Solution of Institution data Shift Problem using Multi Source Domain Transfer Learning

Jan 17, BME:Intelligent Neuroimaging Laboratory, Lab Meetings

Hyunjae Jeong (KU BME)

Korea University, Department of Biomedical Engineering

Korea University, Department of Artifial Intelligence

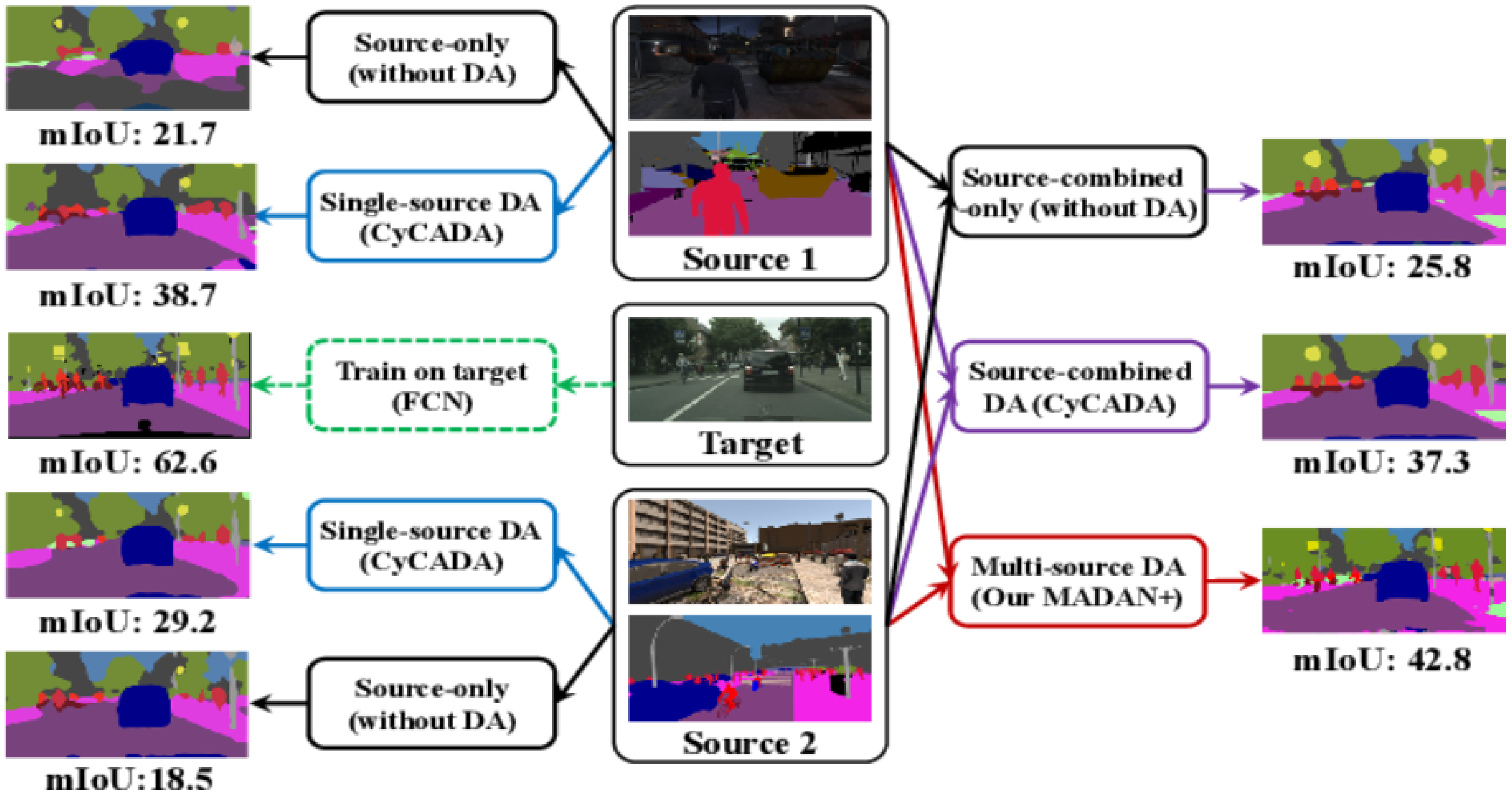
Brain Reverse Engineering by Intelligent Neuroimaging Laboratory

01 Introduction

Multi Source Domain Adaptation

A good image-to-image translation model should learn a mapping between different visual domains while satis fying the following properties: 1) diversity of generated images and 2) scalability over multiple domains.

Existing methods address either of the issues, having limited diversity or multiple models for all domains.



Multi Source Domain Adaptation

Unsupervised Multi Source Transfer Learning

01 Introduction

Multi-Site MRI Harmonization

- protocols has scanner-induced variability due to factors such as magnetic field strength, coil channels, gradient directi ons, manufacturer, and image resolution

CNN is a statistical method, which learns the statistics of the training data un der the identical indepen-dent distribution (IID) assumption, which implies that the trained CNN is supposed to work on data with identical or similar distributions.

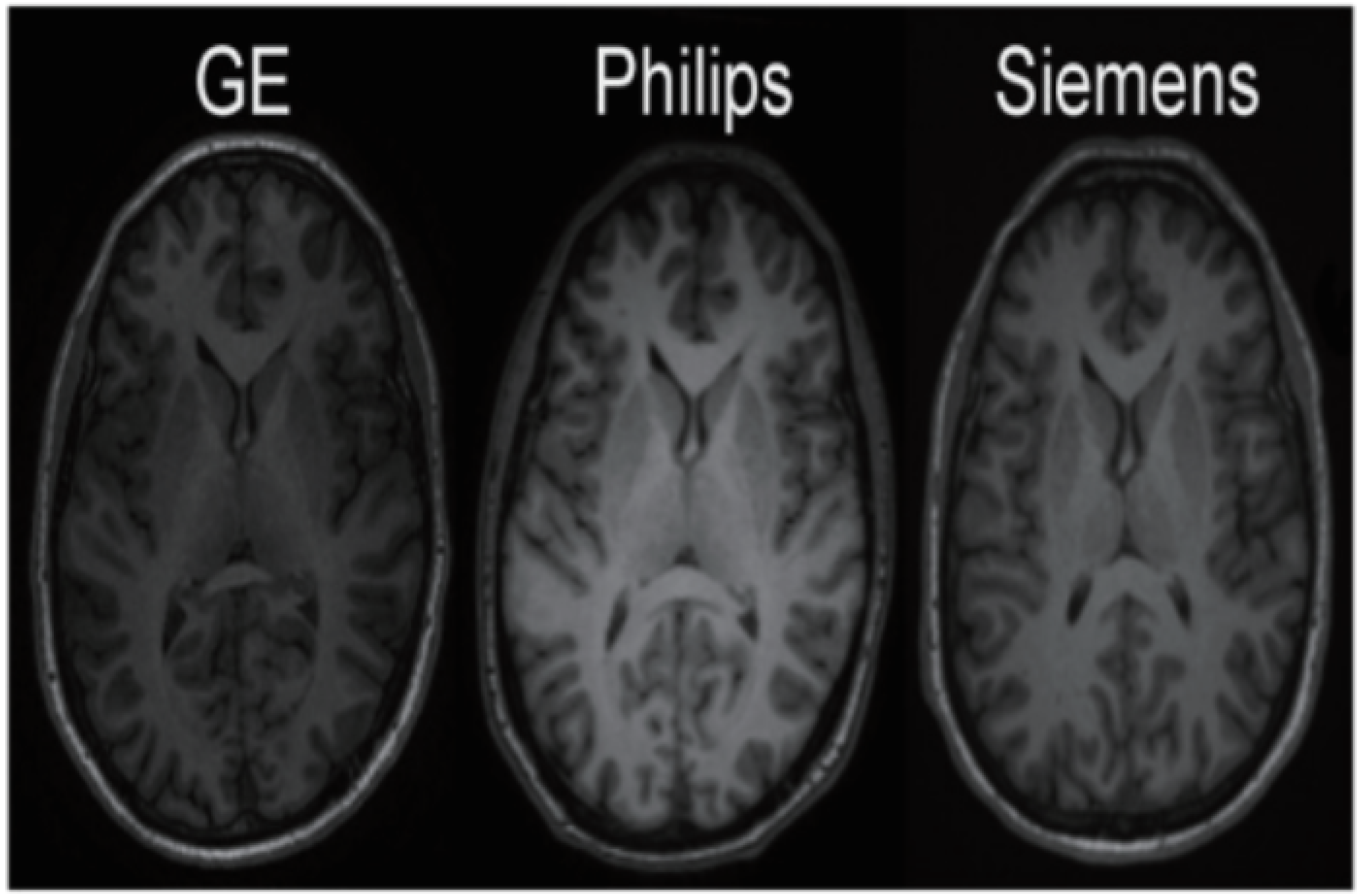


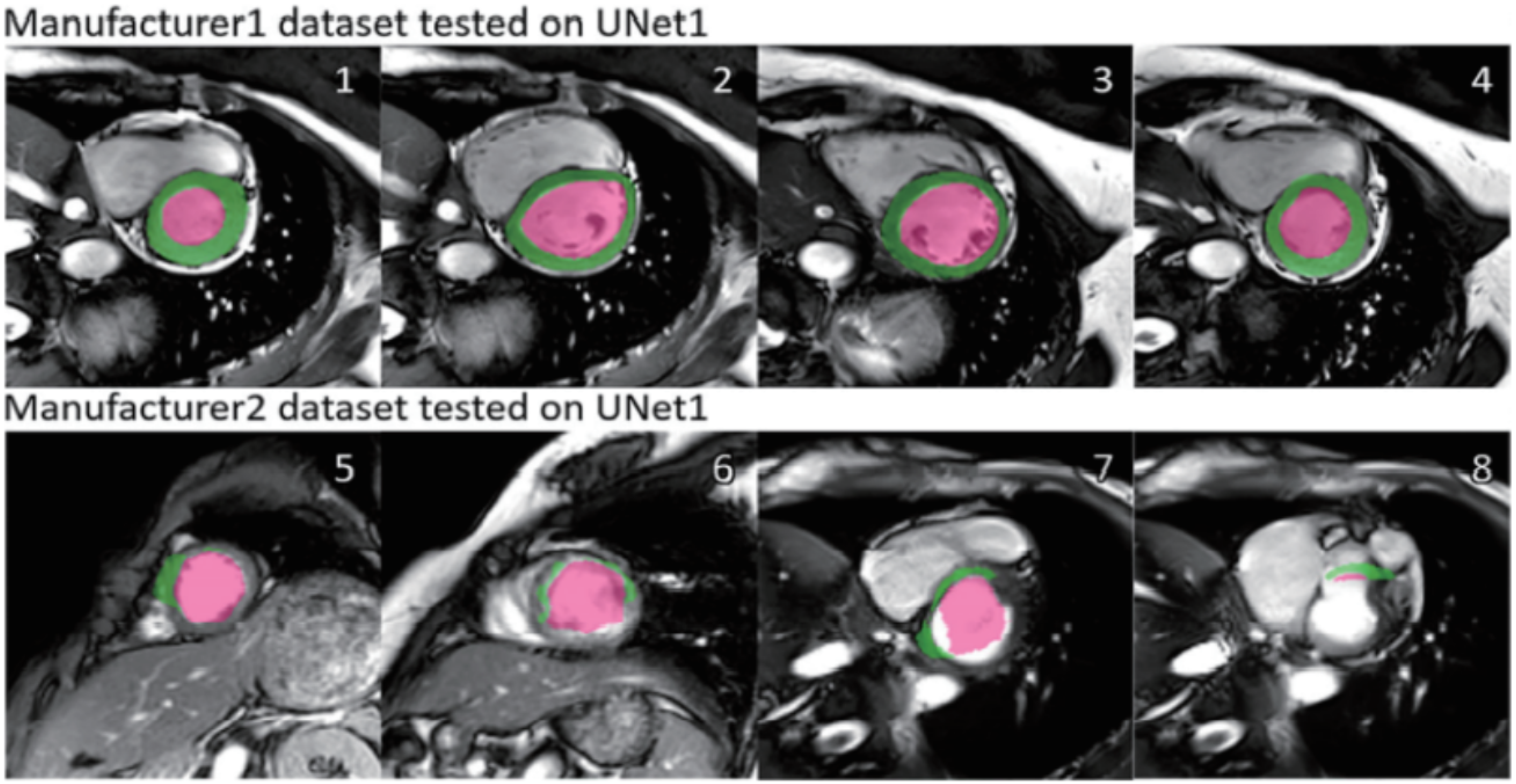
Table 1: Specifications of Cine MRI Datasets Acquired from Scanners of Different Manufacturers

MRI Scanner Manufacturer	Field Strength	In-plane Resolution (mm)	Slice Gap (mm)	Phases per Cardiac Cycle	Total No. of Frames	No. of Annotated Training Frames	No. of Annotated Testing Frames
Manufacturer 1	3.0 T	1.2 × 1.2	10	30	24905	2520	923
Manufacturer 2	1.5 T	1.17 × 1.17	9.6	20	14746	1680	924
Manufacturer 3	3.0 T	1.25 × 1.25	10	20	10640	1320	764

Note.—All manufacturer datasets had 50 patients each. For each dataset, 33 patient datasets were used for training and 17 were used for testing.

01 Introduction

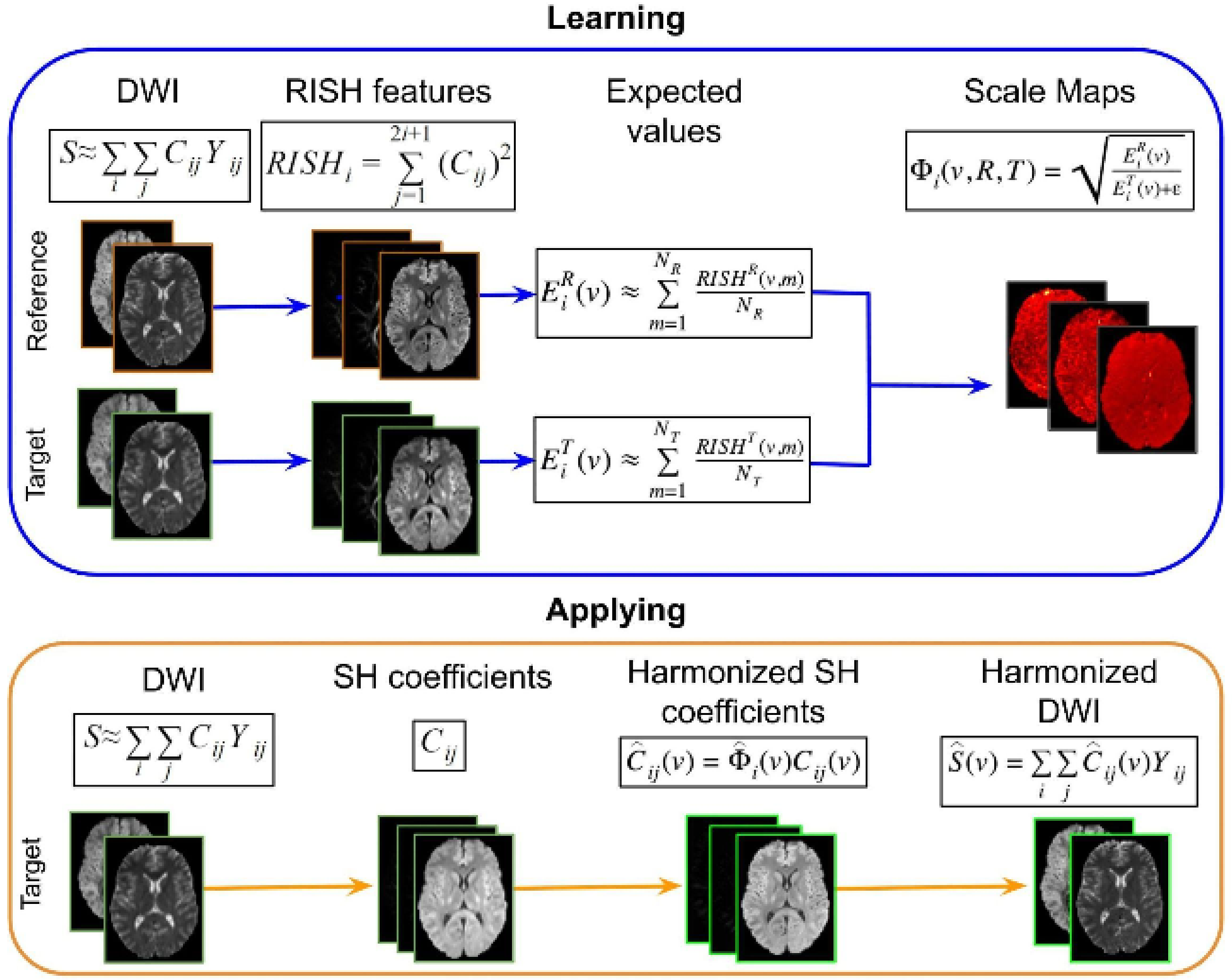
Results - Performance Drop



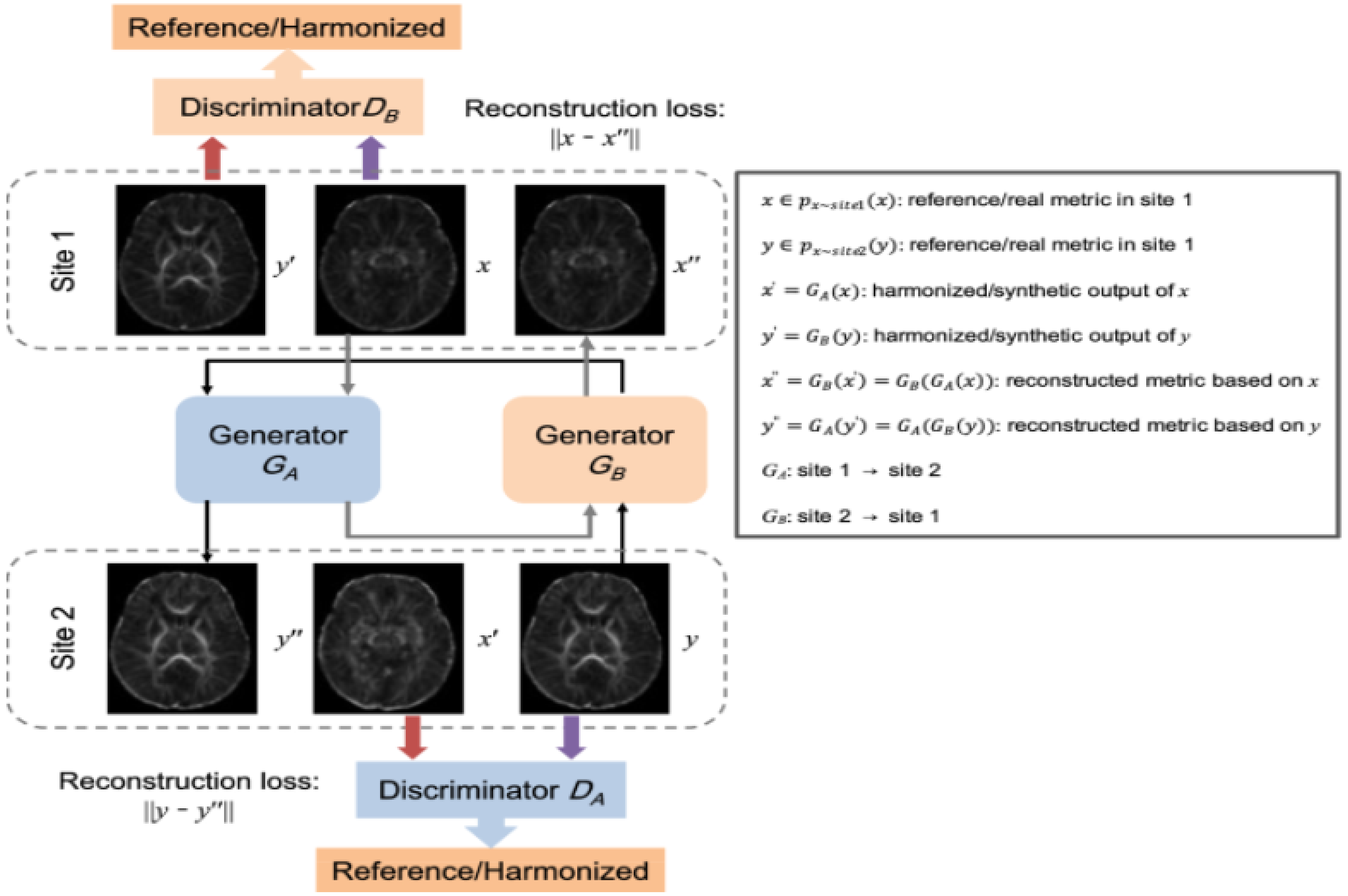
01 Introduction

Application to Medical Field - MRI Harmonization

- Large data initiatives and high-powered brain imaging analyses require the pooling of MR images acquired acr oss multiple scanners, often using different protocols.
- Several retrospective harmonization techniques - cannot distinguish between image acquisition based variabili ty and cross-site population variability



Radiomics Feature Based Adaptation



Unsupervised Single Source Transfer Learning

01 Introduction

Contribution

- Most harmonization methods fall into two broad categories:

  1) harmonization of image-derived features using statistical properties of the distribution, for example ComBat

  2) harmonization of the task output of the MR images, i.e. segmentation, classification, and prediction

↓

- A wide range of tasks were to be performed on images, harmonization would need to be performed separately for each task.

- Think anatomical patterns (or contents) from MR images collected from different sites share the same latent s pace, and it is not necessary to separate them into different “domains".

01 Introduction

Research Way

- 1. Multi by Multi Domain Adaptation Process
  Methods & Framework
  Four Protocols MRI Harmonization Experiments (ADNI, SMC, GMC, GSEV)
  Discussion

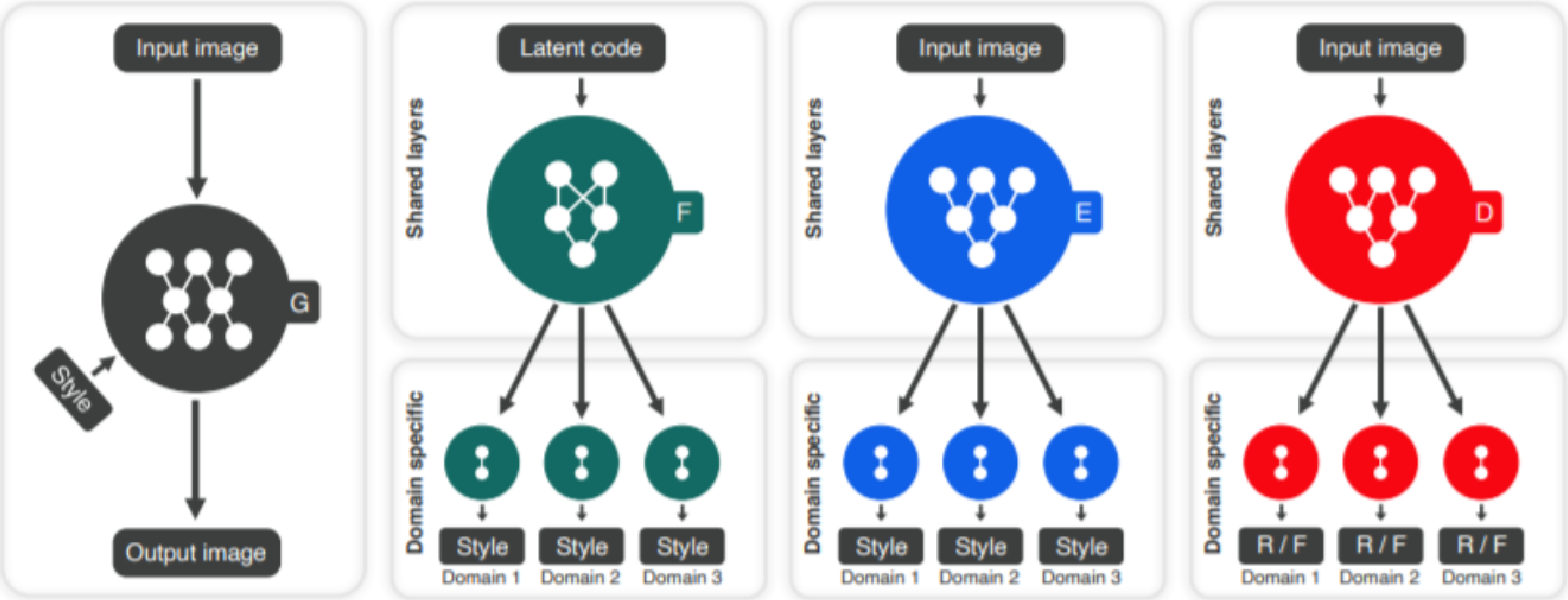
- 2. Validation Process
  2D MRI Synthesize & 3D MRI Reconstruction
  Fast Surfer Cortical Values (Thickness or Volume) Difference
  Alzheimer / Healthy Control Classification per protocols
  Discussion

- 3. Future Work

02 Methods

Methods - Proposed Framework

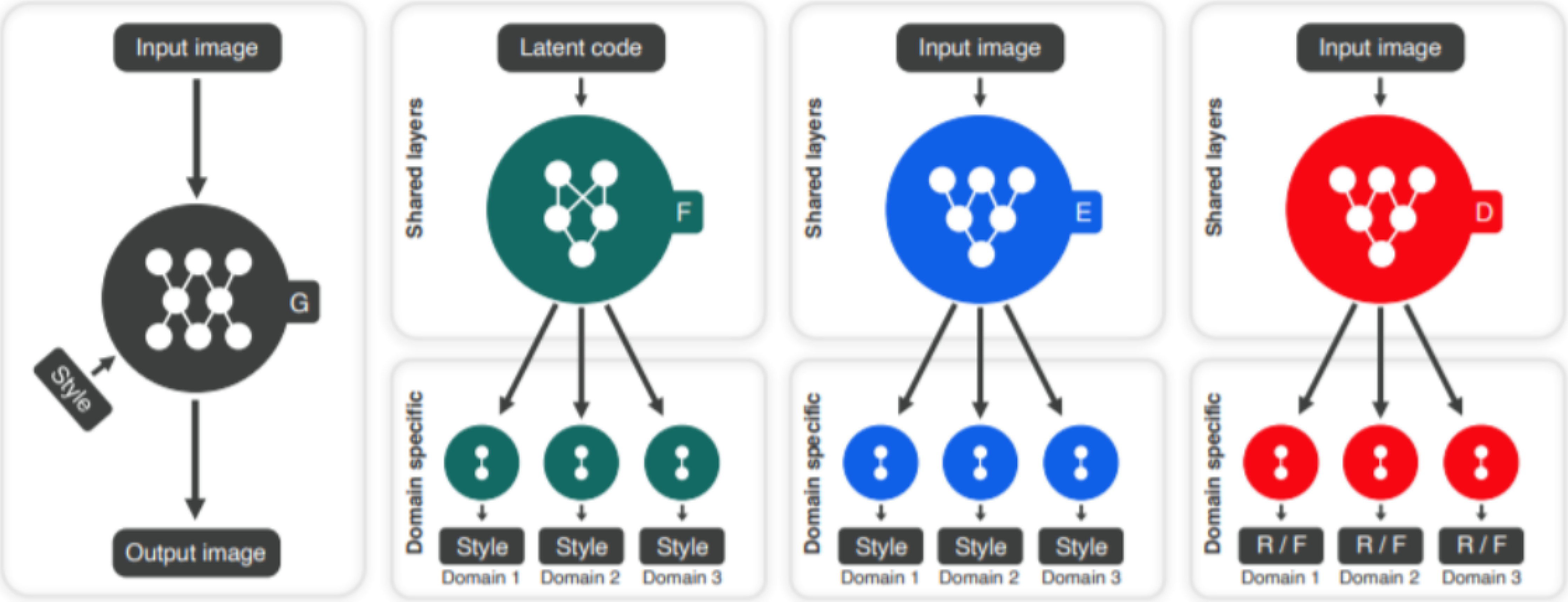
- The architecture of the style-encoding GAN (Star GAN v2). Generators learns to generate image by inputting a source image and a style code. The learning process is driven by cycle consist loss, adversarial loss, style recon struction loss and style diversification loss
- The detailed architecture of the generator in network. In each of the block, the three number means number of input channels, number of output channels and the image size.



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

Methods - The Architecture of Framework

1. The generator translates an input image into an output image reflecting the domain-specific style code.
2. The mapping network transforms a latent code into style codes for multiple domains, one of which is randomly selected during training.
3. The style encoder extracts the style code of an image, allowing the generator to perform reference guided i mage synthesis.
4. The discriminator distinguishes between real and fake images from multiple domains. Note that all modules except the generator contain multiple output branches, one of which is selected when training the correspondi ng domain.



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

02 Methods

Network Training - Seperated Loss

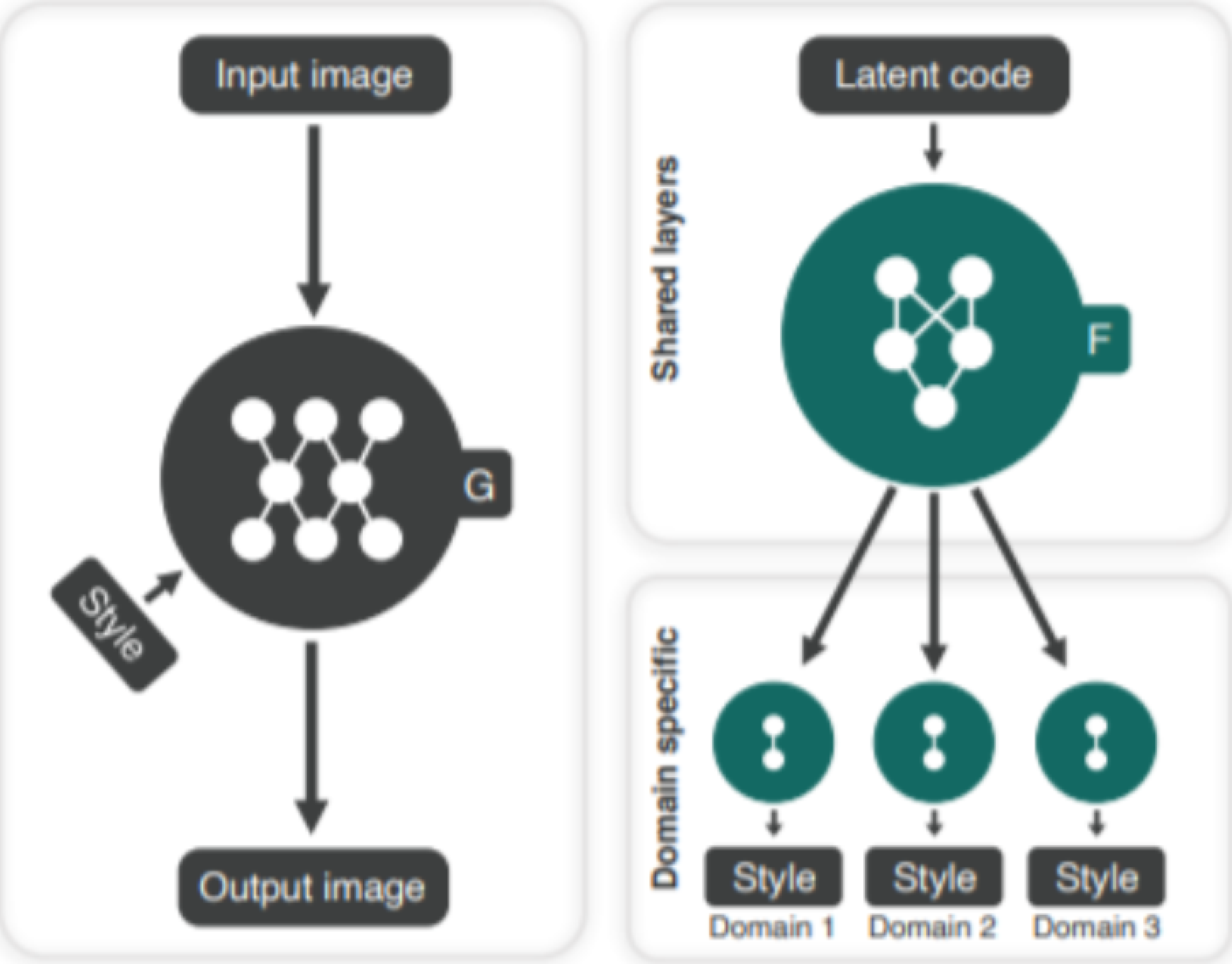
- Directly modeling the space of high-dimensional convolutional filters in an image generation network is practically prohibitive in terms of both computation and parameter scale

Adversarial loss. During training, we sample a latent code $z \in Z$ randomly, and the mapping network M learns to generate a target style code $s = M(z)$. The generator G takes an image x and s as inputs and learns to generate an output image $G(x, s)$ that is indistinguishable by the discriminator D from real images via an adversarial loss:

$$L_{GAN} = \mathbb{E}_x[\log D(x)] + \mathbb{E}_{x,z}[\log(1 - D(G(x,s)))]$$

Cycle-Consistency Loss. To guarantee that generated images are meaningful to the original images and properly preserving the style-irrelevant characteristics (e.g. anatomical patterns) of input x, an additional cycle consistency loss (Zhao et al., 2019) is defined as the difference between original and reconstructed images:

$$L_{cyc} = \mathbb{E}_{x,z}[\|x - G(G(x,s), s_x)\|_1]$$



(a) Generator (b) Mapping network

02 Methods

Network Training - Seperated Loss

- Directly modeling the space of high-dimensional convolutional filters in an image generation network is practically prohibitive in terms of both computation and parameter scale

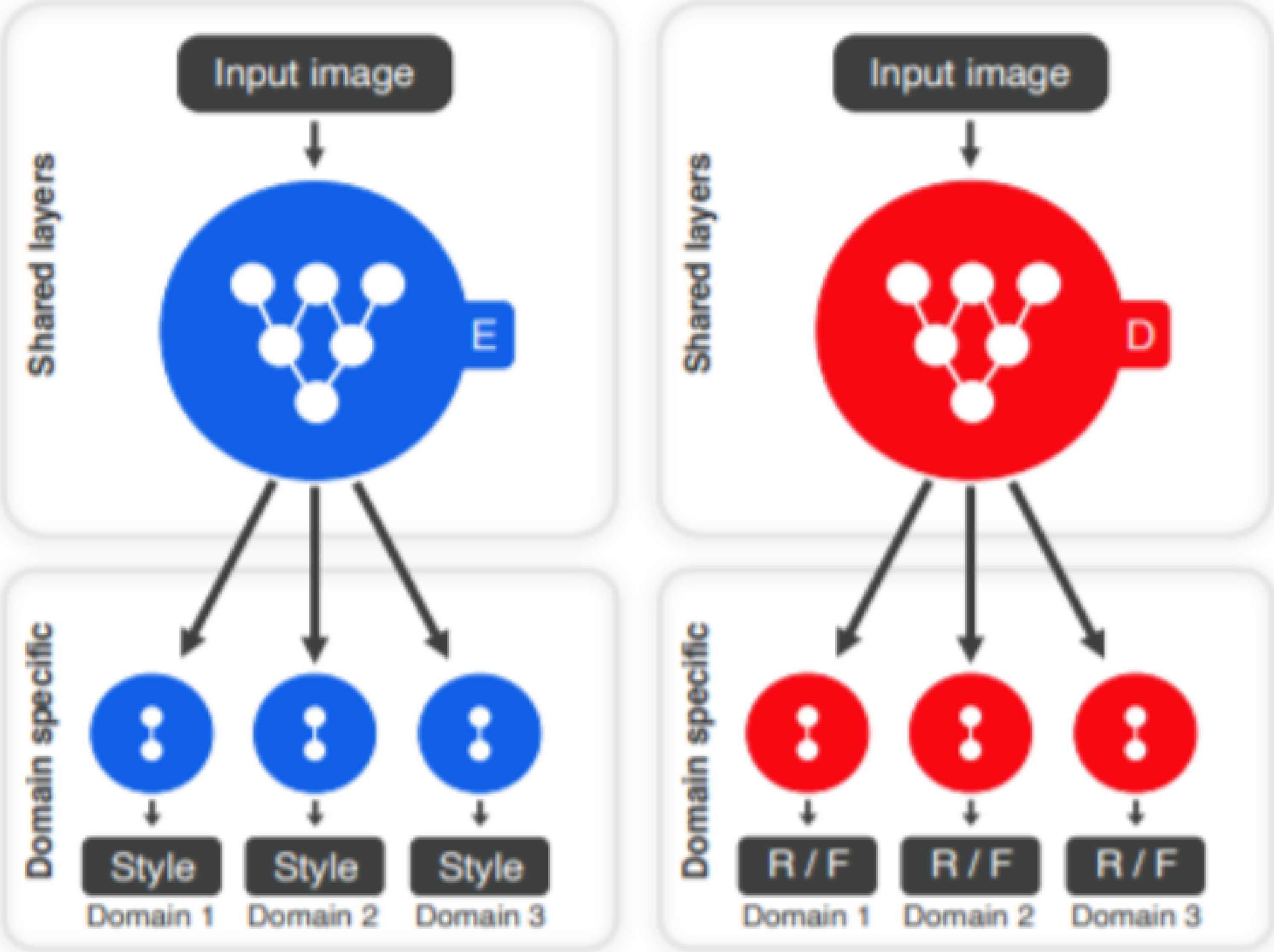
Style reconstruction loss. In order to enforce the generator G to use the style code while generating the image $G(x, s)$, we incorporate a style reconstruction loss:

$$L_{sty} = \mathbb{E}_{x,z}[\|s - E(G(x, s))\|_1]$$

Our learned encoder E allows G to transform an input image x, to reflect the style of a reference image.

Style diversification loss. To further enable the generator G to produce diverse images, we explicitly regularize G with the diversity sensitive loss (Wang et al., 2018):

$$L_{div} = \mathbb{E}_{x,z_1,z_2}[\|G(x, s_1) - G(x, s_2)\|_1]$$



(c) Style encoder (d) Discriminator

02 Methods

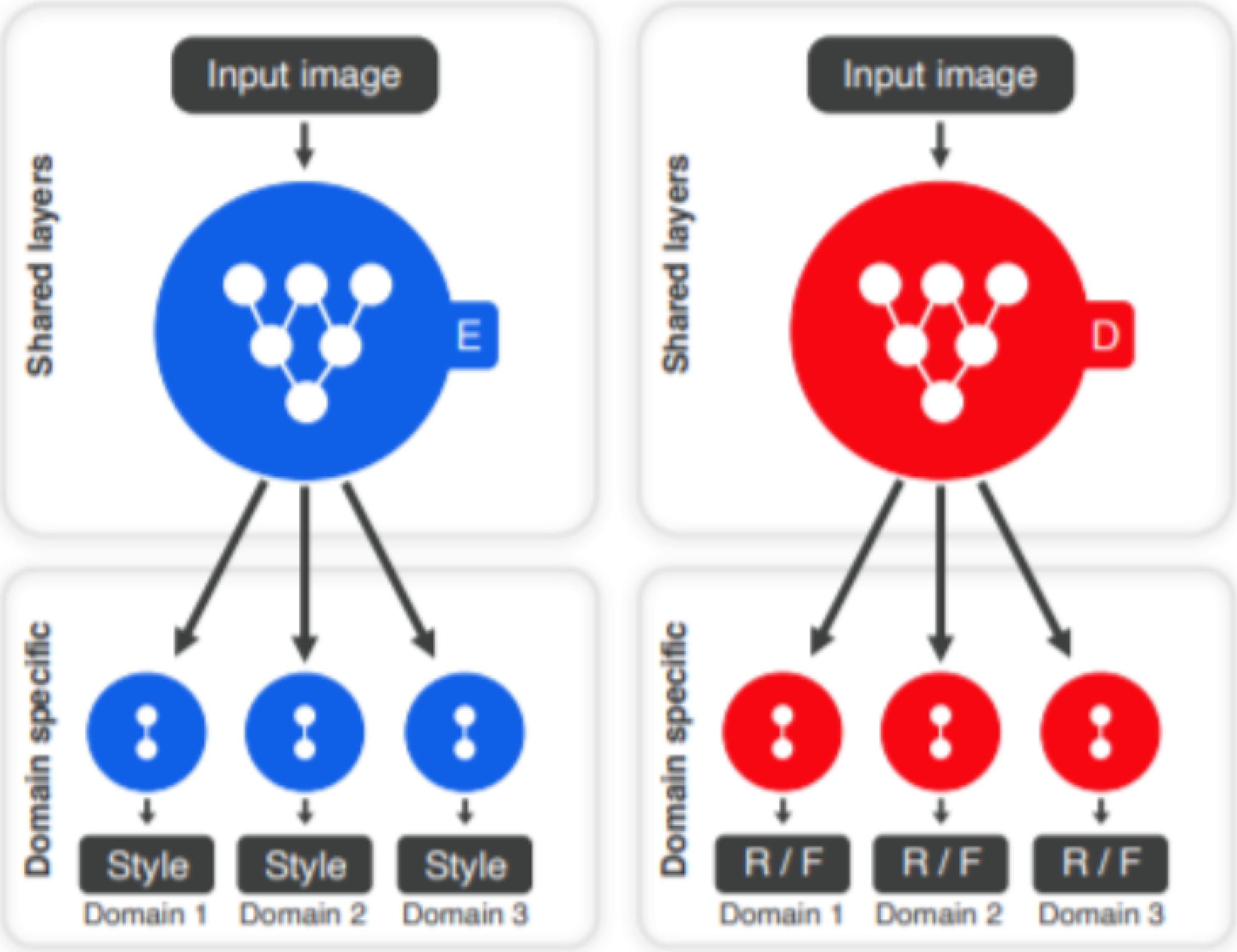
Network Training - Seperated Loss

- Directly modeling the space of high-dimensional convolutional filters in an image generation network is practic ally prohibitive in terms of both computation and parameter scale

Put together, our full objective function can be summarized as follows:

$$L(G, M, E, D) = L_{GAN} + \lambda_{cyc} L_{cyc} + \lambda_{sty} L_{sty} - \lambda_{div} L_{div}$$

Where λ_{cyc}, λ_{sty} and λ_{div} are hyperparameters for each term.



(c) Style encoder (d) Discriminator

02 Methods

Experimental Set Up - Dataset

- Healthy Control Domain Adaptation
  Original Datasets : ADNI, GSEV, SMC, GMC Dataset
  ADNI Control Normal: 4977 Patients , (256,256,192), GSEV Control Normal: 256 Patients , (256,256,192)
  SMC Control Normal: 2804 Patients, (256,256,192), GMC Control Normal : 96 Patients, (256,256,192)
- New Generated Datasets : Stack Channel, Random Selections at Patients Images, Total Train : 1200 MR Slices per protocol

ADNI Database
N = 4977
(256,256,192)

SMC Database
N = 2804
(256,256,192)

GSEV Database
N = 256
(256,256,192)

GMC Database
N = 96
(256,256,192)

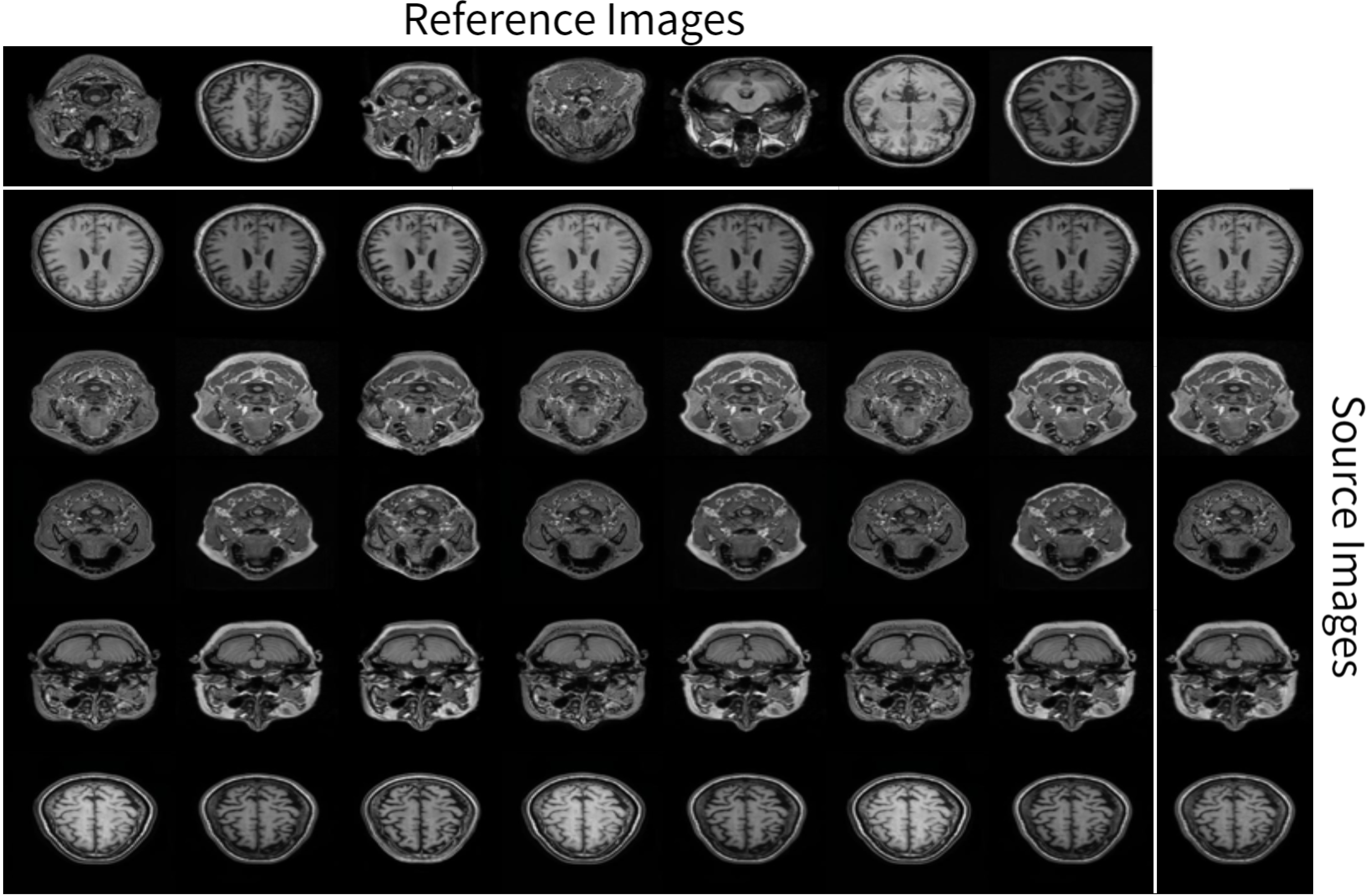
Random Selection

New Database
N = 1200 Slice each Protocols (ADNI, SMC, GSEV, GMC)
Extract MR Images each Patients

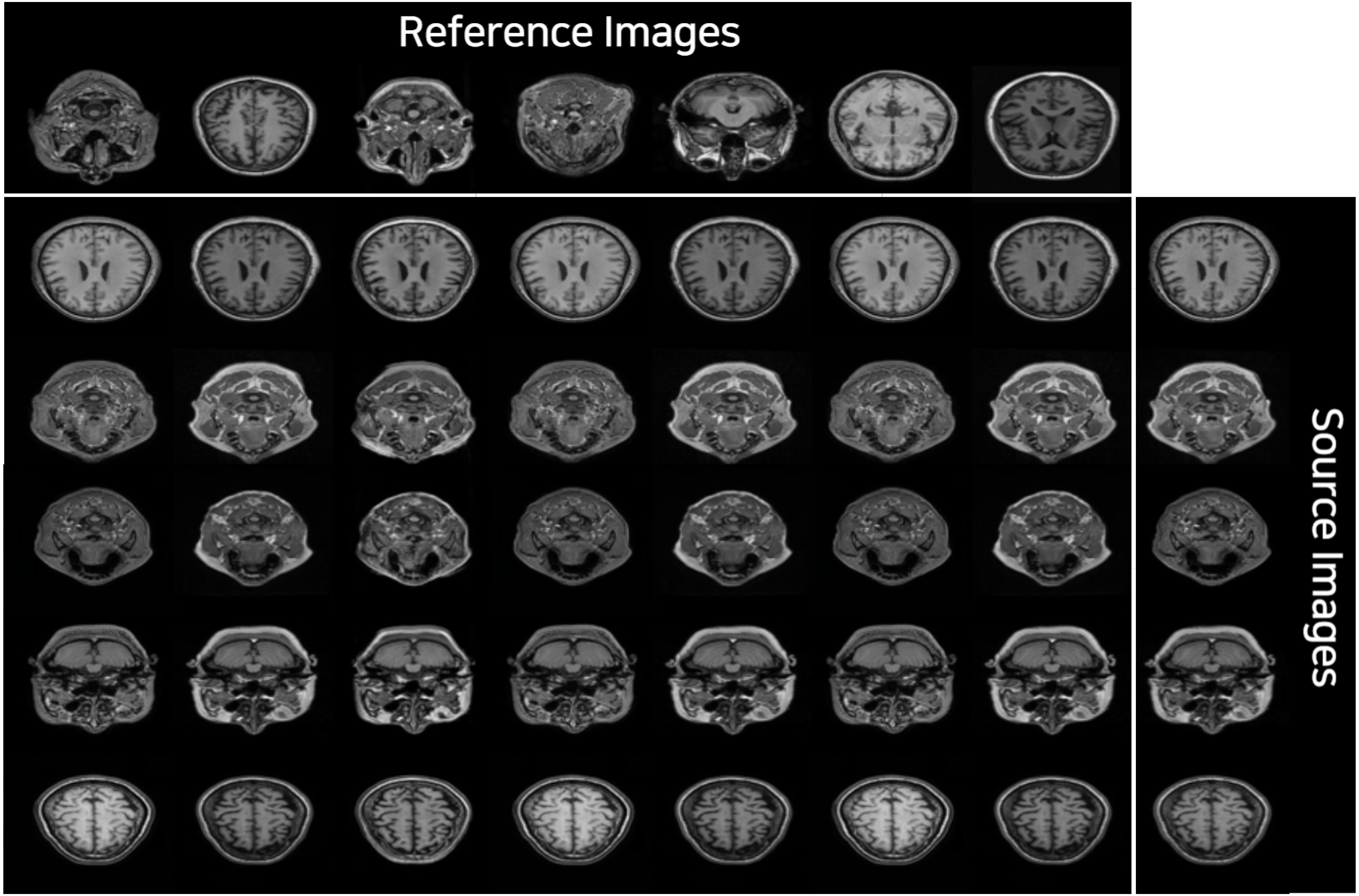
1. Z-Score Normalization
normalized by dividing by the maximum a fter clipping.

2. Data Augmentation
Random Image Selection per patients

3. Hyperparameter
Optimizer = Adam, Epoch = 16000, Los s Function = ADV Loss + Sty Rec Loss + Cycle Consistency Loss + Style Diver sification Loss

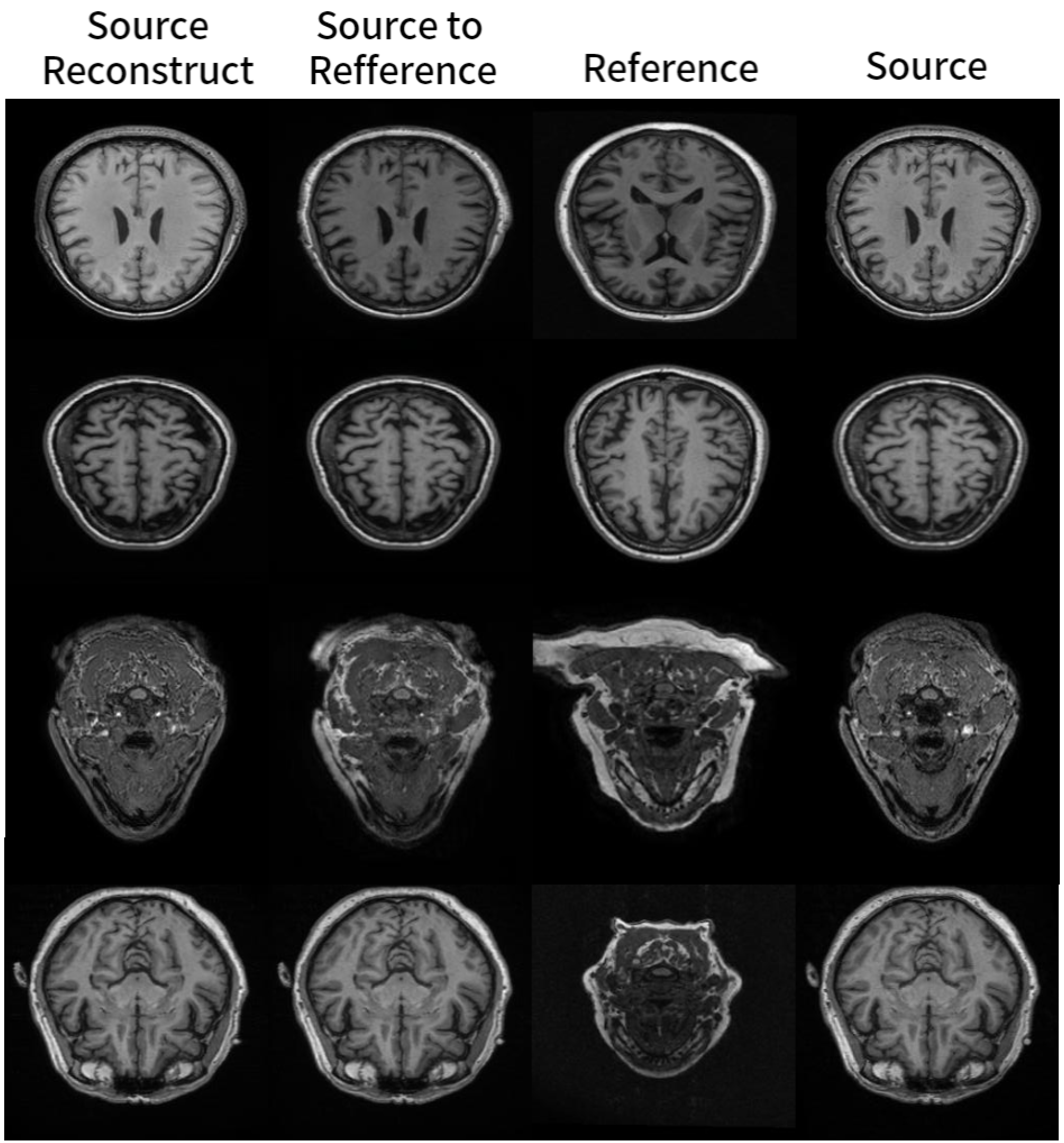


Reference Images
Source Images


Reference Images
Source Images

Experiments

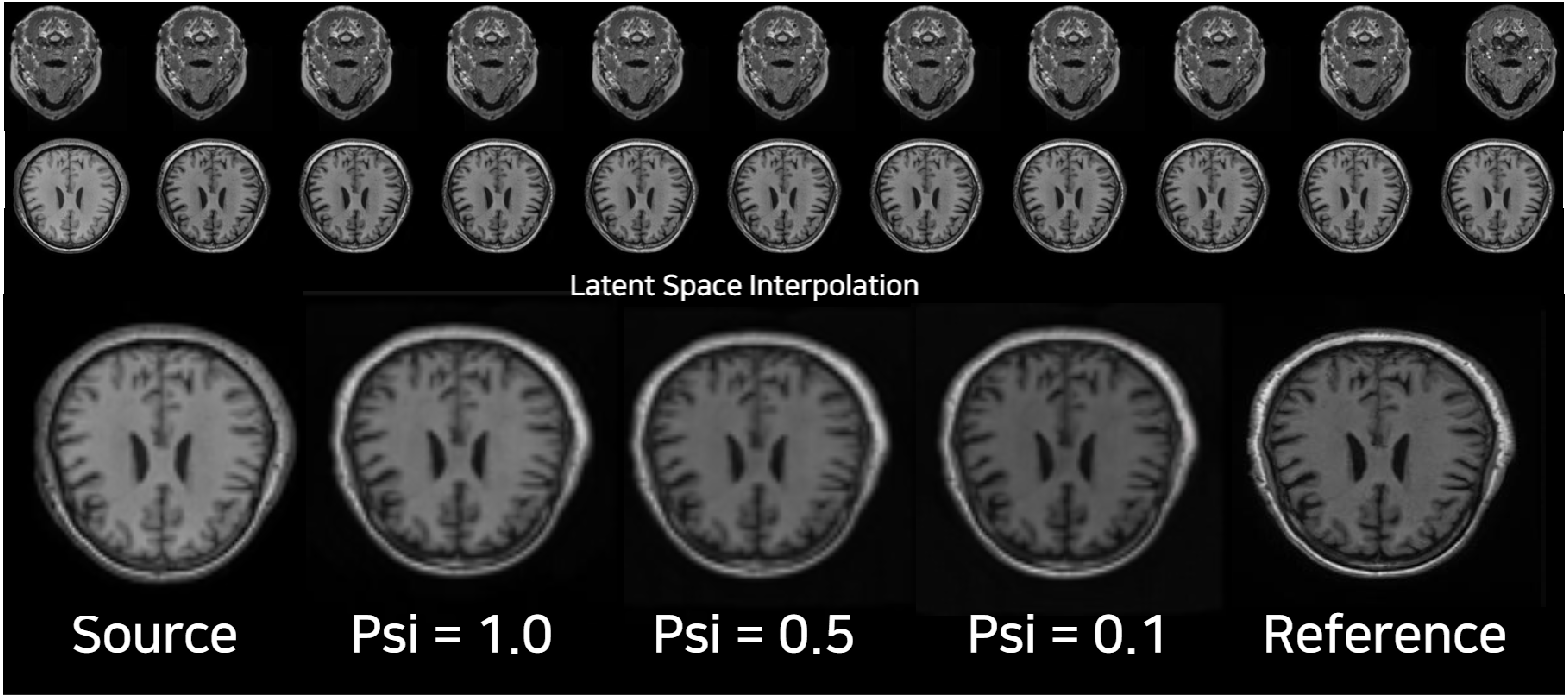
- Results - reference images with cycle consistency images



Experiments

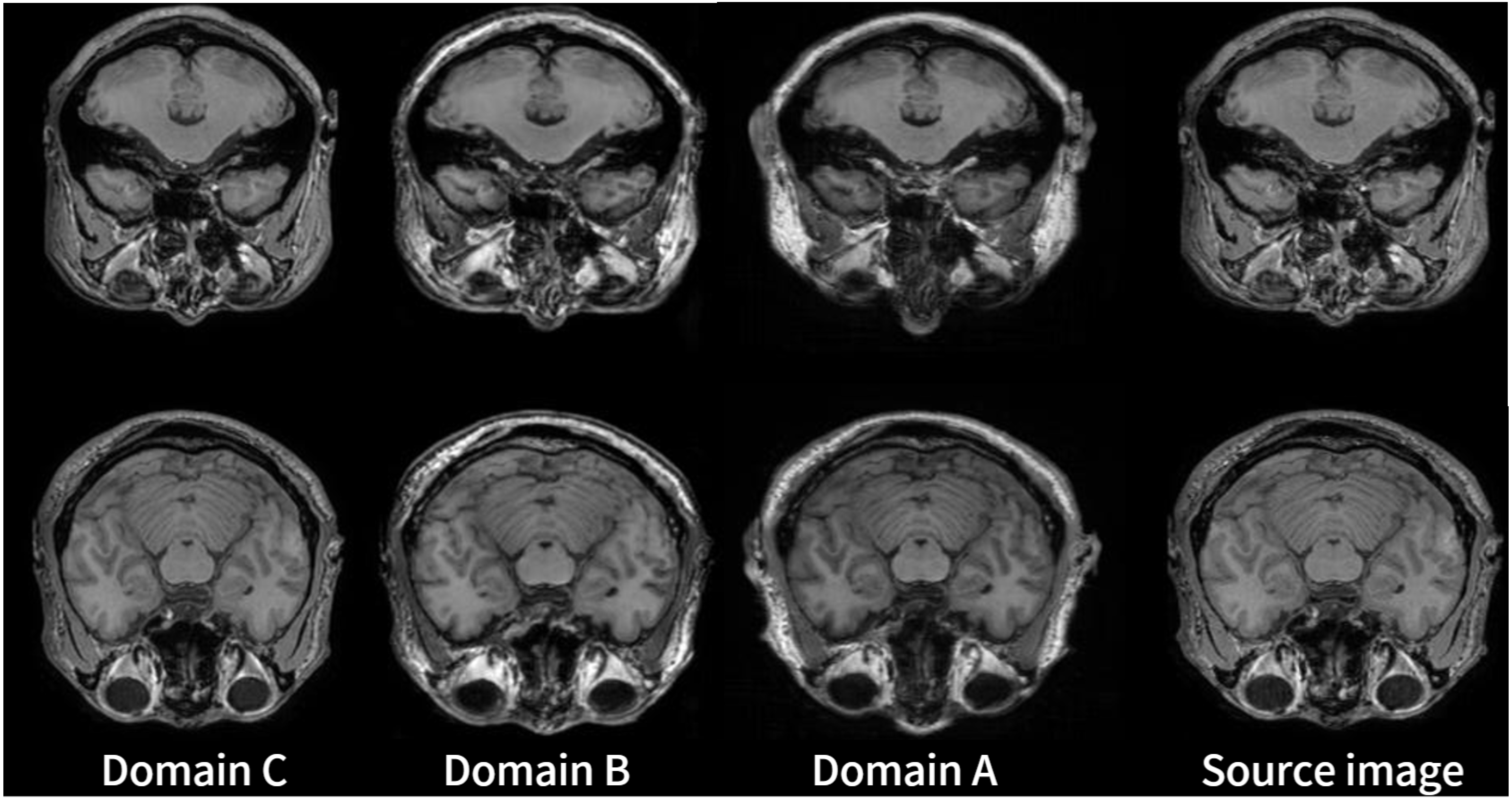
$$L(G, M, E, D) = L_{GAN} + \boxed{\lambda_{cyc}} L_{cyc} + \lambda_{sty} L_{sty} - \lambda_{div} L_{div}$$

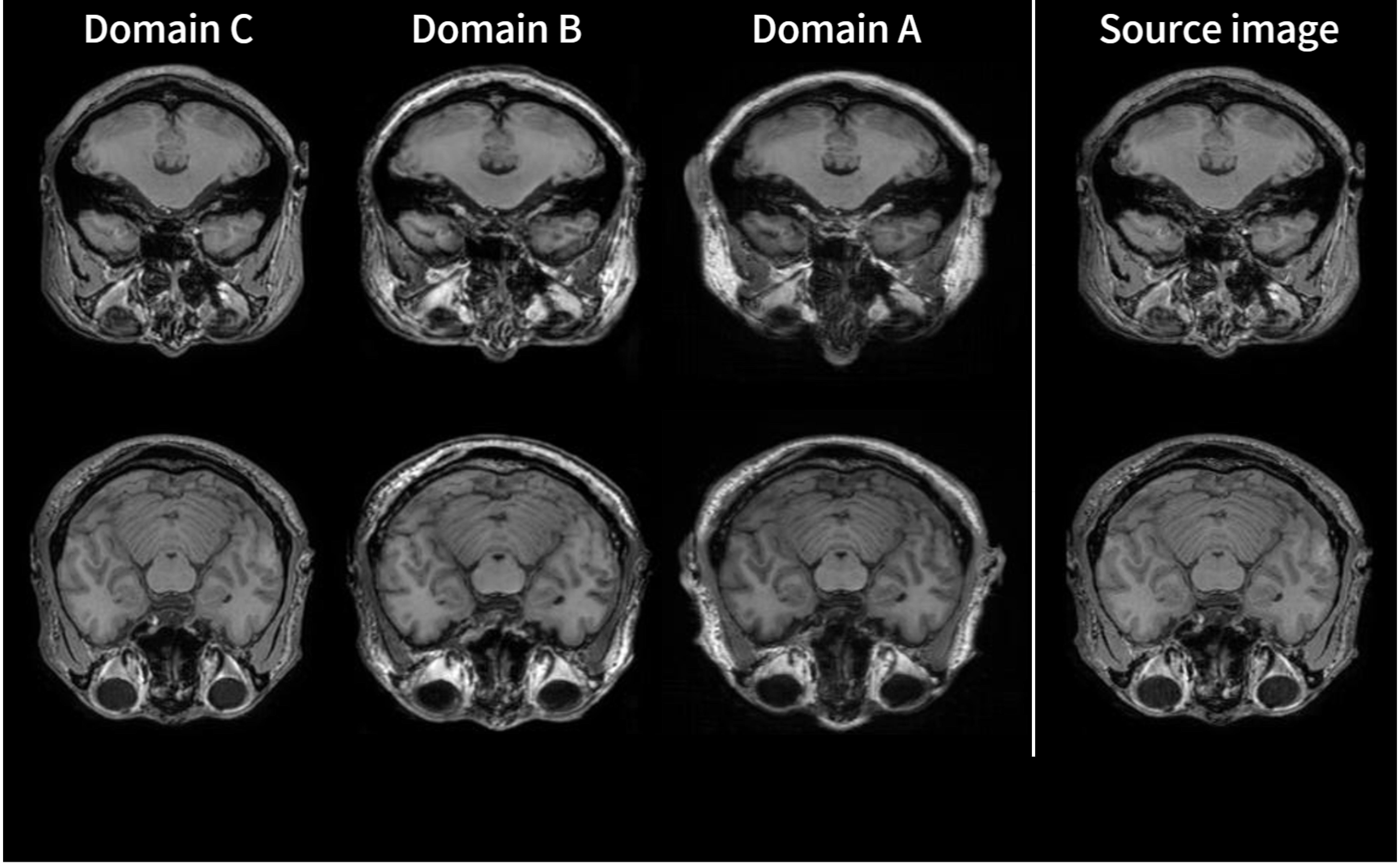
- Results - Cycle Consistency Coefficients (lambda cyc) difference

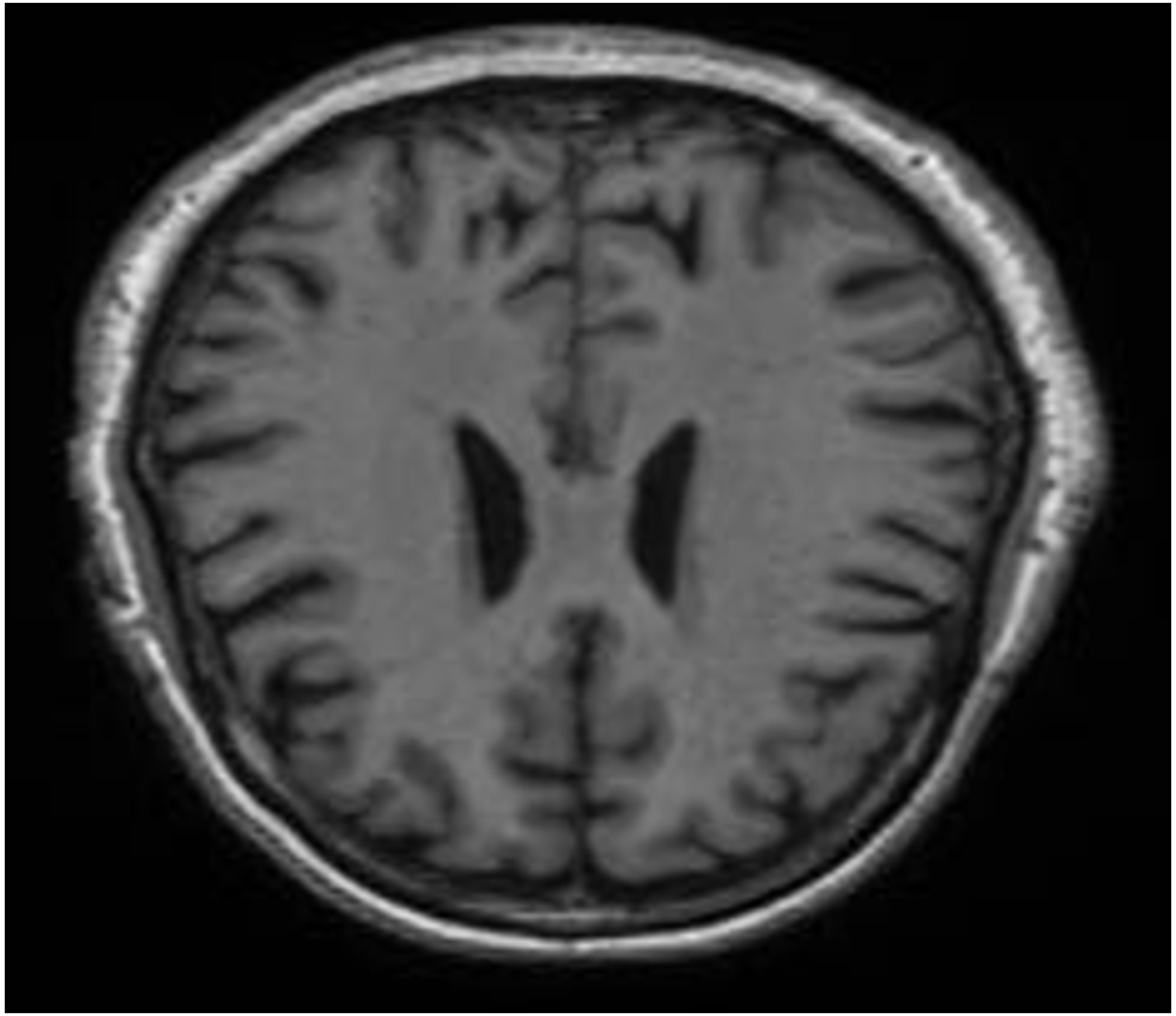
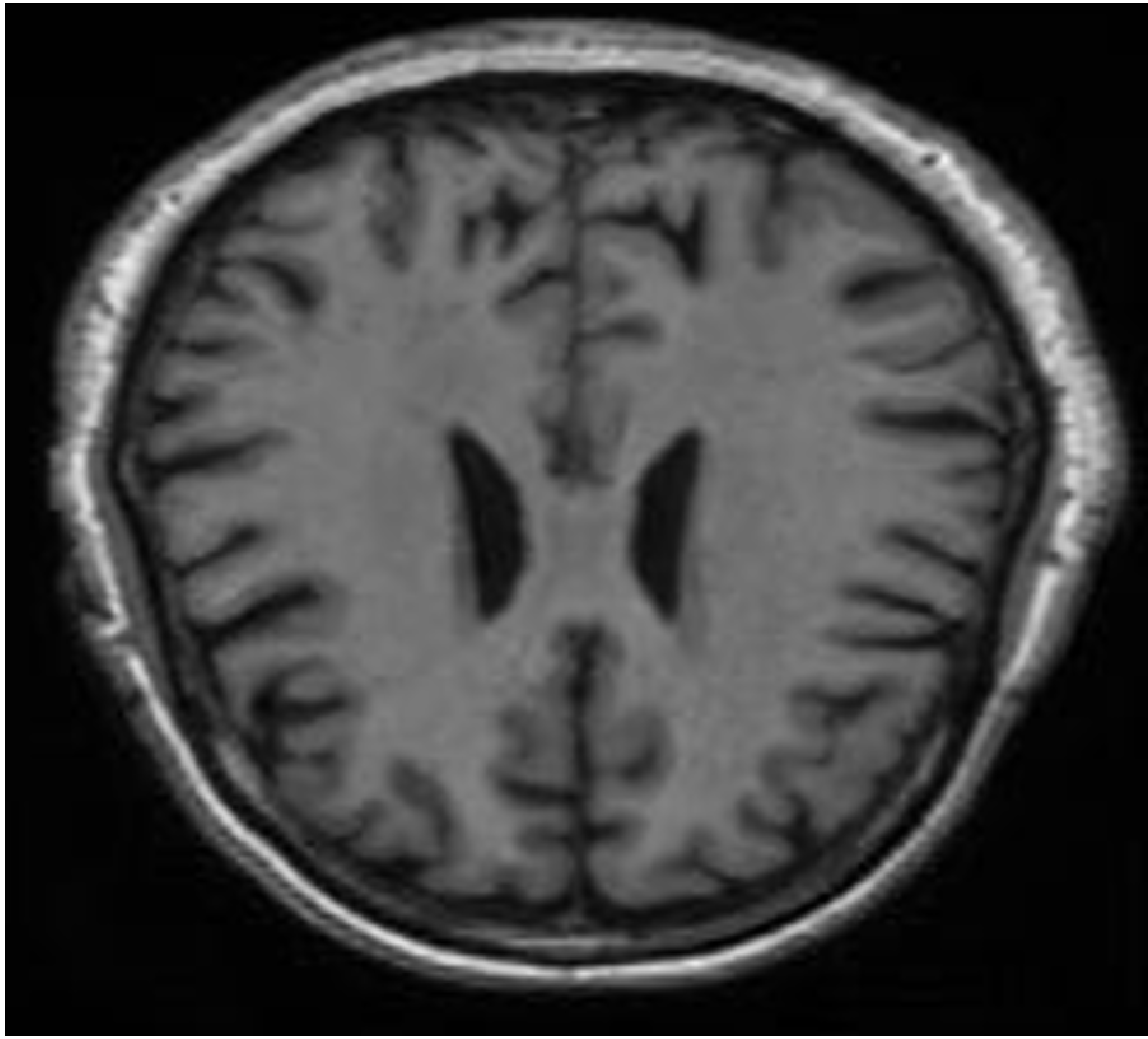


Experiments

- Results - Domain Differences

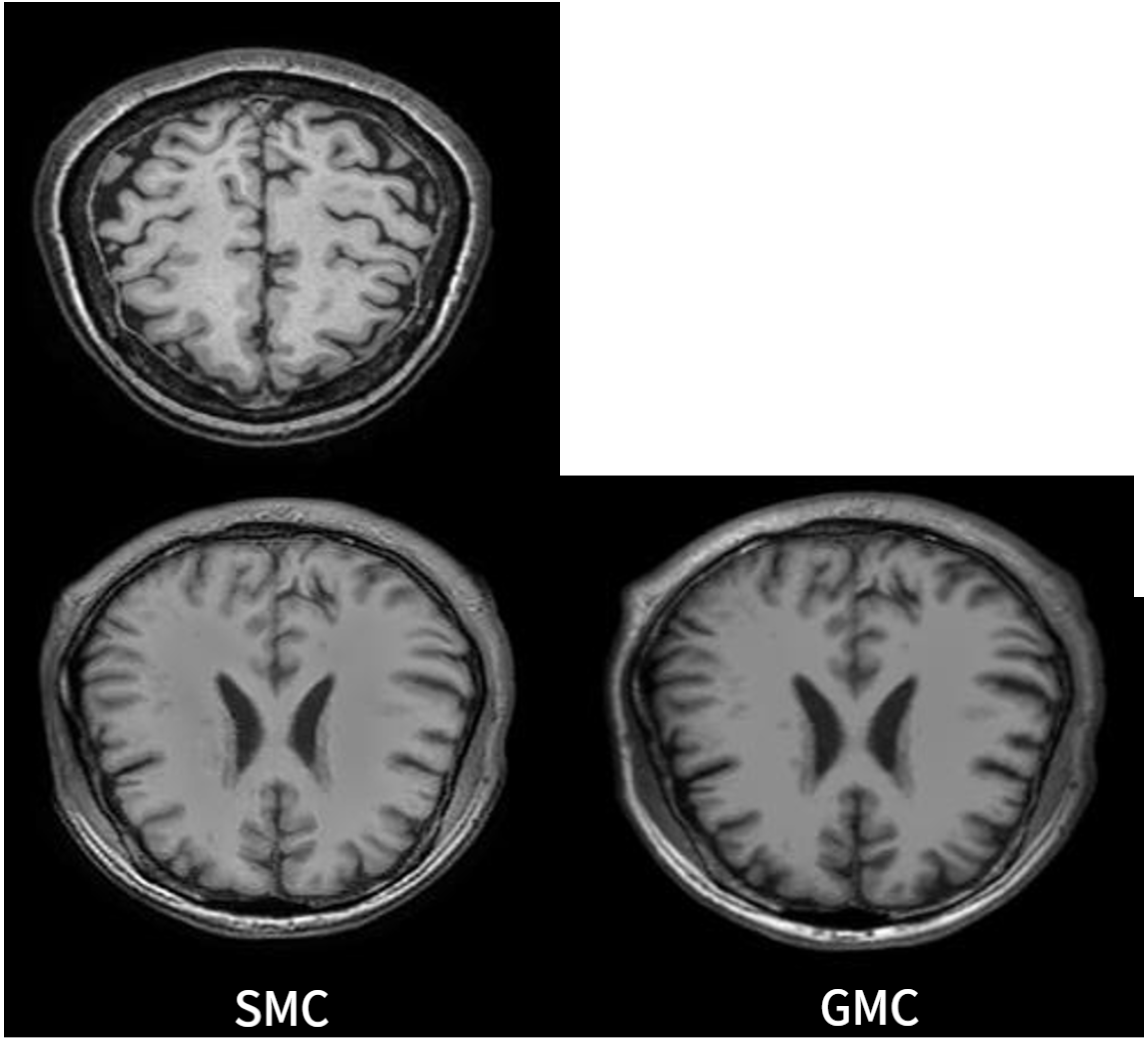



Domain C
Domain B
Domain A
Source image

Experiments

- Results - Domain Differences

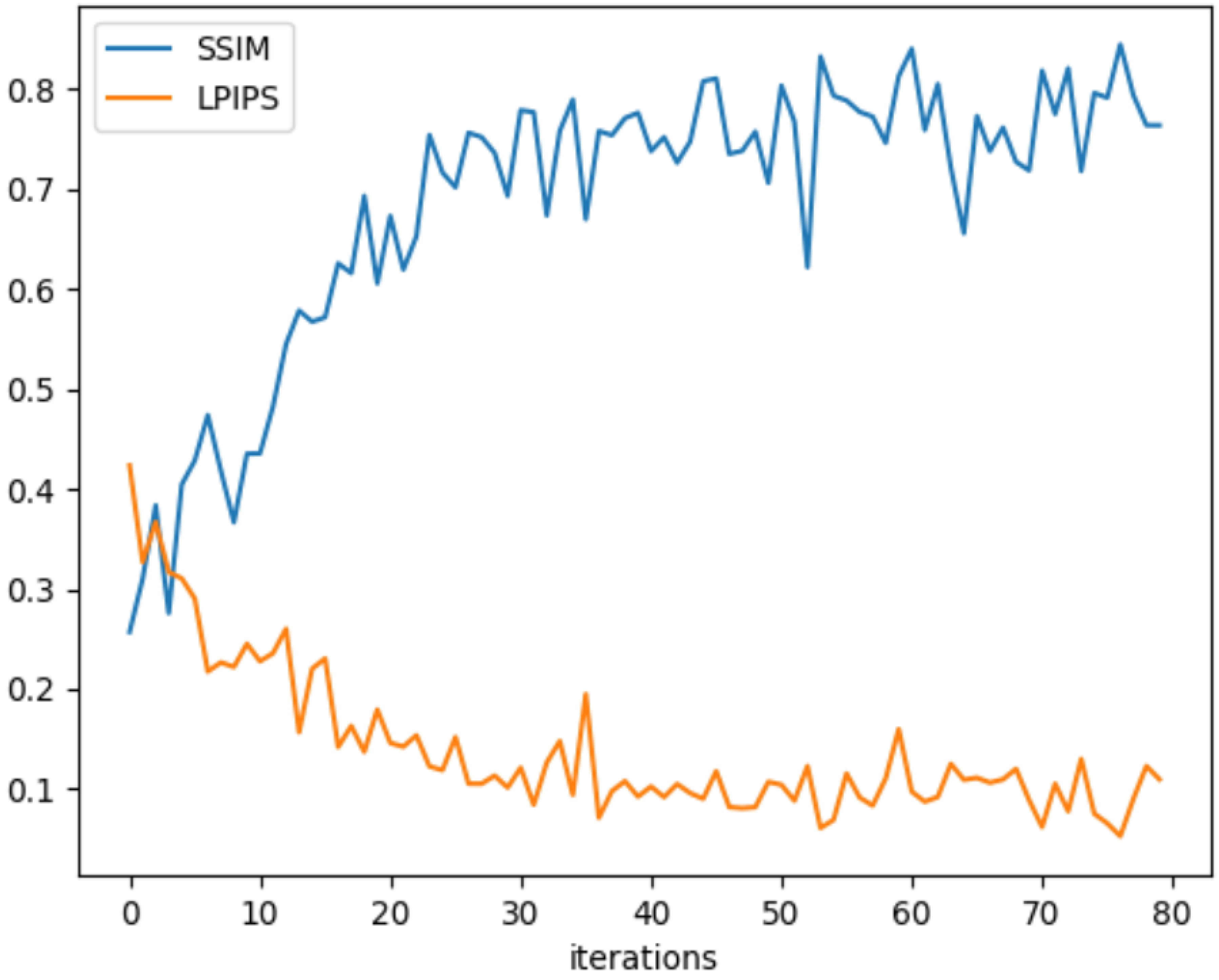


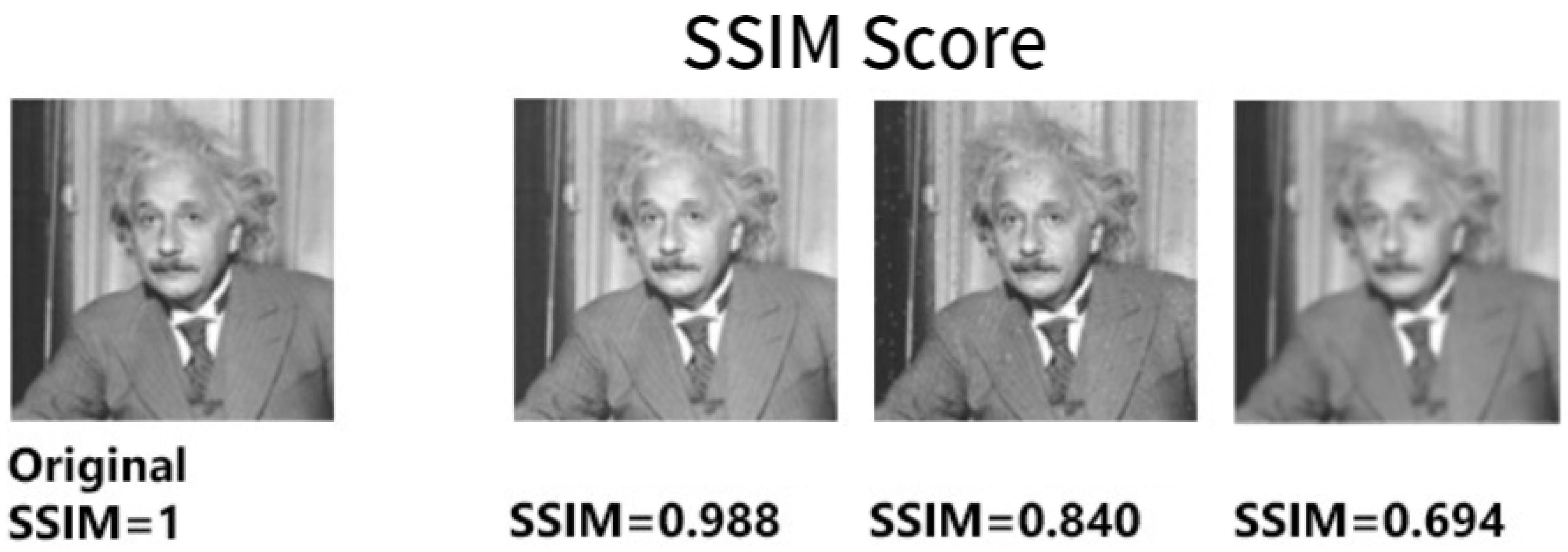
Experiments

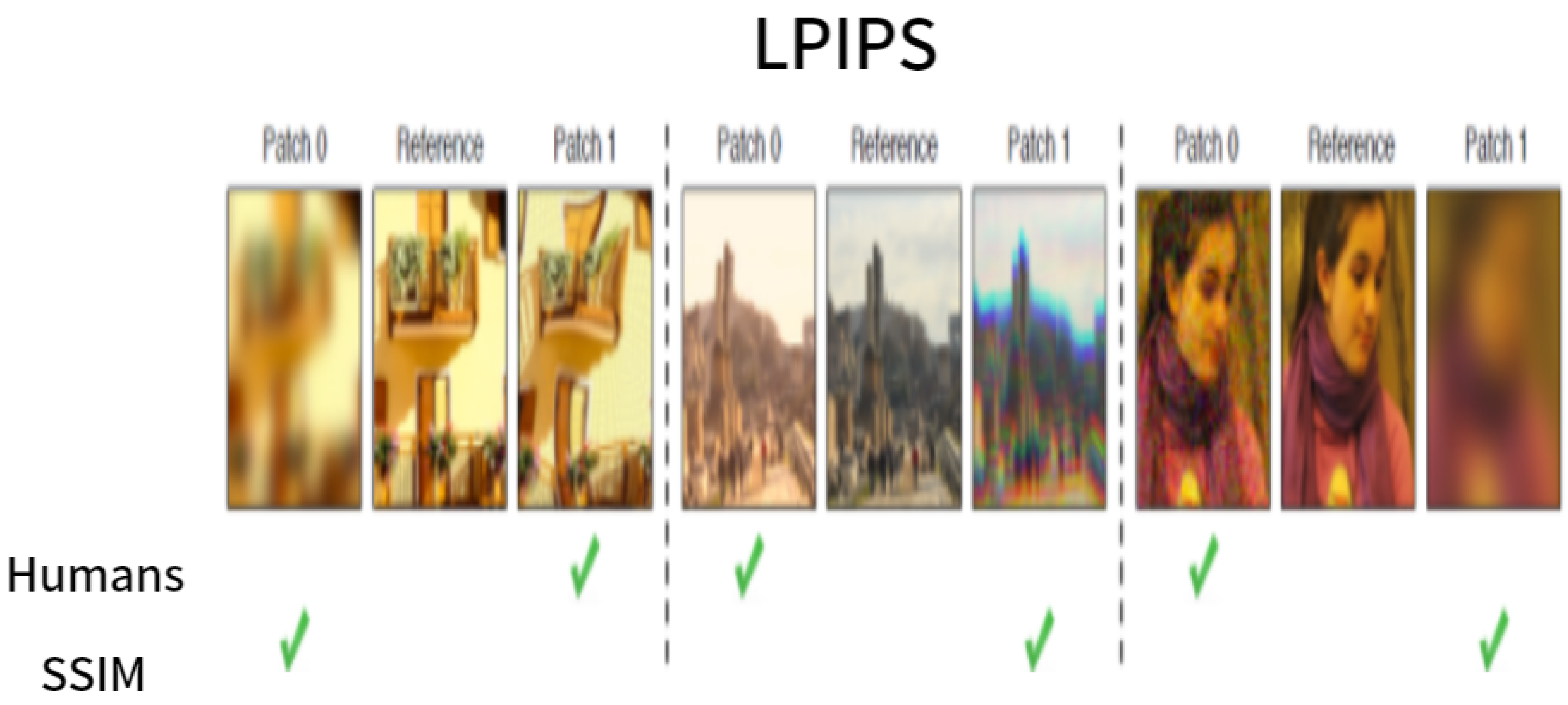
- Results - Unseen Dataset

03 Experiments

Domain Adaptation GAN - Evaluation Metrics







03 Experiments

Research Way

- 1. Multi by Multi Domain Adaptation Process
  Methods & Framework
  Four Protocols MRI Harmonization Experiments (ADNI, SMC, GMC, GSEV)
  Discussion

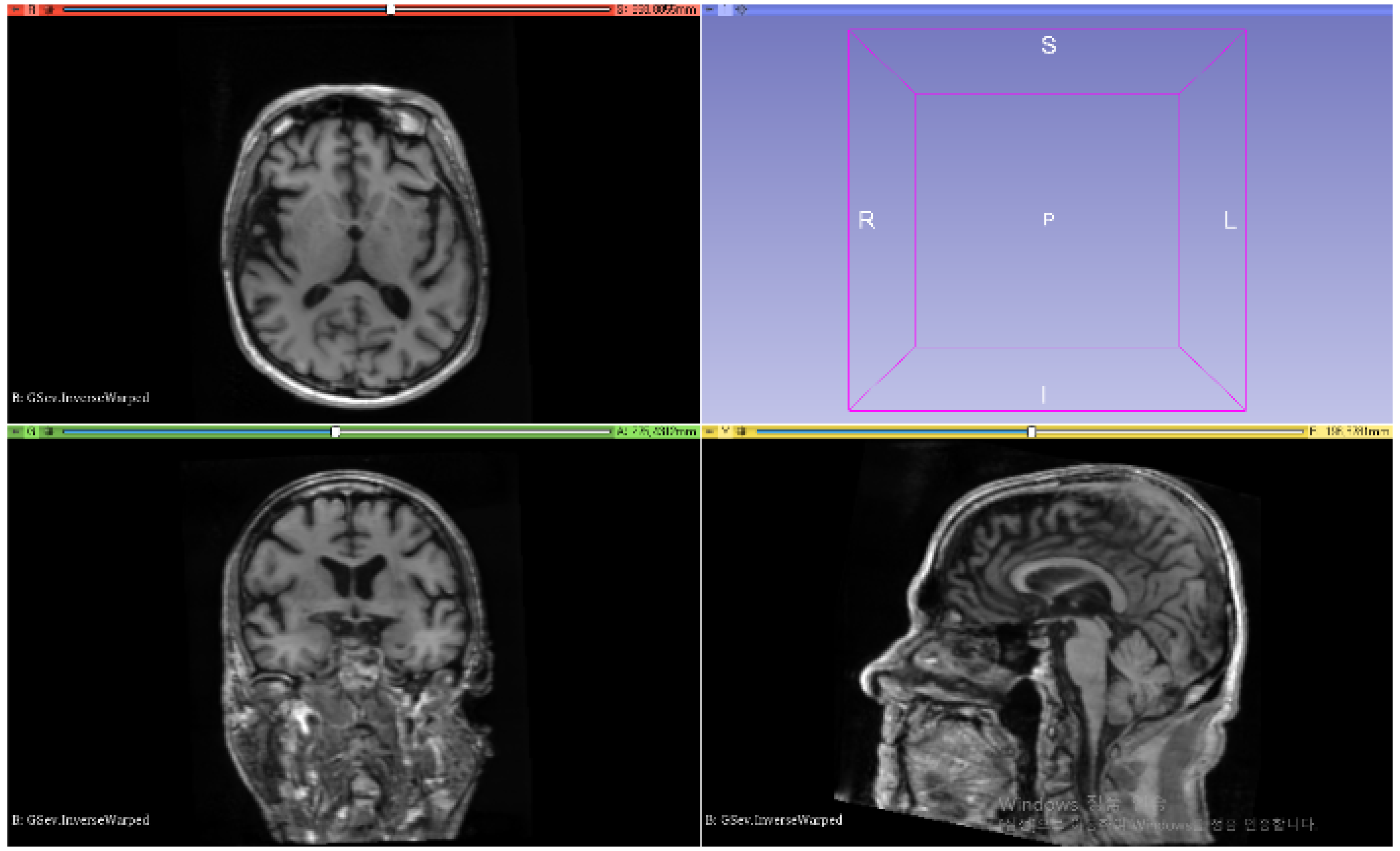
- 2. Validation Process
  2D MRI Synthesize & 3D MRI Reconstruction
  Fast Surfer Cortical Values (Thickness or Volume) Difference
  Alzheimer / Healthy Control Classification per protocols
  Discussion

- 3. Future Work

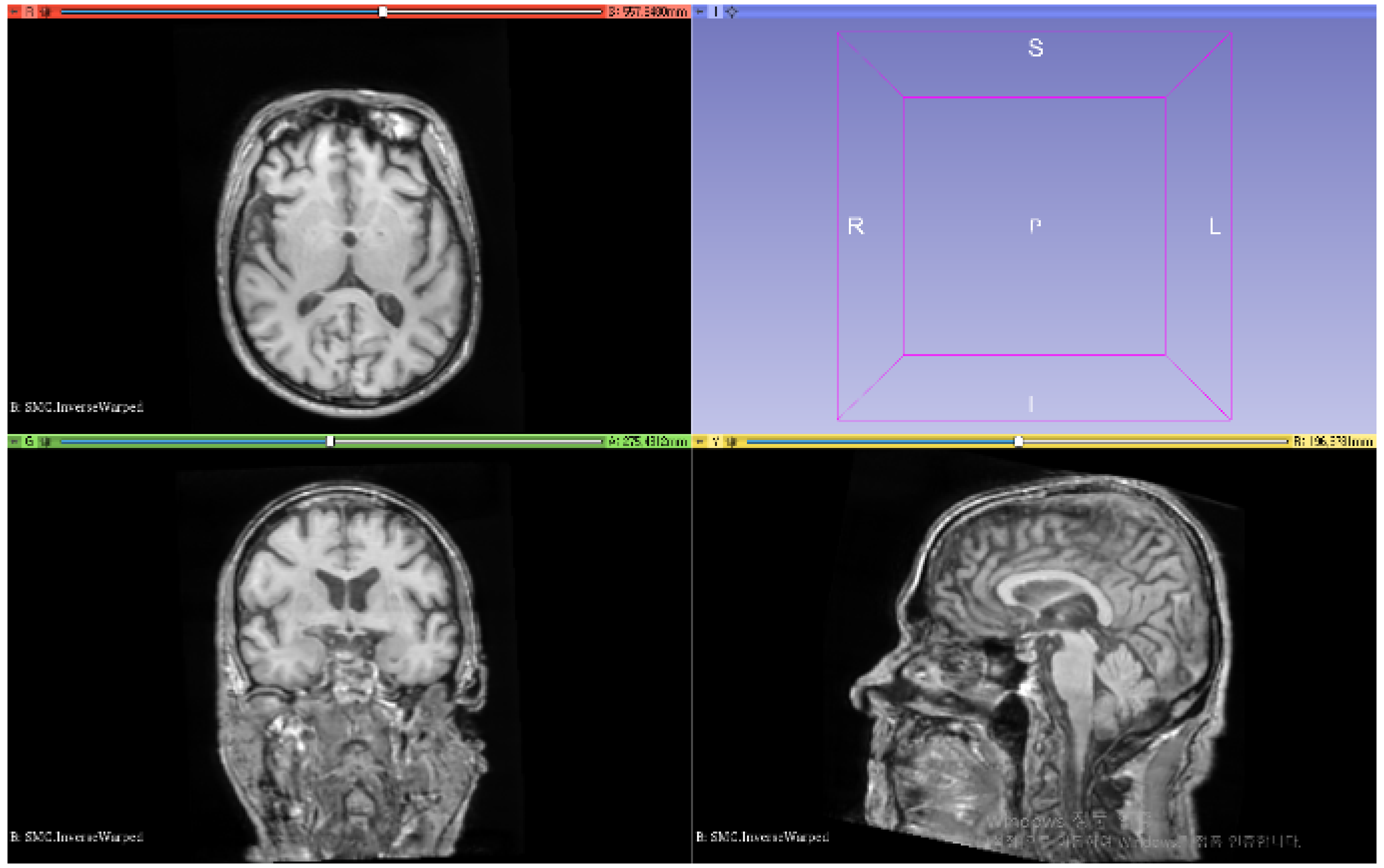
Experiments

- Results
  3D Reconstruction

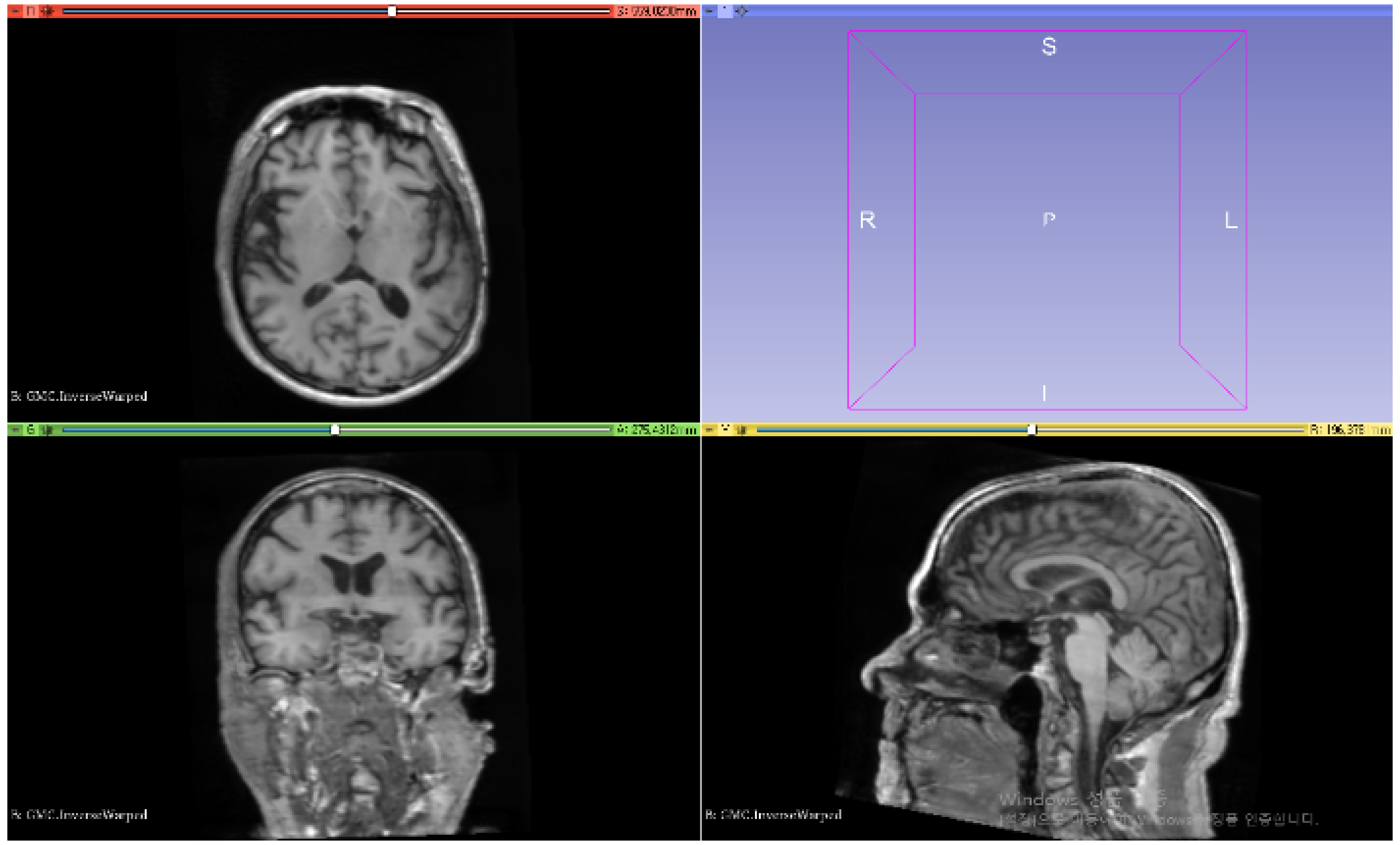
GSEV



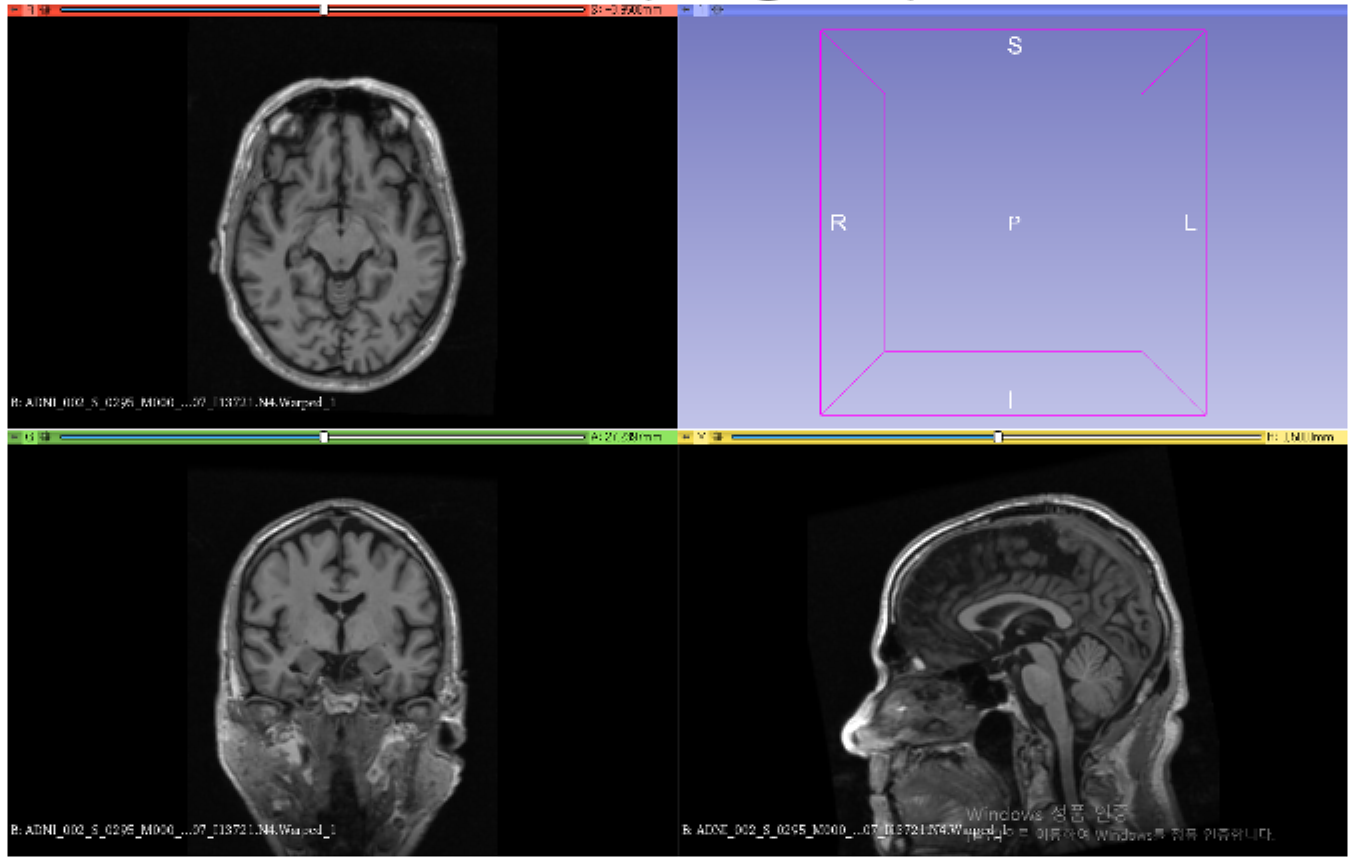
SMC



GMC

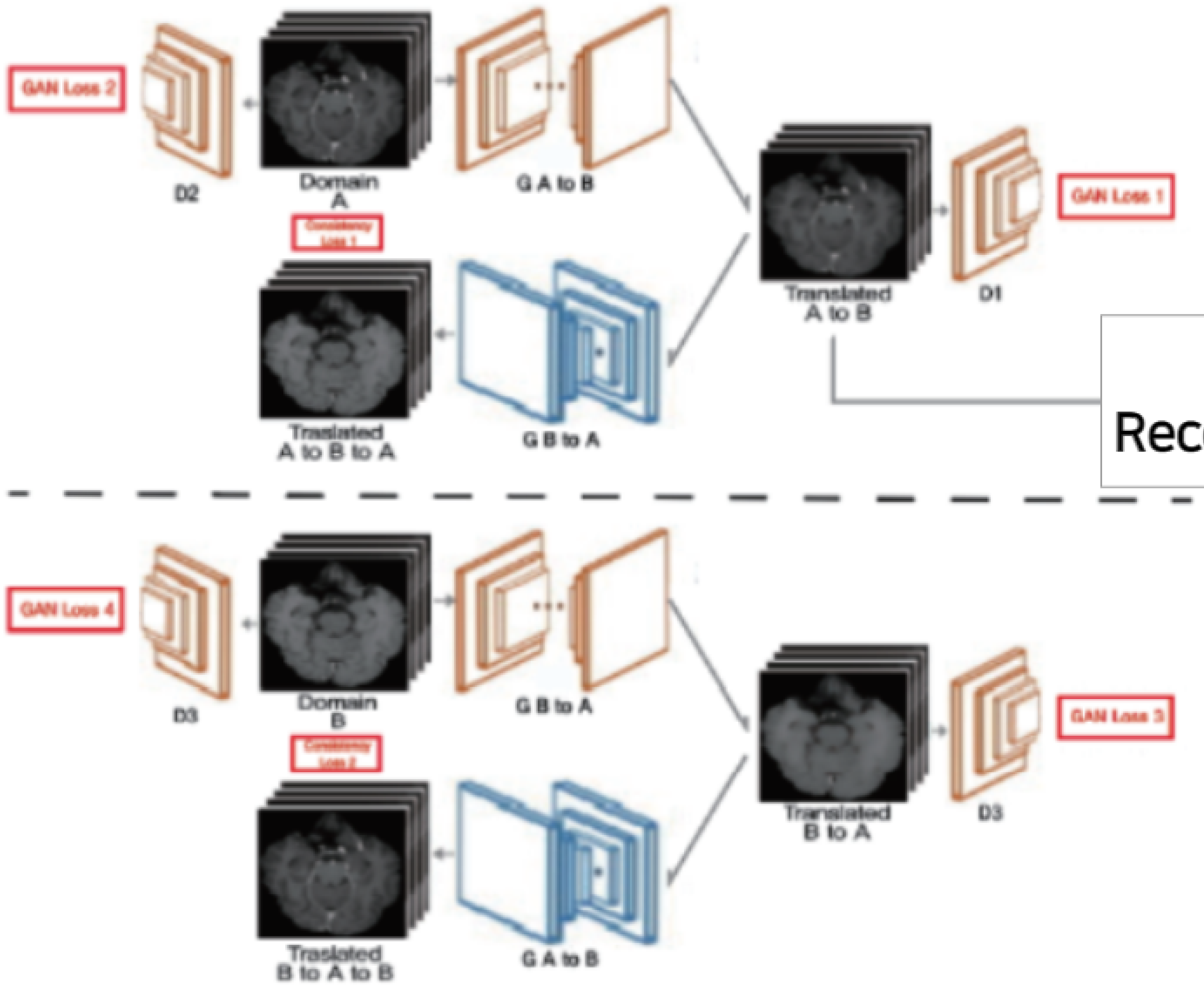


ADNI (Original)



03 Experiments

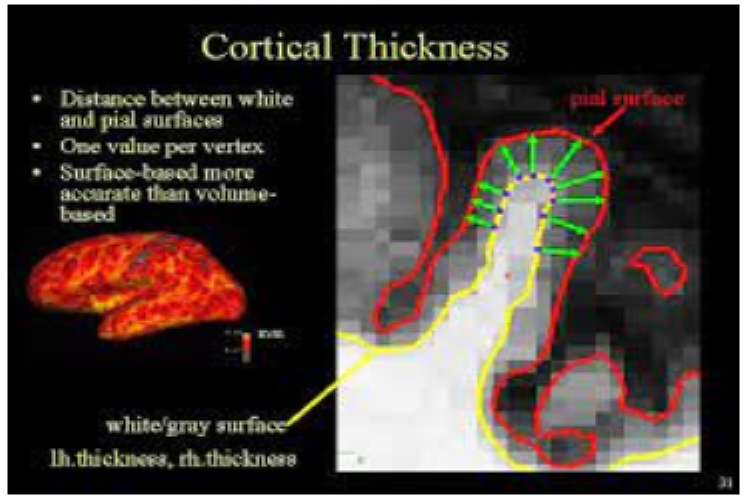
Validation Process



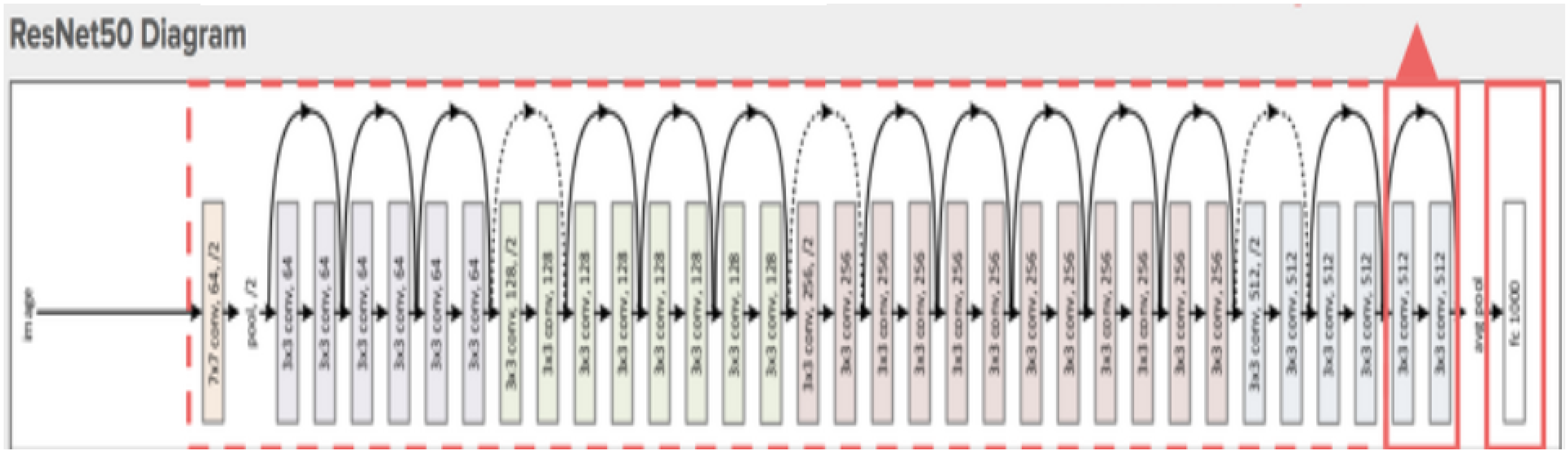
Institutions Harmonization

3D Reconstruction

Evaluation 1 :
Cortical Thickness Maintainance



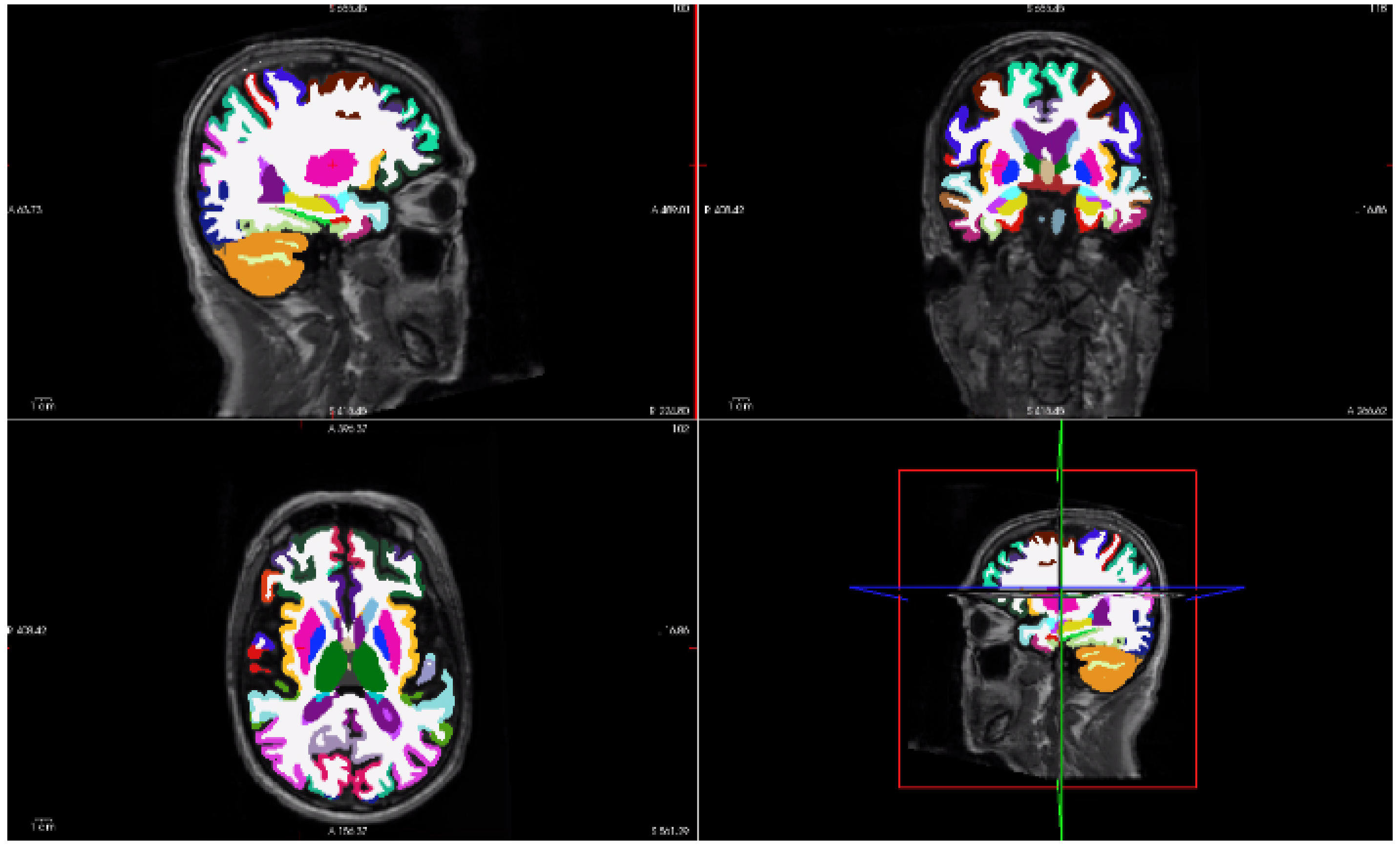
Evaluation 2 : Protocols Classification



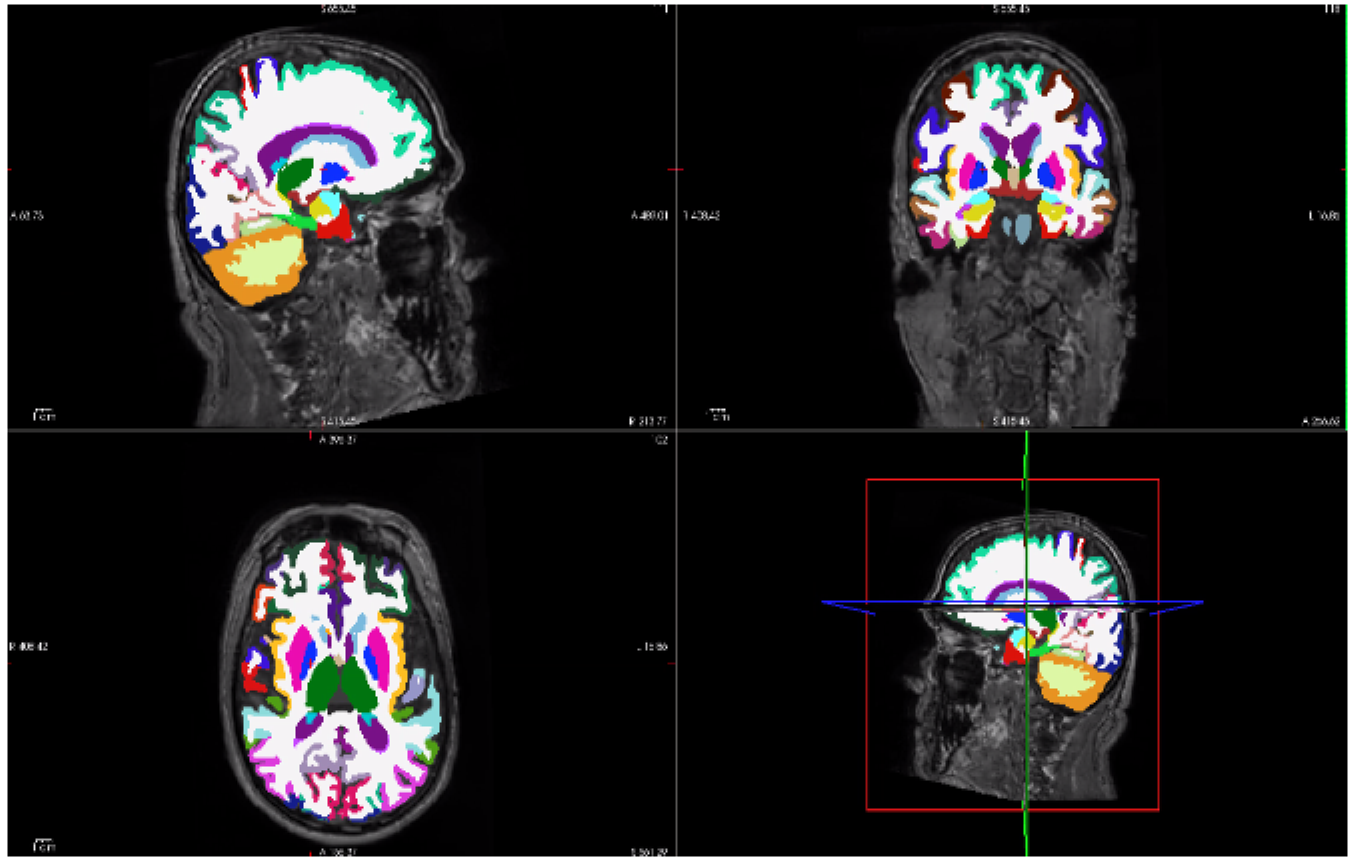
Experiments

- Cortical Thickness Maintainance (Fast Surfer)

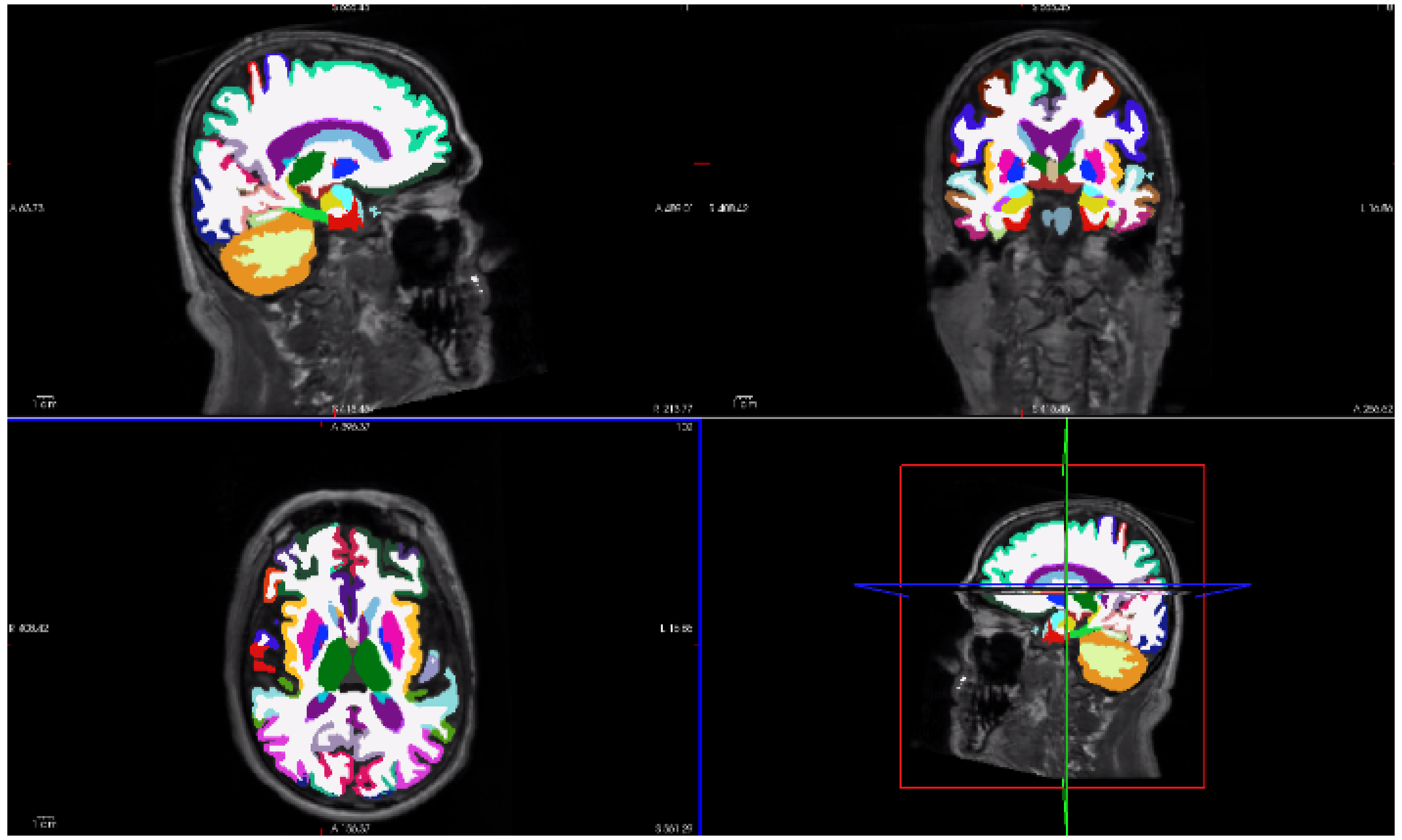
GSEV



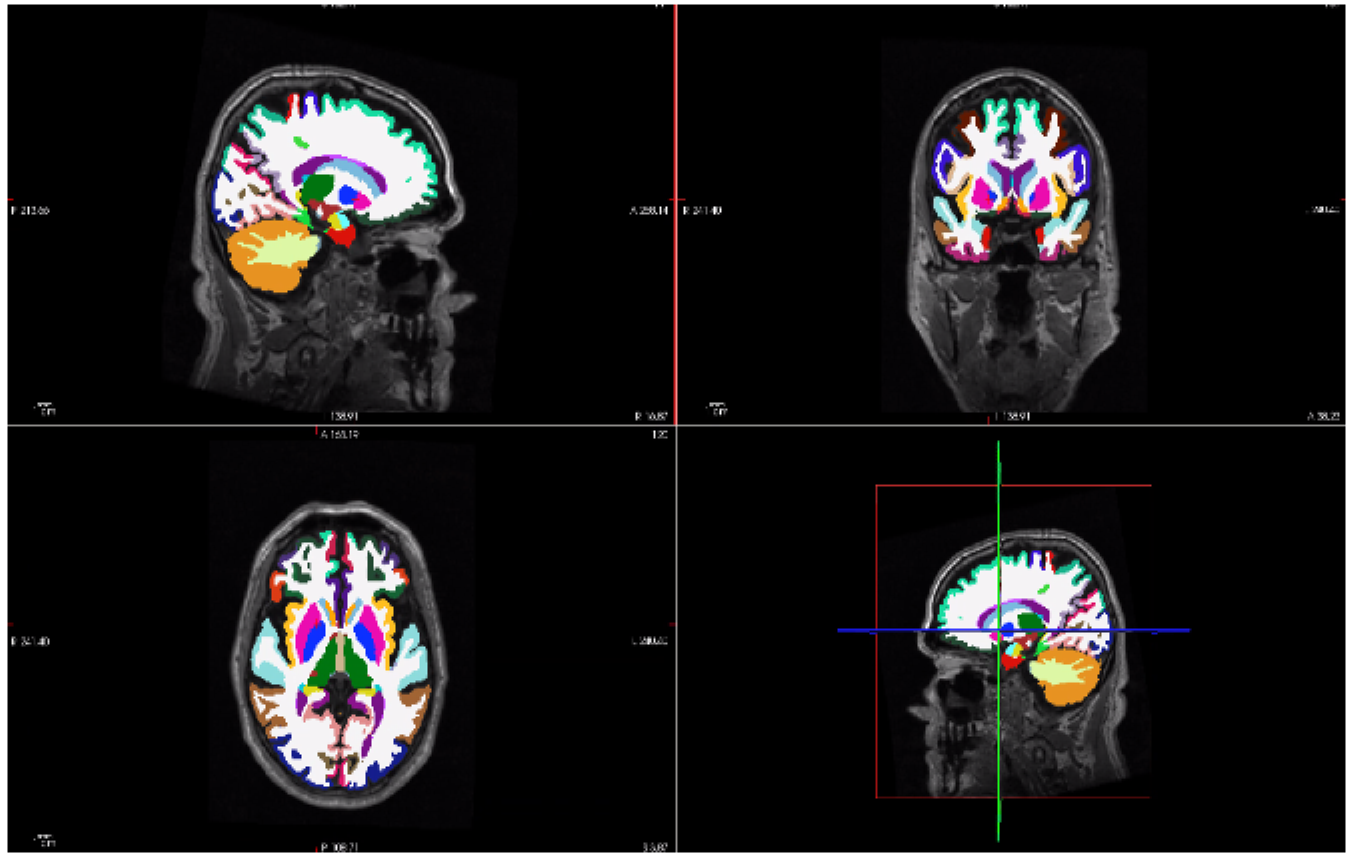
SMC



GMC



ADNI (Original)



Experiments

- Classification Output

```
tensor(8.3182, device='cuda:0', grad_fn=<NllLossBackward>)
 87%|                                        | 999/1152 [00:57<00:08, 17.56it/s]
tensor(0.0492, device='cuda:0', grad_fn=<NllLossBackward>)
100%|                                        | 1152/1152 [01:06<00:00, 17.45it/s]
Train ACC : 85.50347137451172
100%|                                        | 288/288 [00:03<00:00, 90.81it/s]
Valid ACC : 84.72222137451172
  0%|                                        | 0/1152 [00:00<?, ?it/s]
tensor(0.0130, device='cuda:0', grad_fn=<NllLossBackward>)
 87%|                                        | 999/1152 [00:55<00:08, 18.34it/s]
tensor(2.0882, device='cuda:0', grad_fn=<NllLossBackward>)
100%|                                        | 1152/1152 [01:03<00:00, 18.00it/s]
Train ACC : 89.72222137451172
100%|                                        | 288/288 [00:03<00:00, 88.62it/s]
Valid ACC : 86.80555725097656
  0%|                                        | 0/1152 [00:00<?, ?it/s]
tensor(0.0122, device='cuda:0', grad_fn=<NllLossBackward>)
 87%|                                        | 999/1152 [00:56<00:08, 17.73it/s]
tensor(0.0562, device='cuda:0', grad_fn=<NllLossBackward>)
100%|                                        | 1152/1152 [01:04<00:00, 17.81it/s]
Train ACC : 92.6388931274414
100%|                                        | 288/288 [00:03<00:00, 86.54it/s]
Valid ACC : 92.36111450195312
  0%|                                        | 0/1152 [00:00<?, ?it/s]
```

	Before	After
ADNI, GMC =	96.52996826171875	66.66666412353516
ADNI, SMC =	91.2000732421875	64.3533172607421 9
ADNI, GSEV =	94.69696807861328	69.08517456054688
GSEV, GMC =	83.4595947265625	61.51419830322656
SMC, GSEV =	88.19445037841797	55.81597137451172
SMC, GMC =	87.84722137451172	57.89141464233398 4