- Huge dataset
- Long training time
- High resolution images

Thank you for your attention!

QnA