CollaGAN: Collaborative GAN for Missing Image Data Imputation

Dongwook Lee[1], Junyoung Kim[1], Won-Jin Moon[2], Jong Chul Ye[1]
[1]: Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea
{dongwook.lee, junyoung.kim, jong.ye}@kaist.ac.kr
[2]: Konkuk University Medical Center, Seoul, Korea
mdmoonwj@kuh.ac.kr

Presented by 이창선

Image Imputation

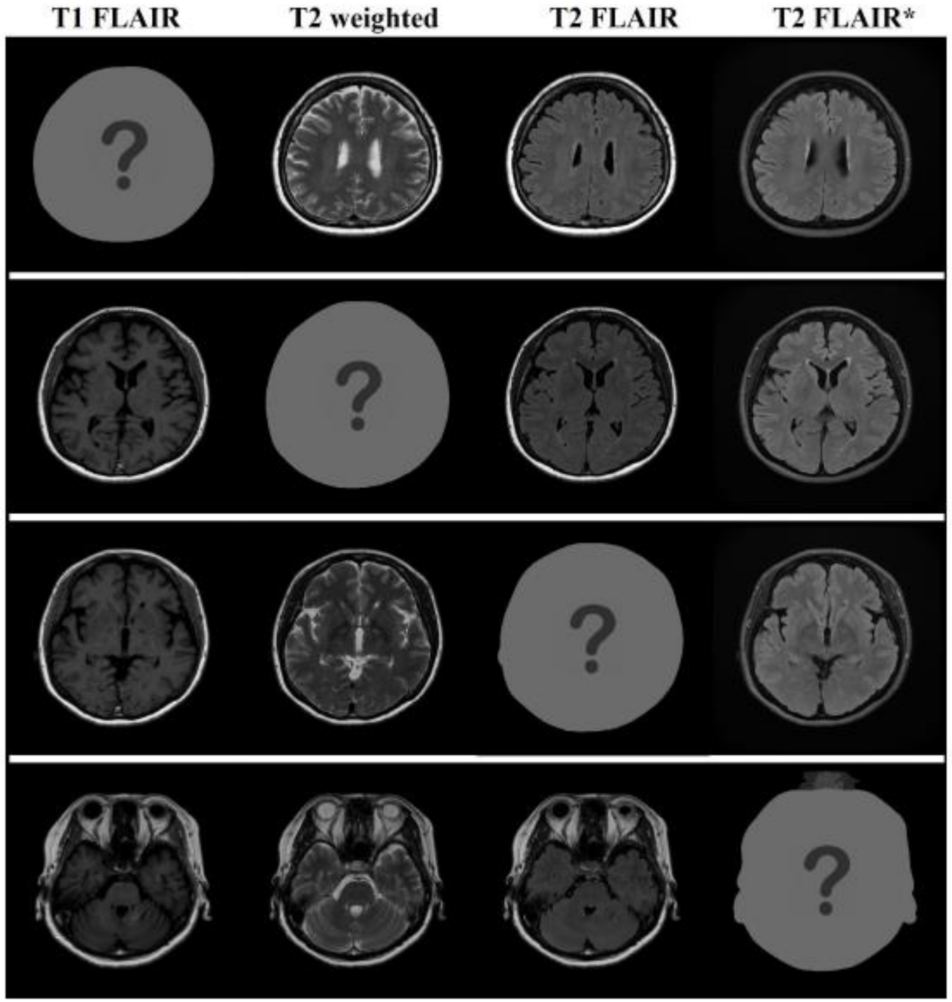
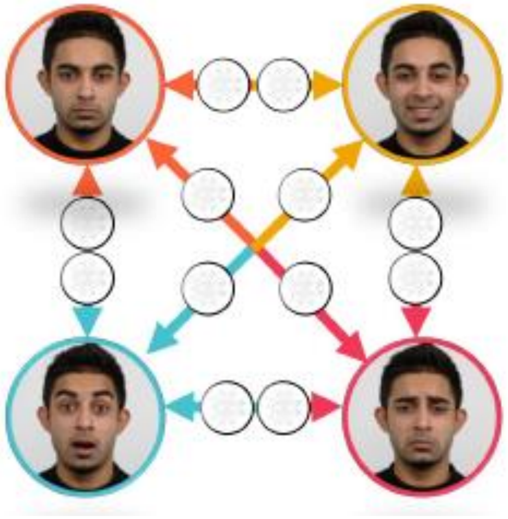
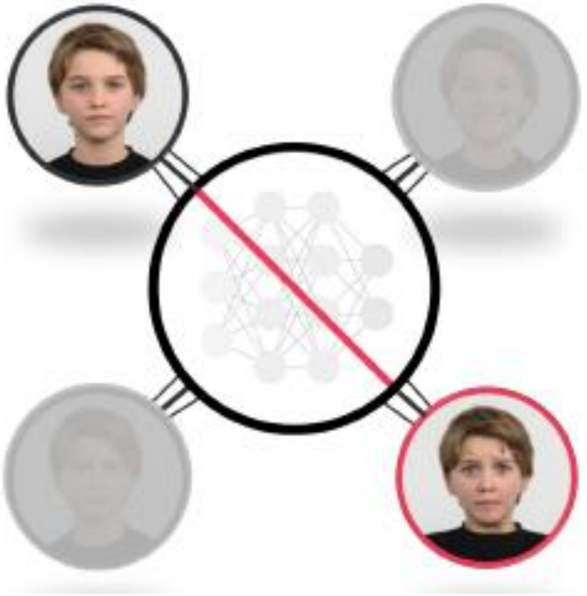


Image Translation vs. Imputation

(a) Cross-domain models

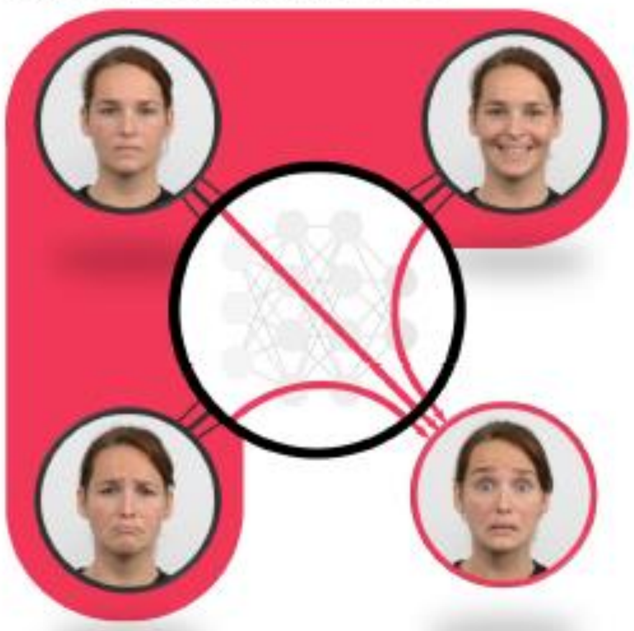


(b) StarGAN



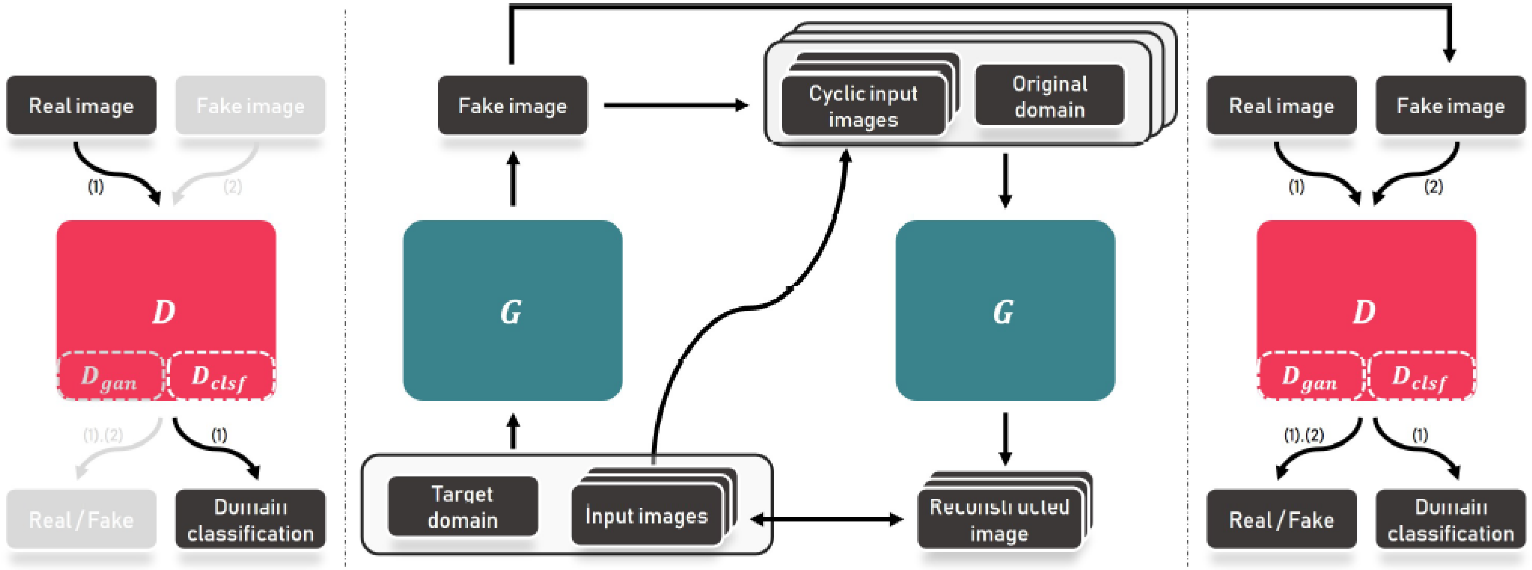
vs.

(c) Collaborative GAN

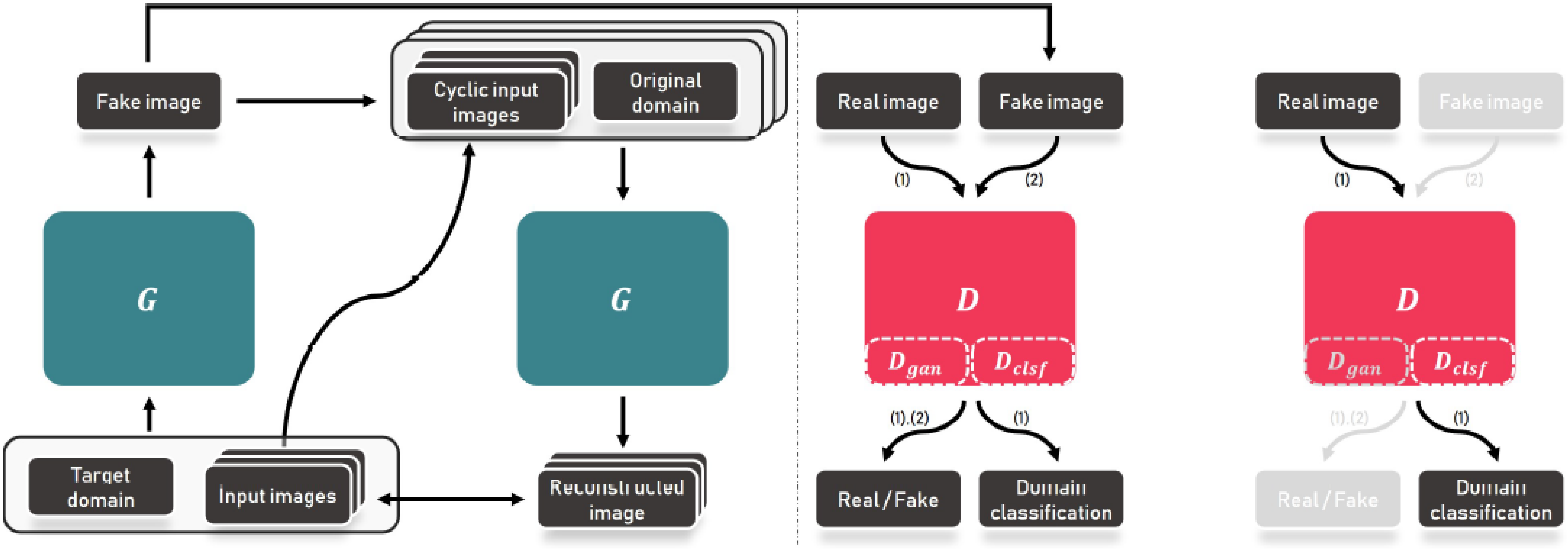


CollaGAN

- **ONE** algorithm
- Estimate the Missing Data in **ANY** domain
- Exploits the data for the **REST** of the domains



Method (Overview)

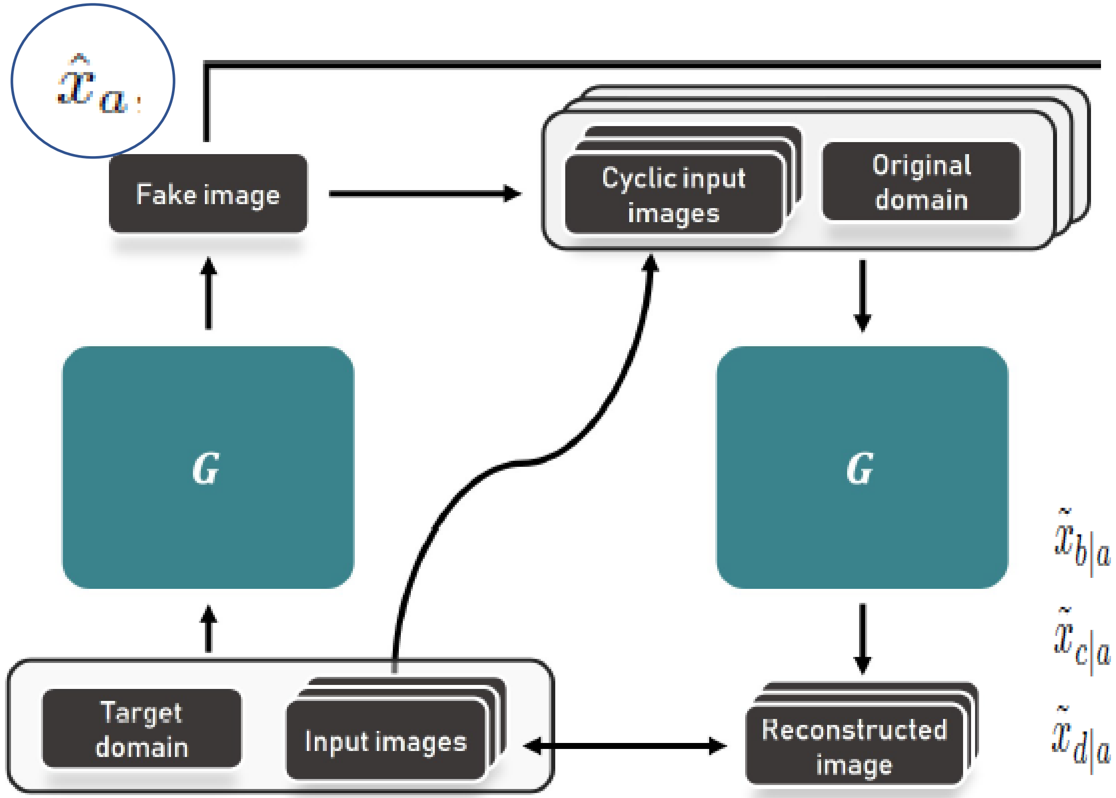


Method (Network Loss)

- Multiple Cycle Consistency Loss

e.g) Four types (N = 4) of domains: a, b, c, d

Let’s synthesize the fake image in the target domain a using b, c and d, called $\hat{x}_a$



$$\begin{aligned}\tilde{x}_{b|a} &= G(\{\hat{x}_a, x_c, x_d\}; b)\\ \tilde{x}_{c|a} &= G(\{\hat{x}_a, x_b, x_d\}; c)\\ \tilde{x}_{d|a} &= G(\{\hat{x}_a, x_b, x_c\}; d)\end{aligned}$$

In general,

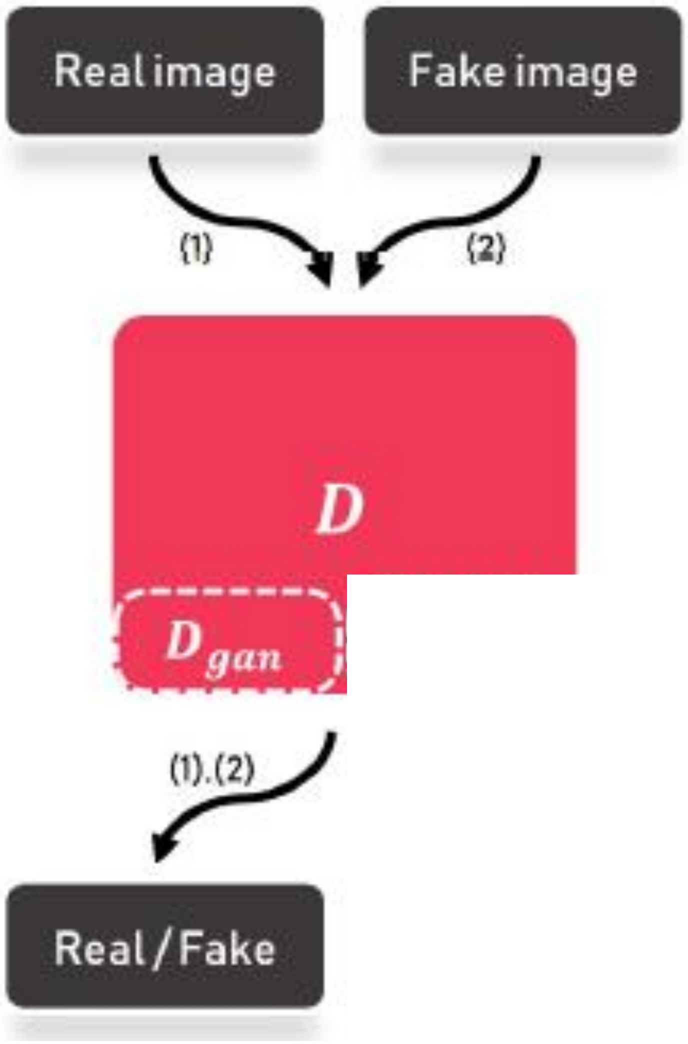
$$\mathcal{L}_{mcc,\kappa} = \sum_{\kappa' \neq \kappa} ||x_{\kappa'} - \tilde{x}_{\kappa'|\kappa}||_1$$

Method (Network Loss)

- Discriminator Loss
  - Adversarial Loss of LSGAN
    - Overcome vanishing gradient problem

$$\mathcal{L}_{gan}^{dsc}(D_{gan}) = \mathbb{E}_{x_\kappa}[(D_{gan}(x_\kappa)-1)^2] + \mathbb{E}_{\tilde{x}_{\kappa|\kappa}}[(D_{gan}(\tilde{x}_{\kappa|\kappa}))^2]$$

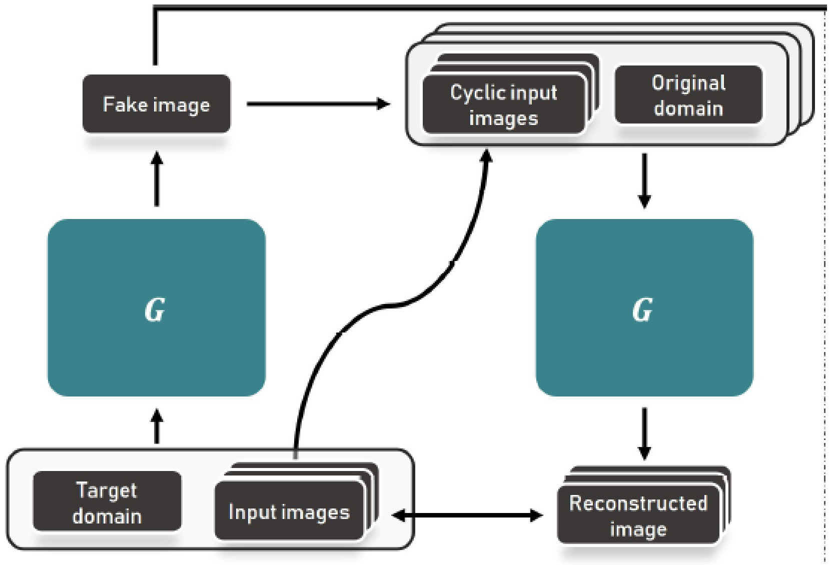
$$\mathcal{L}_{gan}^{gen}(G) = \mathbb{E}_{\tilde{x}_{\kappa|\kappa}}[(D_{gan}(\tilde{x}_{\kappa|\kappa}) - 1)^2]$$

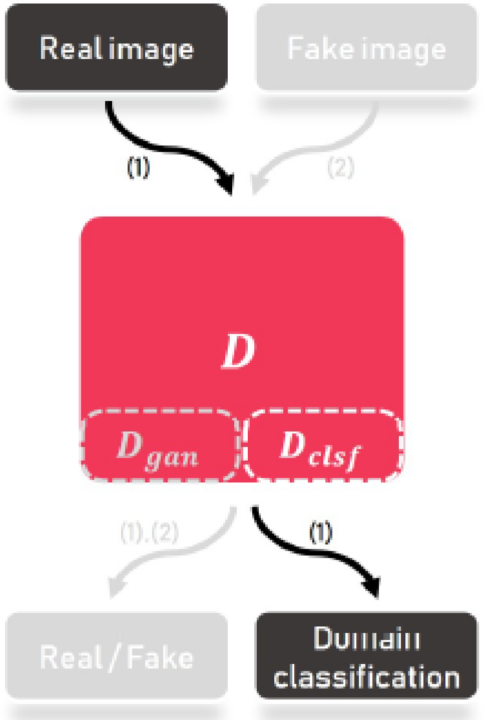


Method (Network Loss)

- Discriminator Loss
  - Domain Classification Loss

$$\mathcal{L}_{clsf}^{real}(D_{clsf}) = \mathbb{E}_{x_\kappa}[-\log(D_{clsf}(\kappa; \boxed{x_\kappa}))]$$





$$\mathcal{L}_{clsf}^{fake}(G) = \mathbb{E}_{\hat{x}_{\kappa|\kappa}}[-\log(D_{clsf}(\kappa; \boxed{\hat{x}_{\kappa|\kappa}}))]$$

Method (Network Loss)

- Structural Similarity Index Loss

$$\mathrm{SSIM}(p) = \frac{2\mu_X\mu_Y + C_1}{\mu_X^2 + \mu_Y^2 + C_1} \cdot \frac{2\sigma_{XY} + C_2}{\sigma_X^2 + \sigma_Y^2 + C_2}$$

  - Luminance

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

  - Contrast

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2\right)^{\frac{1}{2}}$$

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$\mathrm{SSIM}(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^{\alpha} \cdot [c(\mathbf{x}, \mathbf{y})]^{\beta} \cdot [s(\mathbf{x}, \mathbf{y})]^{\gamma}$$

  - Structure

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

$$\mathcal{L}_{\mathrm{SSIM}}(X, Y) = -\log\left(\frac{1}{2|P|}\sum_{p \in P(X,Y)}(1 + \mathrm{SSIM}(p))\right) \longrightarrow \mathcal{L}_{mcc-\mathrm{SSIM},\kappa} = \sum_{\kappa' \neq \kappa} \mathcal{L}_{\mathrm{SSIM}}\left(x_{\kappa'}, \tilde{x}_{\kappa'|\kappa}\right)$$

Full Objective

- Finally, the objective (loss) functions to optimize G and D are written , respectively, as

Generator

$$\mathcal{L}_{mcc,\kappa} = \sum_{\kappa' \neq \kappa} ||x_{\kappa'} - \tilde{x}_{\kappa'|\kappa}||_1$$

$$\mathcal{L}_{gan}^{gen}(G) = \mathbb{E}_{\tilde{x}_{\kappa|\kappa}}[(D_{gan}(\tilde{x}_{\kappa|\kappa}) - 1)^2]$$

$$\mathcal{L}_{clsf}^{fake}(G) = \mathbb{E}_{\hat{x}_{\kappa|\kappa}}[-\log(D_{clsf}(\kappa; \hat{x}_{\kappa|\kappa}))]$$

$$\mathcal{L}_{\mathrm{SSIM}}(X, Y) = -\log\left(\frac{1}{2|P|}\sum_{p \in P(X,Y)}(1 + \mathrm{SSIM}(p))\right)$$

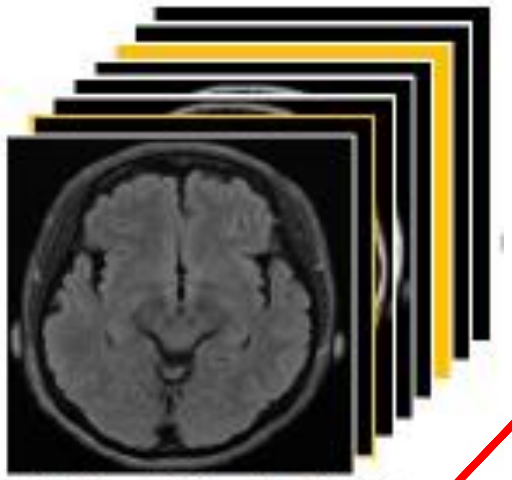
Discriminator

$$\mathcal{L}_{gan}^{dsc}(D_{gan}) = \mathbb{E}_{x_\kappa}[(D_{gan}(x_\kappa) - 1)^2] + \mathbb{E}_{\tilde{x}_{\kappa|\kappa}}[(D_{gan}(\tilde{x}_{\kappa|\kappa}))^2]$$

$$\mathcal{L}_{clsf}^{real}(D_{clsf}) = \mathbb{E}_{x_\kappa}[-\log(D_{clsf}(\kappa; x_\kappa))]$$

Method (Mask Vector)

- Selecting the Target Domain

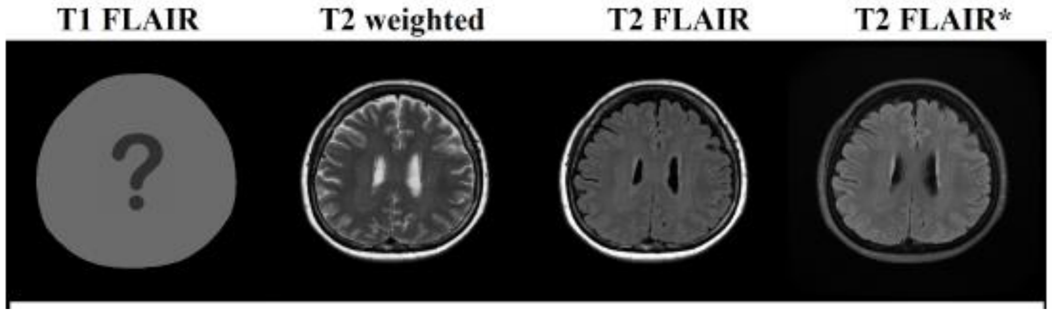



```
self.a_img = tf.placeholder(dtype,[self.nB,self.nCh_out, self.nY, self.nX])
self.b_img = tf.placeholder(dtype,[self.nB,self.nCh_out, self.nY, self.nX])
self.c_img = tf.placeholder(dtype,[self.nB,self.nCh_out, self.nY, self.nX])
self.d_img = tf.placeholder(dtype,[self.nB,self.nCh_out, self.nY, self.nX])

self.a_mask = tf.placeholder(dtype,[self.nB,1, self.nY, self.nX])
self.b_mask = tf.placeholder(dtype,[self.nB,1, self.nY, self.nX])
self.c_mask = tf.placeholder(dtype,[self.nB,1, self.nY, self.nX])
self.d_mask = tf.placeholder(dtype,[self.nB,1, self.nY, self.nX])
```

```
''' generate inputs ( imag + mask ) '''
tmp_zeros = tf.zeros([self.nB,self.nCh_out,self.nY,self.nX],dtype)
inp1 = tf.cond(self.bool0, lambda:tmp_zeros, lambda:self.a_img)
inp2 = tf.cond(self.bool1, lambda:tmp_zeros, lambda:self.b_img)
inp3 = tf.cond(self.bool2, lambda:tmp_zeros, lambda:self.c_img)
inp4 = tf.cond(self.bool3, lambda:tmp_zeros, lambda:self.d_img)

input_contrasts = tf.concat([inp1,inp2,inp3,inp4],axis=ch_dim)
self.inputs = tf.concat([input_contrasts, self.a_mask, self.b_mask,self.c_mask,self.d_mask],axis=ch_dim)
```

Experiments

1) **MR Contrast Synthesis**

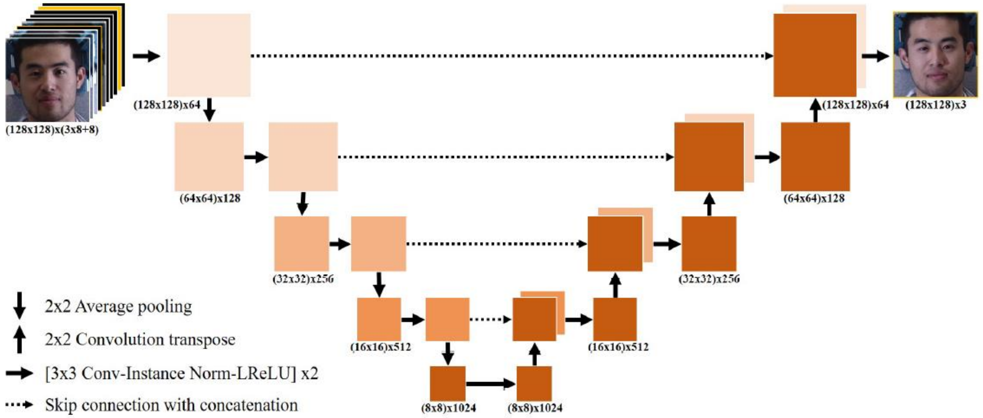


2) **CMU Multi-PIE**

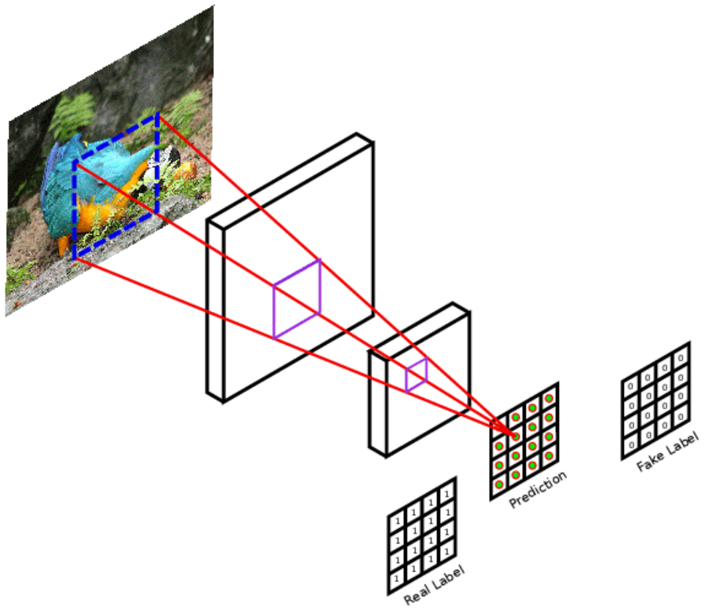
3) **RaFD**

Network Implementation

- Generator
  - U-net structure
  - Instance normalization and Leaky-Relu

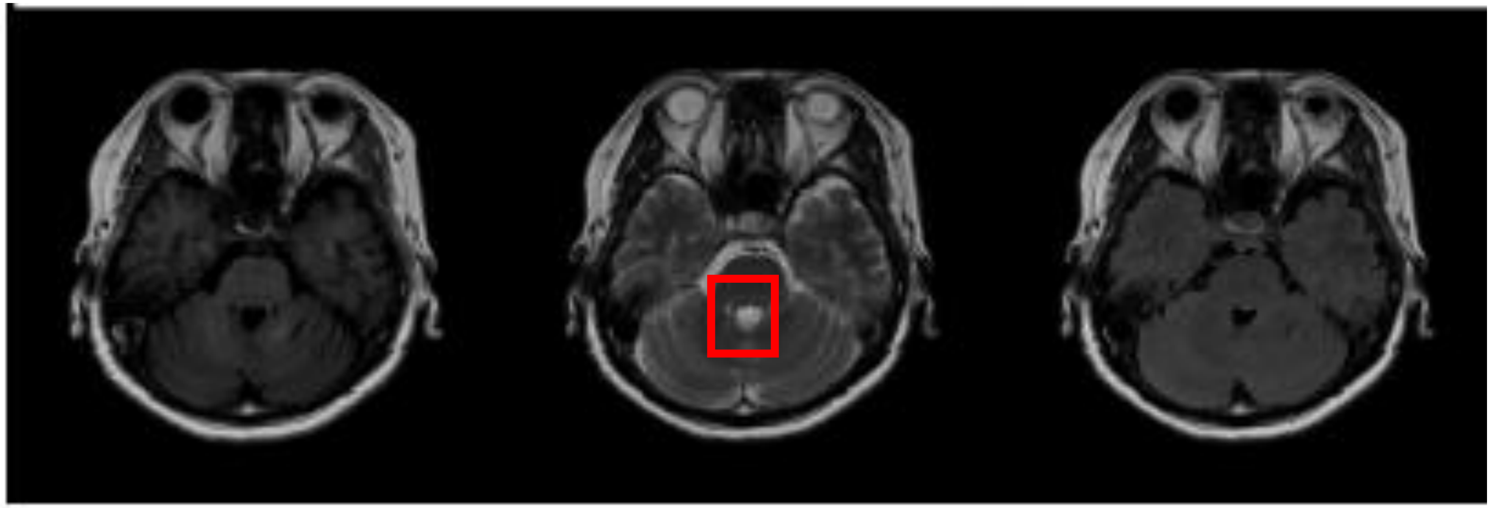


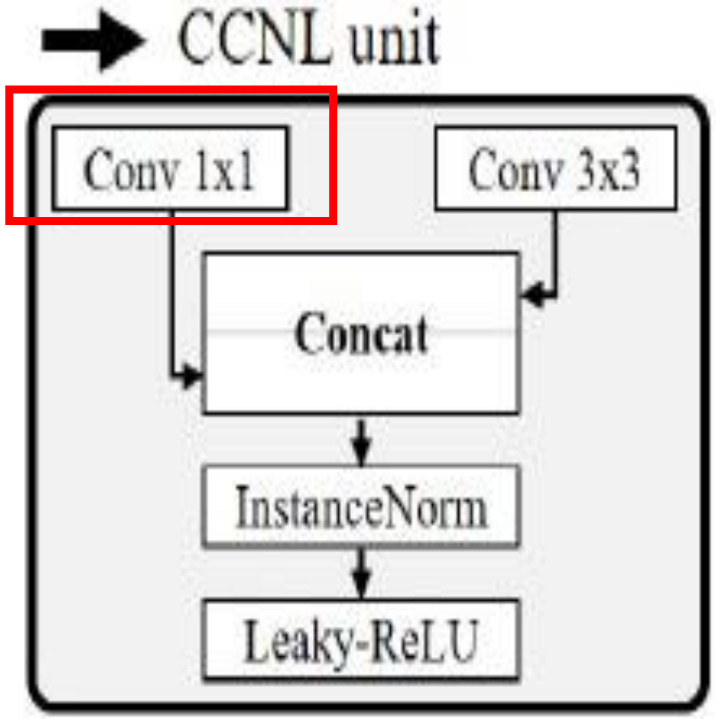
- Discriminator
  - PatchGAN



MR contrast translation

- Network Implementation (Generator)
  - Each pixel determined by MR parameters
  - MR parameters is a voxel-wise property





MR contrast translation

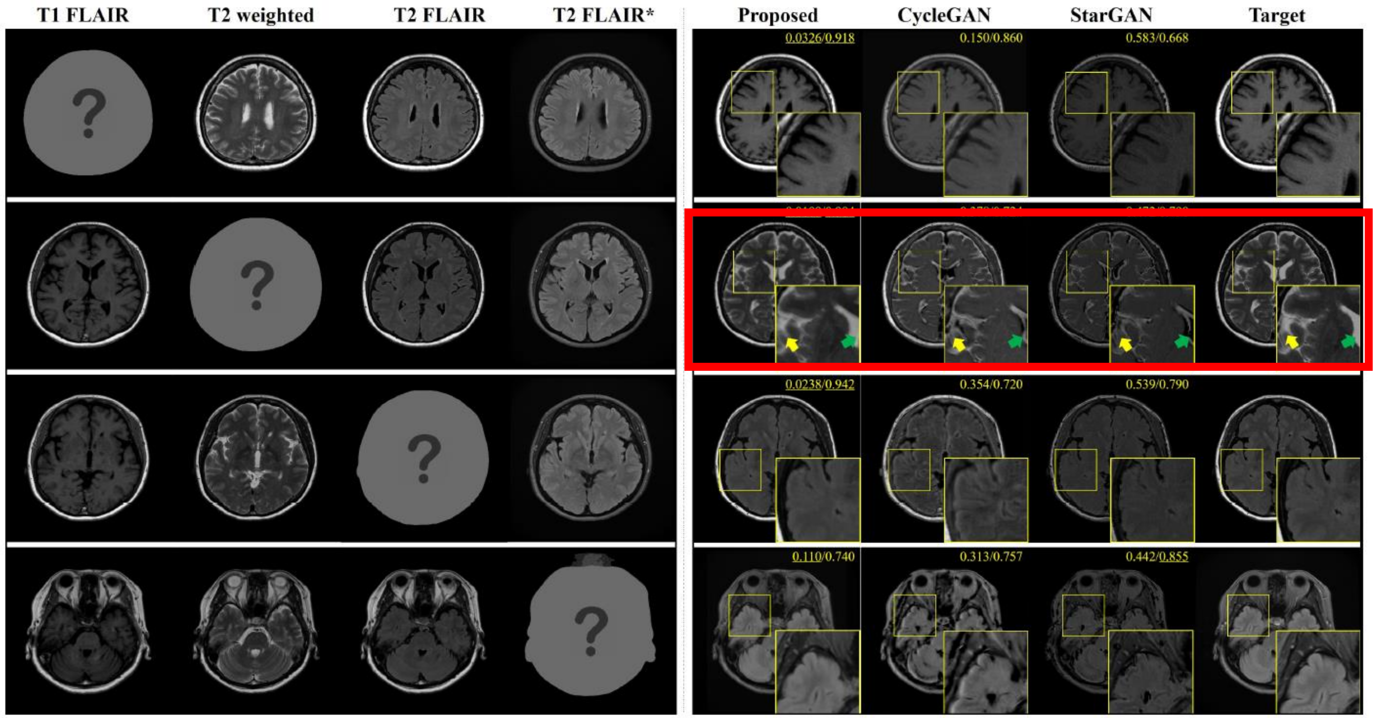
- Network Implementation (Discriminator)

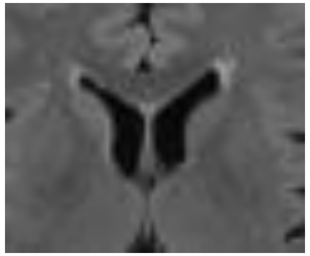
Order	Layers					k
1a	C(n4,s1)-L	C(n4,s1)-L	C(n4,s1)-L	C(n4,s1)-L	C(n16,s4)-L	4
1b	C(n4,s1)-L	C(n8,s2)-L	C(n8,s1)-L	C(n16,s2)-L	C(n16,s1)-L	4
1c	C(n16,s4)-L	C(n16,s1)-L	C(n16,s1)-L	C(n16,s1)-L	C(n16,s1)-L	4
2	1a 1b 1c	Cat	C(n32,s2)-L	C(n64,s2)-L	C(n128,s2)-L	4
3a	C(n1,s1)	Sigmoid (D_{gan})				3
3b	FC(n4)	Softmax (D_{cls})				8

Table 3: Architecture of the descriminator used for MR contrast translation. k is the kernel size for the convolution and C(n,s) represents the convolution layer with n channels and s strides. Cat, L and FC represent the concatenate layer, the leaky-ReLU layer and the fully-connected layer, respectively.

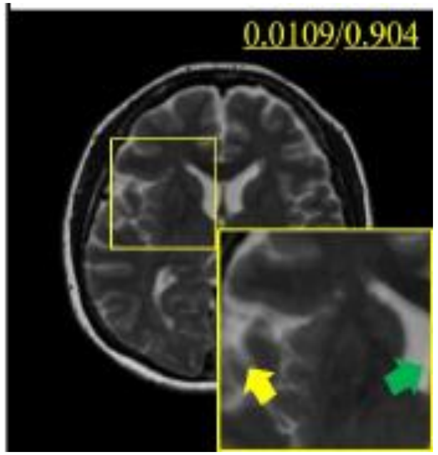
MR Contrast Translation

- Results

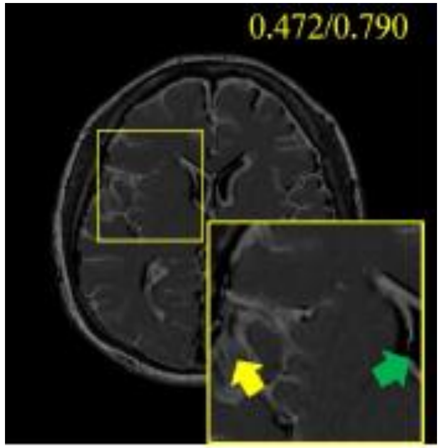




CSF
shown in
T2 FLAIR*



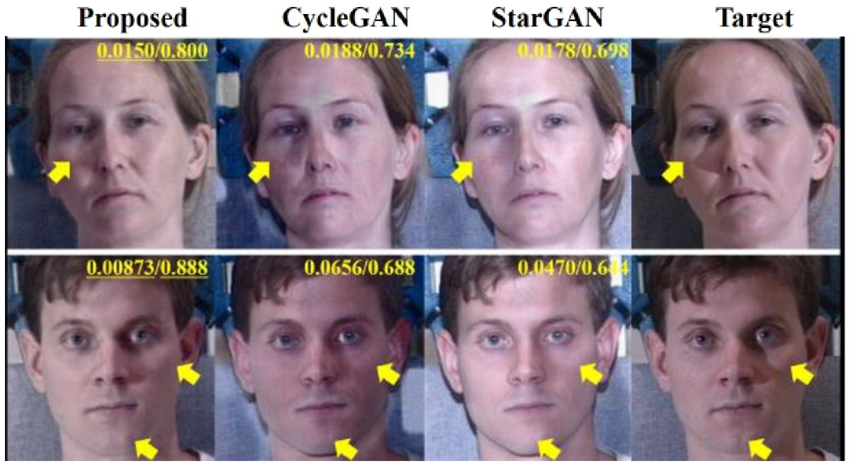
collaGAN



starGAN

Illumination Translation

- Results

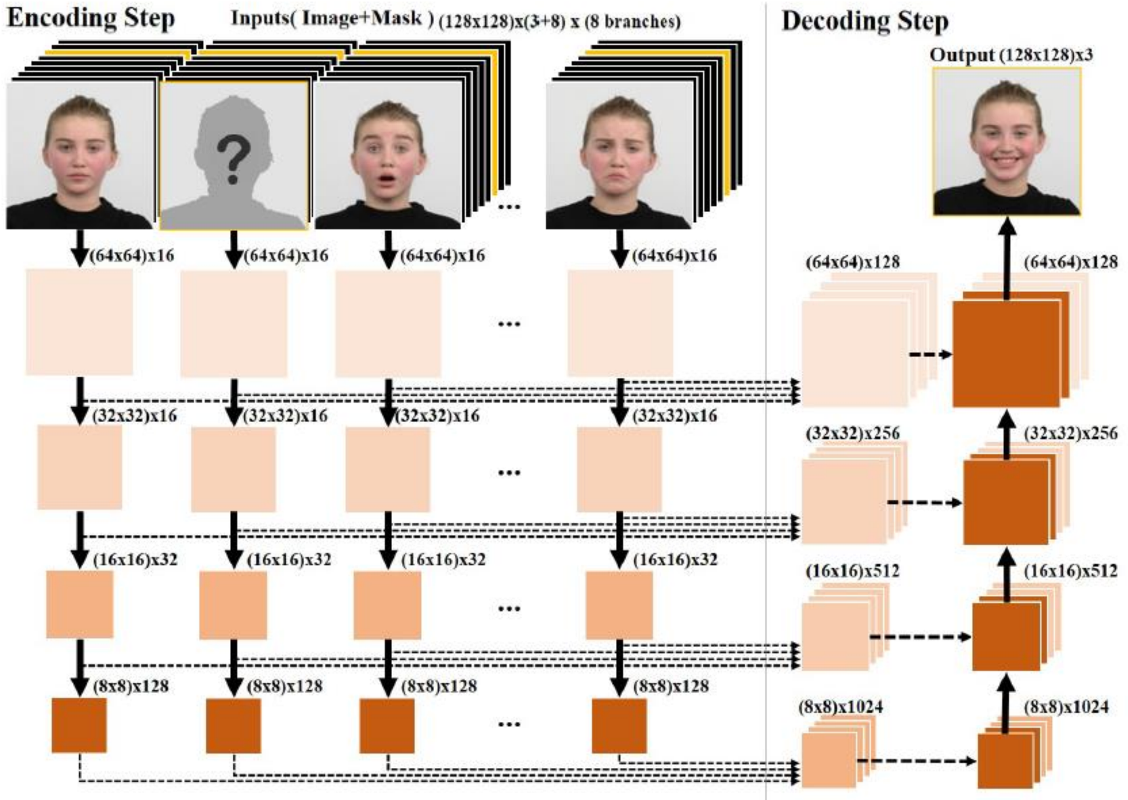


		pix2pix	CycleGAN	StarGAN	Proposed
$-90°$	NMSE	0.0334	0.0777	0.0545	**0.0122**
	SSIM	0.799	0.640	0.606	**0.876**
$-45°$	NMSE	0.0181	0.0656	0.0470	**0.00873**
	SSIM	0.840	0.688	0.644	**0.888**
$45°$	NMSE	0.0151	0.0188	0.0178	**0.0150**
	SSIM	0.607	0.734	0.698	**0.800**
$90°$	NMSE	0.0680	0.0868	0.0481	**0.00839**
	SSIM	0.708	0.665	0.668	**0.894**

Table 7: Quantitative results for illumination imputation. The NMSE/SSIM (upper/lower part for each row, respectively) are calculated from the target domain.

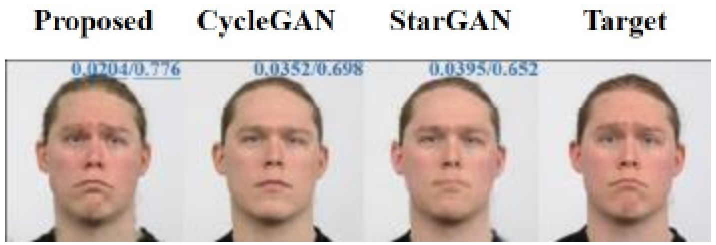
Facial Expression Translation

- Network Implementation (Generator)
  - Images are not strictly aligned pixel-wise manner
  - Features from each facial expressions mixed up in the middle stage



Facial Expression Translation

- Results

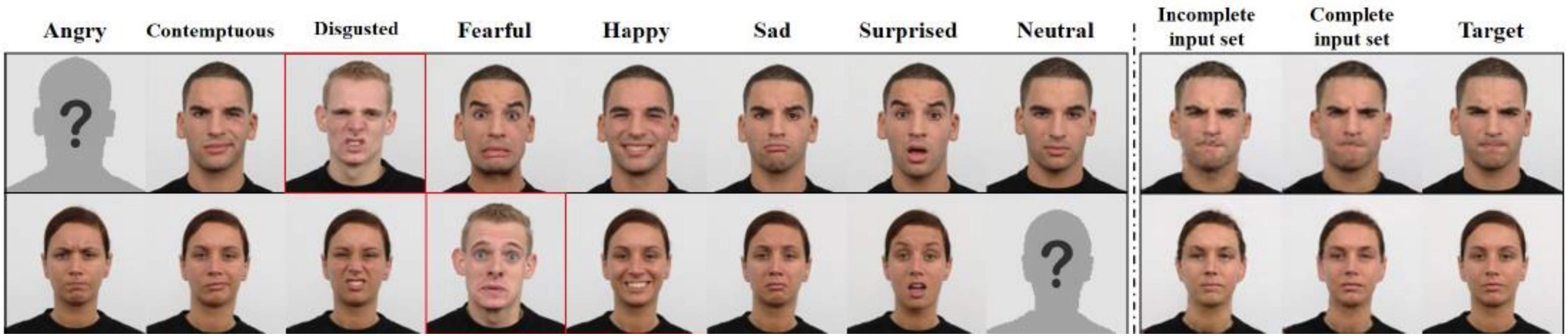


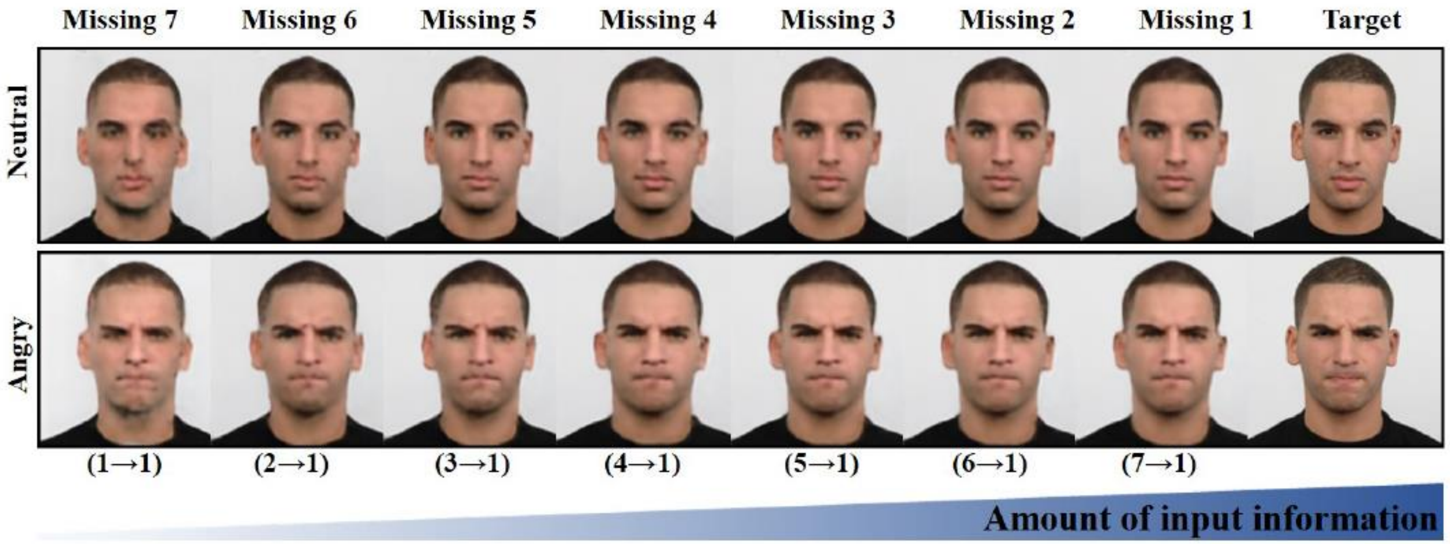
	pix2pix	CycleGAN	StarGAN	Proposed
A	0.0247	0.0301	0.0306	**0.0197**
	0.765	0.732	0.698	**0.794**
C	0.0283	0.0327	0.0421	**0.0105**
	0.724	0.0700	0.696	**0.840**
D	0.0333	0.0362	0.0397	**0.0172**
	0.716	0.694	0.683	**0.802**
F	0.0395	0.0329	0.0487	**0.0213**
	0.677	0.685	0.670	**0.761**
H	0.0345	0.0350	0.0420	**0.0211**
	0.697	0.682	0.606	**0.778**
S	0.0335	0.0268	0.0363	**0.0122**
	0.697	0.729	0.692	**0.803**
Sad	0.0349	0.0352	0.0395	**0.0204**
	0.679	0.6975	0.652	**0.776**

Table 6: Quantitative results for facial expression imputation. The NMSE/SSIM (lower/upper part for each facial expression, respectively) are calculated from each target domain (A:angry, C:contemptuous, D:disgusted, F:fearful, H:happy, S:surprised, Sad:sad).

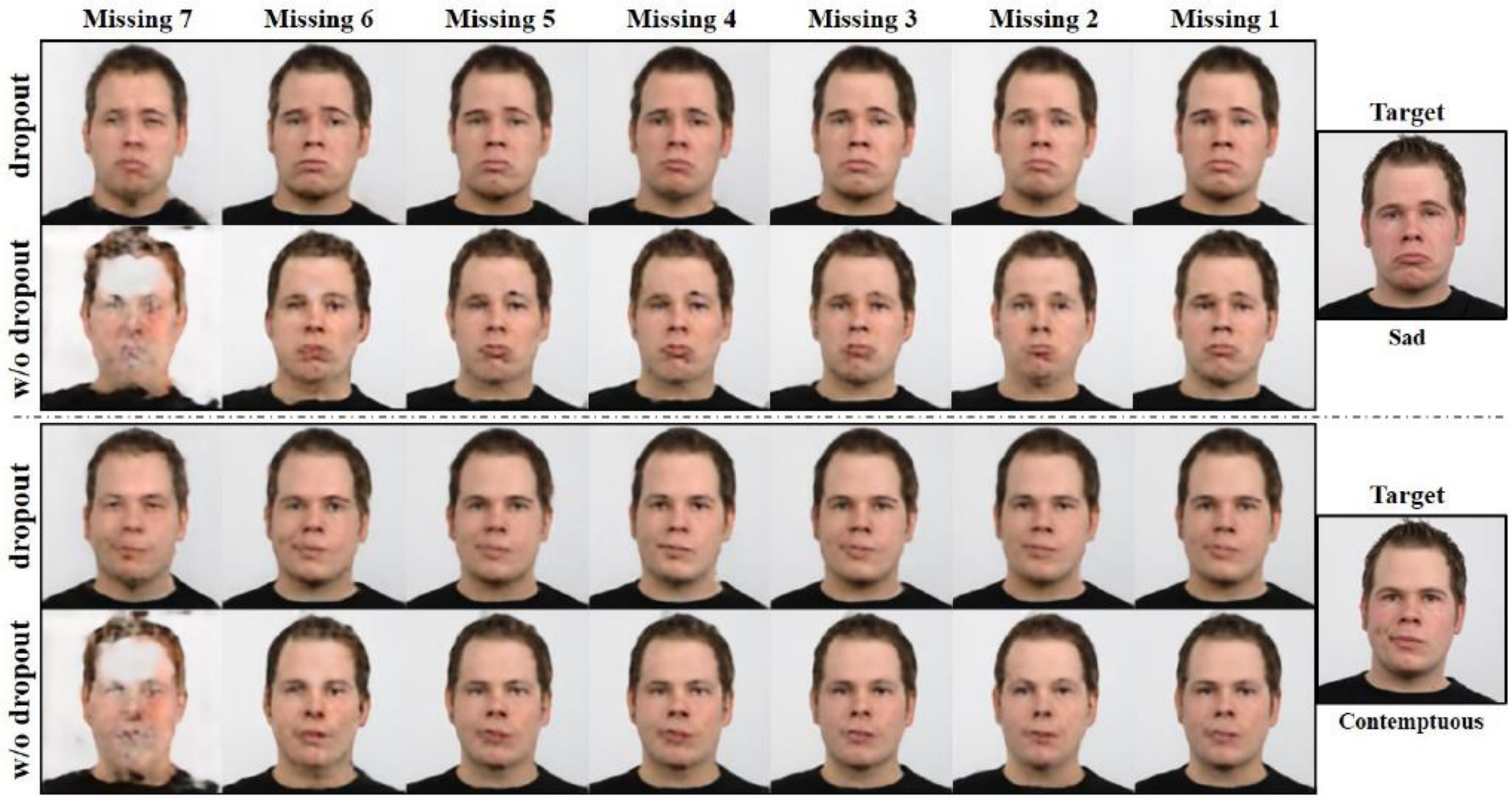
Chosen as the best	pix2pix	CycleGAN	StarGAN	Proposed
	3.8%	17.9%	7.4%	**70.8%**

Robustness of Collaborative Training





Effect of Input Dropout



Ablation Study

- Investigate the advantage of MCC loss and SSIM loss
- Using RaFD dataset

(Mean±std)	l_1 w/o L_{MCC}	w/o L_{SSIM}	Proposed
NMSE	0.0372±0.00653	0.0200±0.00391	**0.0178±0.00419**
SSIM	0.714±0.0211	0.779±0.0243	**0.793±0.0237**

Table 9: Quantitative results for the ablation study.

Q & A

Thank you for listening