温州肯恩大学
WENZHOU-KEAN UNIVERSITY

Domain Robust Medical Image Segmentation

In Partial Fulfillment of the Requirements

for the **Bachelor** of Degree in Applied Mathematics

by

Changjie Lu

1129503

May, 2022

Contents

Acknowledgement

During my past three years in the department of mathematics, Wenzhou-Kean University, I gained considerable growth both in life and academics.

First of all, I would like to thank Liwei Wang, my encyclopedic and unassertive roommate during my sophomore year. Liwei taught me nearly all mathematics problems and untied my confusion within a few words. Every night before I went to bed listening to what you found in the wormhole, it took my mind off the temporal lives. Liwei always told me that learning needs to sit on the cold bench with perseverance. Meanwhile, thank you for your introduction to senior student Shen Zheng who took me into the beautiful journey of the computer vision area, changing and shaping my future. I never forget that we work on one paper submission deadline until midnight and achieve several state-of-the-art in the low-level vision field. I will keep on going in computer vision research in the future. I hope everything goes well in your graduate life.

Then, I have to thank my advisor, Prof. Gaurav Gupta. You provide me even all kinds of resources in the math department, including GPU, the opportunity to reach other professors, and life guidance. I felt delighted during my college work!

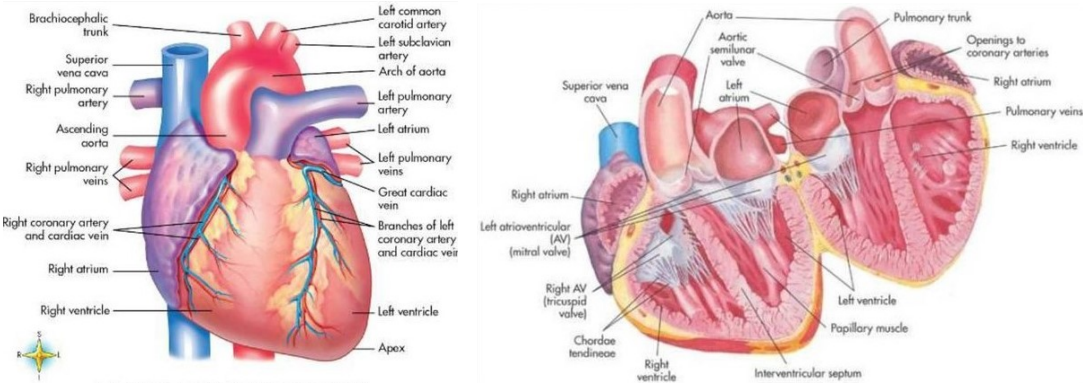
Last but not least, I would like to thank my family, who gave me unlimited financial aid and spiritual support for my research. Without their encouragement and help, I could not finish my bachelor thesis. I would also thank myself for persistence, endeavoring alone in the unknown area most of my college time.

Chapter 1

Introduction

Medicine is not only a science. It is also an art.

Paracelsus



(a) Cardiac Appearance (b) Anatomical Cardiogram

Figure 1.1: The images of Cardiac Appearance, Anatomical Cardiogram.[1] In segmentation part, we mainly focus on left atrial, right atrial, left ventricle and right ventricle.

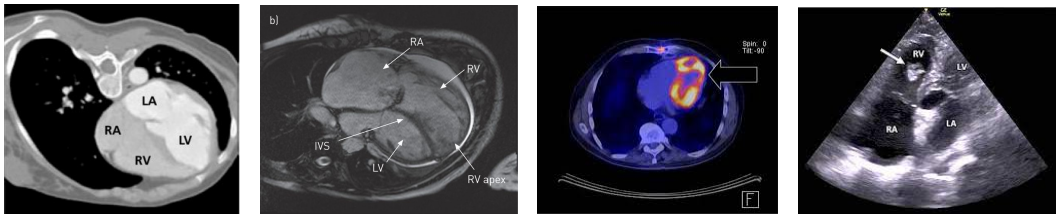
1.1 Motivation

1.1.1 About Cardiac

As one of the essential organs, the cardiac maintains normal metabolism and function for every organ and tissue. The leading cause of death globally, according to World Health Organization (WHO), is cardiovascular disease (CVD). In 2016, more than 17 million people reported CVD deaths, and heart disease was the primary reason. [2] The common cardiac diseases include arrhythmia, premature beat, atrial fibrillation, ventricular fibrillation, etc. People could get these diseases due to congenital cardiovascular defects in young children, sudden cardiac arrest after overworking in adults, or atherosclerosis in older people. [3] Heart disease is usually accompanied by organic abnormalities. Therefore, pinpointing abnormalities in the heart is a crucial step in starting the treatment process.

1.1.2 Why Deep Medical Image Segmentation

As mentioned above, the critical part before the treatment process is pinpointing. But how can we know what happens in the invisible area of the body? Fortunately, since Wilhelm Conrad Roentgen discovered X-ray radiation in 1895, medi-



(a) CT (b) MRI (c) PET (d) US

Figure 1.2: Four types of medical imaging of cardiac. From left to right: computerized tomography, magnetic resonance Imaging, positron emission tomography, and ultrasound

cal imaging has grown into a vast scientific discipline. The typical way of diagnosis is to analyze the patient's organ medical imaging, including the modalities of computerized tomography (CT), magnetic resonance tomography (MRT), positron emission tomography (PET), or ultrasound (US).[4] With this imaging, doctors can develop a detailed diagnostic assessment and clinical plan.

Determining what areas are the focused parts is called image segmentation. Whole heart segmentation targets extracting the substructures of the heart, including the four-chamber blood cavities of the left ventricular myocardium, [5] including left atrium (LA), left ventriculus (LV), right atrium (RA), right ventriculus (RV), and myocardium (Myo). These tasks are both skillful experience needed and time-consuming. Researchers make much effort both at the machine and algorithm sides to analyze the check-up. Bloomgarden et al. introduced an analysis strategy in rotating plane long-axis orientations to define the valve planes and the apex better while reducing the number of slices. [6] Gupta et al. built a deformable model for the segmentation of ventricular boundaries in cardiac MR images.[7] Singleton et al. used edge detection by tissue classification in pixel neighborhoods for MRI. [8] Zhuang et al. proposed a registration-based propagation framework, including the locally affine registration method (LARM) and the free-form deformations with adaptive control point status. [9] The other standard methods are mentioned in [10, 11, 12].

However, the data problems such as limitation, low quality, and cross-modality are

too intractable for the traditional way to deal with. Finding the complex mapping function for reconstruction, repair, or register in an analytical form is not generalizable. Noting that a deep neural network is an innovative tool to estimate the real function, combined with the fast development of hardware, the deep learning approach has achieved promising results in many vision areas such as classification, segmentation, and reconstruction. Deep learning tools also show their impressive performance on medical images. Due to the flexibility of deep learning, it does well in dealing with the difficulty of cross-modality, low definition, or data limitation issues. Once we find some mapping challenging to define analytically, we can approximate them by optimizing the neural network. Nowadays, we can discover neural networks in almost every area, including unsupervised, semi-supervised, unsupervised, and zero/few-shot learning. We will discuss more in the chapter on Background.

Chapter 2

Background

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

Brook Taylor

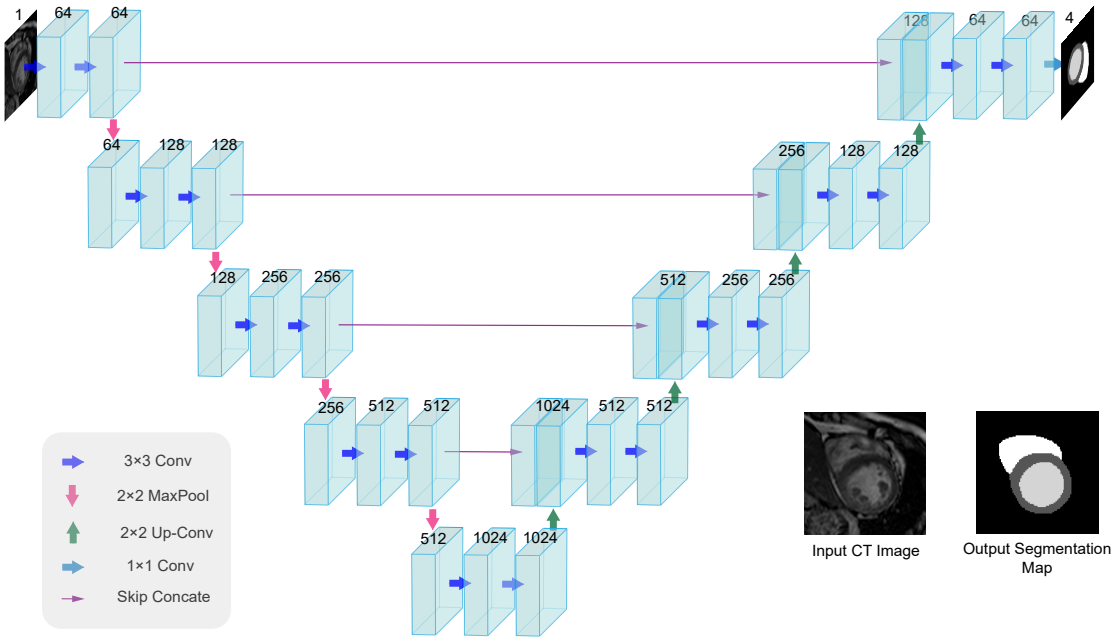


Figure 2.1: The architecture of U-Net. The shape is like the letter, so that called U-Net.

2.1 Deep Learning Development

Convolutional Neural Network[13] shown the promising performance on image processing. With the fast development of hardware, the design of CNN became deeper and achieved the accuracy closer or even higher than human recognition[14, 15, 16].

2.2 Deep Learning for Medical Images

2.2.1 Backbone: U-Net

For medical image segmentation, U-Net [17] turned out to be the most popular backbone since 2015. U-Net contained several downsampling and several symmetrical upsampling with skip connections. In the encoder part, U-Net had four times down-sampling, applying two convolutional operations and ReLU activation function. The role of the decoder was symmetric as the encoder. To get the in-

formation from the encoder, the decoder will accept half of the information from the encoder. The operation was concatenation. U-Net was good at dealing with a small amount of data. However, CNNs were still sensitive to the position lightness of the data. Therefore, the author also used smooth deformation to augment the data. At the end of the encoder, the author applied the drop-out layer.

The authors designed two types of the loss function. The first one is Cross-Entropy loss, which calculated the cross-entropy between mask and output segmentation.

$$L_{\text{CE}} = -\sum_{i=1}^{n} t_i \log\left(p_i\right), \text{ for n classes} \tag{2.1}$$

where t_i is the truth label and p_i is the Softmax probability for i^{th} class.

The second one is energy function. First, define the Softmax function:

$$p_k(\mathbf{x}) = \exp\left(a_k(\mathbf{x})\right) / \left(\sum_{k'=1}^{K} \exp\left(a_{k'}(\mathbf{x})\right)\right) \tag{2.2}$$

where a_k^x represents the activated pixels in the channel, K is the number of classes. The range of the function is between 0 and 1. Combined with Cross-Entropy Loss, the error of each mask was calculated, and finally got the total energy E.

$$E = \sum_{\mathbf{x}\in\Omega} w(\mathbf{x}) \log\left(p_{\ell(\mathbf{x})}(\mathbf{x})\right) \tag{2.3}$$

The author utilized batch gradient descent with a momentum algorithm with 0.99 as the initial value and learning rate decay. To reduce the fluctuation of gradient descent, RMSProp was applied. Since we don't have enough space to talk about this, readers could refer to pytorch official website.

Here, we will introduce two important metrics for evaluation. The first is Intersection Over Union (IOU).

$$IOU = \frac{A \cap B}{A \cup B} \tag{2.4}$$

where A and B are the ground truth and output segmentation separately.
The second one is Dice coefficient.

$$Dice = \frac{2|A \cap B|}{|A| + |B|} \tag{2.5}$$

It usually calculates the similarity of two samples.
To test the efficiency of its design, we experiment on U-Net. The dataset is from [18], containing 100 MRI samples for training and 54 samples for testing. We take them slice as the input. Each sample has 88 slices. The champion design is cascaded 3D V-Net[19]. In comparison, we use naive U-Net without any modification. In this experiment, we found that the 3D sequential property was important in-

Experiment Method	Setting	Paramaters	DICE
3D VNet	2*3D V-Net	38M	**0.923**(train validation) **0.932**(test)
Multi-task Learning (whether post/pre)	U-Net + post/pre label	18M+	0.901(train validation) 0.921(test)
Naive U-Net	Original	17M	0.883(train validation)
Naive U-Net	0.6*training sample	17M	0.882(train validation)
Naive U-Net	Half Channel	4M	**0.899**(train validation)
Naive U-Net	Cut the deepest layer	4M	0.870(train validation)
Naive U-Net	w/o skip connection	25M	0.854(train validation)

formation to learn. If we used the 2d slice, each slice was independent. The champion's DICE reached 0.923. The other method[20] uses multi-task learning, adding one branch to classify whether the patient toke radiofrequency ablation or not.
In the experiment of the naive net, we trained five epochs with batch size 12 on single 2080TI. When reducing half channel of the network, the result unexpectedly increased 1.7% with 4M parameters. This indicated the overfitting of the original network. When we deleted the deepest layer of the encoder, the accuracy came down to 0.870. Deleting the skip connection damages the results, illustrating the importance of skip concatenate.

2.2.2 U-Net Family

To make it effective in specific tasks, researchers proposed a variety of U-Net, including U-Net++, Double U-Net, nnUNet, ResUNet++ etc. [21, 22, 23, 24, 25] Moreover, combined the recent most popular architecture including transformer[26], TransUNet[27] and SwinUNet[28]. These algorithms have been proposed and achieved highest performance in almost every medical image segmentation tasks.[28, 27] Although the transformer-based method has impressive results, considerable computation is needed. Moreover, the workflow with the pure network can not solve all of the problems, for instance, image degeneration, cross-modality, multi-tasks. Based on the mathematical definition and deduction, a neural network serves as a function that can approximate complex expressions to measure.

2.2.3 Deep Neural Network as an Estimator

For instance, cross-modality learning, i.e., transfer the knowledge between CT and MRI. For unpaired images, classifying the unlabeled data is quite challenging. In the domain adaptation approach, the most important job is to measure the discrepancy in extract feature space (detailed explained in preliminary). Tzeng et al. utilized the Maximum Mean Discrepancy (MMD) as a domain fusion metric.[29] Wu et al. proposed a CF distance to measure the domain discrepancy.[30] Dou et al. presented a plug-and-play adversarial domain adaptation network, achieving an average of 63.9% Dice compared with the 13.2% without domain adaptation method.[31] Chen et al. present effectively adaptation framework Deeply Synergistic Image and Feature Alignment (SIFA).[32]. These papers show deep neural networks have a strong performance in domain adaptation. The neural networks always serve as an estimator. We will talk more in the chapter Preliminary.

Chapter 3

Preliminary

A workman must first sharpen his tools if he is to do his work well.

Confucius

3.1 Unsupervised Domain Adaptation for Cardiac Segmentation

3.1.1 Definition

Due to the privacy of medical imaging, researchers can only get a small amount of data. Moreover, the gold-standard segmentation from the expert is hard to collect and time-consuming. Therefore, it is difficult for researchers to train models in each type of data, such as Computerized Tomography (CT) or Magnetic Resonance Imaging (MRI). Moreover, imaging of the same type probably belongs to different distributions because of the distinction of machine parameter settings, for instance, Tesla C5, C7. In this problem, the cardiac data from CT has been labeled as MRI without labels(or inverse). We want to segment the MRI image by using the knowledge of labeled data in the CT, i.e., source domain. Unsupervised Domain Adaptation (UDA) provides the opportunity to allow a model which is trained perfectly on labeled data can be transferred to the unlabeled data if the source domain and target domain show similar features.

3.1.2 Preliminary

This section will define the problem and introduce the model utilized in the following chapter.

Unsupervised Domain Adaptation

Firstly, define a classification problem. X is the input space, $Y = \{0, 1, \ldots, L-1\}$ is L possible label. There are data from two domain subject to $X \times Y$, calling them $\mathcal{D}_{\mathrm{S}}$ and $\mathcal{D}_{\mathrm{T}}$ respectively. There are independent identically distributed with the label in source domain while no label in target domain. Our aim is to train a discriminator $\eta : X \to Y$, minimizing target risk, i.e. $Min(R_{\mathcal{D}_{\mathrm{T}}}(\eta) =$

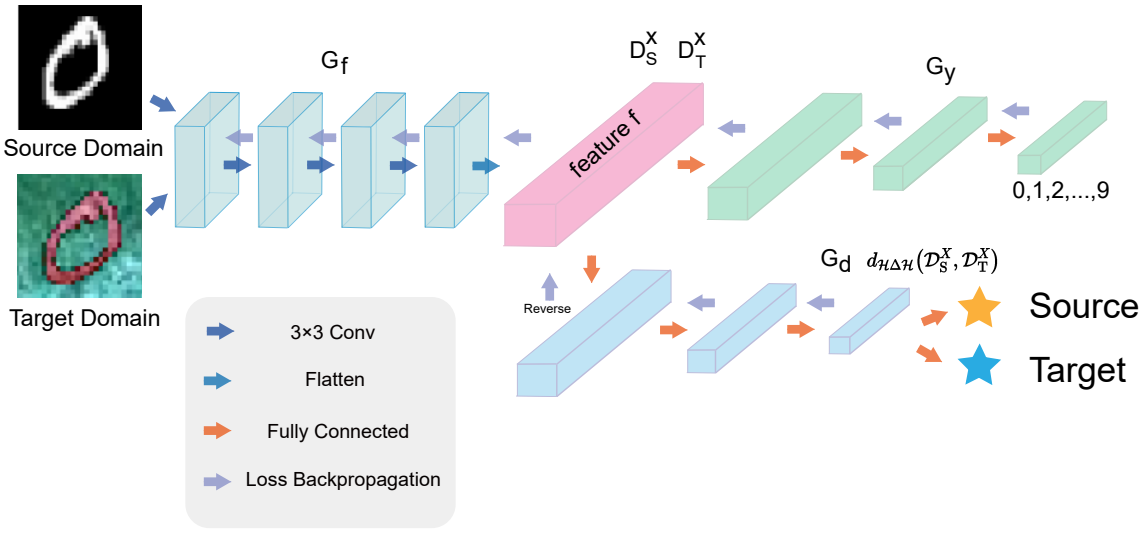


Figure 3.1: Adversarial Domain Adaptation Workflow

$\Pr_{(\mathbf{x},y)\sim\mathcal{D}_{\mathrm{T}}}(\eta(\mathbf{x}) \neq y))$

Therefore, we design a map G_f, mapping $\mathcal{D}_{\mathrm{S}}$ and $\mathcal{D}_{\mathrm{T}}$ to target feature space. In this space, the feature from the source domain and target domain should be as close as possible. To get the criteria to measure the distance, we have to adversarially design a map G_d to discriminate the class in this feature space.

The design of G_d is as follows:

Define a hypothesis class $\mathcal{H}$ in which there exists one best classifier that can be found. Find the upper bound of distance:

$$d_{\mathcal{H}}\left(\mathcal{D}_{\mathrm{S}}^{X}, \mathcal{D}_{\mathrm{T}}^{X}\right) = 2 \sup_{\eta \in \mathcal{H}} \left| \Pr_{\mathbf{x}\sim\mathcal{D}_{\mathrm{S}}^{X}}[\eta(\mathbf{x}) = 1] - \Pr_{\mathbf{x}\sim\mathcal{D}_{\mathrm{T}}^{X}}[\eta(\mathbf{x}) = 1] \right| \tag{3.1}$$

The greatest distance of probability is $\mathcal{H}$ divergence. Towards the discrete data in practice, given a symmetric $\mathcal{H}$, $S \sim \left(\mathcal{D}_{\mathrm{S}}^{X}\right)^{n}$,$T \sim \left(\mathcal{D}_{\mathrm{T}}^{X}\right)^{n'}$, the distance can be calculated as:

$$d_{\mathcal{H}}\left(\mathcal{D}_{\mathrm{S}}^{X}, \mathcal{D}_{\mathrm{T}}^{X}\right) = 2\left(1 - \min_{\eta \in \mathcal{H}}\left[\frac{1}{n}\sum_{i=1}^{n} I\left[\eta\left(\mathbf{x}_i\right) = 0\right] + \frac{1}{n'}\sum_{i=n+1}^{N} I\left[\eta\left(\mathbf{x}_i\right) = 1\right]\right]\right) \tag{3.2}$$

The distance of $\mathcal{H}\Delta\mathcal{H}$:

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}_{\mathrm{S}}^{X}, \mathcal{D}_{\mathrm{T}}^{X}\right) &= 2 \sup_{\eta \in \mathcal{H}\Delta\mathcal{H}} \left| \Pr_{\mathbf{x}\sim\mathcal{D}_{\mathrm{D}}^{X}}[I[\eta(\mathbf{x}_i)=1] - \Pr_{\mathbf{x}\sim\mathcal{D}_{\mathrm{T}}^{X}}[I[\eta(\mathbf{x}_i)=1]] \right| \\ &\leq 2 \sup_{\eta \in \mathcal{H}_d} \left| \Pr_{\mathbf{x}\sim\mathcal{D}_{\mathrm{D}}^{X}}[I[\eta(\mathbf{x}_i)=1] - \Pr_{\mathbf{x}\sim\mathcal{D}_{\mathrm{T}}^{X}}[I[\eta(\mathbf{x}_i)=1]] \right| \\ &= 2 \sup_{\eta \in \mathcal{H}_d} \left| \Pr_{\mathbf{x}\sim\mathcal{D}_{\mathrm{D}}^{X}}[I[\eta(\mathbf{x}_i)=1] - \Pr_{\mathbf{x}\sim\mathcal{D}_{\mathrm{T}}^{X}}[I[\eta(\mathbf{x}_i)=0]] - 1 \right| \end{aligned} \tag{3.3}$$

Under the circumstance of enough samples, we can specifically calculate the value of upper bound $d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}_{\mathrm{S}}^{X}, \mathcal{D}_{\mathrm{T}}^{X}\right)$. The classification whose complexity is designed to be higher than the samples could successfully classify the two domains. Then we can find the function η i.e. G_d.

During the process of training, G_f and G_d play the adversarial game to improve the performance of G_f, that is, when the distance of the source domain and target domain in the feature space become closer and closer. It is hard for the discriminator to classify. Now, we have achieved the domain adaptation in this feature space.

Then, we have to design a classifier G_y that can classify the label in each domain.

Theorem 1 *For every $h \in \mathcal{H}$:*

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}_{\mathrm{S}}^{X}, \mathcal{D}_{\mathrm{T}}^{X}\right) + \lambda \tag{3.4}$$

where $\lambda = \epsilon_S\left(h^*\right) + \epsilon_T\left(h^*\right), h^* = \arg\min_{h\in\mathcal{H}} \epsilon_S(h) + \epsilon_T(h)$

The error of G_y is constrained by the error $\epsilon_S(h)$ on the source domain, the distance $\mathcal{H}\Delta\mathcal{H}$ of feature between the source domain and target domain, and one constant λ.

Proof 3.1.1 *Suppose the $\mathcal{Z}$ is the feature space after the mapping of G_y. $\mathcal{Z}_{Gy} = \{z \in \mathcal{Z} : G_y(z) = 1\}$ represents the set classified as 1 (source domain 0, target domain 1). Then*

$$\epsilon_T(h) \leq \lambda_T + Pr_{\mathcal{D}_T}\left[Z_{G_y} \Delta Z_{G_y^*}\right] \tag{3.5}$$

where G_y and G_y^* is the set of inconsistent classification. Then

$$\Pr_{\mathcal{D}_T}\left[Z_h \Delta Z_{h^*}\right] = \Pr_{\mathcal{D}_T}\left[\{z \in \mathcal{Z} : h(z) = 1\} \oplus \{z \in \mathcal{Z} : h^*(z) = 1\}\right] \tag{3.6}$$

$$\lambda_T + \Pr_{\mathcal{D}_T}\left[Z_h \Delta Z_{h^*}\right] \leq \lambda_T + \Pr_{\mathcal{D}_S}\left[Z_h \Delta Z_{h^*}\right] + \left|\Pr_{\mathcal{D}_S}\left[Z_h \Delta Z_{h^*}\right] - \Pr_{\mathcal{D}_T}\left[Z_h \Delta Z_{h^*}\right]\right| \tag{3.7}$$

The inconsistent classification is smaller than the total error from G_y and G_y^*:

$$\Pr_{\mathcal{D}_S}\left[Z_h \Delta Z_{h^*}\right] \leq \lambda_S + \epsilon_S(h) \tag{3.8}$$

According to the 3.2:

$$d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}_{\mathrm{S}}^X, \mathcal{D}_{\mathrm{T}}^X\right) = 2 \sup_{h_1, h_2 \in \mathcal{H}} \left| Pr_{\mathcal{D}_{\mathrm{S}}^X}\left[Z_{h_1} \Delta Z_{h_2}\right] - \Pr_{\mathcal{D}_{\mathrm{T}}^X}\left[Z_{h_1} \Delta Z_{h_2}\right] \right| \tag{3.9}$$

Combined with 3.6:

$$\mid Pr_{\mathcal{D}_S}\left[Z_h \Delta Z_{h^*}\right] - \Pr_{\mathcal{D}_T}\left[Z_h \Delta Z_{h^*}\right] \leq \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}_{\mathrm{S}}^X, \mathcal{D}_{\mathrm{T}}^X\right) \tag{3.10}$$

Finally, with 3.5,3.6,3.7,3.10,

$$\epsilon_T(h) \leq \lambda_T + \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}_{\mathrm{S}}^X, \mathcal{D}_{\mathrm{T}}^X\right) \tag{3.11}$$

Proved.

In practice, we have to introduce the theory from VC dimension. Suppose hypothesis class H has d VC dimension. For $S \sim \left(\mathcal{D}_{\mathrm{S}}^X\right)^n$,$T \sim \left(\mathcal{D}_{\mathrm{T}}^X\right)^{n'}$, the generalization error is defined as:

$$\epsilon_S(h) \leq \hat{\epsilon}_S(h) + \sqrt{\frac{4}{m}\left(d \log \frac{2em}{d} + \log \frac{4}{\delta}\right)} \tag{3.12}$$

with the probability of $1 - \delta$. Consider the discrete form of 3.3 from 3.2, we can

get the upper bound of error:

$$R_{\mathcal{D}_{\mathrm{T}}}(\eta) \leq R_S(\eta)+\sqrt{\frac{4}{n}\left(d\log\frac{2en}{d}+\log\frac{4}{\delta}\right)}+\hat{d}_{\mathcal{H}}(S,T)+4\sqrt{\frac{1}{n}\left(d\log\frac{2n}{d}+\log\frac{4}{\delta}\right)}+\lambda \tag{3.13}$$

where $\beta \geq \inf_{\eta^* \in \mathcal{H}}\left[R_{\mathcal{D}_{\mathrm{S}}}\left(\eta^*\right)+R_{\mathcal{D}_{\mathrm{T}}}\left(\eta^*\right)\right]$, $R_S(\eta) = \frac{1}{n}\sum_{i=1}^{m} I\left[\eta\left(\mathbf{x}_i\right) \neq y_i\right]$, $R_S(\eta)$ represents the error of source domain. Finally, we can design a label classifier G_y. According to the definition above, now design the neural network. The prediction results are evaluated from G_f and G_y, calculating the loss L_y. The data from the source domain and target domain are mapped into feature f by G_f. The domain classifier G_d will do the classification job, calculating the loss L_d. The gradient of loss Ld is reversed when backpropagated to feature f so that G_f and G_d play the adversarial game. After training several times alternately, the distribution of source domain and target domain in feature f becomes the same. Finally, we can use the well-trained network G_f and G_y to classify on the target domain.

Variational Auto Encoder

Suppose x is the input. VAE[33] wants to get the reconstructed images $\hat{x}$, which are close to x input but could generate the continuous images with the property of x. First, define the latent variable z and prior distribution $p(z)$, where the latent variable is a part between the encoder and decoder. Decoder sampling from z, get the likelihood distribution $p(x|z)$. The latent variable is trained from the input data point so that there is a posterior distribution $p(z|x)$.

Suppose $p(z) \sim \mathcal{N}(0, I)$, where I is an identity metrix.

Before decoding, we must have a posterior distribution $p(z|x)$ with a good representation of x, which is approximated by neural network $q_x(z)$. Therefore, the target is to minimize the discrepancy of posterior distribution $p(z|x)$ and $q_x(z)$. To

measure the distance of two distributions, we introduce Kullback-Leibler divergence(KLD)[34]:

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} \mathrm{d}x \tag{3.14}$$

With Bayesian Rule:

$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)} \tag{3.15}$$

Find g^* and h^* :

$$(g^*, h^*) = \underset{(g,h)\in G\times H}{\arg\min}\, KL\left(q_x(z)\|p(z \mid x)\right) \tag{3.16}$$

$$\begin{aligned}
&= \underset{(g,h)\in G\times H}{\arg\min} \Big(\int q_x(z) \log \frac{q_x(z)}{\frac{p(z(x)p(x)}{p(x)}} dz \Big) \\
&= \underset{(g,h)\in G\times H}{\arg\min} \Big(\int q_x(z) \log q_x(z) + \log p(x) \int q_x(z) dz - \int q_x(z) \log[p(x \mid z)p(z)] dz \Big) \\
&= \underset{(g,h)\in G\times H}{\arg\min} \Big(\log p(x) + \int q_x(z) \log q_x(z) dz - \int q_x(z) \log[p(x \mid z)p(z)] dz \Big) \\
&= \underset{(g,h)\in G\times H}{\arg\min} \Big(\int q_x(z) \log \frac{q_x(z)}{p(z)} dz - \int q_x(z) \log p(x \mid z) dz \Big) \\
&= \underset{(g,h)\in G\times H}{\arg\min} \big(KL(q_x(z)\|p(z)) - \mathbb{E}_{z\sim q_x}(\log p(x \mid z)) \big) \\
&= \underset{(g,h)\in G\times H}{\arg\max} \big(\mathbb{E}_{z\sim q_x}(\log p(x \mid z)) - KL(q_x(z)\|p(z)) \big)
\end{aligned}$$

We have to maximize the log-likelihood of the output and minimize the KL divergence between the encoder and latent variable z with a gaussian distribution. For the first term, cross-entropy loss or mean square error loss can be applied. Here we use MSE as an example.

$$(g^*, h^*) = \underset{(f,g,h)\in F\times G\times H}{\arg\max} \left(\mathbb{E}_{z\sim q_x} \left(-\frac{\|x - f(z)\|^2}{2c} \right) - KL\left(q_x(z)\|p(z)\right) \right) \tag{3.17}$$

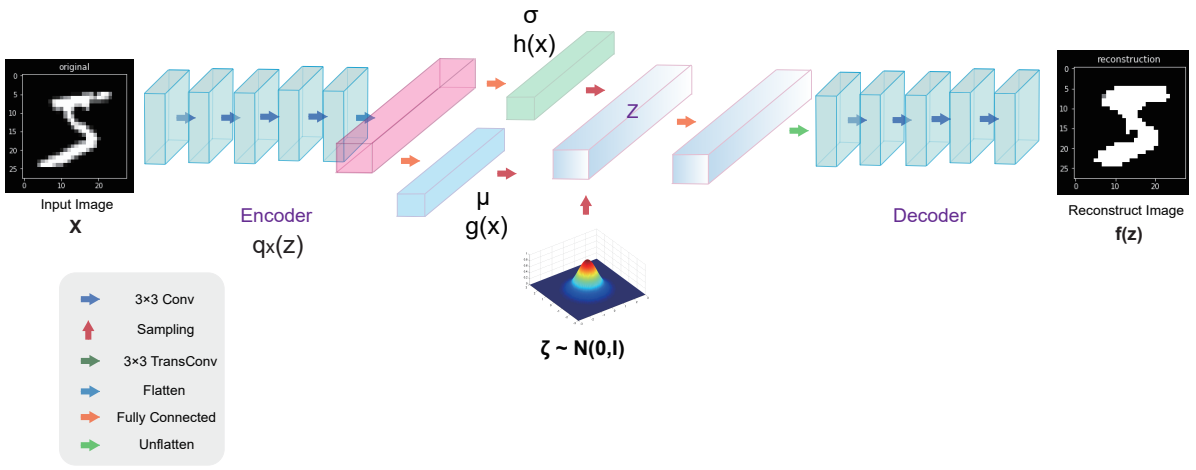


Figure 3.2: The architecture of Variational Auto Encoder (VAE)

Where c is a constant.

For the second term, noting that $p(z)$ is normal distribution, $q_x(z)$ should have two parts, one is to estimate the mean, and the other one is to estimate the variance. We design $q_x(z) \sim \mathcal{N}(g(x), h(x))$, where $g(x)$ and $h(x)$ are of encoder's variation.

Meanwhile, to make sure every process in this training is derivable to backpropagate the gradient, and the process of sampling $q_x(z) \sim \mathcal{N}(g(x), h(x))$ is not derivable. We reparameterize it as $z = h(x)\zeta + g(x)$, where $\zeta \sim \mathcal{N}(0, I)$. Then the loss function $\mathcal{L}$ can be defined as:

$$\mathcal{L} = C||x - f(z)||^2 - KL(N(g(x), h(x))\|N(0, I)) \tag{3.18}$$

Where f is the decoder. As every number in metric I is independent, we only consider the one variable normal distribution in deduction. With $\mathcal{N}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

, to make it convenience, replace the $g(x)$, $h(x)$ to μ, σ^2

$$\begin{aligned}
\mathcal{L}_{\mathcal{KL}} &= \int \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \left(\log \frac{e^{-(x-\mu)^2/2\sigma^2}/\sqrt{2\pi\sigma^2}}{e^{-x^2/2}/\sqrt{2\pi}} \right) dx \\
&= \int \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \log \left\{ \frac{1}{\sqrt{\sigma^2}} \exp \left\{ \frac{1}{2} \left[x^2 - (x-\mu)^2/\sigma^2 \right] \right\} \right\} dx \\
&= \frac{1}{2} \int \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \left[-\log \sigma^2 + x^2 - (x-\mu)^2/\sigma^2 \right] dx \\
&= \frac{1}{2} \left(-\log \sigma^2 + \mu^2 + \sigma^2 - 1 \right)
\end{aligned} \tag{3.19}$$

Finally, the loss function $\mathcal{L}$ is:

$$\mathcal{L} = C||x - f(z)||^2 - \frac{1}{2}(-log(h(x)) + g^2(x) + h(x) - 1) \tag{3.20}$$

Where C is a constant. Hence, 3.20 is the objective for neural network to minimize.

Mutual Information Neural Estimation

Mutual information is to measure the relationship between random variables. In biomedical sciences, blind source separation, feature selection, and causality, mutual information is widely used.[35] However, mutual information is often related to higher-order functions. For instance, the mutual information of latent space and output. It is hard to estimate the mutual information without the joint and marginal distribution formula. This paper[35] uses a neural network to evaluate mutual information.

Mutual information of two random variables X and Z can be estimated as:

$$I(X;Z) = \int_{\mathcal{X}\times\mathcal{Z}} \log \frac{d\mathbb{P}_{XZ}}{d\mathbb{P}_X \otimes \mathbb{P}_Z} d\mathbb{P}_{XZ} \tag{3.21}$$

where $\mathbb{P}_{XZ}$ is the joint probability distribution, $\mathbb{P}_X$ and $\mathbb{P}_Z$ are the marginals. It

can be understood as the decrease of the uncertainty in X given Z:

$$I(X;Z) := H(X) - H(X \mid Z) \tag{3.22}$$

where H is the Shannon entropy.

Moreover, we also can write it in KL divergence[34]:

$$I(X;Z) = D_{KL}\left(\mathbb{P}_{XZ} \| \mathbb{P}_X \otimes \mathbb{P}_Z\right) \tag{3.23}$$

Proof 3.1.2

$$\begin{aligned}
I(X,Z) &= H(X) - H(X \mid Z) \\
&= -\int_x p(x) \log p(x) dx + \int_{x,z} p(x,z) \log p(x \mid z) dx dz \\
&= -\int_{x,z} p(x,z) \log p(x) dx dz + \int_{x,z} p(x,z) \log p(x \mid z) dx dz \\
&= \int_{x,z} p(x,z)(-\log p(x) + \log p(x \mid z)) dx dz \\
&= \int_{x,z} p(x,z) \log \frac{p(x \mid z)}{p(x)} dx dz \\
&= \int_{x,z} p(x,z) \log \frac{p(x,z)}{p(x)p(z)} dx dz \\
&= D_{KL}(P(X,Z) \| P(X) \otimes P(Z))
\end{aligned} \tag{3.24}$$

KL divergence was mentioned in 3.14.

The paper MINE mentions two types of mutual information, one is Donsker-Varadhan representation[36] with a tighter bound, and the other one is f-divergence representation[37] with a loose bound.

Theorem 2 *The mutual information form of KL divergence can be written in the*

following dual representation:

$$D_{KL}(\mathbb{P}\|\mathbb{Q}) = \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}_{\mathbb{P}}[T] - \log\left(\mathbb{E}_{\mathbb{Q}}\left[e^{T}\right]\right) \tag{3.25}$$

where the supremum is considering all functions T such that the two expectations are finite.

Proof 3.1.3 *Firstly, we define the distance between the left side and right in the 3.25:*

$$\Delta := D_{KL}(\mathbb{P}\|\mathbb{Q}) - \left(\mathbb{E}_{\mathbb{P}}[T] - \log\left(\mathbb{E}_{\mathbb{Q}}\left[e^{T}\right]\right)\right) \tag{3.26}$$

We have to verify $\Delta \geq 0$. Now, we construct the right side of 3.26. Given any function T, define Gibbs distribution $\mathbb{G}$ such that $d\mathbb{G} = \frac{1}{Z}e^{T}d\mathbb{Q}$, where $Z = \mathbb{E}_{\mathbb{Q}}\left[e^{T}\right]$. Then we can get:

$$\begin{aligned} \mathbb{E}_{\mathbb{P}}[T] - \log\left(\mathbb{E}_{\mathbb{Q}}\left[e^{T}\right]\right) &= \mathbb{E}_{\mathbb{P}}[T] - \log Z \\ &= \mathbb{E}_{\mathbb{P}}\left[\log\frac{d\mathbb{G}}{d\mathbb{Q}}\right] \end{aligned} \tag{3.27}$$

Substitute 3.27 into 3.26:

$$\begin{aligned} \Delta &= \mathbb{E}_{\mathbb{P}}\left[\log\frac{d\mathbb{P}}{d\mathbb{Q}}\right] - \mathbb{E}_{\mathbb{P}}\left[\log\frac{d\mathbb{G}}{d\mathbb{Q}}\right] \\ &= \mathbb{E}_{\mathbb{P}}\left[\log\frac{d\mathbb{P}}{d\mathbb{Q}} - \log\frac{d\mathbb{G}}{d\mathbb{Q}}\right] \\ &= \mathbb{E}_{\mathbb{P}}\log\left[\frac{d\mathbb{P}}{d\mathbb{Q}}\frac{d\mathbb{Q}}{d\mathbb{G}}\right] \\ &= \mathbb{E}_{\mathbb{P}}\log\frac{d\mathbb{P}}{d\mathbb{G}} = D_{KL}(\mathbb{P}\|\mathbb{G}) \end{aligned} \tag{3.28}$$

Because KL divergence is always greater than 0, hence, $\Delta \geq 0$.

Proved.

The other dual representation is f-divergence representation:

$$D_{KL}(\mathbb{P}\|\mathbb{Q}) \geq \sup_{T\in\mathcal{F}} \mathbb{E}_{\mathbb{P}}[T] - \mathbb{E}_{\mathbb{Q}}\left[e^{T-1}\right] \tag{3.29}$$

Noting that $x \geq e \log x$, it can be easily proved.

The neural estimation is to estimate the function T. With neural network parameters $\theta \in \Theta$. We exploit the bound:

$$I(X;Z) \geq I_{\Theta}(X,Z) \tag{3.30}$$

where $I_{\Theta}(X,Z)$ is the neural information measure defined as:

$$I_{\Theta}(X,Z) = \sup_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}_{XZ}}\left[T_\theta\right] - \log\left(\mathbb{E}_{\mathbb{P}_X\otimes\mathbb{P}_Z}\left[e^{T_\theta}\right]\right) \tag{3.31}$$

Let $\mathcal{F} = \{T_\theta\}_{\theta\in\Theta}$ be the set of neural networks. The mutual information can be estimated as:

$$\widehat{I(X;Z)}_n = \sup_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}_{XZ}^{(n)}}\left[T_\theta\right] - \log\left(\mathbb{E}_{\mathbb{P}_X^{(n)}\otimes\mathbb{P}_Z^{(n)}}\left[e^{T_\theta}\right]\right) \tag{3.32}$$

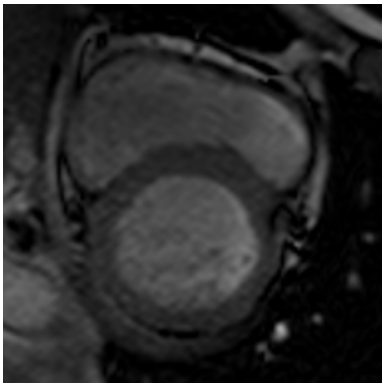
We draw the sample from the Z marginal distribution. Apply the stochastic gradient:

$$\widehat{G}_B = \mathbb{E}_B\left[\nabla_\theta T_\theta\right] - \frac{\mathbb{E}_B\left[\nabla_\theta T_\theta e^{T_\theta}\right]}{\mathbb{E}_B\left[e^{T_\theta}\right]} \tag{3.33}$$

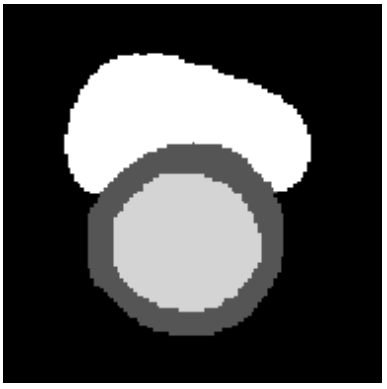
We use an exponential moving average to reduce the bias estimate of the whole batch gradient. The author also proves its strong consistency so we can ignore the error. Please refer [35] for detailed information.

Chapter 4

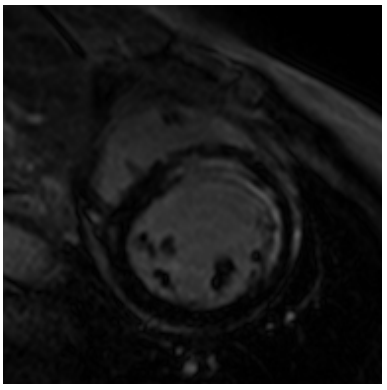
Unsupervised Domain Adaptation for Cardiac Segmentation



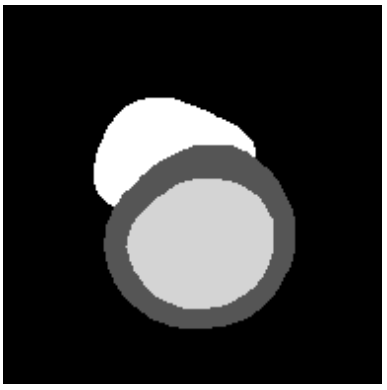
(a) CT



(b) CT Mask



(c) LGE MRI



(d) LGE MRI Mask

Unsupervised Domain Adaptation for Cardiac Segmentation: Towards Structure Mutual Information Maximization

4.1 Abstract

Unsupervised domain adaptation approaches have recently succeeded in various medical image segmentation tasks. The reported works often tackle the domain shift problem by aligning the domain-invariant features and minimizing the domain-specific discrepancies. That strategy works well when the difference between a specific domain and between different domains is slight. However, the generalization ability of these models on diverse imaging modalities remains a significant challenge. This paper introduces UDA-VAE++, an unsupervised domain adaptation framework for cardiac segmentation with a compact loss function lower bound. To estimate this new lower bound, we develop a novel Structure Mutual Information Estimation (SMIE) block with a global estimator, a local estimator, and a prior information matching estimator to maximize the mutual information between the reconstruction and segmentation tasks. Specifically, we design a novel sequential reparameterization scheme that enables information flow and variance correction from the low-resolution latent space to the high-resolution latent space. Comprehensive experiments on benchmark cardiac segmentation datasets demonstrate that our model outperforms previous state-of-the-art qualitatively and quantitatively. The code is available at https://github.com/LOUEY233/Toward-Mutual-Information

4.2 Related Work

4.2.1 Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (UDA) has been widely used for biomedical image segmentation tasks. The early works [38] and [39] leverage unsupervised domain adaptation with adversarial training for multi-modal biomedical image segmentation. Specifically, both papers utilize a plug-and-play domain adaptation module to align the features in the source and the target domain.

Due to the promising generalization ability of Generative Adversarial Network (GAN) [40], recent research has begun to incorporate GAN in UDA for biomedical image segmentation. For example, [41] utilizes CycleGAN [42] with a shape-consistency loss to realize cross-domain translation between CT and MRI images. SIFA [32] presents a synergistic domain alignment at both image-level and feature-level using the adversarial learning of CycleGAN to exploit domain-invariant characteristics. DUDA [43] further incorporates a cross-domain consistency loss to improve the segmentation performances.

Another faithful research direction is to use Variational Autoencoder (VAE) [44]. That strategy is advantageous when there are few images in the target domain. For instance, [45] follows the few-shot learning strategy, integrating a VAE-based feature prior to matching with adversarial learning to exploit the domain-invariant features. FUDA [46] further incorporates Random Adaptive Instance Normalization to explore diverse target styles where there is only one unlabeled image in the target domain. The recent work CFDNet [30] proposes an effective metric, dubbed CF Distance, which enables explicit domain adaptation with image reconstruction and prior distribution matching. Another work UDA-VAE [47] goes even further: it drives the latent space of the source and target domains towards a common, parameterized variational form following Gaussian Distribution.

Compared with previous UDA approaches, our method is the first that sequen-

tially integrates multi-scale latent space features. That design enables our network to effectively minimize the domain-specific discrepancy according to the information flow from the low-resolution latent space to the high-resolution latent space.

4.2.2 Mutual Information Neural Estimation

Mutual Information Neural Estimation (MINE) is first introduced in [35], where the author utilizes gradient descent algorithms over neural networks to approximate the mutual information between continuous random variables. Based upon MINE, Deep InfoMax (DIM) [48] explores unsupervised visual representation learning by maximizing the mutual information for the network input and the encoded output under statistical constrain. A recent work [49] utilizes MINE to address the domain shift problem in unsupervised domain adaptation. Specifically, that paper integrates network predictions and local features into global features by simultaneously maximizing the mutual information.

Recently, MINE has been applied in biomedical image processing tasks. For example, based on MINE, [50] maximizes the mutual information between source and fused images from Multiview 3-D Echocardiography. [51] tackle the challenging unsupervised multimodal brain image segmentation task by estimating the mutual information using a lightweight convolutional neural network.

Different from previous MINE approaches, our framework is the first that conducts mutual information estimation and maximization with both image reconstruction and image segmentation. Our unique design enables image reconstruction and image segmentation to be mutually beneficial during model learning.

4.3 Methodology

In this section, we will discuss our UDA-VAE++'s workflow, explain the proposed structure mutual information estimation block, and display the loss functions.

Symbols	Description
S	Source domain
T	Target domain
z	Latent variable
x	Input image data point
$p_\theta()$	PDF of variables with parameter θ
$q_\phi()$	Neural network with parameter ϕ
$D(\phi_S, \phi_T)$	Domain distance between source and target
$\hat{y}$	Predicted segmentation
y	Ground truth segmentation
R_S	Reconstructed image in the source domain
R_T	Reconstructed image in the target domain
D_{KL}	KL Divergence
ϵ	Reconstruction error
H	Entropy

Table 4.1: Preliminary for Important Symbols

4.3.1 UDA-VAE++ Model Workflow

As shown in Fig. 4.1, we use U-Net [52] as our backbone due to its remarkable success in medical image segmentation. Firstly, The network performs four downsamplings. Each of the downsampling operations uses two convolutional layers. Secondly, the network uses upsampling symmetrically with skip connection. We then obtain a multi-scale encoding output with channels of 256, 128, 64, and image sizes of 40×40, 80×80, 160×160, respectively. Each encoding output will be followed by variational reasoning [44, 53]. Using the reparameterization trick [44] with the latent mean variable, the latent log variance variable, and the standard normal distribution, we obtain three latent variables z_1, z_2, z_3. After that, We use a single convolutional layer to obtain the predicted segmentation $\hat{y}$.

Finally, we leverage a fully convolutional network with 7 layers for image reconstruction. The input for the source domain includes the ground truth segmentation y and the latent variable z, whereas the input for the target domain is the predicted segmentation $\hat{y}$.

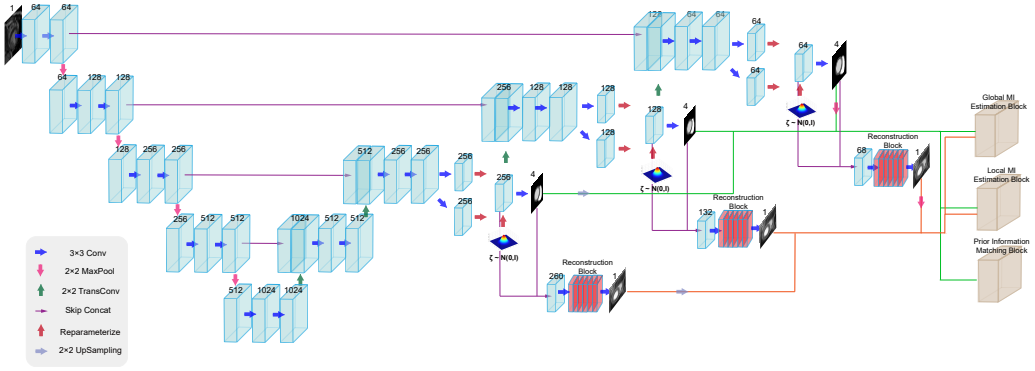


Figure 4.1: The Model Architecture of UDA-VAE++. The backbone of UDA-VAE++ is U-Net (blue boxes) with three scales of variational blocks. The green line refers to the concatenation of the segmentation output, whereas the orange line indicates the concatenation of the reconstruction output. The reconstruction blocks (red boxes) contain seven convolution layers. The grey box refers to the MI estimation block detailed in Fig. 4.3

4.3.2 Structure Mutual Information Estimation

In this subsection, we aim to estimate the mutual information between the segmentation outcome $\hat{y}$ and the reconstruction output R in the source and target domains. The mutual information can be formulated as:

$$\widehat{\mathcal{I}}\left(\hat{y}; R\right) = D_{KL}\left(\mathbb{P}_{\hat{y}R} \| \mathbb{P}_{\hat{y}} \otimes \mathbb{P}_{R}\right) \tag{4.1}$$

The KL divergence between joint distribution $\mathbb{P}_{\hat{y}R}$ and marginal distribution $\mathbb{P}_{\hat{y}} \otimes \mathbb{P}_{R}$ can be written as its dual representation[54] as below:

$$D_{KL}(\mathbb{P}_{\hat{y}R} \| \mathbb{P}_{\hat{y}} \otimes \mathbb{P}_{R}) = \sup_{T:\Omega \to \mathbb{R}} \left(\mathbb{E}_{\mathbb{P}_{\hat{y}R}}[T] - \log\left(\mathbb{E}_{\mathbb{P}_{\hat{y}} \otimes \mathbb{P}_{R}}\left[e^{T}\right]\right)\right) \tag{4.2}$$

where T is the set of all possible neural network.

Inspired by [48], we are interested in automatically maximizing the mutual information rather than manually obtaining the exact value for mutual information. The mutual information maximization process can be formulated as:

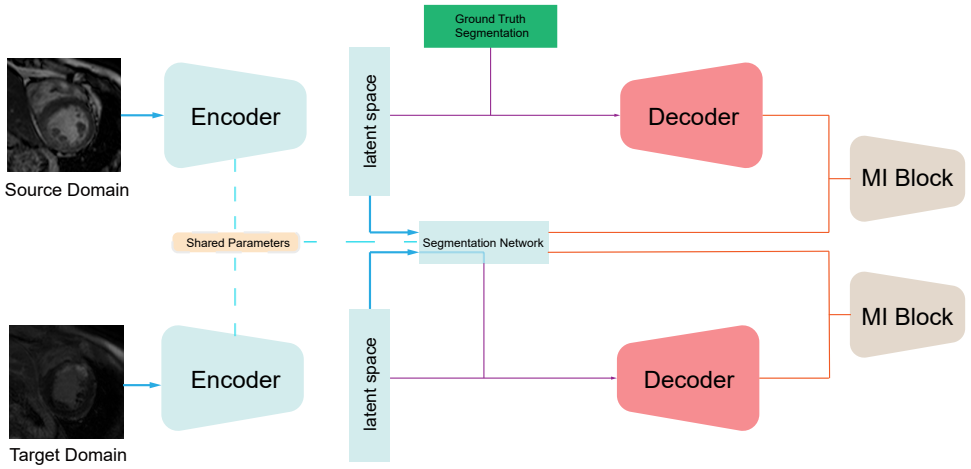


Figure 4.2: The workflow for Unsupervised Domain Adaptation. The image from the source and target domain will first be encoded in the shared parameters downsampling part of the U-Net backbone. Next, each scale output will go through the same segmentation network. In the source domain, the ground truth segmentation masks combining the variables in latent space will be reconstructed by the upsampling part of U-Net. The MI block will maximize the mutual information of the segmentation output and the reconstruction output.

$$\widehat{\mathcal{I}}\left(\hat{y};R\right)=\mathbb{E}_{\mathbb{P}_{\hat{y}R}}\left[-\operatorname{sp}\left(-T\left(\hat{y},R\right)\right)\right]-\mathbb{E}_{\mathbb{P}_{\hat{y}}\otimes\mathbb{P}_{R}}\left[\operatorname{sp}\left(T\left(\hat{y},R'\right)\right)\right] \tag{4.3}$$

where R' is an input sampled from R, and $\operatorname{sp}(z)=\log\left(1+e^{z}\right)$ is the softplus function.

The next step is to estimate the joint and marginal distribution of $\hat{y}$ and R using contrastive learning. First, we design three estimators in the MI block [48]. The original paired R and $\hat{y}$ serve as the anchor and the positive point, respectively. We then shuffle R randomly to obtain the negative point. To fuse the data together, we upsample the 40×40 feature map and downsample the 160×160 feature map. Before entering the estimator block, the anchor and negative point will go through two convolutional layers, whereas the positive point will go through three convolutional layers.

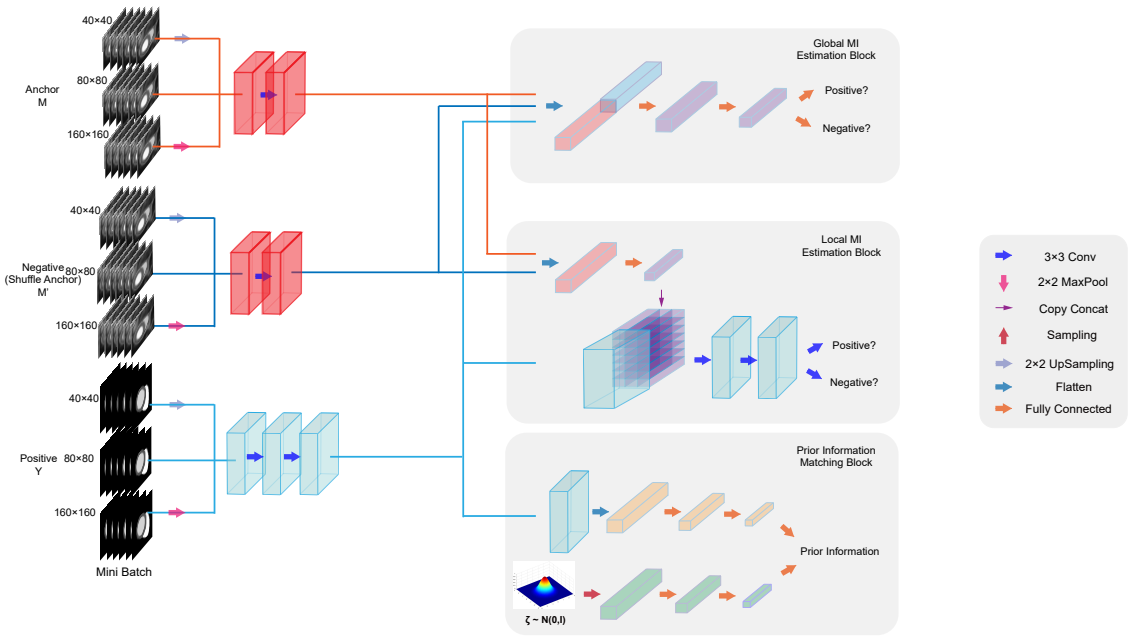


Figure 4.3: The architecture of Structure Mutual Information Estimation (SMIE) block. We use the reconstruction image as anchor, the shuffle reconstructed images as negative points, and the segmented image as positive points. The Global & Local Mutual Information (MI) Estimation Block follows contrastive learning schemes to maximize mutual information, whereas the prior information matching block align the positive point with the standard normal distribution. Finally, the sum of the outputs score from these three blocks serves as the loss function for $\mathcal{L}_{MI}$.

For the Global MI Estimation block, we concatenate the positive points with anchor and negative points, pushing the anchor away from the negative points and pulling the anchor towards the positive point. For the Local MI Estimation block, we extract the high-level semantics using fully connected layers. Next, we concatenate the semantic information with the positive point to acquire the locality information, followed by two convolutional layers for contrastive learning. Finally, motivated by [48, 49], we adopt the prior matching [55] strategy to constrain the visual representations according to standard normal distribution. Specifically, in the prior information estimation block, the positive point will go through fully connected layers and output the prior information.

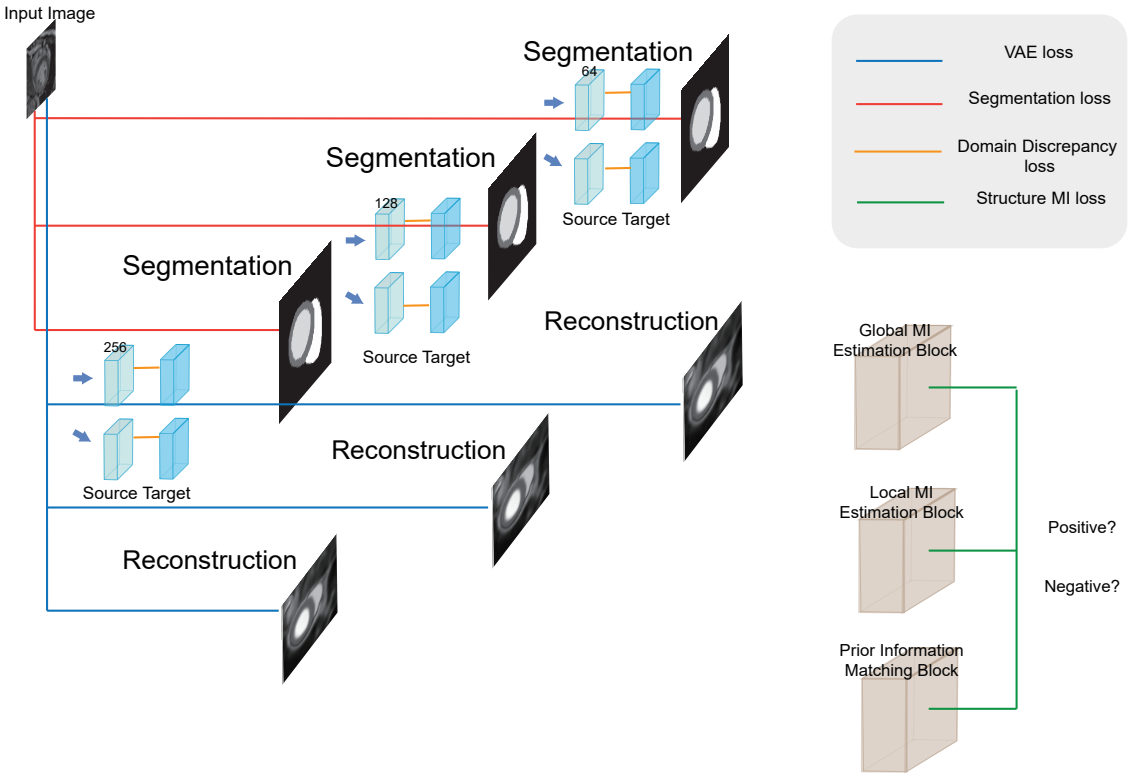


Figure 4.4: The loss function of the proposed method. The blue line refers to the reconstruction loss. The red line indicates the segmentation loss between the predicted segmentation and ground truth segmentation. The orange line illustrates the domain discrepancy loss in the latent space using the l2 norm of Gaussian distribution distance explicitly. The green line refers to the structure mutual information loss. The other loss will be calculated respectively in domain and target, except for the domain discrepancy loss.

4.3.3 Loss function

For the segmentation part, we aim to maximize the joint log-likelihood $\log p_{\theta_S}(x, y)$ of the dataset.

Theorem 3

$$
\begin{aligned}
&\log p_{\theta_S}(x, y) \\
\geq &\left(\epsilon + \widehat{\mathcal{I}}_{q\phi_S}(x, y, z) - H_{q\phi_S}(z) + \log \frac{p_{\theta_S}(x, y)}{q_{\phi_S}(x, y)}\right) \\
&- D_{KL}\left(q_{\phi_S}(z \mid x) \| p_{\theta_S}(z)\right) \\
&+ E_{q_{\phi_S}(z|x)}[\log p_{\theta_S}(x \mid y, z)] \\
&+ E_{q_{\phi_S}(z|x)}[\log p_{\theta_S}(y \mid z)]
\end{aligned}
\tag{4.4}
$$

where $\epsilon, H_{q\phi_S(z)}, \log \frac{p_{\theta_S}(x,y)}{q_{\phi_S}(x,y)}$ *are all constant.*

Proof 4.3.1 *Detailed proof will be in the supplementary material.*

For the domain discrepancy loss, we minimize it explicitly as the latent space obeys normal distribution.

Therefore, our loss function (Fig. 4.4) contains structure mutual information estimation loss $\mathcal{L}_{MI}$(Eq.4.4 line 1) reconstruction loss $\mathcal{L}_{recon}$(Eq.4.4 line 2,3), segmentation loss $\mathcal{L}_{seg}$(Eq.4.4 line 4), and domain discrepancy loss $\mathcal{L}_D$.

Reconstruction Loss

The reconstruction loss is same as the design in VAE. We use neural network $q_\phi(z|x)$ with parameter ϕ to approximate the posterior distribution $p_\theta(z|x)$ for latent variable z. In other words, we attempt to minimize the KL divergence of $q_\phi(z|x)$ and $p_\theta(z|x)$:

$$
\begin{aligned}
&D_{KL}\left(q_\phi(z|x) \| p_\theta(z|x)\right) \\
&= D_{KL}\left(q_\phi(z|x) \| p_\theta(z)\right) - E_{z\sim q_\phi}[\log p_\theta(x|z)]
\end{aligned}
\tag{4.5}
$$

The first term aims to minimize the KL divergence between the neural network $q_\phi(z|x)$ and the prior distribution $p_\theta(z) \sim N(0, I)$, where I is the identity matrix.

The neural network $q_\phi(z|x)$ performs variational reasoning upon u and σ^2 to approximate 0 and I, respectively. With the reparameterization trick[33](red arrows in Fig. 4.1), we can get:

$$D_{KL}\left(q_\phi(z|x)\|p_\theta(z)\right) = \frac{1}{2}\left(\sigma^2 + u^2 - \log\sigma^2 - 1\right) \tag{4.6}$$

The second term in equ[4.5] is to maximize the likelihood of x. This can be calculated by cross entropy loss between the input x and the reconstruction output R:

$$\mathcal{L}_{ce} = -(x\log(R) + (1-x)\log(1-R)) \tag{4.7}$$

Finally, we get the reconstruction loss:

$$\mathcal{L}_{recon} = D_{KL} + \mathcal{L}_{ce} \tag{4.8}$$

Segmentation Loss

The segmentation loss helps us minimize the loss between the predicted segmentation $\hat{y}$ and the ground truth segmentation y. We apply cross-entropy loss, which is formulated as below:

$$\mathcal{L}_{seg} = -(y\log(\hat{y}) + (1-y)\log(1-\hat{y})) \tag{4.9}$$

Domain Discrepancy Loss

The Domain Discrepancy Loss helps reduce the domain discrepancy between the source and the target domain in the latent space. In the UDA-VAE framework, [47] has proved that optimizing the distance explicitly would have better accuracy than adversarial training. As the latent space is regularized into a standard normal distribution, we can calculate the distance analytically. The Domain Discrepancy Loss is formulated as below:

$$\begin{aligned}\mathcal{L}_D &= D(q_{\phi_S}(z), q_{\phi_T}(z)) \\ &= \int [q_{\phi_S}(z) - q_{\phi_T}(z)]^2 dz \\ &= \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \left[k\left(x_{S_i}, x_{S_j}\right) + k\left(x_{T_i}, x_{T_j}\right) - 2k\left(x_{S_i}, x_{T_j}\right) \right]\end{aligned} \tag{4.10}$$

where M is the batch size. i, j are ith, jth element in one batch. As the variables in latent space obey standard normal distribution. The kernel function k is:

$$k\left(x_{S_i}, x_{T_j}\right) = (2\pi)^{-\frac{1}{2}} e^{-\frac{1}{2}\left[\frac{\left(u_{S_i} - u_{T_j}\right)^2}{\sigma_{S_i}^2 + \sigma_{T_j}^2} + \log(\sigma_{S_i}^2 + \sigma_{T_j}^2)\right]} \tag{4.11}$$

Structure Mutual Information Loss

As discussed in equ[4.3], we design a contrastive learning framework to estimate the joint and marginal distribution of $\hat{y}$ and R. To maximize $\widehat{\mathcal{I}}(\hat{y}; R)$, we design a global MI estimation block, a local MI estimation block, and a prior information matching block.

$$\mathcal{L}_{MI} = -(\alpha \widehat{\mathcal{I}}(\hat{y}; R)_{Global} + \beta \widehat{\mathcal{I}}(\hat{y}; R)_{Local} + \gamma \widehat{\mathcal{I}}_{Prior}) \tag{4.12}$$

where α, β, γ are set as 0.5, 1.0, 0.1. $\widehat{\mathcal{I}}_{Prior} = log(\mathcal{N}) + log(1 - \hat{y})$, where $\mathcal{N}$ is the standard normal distribution.

Total Loss

The total loss is defined as:

$$\begin{aligned}\mathcal{L}_{total} = &(c1\mathcal{L}_{recon} + c2\mathcal{L}_{seg} + c3\mathcal{L}_{MI})_{source} \\ &+ (c1\mathcal{L}_{recon} + c2\mathcal{L}_{seg} + c3\mathcal{L}_{MI})_{target} \\ &+ c4\mathcal{L}_D\end{aligned} \tag{4.13}$$

where c1, c2, c3, c4 are empirically set as 1e-2, 1, 1e-1, 1e-5, respectively.

Model Components						Dice		
Base	CN	Att	Global	Local	Prior	MYO	LV	RV
yes						68.42	84.41	72.59
yes	yes					68.56	84.07	74.06
yes	yes	yes				68.30	84.91	74.72
yes	yes	yes	yes			69.25	84.70	75.63
yes	yes	yes	yes	yes		68.49	87.50	**77.37**
yes	yes		yes	yes	yes	**70.75**	**88.64**	75.82
yes	yes	yes	yes	yes	yes	69.81	87.54	77.13

Table 4.2: The Ablations of model components for MS-CMRSeg Dataset from **CT to MRI**. Base: UDA-VAE [47]. CN: changing the up-sampling network of U-Net. Att: Attention. Global: Global MI Estimation Block. Local: Local MI Estimation Block. Prior: Prior Matching. The best score for UDA from CT to MRI is in **bold** while the second-best score is in blue.

	Dice (%)			ASSD (mm)		
	MYO	LV	RV	MYO	LV	RV
NoAdapt	14.50	34.51	31.10	21.6	11.3	14.5
CFDNet [30]	64.21	81.39	72.30	2.81	3.41	4.91
SIFA [32]	67.69	83.31	**79.04**	2.56	3.44	**2.13**
UDA-VAE [47]	68.42	84.41	72.59	2.39	2.59	3.97
UDA-VAE++	**70.75**	**88.64**	75.82	**2.02**	**2.27**	3.62

Table 4.3: Unsupervised Domain Adaptation for MS-CMRSeg Dataset from **CT to MRI**. The best score for Dice↑ and ASSD↓ are in **bold**.

4.4 Experiments

4.4.1 Implementation Details

We use Adam optimizer [56] and Pytorch framework [57] to train our model for 30 epochs. The learning rate is initialized at 1e-4 and is reduced by 10 % after every epoch. The batch size is 12, which takes about 1 hour to converge on a single NVIDIA Tesla V100 GPU. The network weight follows Xavier initialization [58]. Neither gradient scaling nor gradient clipping is applied during training.

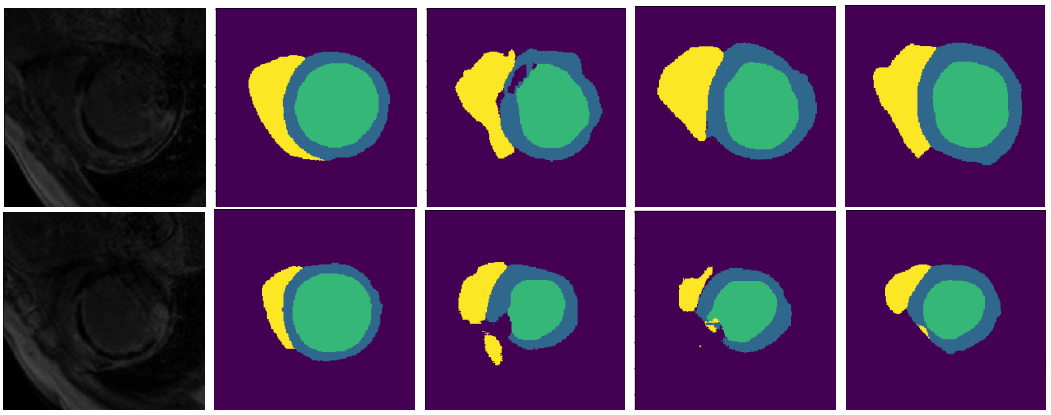


Figure 4.5: Segmentation output from MS-CMRSeg Dataset (CT to MRI). From left to right: MRI, Ground truth, CFDNet[30], UDA-VAE[47], UDA-VAE++. For the segmentation, we use yellow, green, and dark green to represent RV, MYO, and LV, respectively.

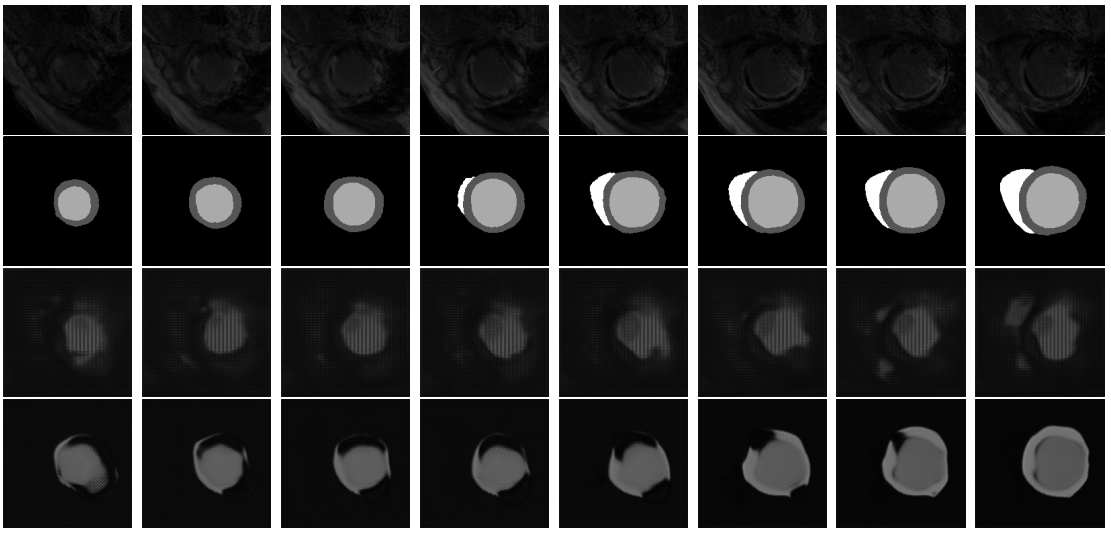


Figure 4.6: Reconstruction Images from MS-CMRSeg Dataset (CT to MRI). From top to bottom row: MRI images, corresponding segmentation ground truth, UDA-VAE, UDA-VAE++.

	Dice (%)			ASSD (mm)		
	MYO	LV	RV	MYO	LV	RV
NoAdapt	12.32	30.24	37.25	24.9	10.4	16.7
CFDNet [30]	57.41	78.44	77.63	3.61	3.87	2.49
SIFA [32]	60.89	79.32	**82.39**	3.44	3.65	1.80
UDA-VAE [47]	58.58	79.43	80.43	3.53	3.27	2.04
UDA-VAE++	**68.74**	**85.08**	81.42	**2.34**	**2.61**	**1.71**

Table 4.4: Unsupervised Domain Adaptation for MS-CMRSeg Dataset from **MRI to CT**. The best score for Dice ↑ and ASSD↓ are in **bold**.

Methods	Dice (%)					ASSD (mm)				
	MYO	LA	LV	RA	RV	MYO	LA	LV	RA	RV
NoAdapt	0.08	3.08	0.00	0.74	23.9	–	–	–	–	–
PnP-AdaNet [39]	32.7	49.7	48.4	62.4	44.2	6.89	22.6	9.56	20.7	20.0
SIFA [32]	37.1	65.7	61.2	51.9	18.5	11.8	5.47	16.0	14.7	21.6
UDA-VAE [47]	47.0	63.1	73.8	71.1	73.4	4.73	5.33	4.30	6.97	4.56
UDA-VAE++	**51.4**	**65.9**	**76.5**	**73.0**	**75.5**	**3.88**	**5.23**	**3.78**	**6.25**	**4.06**

Table 4.5: Unsupervised Domain Adaptation for MM-WHS Dataset from **CT to MRI**. The best score for Dice↑ and ASSD↓ are in **bold**.

4.4.2 Datasets

We consider two benchmark datasets for model performance comparison, including Multi-Modality Whole Heart Segmentation (MM-WHS) Challenge dataset [5] and Multi-Sequence Cardiac MR Segmentation (MS-CMRSeg) Challenge dataset [59]

MM-WHS Dataset contains 20 labeled CT images and 20 labeled LGE-MRI images, which are unpaired. Each image is cropped to a size of 240×220.

MS-CMRSeg Dataset contains 35 labeled CT images and 45 labeled LGE-MRI images, which are also not paired. Each image is cropped to a size of 192×192. Similar to [30, 47], we include the following three structures in the given images for segmentation: the myocardial (MYO), the left ventriculus (LV), and the right ventriculus (RV). For both datasets, We remove the MRI ground truth during CT to MRI experiments and remove the CT ground truth during MRI to CT experiments. The train-test split strategy is consistent with [39, 32, 30, 47]

4.4.3 Evaluation Metrics

We use three commonly used evaluation metrics for segmentation, including Dice coefficient (%) and Average Symmetric Surface Distance (ASSD) (mm). The Dice coefficient calculates the agreement between the predicted segmentation and ground truth segmentation by dividing the intersection area by the total pixels in both images. ASSD measures the segmentation accuracy at boundary-level using the Euclidean distance of the closest surface voxels between two segmentations [60]. All metrics are in the format of the mean. A higher Dice and a lower ASSD score indicate better segmentation performances.

4.4.4 Ablation Study

In this subsection, we investigate the contribution of our model components via an ablation study, using the Dice coefficient as the evaluation metric. Specifically, we gradually add individual components and see how the presence of that component will affect the model performances.

Table 4.2 shows the quantitative results of the ablation study. It is shown that most proposed modules will improve the Dice scores. For example, sequential reparameterization, adding Attention, Global, and Local MI estimation increases the Dice score for MYO, LV, and RV. Besides, prior info matching will slightly decrease RV but significantly increase MYO and LV, indicating overall performance improvement.

4.4.5 Qualitative Comparison

Fig. 4.5 shows the visual comparison for segmentation among different models, including CFDNet, UDA-VAE, and the proposed UDA-VAE++. It is shown that the proposed UDA-VAE++ leads to the best structure representation, the best edge preservation, and is the closest to the ground truth. In contrast, CFDNet

and UDA-VAE have a significant segmentation error between MYO, RV, and the background.

Fig. 4.6 displays the visual comparison for reconstruction between different models. Here we only compare UDA-VAE++ with UDA-VAE since UDA-VAE is the only related work that considers image reconstruction. It is shown that the proposed UDA-VAE++ displays significantly better reconstruction than UDA-VAE. UDA-VAE++ has excellent edge preservation, shape representation, and class segmentation. In comparison, UDA-VAE has a significant amount of blurs and artifacts.

4.4.6 Quantitative Comparison

The quantitative comparison utilize several state-of-the-art models, including PnP-AdaNet [39], SIFA [32], UDA-VAE [47], and the proposed UDA-VAE++.

Table 4.3 shows the quantitative comparison for UDA with MS-CMRSeg Dataset (CT to MRI). We can find that the proposed UDA-VAE++ has the best Dice and ASSD score in terms of MYO and LV segmentation. While SIFA has a slight advantage for RV segmentation, it underperforms our model for all other metrics in the table. Therefore, we can conclude that the proposed UDA-VAE++ has the best performance in this experiment.

Table 4.4 shows the quantitative comparison for UDA with MM-WHS Dataset (CT to MRI). We can observe that the proposed UDA-VAE++ has the best Dice and ASSD score in terms of MYO and LV segmentation. Despite SIFA's success in Dice score at RV segmentation, it significantly underperforms our method for all other metrics. Overall, the proposed UDA-VAE++ has the best result in this comparison.

Table 4.5 shows the quantitative comparison for UDA with MS-CMRSeg Dataset (MRI to CT). We can see that the proposed UDA-VAE++ has the best Dice and ASSD score in terms of all segmentations (MYO, LA, LV, RA, RV).

4.5 Conclusion

This paper introduces UDA-VAE++, an unsupervised domain adaptation framework for cardiac segmentation. Through mutual information estimation and maximization, we make the reconstruction and segmentation task mutually beneficial. Moreover, we introduce the sequential reparameterization design, allowing information flow between multi-scale latent space features. Extensive experiments demonstrate that our model achieved state-of-the-art performances on benchmark datasets. Our future work will integrate the proposed mutual information estimation block with self-supervised domain adaptation methods. We also aim to extend our framework to other medical image segmentation tasks (e.g., brain image segmentation).

4.6 Supplementary Material

Proof of Eq.4:

Firstly, We follow the deduction from UDA-VAE[47].

$$
\begin{aligned}
&\log p_{\theta_S}(x,y) \\
=&\int q_{\phi_S}(z \mid x,y)\cdot \\
&\log\left[\frac{q_{\phi_S}(z \mid x,y)}{p_{\theta_S}(z \mid x,y)}\cdot\frac{p_{\theta_S}(z)}{q_{\phi_S}(z \mid x,y)}\cdot p_{\theta_S}(x,y \mid z)\right]dz \\
=&D_{KL}\left(q_{\phi_S}(z \mid x,y)\|p_{\theta_S}(z \mid x,y)\right)- \\
&D_{KL}\left(q_{\phi_S}(z \mid x,y)\|p_{\theta_S}(z)\right)+ \\
&E_{q_{\phi_S}(z|x,y)}\log\left[p_{\theta_S}(x,y \mid z)\right]
\end{aligned}
\tag{4.14}
$$

Note that UDA-VAE[47] neglects the term $D_{KL}\left(q_{\phi_S}(z \mid x,y)\|p_{\theta_S}(z \mid x,y)\right)$ as it is greater than 0.

In comparison, we deduce a compact lower bound with the following term.

$$
\begin{aligned}
&D_{KL}\left(q_{\phi_S}(z \mid x,y)\|p_{\theta_S}(z \mid x,y)\right) \\
&=\int q_{\phi_S}(z \mid x,y)\log\frac{q_{\phi_S}(z \mid x,y)}{p_{\theta_S}(z \mid x,y)}dz \\
&=\int\frac{q_{\phi_S}(x,y,z)}{q_{\phi_S}(x,y)}\log\frac{q_{\phi_S}(x,y,z)}{p_{\theta_S}(x,y,z)}\frac{p_{\theta_S}(x,y)}{q_{\phi_S}(x,y)}dz \\
&=\frac{1}{q_{\phi_S}(x,y)}[\int q_{\phi_S}(x,y,z)\log\frac{q_{\phi_S}(x,y,z)}{p_{\theta_S}(x,y,z)}+q_{\phi_S}(x,y,z)\log\frac{p_{\theta_S}(x,y)}{q_{\phi_S}(x,y)}dz] \\
&=\frac{1}{q_{\phi_S}(x,y)}\int q_{\phi_S}(x,y,z)\log\frac{q_{\phi_S}(x,y,z)}{p_{\theta_S}(x,y,z)}dz+\log\frac{p_{\theta_S}(x,y)}{q_{\phi_S}(x,y)} \\
&=\frac{1}{q_{\phi_S}(x,y)}D_{KL}(q_{\phi_S}(x,y,z)\|p_{\theta_S}(x,y,z))+\log\frac{p_{\theta_S}(x,y)}{q_{\phi_S}(x,y)} \\
&\geq D_{KL}(q_{\phi_S}(x,y,z)\|p_{\theta_S}(x,y,z))+\log\frac{p_{\theta_S}(x,y)}{q_{\phi_S}(x,y)}
\end{aligned}
\tag{4.15}
$$

Consider the reconstruction error[35]:

$$\begin{aligned}\mathcal{R} =& \mathbb{E}_{(x,y,z)\sim q_{\phi_S}(x,y,z)} \log \frac{q_{\phi_S}(x,y,z)}{p_{\theta_S}(x,y,z)} - \\ & \mathbb{E}_{(x,y,z)\sim q_{\phi_S}(x,y,z)} \log q_{\phi_S}(x,y,z) + \mathbb{E}_{z\sim q_{\phi_S}(z)} \log p_{\theta_S}(z)\end{aligned} \tag{4.16}$$

The second term is the joint entropy $H_q(x, y, z)$.

The third term can be written as:

$$\mathbb{E}_{z\sim q_{\phi_S}(z)} \log p_{\theta_S}(z) = -D_{KL}(q_{\phi_S(z)} \| p_{\theta_S}) - H_{q_{\phi_S}}(z) \tag{4.17}$$

With

$$H_{q_{\phi_S(z)}}(x, y, z) - H_{q_{\phi_S}}(z) = H_{q_{\phi_S}}(z) - I_{q_{\phi_S}}(x, y, z) \tag{4.18}$$

where I is mutual information.

The reconstruction error can be written as:

$$\mathcal{R} \leq D_{KL}(q_{\phi_S(x,y,z)} \| p_{\theta_S(x,y,z)}) - I_{q_{\phi_S}}(x, y, z) + H_{q_{\phi_S}}(z) \tag{4.19}$$

which is compact when $q_{\phi_S(z)}$ matches the prior distribution $p_{\theta_S}(z)$.

$$D_{KL}(q_{\phi_S(x,y,z)} \| p_{\theta_S(x,y,z)}) \geq \mathcal{R} + I_{q_{\phi_S}}(x, y, z) - H_{q_{\phi_S}}(z) \tag{4.20}$$

Thus, we obtain the bound,

$$\begin{aligned}& D_{KL}\left(q_{\phi_S}(z \mid x, y) \| p_{\theta_S}(z \mid x, y)\right) \\ & \geq D_{KL}(q_{\phi_S}(x, y, z) \| p_{\theta_S}(x, y, z)) + \log \frac{p_{\theta_S}(x, y)}{q_{\phi_S}(x, y)} \\ & \geq \mathcal{R} + I_{q_{\phi_S}}(x, y, z) - H_{q_{\phi_S}}(z) + \log \frac{p_{\theta_S}(x, y)}{q_{\phi_S}(x, y)}\end{aligned} \tag{4.21}$$

From, Eq.4.14 and Eq.4.21,

$$
\begin{aligned}
&\log p_{\theta_S}(x, y) \\
\geq &(\mathcal{R} + I_{q_{\phi_S}}(x, y, z) - H_{q_{\phi_S}}(z) + \log \frac{p_{\theta_S}(x, y)}{q_{\phi_S}(x, y)}) - \\
&D_{KL}\left(q_{\phi_S}(z \mid x) \| p_{\theta_S}(z)\right) + E_{q_{\phi_S}(z|x)} \log p_{\theta_S}(x, y \mid z) \\
= &(\mathcal{R} + I_{q_{\phi_S}}(x, y, z) - H_{q_{\phi_S}}(z) + \log \frac{p_{\theta_S}(x, y)}{q_{\phi_S}(x, y)}) - \\
&D_{KL}\left(q_{\phi_S}(z \mid x) \| p_{\theta_S}(z)\right) + E_{q_{\phi_S}(z|x)} \log p_{\theta_S}(x \mid y, z) \\
&+ E_{q_{\phi_S}(z|x)} \log p_{\theta_S}(y \mid z)
\end{aligned}
\tag{4.22}
$$

where R, $\log \frac{p_{\theta_S}(x,y)}{q_{\phi_S}(x,y)}$ and $H_{q_{\phi_S}}(z)$ are constant. The equation holds, as $p_{\theta_S}(x, y \mid z) = p_{\theta_S}(y \mid z) \cdot p_{\theta_S}(x \mid y, z)$. Meanwhile, y_S and z_s are conditionally independent on x_S for distribution q_{ϕ_S}, so that $q_{\phi_S}(z \mid x, y) = q_{\phi_S}(z \mid x)$.

Finally, We get the compact lower bound (plus red terms) than UDA-VAE .

The UDA-VAE++ maximizes the mutual information of $I_{q_{\phi_S}}(x, y, z)$.

Proved.

Bibliography

[1] https://slideplayer.com/slide/14408055/.

[2] https://www.who.int/cardiovascular_diseases/about_cvd/en/.

[3] C. W. Tsao, A. W. Aday, Z. I. Almarzooq, A. Alonso, A. Z. Beaton, M. S. Bittencourt, A. K. Boehme, A. E. Buxton, A. P. Carson, Y. Commodore-Mensah *et al.*, "Heart disease and stroke statistics—2022 update: A report from the american heart association," *Circulation*, vol. 145, no. 8, pp. e153–e639, 2022.

[4] F. Ritter, T. Boskamp, A. Homeyer, H. Laue, M. Schwier, F. Link, and H.-O. Peitgen, "Medical image analysis," *IEEE Pulse*, vol. 2, no. 6, pp. 60–70, 2011.

[5] X. Zhuang and J. Shen, "Multi-scale patch and multi-modality atlases for whole heart segmentation of mri," *Medical image analysis*, vol. 31, pp. 77–87, 2016.

[6] D. C. Bloomgarden, Z. A. Fayad, V. A. Ferrari, B. Chin, M. G. St. John Sutton, and L. Axel, "Global cardiac function using fast breath-hold mri: validation of new acquisition and analysis techniques," *Magnetic resonance in medicine*, vol. 37, no. 5, pp. 683–692, 1997.

[7] A. Gupta, L. Von Kurowski, A. Singh, D. Geiger, C.-C. Liang, M.-Y. Chiu, L. Adler, M. Haacke, and D. Wilson, "Cardiac mr image segmentation using deformable models," in *Proceedings of Computers in Cardiology Conference*. IEEE, 1993, pp. 747–750.

[8] H. R. Singleton and G. M. Pohost, "Automatic cardiac mr image segmentation using edge detection by tissue classification in pixel neighborhoods," *Magnetic resonance in medicine*, vol. 37, no. 3, pp. 418–424, 1997.

[9] X. Zhuang, K. S. Rhode, R. S. Razavi, D. J. Hawkes, and S. Ourselin, "A registration-based propagation framework for automatic whole heart segmentation of cardiac mri," *IEEE transactions on medical imaging*, vol. 29, no. 9, pp. 1612–1625, 2010.

[10] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Annual review of biomedical engineering*, vol. 2, no. 1, pp. 315–337, 2000.

[11] D. Kang, J. Woo, C. J. Kuo, P. J. Slomka, D. Dey, and G. Germano, "Heart chambers and whole heart segmentation techniques," *Journal of Electronic Imaging*, vol. 21, no. 1, p. 010901, 2012.

[12] X. Zhuang, "Challenges and methodologies of fully automatic whole heart segmentation: a review," *Journal of healthcare engineering*, vol. 4, no. 3, pp. 371–407, 2013.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[18] http://atriaseg2018.cardiacatlas.org/.

[19] Q. Xia, Y. Yao, Z. Hu, and A. Hao, "Automatic 3d atrial segmentation from ge-mris using volumetric fully convolutional networks," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2018, pp. 211–220.

[20] C. Chen, W. Bai, and D. Rueckert, "Multi-task learning for left atrial segmentation on ge-mri," in *International workshop on statistical atlases and computational models of the heart*. Springer, 2018, pp. 292–301.

[21] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2018, pp. 3–11.

[22] D. Jha, P. H. Smedsrud, M. A. Riegler, D. Johansen, T. De Lange, P. Halvorsen, and H. D. Johansen, "Resunet++: An advanced architecture for medical image segmentation," in *2019 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2019, pp. 225–2255.

[23] F. Isensee, J. Petersen, S. A. Kohl, P. F. Jäger, and K. H. Maier-Hein, "nnu-net: Breaking the spell on successful medical image segmentation," *arXiv preprint arXiv:1904.08128*, vol. 1, pp. 1–8, 2019.

[24] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation," *arXiv preprint arXiv:1802.06955*, 2018.

[25] D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, and H. D. Johansen, "Doubleu-net: A deep convolutional neural network for medical image segmentation," in *2020 IEEE 33rd International symposium on computer-based medical systems (CBMS)*. IEEE, 2020, pp. 558–564.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[27] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[28] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," *arXiv preprint arXiv:2105.05537*, 2021.

[29] E. Tzeng, J. Hoffman, N. Zhang *et al.*, "Deep domain confusion: Maximizing for domain invariance [eb/ol]," *arXiv Preprint*, 2019.

[30] F. Wu and X. Zhuang, "Cf distance: A new domain discrepancy metric and application to explicit domain adaptation for cross-modality cardiac image segmentation," *IEEE Transactions on Medical Imaging*, vol. 39, no. 12, pp. 4274–4285, 2020.

[31] Q. Dou, C. Ouyang, C. Chen, H. Chen, B. Glocker, X. Zhuang, and P.-A. Heng, "Pnp-adanet: Plug-and-play adversarial domain adaptation network at unpaired cross-modality cardiac segmentation," *IEEE Access*, vol. 7, pp. 99 065–99 076, 2019.

[32] C. Chen, Q. Dou, H. Chen, J. Qin, and P. A. Heng, "Unsupervised bidirectional cross-modality adaptation via deeply synergistic image and feature alignment for medical image segmentation," *IEEE transactions on medical imaging*, vol. 39, no. 7, pp. 2494–2505, 2020.

[33] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[34] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[35] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *International conference on machine learning*. PMLR, 2018, pp. 531–540.

[36] M. D. Donsker and S. S. Varadhan, "Asymptotic evaluation of certain markov process expectations for large time, i," *Communications on Pure and Applied Mathematics*, vol. 28, no. 1, pp. 1–47, 1975.

[37] A. Keziou, "Dual representation of φ-divergences and applications," *Comptes rendus mathématique*, vol. 336, no. 10, pp. 857–862, 2003.

[38] Q. Dou, C. Ouyang, C. Chen, H. Chen, and P.-A. Heng, "Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss," *arXiv preprint arXiv:1804.10916*, 2018.

[39] Q. Dou, C. Ouyang, C. Chen, H. Chen, B. Glocker, X. Zhuang, and P.-A. Heng, "Pnp-adanet: Plug-and-play adversarial domain adaptation network at unpaired cross-modality cardiac segmentation," *IEEE Access*, vol. 7, pp. 99 065–99 076, 2019.

[40] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair,

A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[41] Z. Zhang, L. Yang, and Y. Zheng, "Translating and segmenting multimodal medical volumes with cycle-and shape-consistency generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern Recognition*, 2018, pp. 9242–9251.

[42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[43] Y. Liu and X. Du, "Duda: Deep unsupervised domain adaptation learning for multi-sequence cardiac mr image segmentation," in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, 2020, pp. 503–515.

[44] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[45] C. Ouyang, K. Kamnitsas, C. Biffi, J. Duan, and D. Rueckert, "Data efficient unsupervised domain adaptation for cross-modality image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 669–677.

[46] M. Gu, S. Vesal, R. Kosti, and A. Maier, "Few-shot unsupervised domain adaptation for multi-modal cardiac image segmentation," *arXiv preprint arXiv:2201.12386*, 2022.

[47] F. Wu and X. Zhuang, "Unsupervised domain adaptation with variational approximation for cardiac segmentation," *IEEE Transactions on Medical Imaging*, vol. 40, no. 12, pp. 3555–3567, 2021.

[48] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," *arXiv preprint arXiv:1808.06670*, 2018.

[49] Q. Chen and Y. Liu, "Structure-aware feature fusion for unsupervised domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 10 567–10 574.

[50] J. Ting, K. Punithakumar, and N. Ray, "Multiview 3-d echocardiography image fusion with mutual information neural estimation," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 765–771.

[51] G. Snaauw, M. Sasdelli, G. Maicas, S. Lau, J. Verjans, M. Jenkinson, and G. Carneiro, "Mutual information neural estimation for unsupervised multimodal registration of brain images," *arXiv preprint arXiv:2201.10305*, 2022.

[52] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[53] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *arXiv preprint arXiv:1906.02691*, 2019.

[54] M. D. Donsker and S. S. Varadhan, "Asymptotic evaluation of certain markov process expectations for large time, i," *Communications on Pure and Applied Mathematics*, vol. 28, no. 1, pp. 1–47, 1975.

[55] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[57] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[58] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[59] X. Zhuang, "Multivariate mixture model for myocardial segmentation combining multi-source images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 12, pp. 2933–2946, 2018.

[60] T. Heimann, B. Van Ginneken, M. A. Styner, Y. Arzhaeva, V. Aurich, C. Bauer, A. Beck, C. Becker, R. Beichel, G. Bekes *et al.*, "Comparison and evaluation of methods for liver segmentation from ct datasets," *IEEE transactions on medical imaging*, vol. 28, no. 8, pp. 1251–1265, 2009.