Multimedia Tools and Applications (2024) 83:73507–73532
https://doi.org/10.1007/s11042-023-18095-7

1232: HUMAN-CENTRIC MULTIMEDIA ANALYSIS

# A class-aware multi-stage UDA framework for prostate zonal segmentation

**Zibo Ma[1] · Yue Mi[2,3,4] · Bo Zhang[1] · Zheng Zhang[5] · Yu Bai[1] · Jingyun Wu[6] · Haiwen Huang[2,3,4] · Wendong Wang[1]**

Received: 14 June 2023 / Revised: 20 December 2023 / Accepted: 29 December 2023 /
Published online: 18 January 2024


**Abstract**
Unsupervised domain adaptation (UDA) aims to solve the lack of annotation in a new dataset which has non-independent identity distribution compare with training data. It has the potential to help the annotation process in medical image segmentation. Existing self-training based UDA approaches utilize the pseudo labels as ground truth labels for domain adaptation, whereas the generated pseudo labels inevitably introducing the noise when training the model for the target domain, which make the training process unstable and the model is difficult to converge. In the meanwhile, most of the methods ignore the class imbalanced problem. To tackle the issue, we propose a class-aware multi-stage unsupervised domain adaptation framework for prostate zonal segmentation task. We devise a class-specific knowledge guidance strategy for training a better pseudo labels generation model. Extensive experimental results show the effectiveness of our approach against existing state-of-the-art approaches on the UDA problem of prostate zonal segmentation benchmark.

**Keywords** Unsupervised domain adaptation · Prostate zonal segmentation · Meta-learning

Zibo Ma and Yue Mi contributed equally to this work.

✉ Bo Zhang
zbo@bupt.edu.cn

✉ Wendong Wang
wdwang@bupt.edu.cn

1 State Key Laboratory of Networking and Switching Technology, School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing, China

2 Department of Urology, Peking University First Hospital, Beijing, China

3 Institute of Urology, Beijing University, Beijing, China

4 National Urological Cancer Center of China, Beijing, China

5 School of Modern Post, Beijing University of Posts and Telecommunications, Beijing, China

6 Department of Radiology, Peking University First Hospital, Beijing, China

## 1 Introduction

Human-centric multimedia analysis has a wide range of applications, including human re-identification, human activity analysis, and human body pattern analysis. This is an important topic in the field of multimedia tools and applications. The analysis of human-body is highly related to the evaluation of health state, while health issues are of particular concern in today's society. With the advance of technology, Non-invasive computer-aided diagnostic (CAD) techniques have become one of the most important tools for assessing human health. Magnetic resonance imaging (MRI) is one of non-invasive and radiation-free CAD technique, which has been widely used in organ and soft tissues imaging, including various human-body part such as head, neck, chest, and abdomen. In this paper, we are going to conduct a study on the application of multimedia analysis technology to Prostate cancer (PCa). PCa is a common abdominal male disease, which has a serious impact on male's life expectancy and quality of life. To make diagnosis and treatment planning of PCa, doctors demand precise and accurate identification of tumors and surrounding tissues. Therefore, prior to executing PCa clinical diagnostic tasks, it is essential to segment the subdivision of prostate from MRIs. However, the correct contour segmentation of the anatomical structures is time-consuming, that demands proficiency in healthcare. With the success of deep learning in medical image segmentation tasks, deep learning models achieve even surpasses human experts in some applications, such as prostate whole gland segmentation, where one of the important reason is a large number of annotated public datasets [1, 2]. However, diagnosis of PCa requires segmentation of prostate substructures, the lack of refined zonal annotation datasets of the prostate with its surrounding tissues, limits the development of deep learning based PCa diagnosis. As shown in Table 1, there are more than six MRI public datasets for prostate and PCa segmentation tasks, with nearly two hundred cases of multi-center, multi-parametric T2 MRI, whereas only two datasets for prostate substructure (peripheral zone, central zone/transition zone) segmentation. Moreover, only one dataset for prostate substructure and peripheral tissue segmentation studies, which just has a few annotations.

Since medical image segmentation task requires pixel-level annotation, it is time-consuming and labor-intensive, which also requires strong anatomical knowledge, making it impossible to generate large-scale annotations through crowdsourcing as the natural images domain. Meanwhile, data heterogeneity prevents us from effectively using the existing annotation data to train a model that is applicable to the new central data. This is because the distribution drift caused by data heterogeneity, which makes the performance of the model trained on existing annotation data by traditional supervised learning degrade significantly when directly applied it to data from other centers. The problem of data heterogeneity often exists in medical image due to its multi-centric nature, the non-identity distribution (non-IID) as well as Out-of-Distribution (OOD) [3] problems arising from data collected from different medical institutions due to differences in collection devices, physician practices, and individual cases. Figure 1(a) shown the 2D slice sample from T2 prostate MRIs from two publicable datasets, we analyze their intensity distribution by Kernel Density Estimation(KDE) plot. According to the KDE plot, we can see the obvious difference of intensity distribution between two datasets, which has been defined as data heterogeneity in [4].

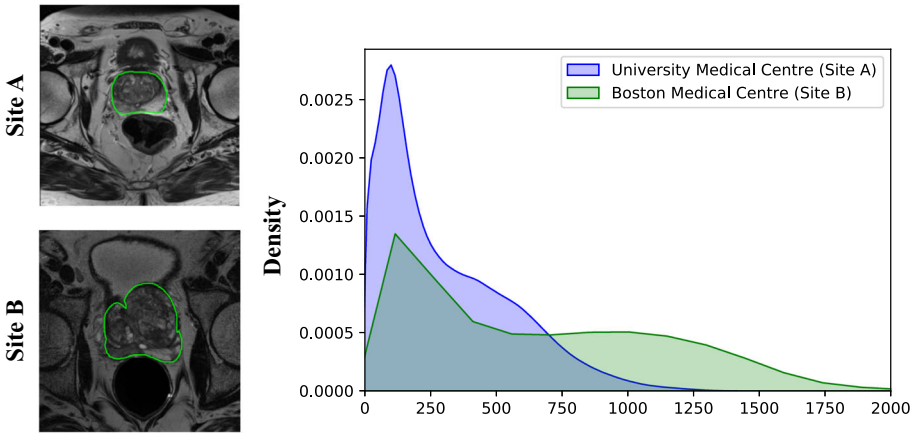
Unsupervised domain adaptation (UDA) aims to adapt a model on unlabeled target domain data by transferring knowledge from labeled source domain data, provides a promising way that can fast adapt on a bunch of new datasets which is non-IID compare with the labeled data, without any manual annotations. The study of the non-IID problem for semantic segmentation tasks has a wide range of application scenarios such as autonomous driving [5] and the

**Table 1** Publicly available datasets for prostate segmentation

| Dataset | MR Modality | Organ / Lesion | Case num |
|---|---|---|---|
| PROMISE12 | T2 | WP | 50 |
| Decathlon | T2, ADC | WP, TZ, PZ | 32 |
| NCI-ISBI13 | T2 | WP, TZ, PZ | 80 |
| I2CVB | T2, DWI | WP, Lesion | 19 |
| Prostate-X | T2, DWI, DCE | Lesion | 182 |
| Prostate-Dianosis | T1, T2, DWI | WP, TZ, PZ, SV, NVB around 11 organs | 5 |

WP-Whole Prostate, TZ-Transition Zone, PZ-Peripheral Zone, NVB-Neurovascular Bundle, SV-Seminal Vesiclexzio

annotation of medical image data [6]. There are two major solutions for the non-IID problem in semantic segmentation: one is adversarial learning based methods and the other is self-training based methods. Adversarial learning based methods approximate the distribution of source and target domain data by implicit feature alignment in the input space or output space, Although the adversarial based methods can learn domain invariant features which are also discriminative for the source domain, while the separability of target samples is always being ignored since the conditional distributions are not explicitly aligned. while. Self-training based methods change the learning of target domain data to supervised learning by generating pseudo labels for the target domain, whereas generated pseudo labels often along with noise, which makes the model training seriously biased (i.e. overfitting on noisy data). Although existing works reduce the noise of pseudo labels either by epistemic uncertainty qualification [7] or by loss correction [5], both of them ignore the class imbalanced problem, and the other thing is that jointly optimize the pseudo generation task and self-training with noisy labels correction task may made the training process unstable. Considering the




**(a) 2D slice example of source and target domain**

**(b) KDE Plot of source and target domain**

**Fig. 1** Visualization of 2D slice example from two prostate T2 MRI Datasets and intensity distribution by KDE plot

problems mentioned above, we propose a class-aware multi-stage UDA framework, which first reduces the gap between source and target domains by an unsupervised image translation model without additonal registration step, and then trains a pseudo labels generation model for the target domain by an adversarial learning based approach, in which multi-level output space adaptation and domain-specific parallel adapters are introduced to enhance domain-specific features; and we devise a class-specific knowledge guidance strategy for solving the foreground class imbalanced problem. In second stage, We retrain the segmentation model by using the target domain pseudo label generated by the model trained in the first stage, a meta-learning based loss correction strategy is introduced to correct the pseudo labels during the training process, and they are used in the training of the new target domain model. Our contributions including:

- We propose a class-aware multi-stage unsupervised domain adaptation (UDA) framework incorporated with input space, output space adaptation, and self-training with noisy labels method, which is able to solve the non-IID problem between train and test data better.
- We design a Class-Specific Knowledge Guidance (CSKG) strategy to solve the class imbalanced problem in foreground classes, and introduce a Domain-Specific Parallel Adapter (DSPA) module to retain the domain discriminative information with very few of parameters non-shared.
- We conduct extensive experiments and ablation studies on the benchmark datasets for prostate multi-zonal segmentation tasks. The results show that our approach has better performance than the state-of-the-art methods.

## 2 Related work

### 2.1 UDA in semantic segmentation

**Adversarial learning based method** Some of methods put image translation as a part of the method, in a tandem way [8] or end-to-end fashion [6]. Concretely, it is an indirect way, which need translate from source domain to target domain, so that the labels from source domain can be used to train segmentation model. While another methods directly use adversarial learning in models training: One network behaves as a generator to obtain the segmentation maps for source and target inputs, the other network serves as a discriminator to derive domain predictions. The generator intends to fool the discriminator to ensure the cross-domain alignment of feature level [9] or output level [10, 11]. However, all of them above retain a shared parameter network, ignored the domain-specific information, weaken the domain discriminative ability [12].

**Self-training based method** The self-training based method entails using highly confident network predictions inferred on unlabeled data to generate pseudo-labels, then use these labels to reinforce the training of the target domain network with the self-taught supervision. While there are two problems in the process of pseudo labels generating and using: the design of filtering rules for getting the high confident predictions to be pseudo labels, and learn with noisy labels.

Some methods [7, 13] rely on various forms of pseudo-label filtering, [13] proposed to threshold the argmax values of predictions and selected high-confidence pseudo-labeled samples. Zheng and Yang [7] utilized uncertainty estimation and enabled the dynamic threshold to obtain rectified pseudo labels. However, these methods only involved confident samples for training, which may result in biased prediction in minor classes and cannot distinguish confused categories. An alternative way is learn with noisy labels, by adjusting the loss

of all training samples before updating model per iteration. It can be categorized into three strategies: i) *loss correction* [14–17] that correct loss in forward or backward process through construct noise transition matrix; ii) *loss reweighting* [16, 18] that imposes every samples have different importance; iii) *label rectify* [19] that adjusts the loss using the rectified label obtained from a convex combination of noisy and predicted labels.

### 2.2 Class imbalanced problem in semantic segmentation

The class imbalance problem not just exist between background and foreground, but also between inter-class of foreground. Some of works [20, 21] focus on the architecture modification of segmentation network. Gao et al. [20] considering the inter-class imbalance problem in foreground, they devise a framework that introduce an auxiliary branches to localize and segment the small organ, using the heat map of small-organ center location to train , improving the segmentation performance on small organ. Feng et al. [21] combining two pyramidal modules to dynamically fuse multi-scale context information. While other works [22, 23] focus on the loss optimization, [22] introduce a class-weighting strategy to weight the vote of each class via condition the weights of loss, and a equally patches extraction strategy for multi-class brain tumor segmentation. Sugino et al. [24] make a comprehensive comparison of five loss weighting strategies and select the optimal one for multi-class brainstem structure segmentation. Yeung et al. [23] define a new hierarchical framework to encompass various Dice and cross entropy-based loss functions, and used this to derive the Unified Focal loss, which is associated with a better recall-precision balance.

Different from the methods mentioned above, [25] devise a 'X' shaped network, which consist of two U-Net architecture, distribution based loss for one U-Net, and region based loss for another U-Net, then the logits of these two sub-network was concatenated to make the final output, it was validated on the cell segmentation task, shown the effectiveness of solving class imbalanced problem about inter-class of foreground. In this paper, we tackle class imbalance problem with an curriculum learning strategy, distill the class specific knowledge to further guide the main segmentation network training.

## 3 Overview of framework

### 3.1 Problem formulation

We focus on the non-IID problem in prostate multi-zonal segmentation tasks. In the source domain, there is a set of images $X_S = \left\{x_s \in \mathbb{R}^{H\times W\times 3}\right\}_{s\in S}$ and the corresponding pixel-wise one-hot labels $Y_S = \left\{y_s \in \{0,1\}^{H\times W\times C}\right\}_{s\in S}$. While in the target domain, only images $X_T = \left\{x_t \in \mathbb{R}^{H\times W\times 3}\right\}_{t\in T}$ are available. The goal is to train a model that can correctly categorize pixels for target data $X_T$. Note that $H, W, C$ denote the height, width, and categories of images respectively.

We propose a framework concurrently learn from labeled source data and unlabeled target data, which can generate more comparable pseudo labels for target data, then use them as ground truth labels for training the target segmentation model. Specifically, We first utilize an input space adaptation module to translate the source domain images to the target domain for mitigating data heterogeneity. After image translation, our framework goes through two training stages, as shown in Fig. 2:

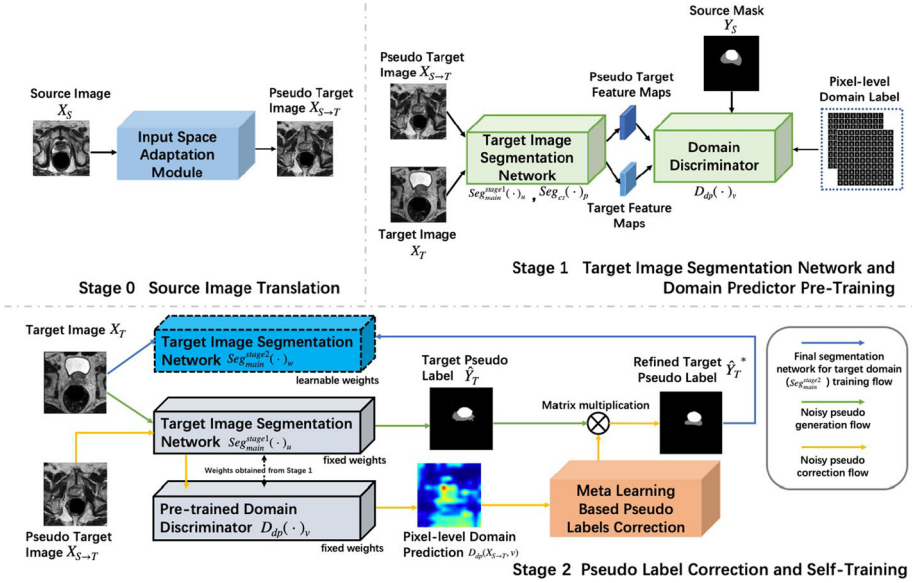



**Fig. 2** Overview of Our Framework. Firstly, we employ an image translation model (in light blue) to transform source domain images into ones approximating the style of the target domain in 'Stage 0'. Subsequently, in 'Stage 1', we conduct adversarial training using labeled source domain data and unlabeled target domain data to train a target domain image segmentation model (in green) and a domain discriminator (in green). Lastly, in 'Stage 2', we utilize the segmentation model (in gray) trained in 'Stage 1' to generate pseudo segmentation labels for target domain data. We employ a domain discriminator (in gray) to filter pixel-level domain features, which is used for correcting pseudo labels. Finally, we retrain a target domain image segmentation model (in teal) using the corrected pseudo labels

**Stage 1:** We concurrently train a model for generating preliminary pseudo labels for the target domain and a domain predictor that could select target-like pixels from source images for each class, which will be used in stage 2.

**Stage 2:** Based on the noisy labels correction, we correct pseudo labels and then use it to retrain the segmentation network for the target domain.

### 3.2 Image translation from source to target domain

Considering the source and target domains are different, which have multiple acquisition protocols, resulting in data heterogeneity, we introduce an input space adaptation module based on MUNIT [26], which is an indeterministic image translation method and was applied to translate MRI between different modalities without pairing [6]. As illustrated in Fig. 3, the input space adaptation module consists of *dual domain disentangled reconstruction (DDDR)* and *source-based pseudo target image distribution matching (SPTIDM)* processes. For convenience, we only describe the process of translating source image to target image. The training objective is given by:

$$\min_{E^c_S,E^s_S,E^c_T,E^s_T,G_S,G_T} \max_{D_S,D_T} \lambda_{\text{GAN}} \left(\mathcal{L}^S_{\text{GAN}} + \mathcal{L}^T_{\text{GAN}}\right) + \lambda_x \left(\mathcal{L}^S_{\text{recon}} + \mathcal{L}^T_{\text{recon}}\right) + \lambda_c \left(\mathcal{L}^{cS}_{\text{recon}} + \mathcal{L}^{cT}_{\text{recon}}\right) + \lambda_s \left(\mathcal{L}^{sS}_{\text{recon}} + \mathcal{L}^{sT}_{\text{recon}}\right) \tag{1}$$

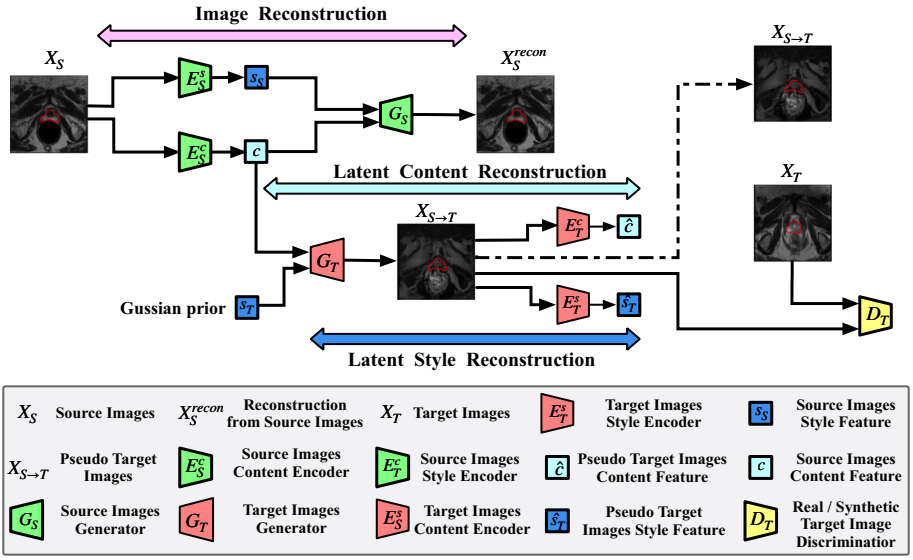



**Fig. 3** Illustration of input space adaptation module. Considering the domain gap between the source and target domain, we first utilize an unpaired image translation network which can translate the source image to target domain for mitigating the domain shift problem

where $\lambda_{\text{GAN}}$ , $\lambda_x$, $\lambda_c$, $\lambda_s$, $\lambda_{cyc}$ are are the weighting factors to balance corresponding loss terms for image translation.

**Dual Domain Disentangled Reconstruction (DDDR)** There are two parallel ways in reconstruction process: i) image reconstruction, which follows the *"image-latent-image"* loop; ii) latent reconstruction, which follows the *"latent-image-latent"* loop. Note that in i), the images from source domain was disentangled into content space $\mathcal{C}$ which is domain-invariant and style space $\mathcal{S}$ which is domain-specific, then the decoder of the source domain $G_S$ reconstruct the source image. In ii), the content feature $c$ of the source image and the style feature $s$ of the target domain were fed into the target domain decoder $G_T$ to generate images that conform the marginal distribution of the target domain. The style feature of target domain $s_T$ is randomly sampled from Gaussian prior $q\ (s_T) \sim \mathcal{N}(0, \mathbf{I})$. Finally the synthetic image $x_{S\to T}$ was encoded by style encoder $E_T^s$ and content encoder $E_T^c$ of the target domain again. The object functions in *DDDR* including $\mathcal{L}_{\text{recon}}^{x_S}$ for image reconstruction:

$$\mathcal{L}_{\text{recon}}^{x_S}\left(E_S^c, E_S^s, G_S, x_S\right) = \mathbb{E}_{x_s\sim X_S}[\|G_S(E_S^c(x_S), E_S^s(x_S)) - x_S\|_1] \tag{2}$$

and $\mathcal{L}_{\text{recon}}^{c_S}$ , $\mathcal{L}_{\text{recon}}^{s_T}$ for latent representation reconstruction:

$$\mathcal{L}_{\text{recon}}^{c_S}\left(E_S^c, G_T, E_T^c, x_S, s_T\right) = \mathbb{E}_{x_S\sim X_S, s_T\sim S_T}[\|E_T^c(G_T(E_S^c(x_S), s_T)) - E_S^c(x_S)\|_1] \tag{3}$$

$$\mathcal{L}_{\text{recon}}^{s_T}\left(E_S^c, G_T, E_T^c, x_S, s_T\right) = \mathbb{E}_{x_S\sim X_S, s_T\sim S_T}[\|E_T^s(G_T(E_S^c(x_S), s_T)) - s_T\|_1] \tag{4}$$

**Source-Based Pseudo Target Image Distribution Matching (SPTIDM)** The distribution of synthetic images was matched to the target domain distribution via adversarial training: $D_T$ discriminates whether the image is real from the target domain or was synthesized from the source domain, and the target domain decoder $G_T$ trying to synthesize images that are

indistinguishable to the discriminator. The training objective is defined as:

$$\begin{aligned}\mathcal{L}_{\mathrm{GAN}}^{x_T}(E_S^c, G_T, D_T, x_S, s_T) &= \mathbb{E}_{x_T \sim X_T}[\log D_T(x_T)] \\ &+ \mathbb{E}_{x_S \sim X_S, s_T \sim S_T}[\log(1 - D_T(G_T(E_S^c(x_S), s_T)))]\end{aligned} \tag{5}$$

### 3.3 Stage 1: target domain pseudo labeling model and pixel-level domain predictor pre-training

For the purpose of generating more credible pseudo labels for the target domain, we introduce an output space adaptation module [11] to align the source and target domain in output space, which has richer semantic information and smaller domain gap. There are two components in stage 1: i) *Segmentation Network* and ii) *Pixel-level Domain Discriminator*. The sketch of stage 1 was shown in Fig. 2 top right part.

The pseudo target images with corresponding ground-truth labels and unlabeled real target images were both used for training. We use an adversarial way to train i) and ii). The training objective of stage 1 is:

$$\min_{Seg_{main-enc}^{stage1}, Seg_{main-dec}^{stage1}, Seg_{cs}} \max_{D_{dp}} \mathcal{L}_{seg} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_d \mathcal{L}_d \tag{6}$$

where $\lambda_{adv}$, $\lambda_d$ are the weighting factors to balance two loss terms for adversarial training. $\mathcal{L}_{adv}$ is the adversarial loss that adapts predicted segmentation masks of target images to the distribution of source predictions, $\mathcal{L}_d$ is the discriminative loss that discriminates the predicted outputs(masks) belong to source domain or target domain, and $\mathcal{L}_{seg}$ is the total loss that makes supervision for segmentation network training:

$$\mathcal{L}_{seg} = \mathcal{L}_{main-seg} + \mathcal{L}_{cs-seg} = (\lambda_{ce}\mathcal{L}_{ce} + \lambda_{cskg}\sum_{i=1}^{I}\mathcal{L}_{cskg}^i) + \lambda_{cs-seg}\sum_{i=1}^{I}\mathcal{L}_{cs-seg}^i \tag{7}$$

where $\lambda_{ce}, \lambda_{cskg}, \lambda_{cs-seg}$ are the weighting factors to balance three loss terms for segmentation network training, and $\lambda_{ce} + \lambda_{cskg} = 1$ for the CSKG strategy. $\mathcal{L}_{ce}$ is the cross entropy loss for learning the class-invariant representations of data, $\mathcal{L}_{cs-seg}^i$ is the dice loss, and $\mathcal{L}_{cskg}^i$ is the class-specific knowledge guidance loss for learning the class-specific knowledge from class-specific decoders indirectly.

Specifically, $Seg_{main-enc}^{stage1}$, $Seg_{main-dec}^{stage1}$, and $Seg_{cs-dec}$ denotes main encoder, main decoder, and class-specific decoders of the segmentation network respectively, $D_{dp}$ denotes a domain predictor. In each iteration, a batch of the pseudo target images $x_{S\to T}$ with its ground-truth labels $y_S$ was forwarded for optimizing the combination of $Seg_{main-enc}^{stage1}$ and $Seg_{cs-dec}$ (i.e. $Seg_{cs}$), and the combination of $Seg_{main-enc}^{stage1}$ and $Seg_{main-dec}^{stage1}$ (i.e. $Seg_{main}^{stage1}$), alternately. Then $Seg_{main}^{stage1}$ predict the softmax output $P_T$ for target image $y_T$, the predictions $P$ of source and target domain (i.e., $P_S$ and $P_T$) was fed to $D_{dp}$ to discriminate the input whether from source or target domain pixel by pixel, which encourage $P_S$ and $P_T$ close to each other. With the adversarial training, the gradients was propagated from $D_{dp}$ to $Seg_{main}^{stage1}$, which would encourage $Seg_{main}^{stage1}$ to predict similar prediction between the target domain and the source domain.

**Class-Specific Knowledge Guidance (CSKG)** The class-imbalanced problem in multi-class semantic segmentation always make model to focus on the majority class which has a big ratio whereas ignore the minority class, so it is necessary to extract robust representations for

domain transfer and class-specific representations to better identify minority foreground class concurrently. The original intention of curriculum learning aims to learn a small network to meet the low-memory and fast execution requirements, it starts with the big teacher network which is deeper and wider, then trains a smaller network to mimic teacher [27, 28]. Learning to mimic teacher turns out to be much easier than directly learning from ground truth, some of them mimic the teacher's class probabilities [27] or feature representation [28], including richer information beyond the traditional supervised learning.

The studies mentioned above inspire us to design a curriculum learning based paradigm for mitigating the class imbalanced problem in semantic segmentation, i.e. CSKG, to enhance the domain-specific feature learning of weight-shared CNN kernels in our segmentation network, especially for minor foreground class. As shown in Fig. 4(a-i), we synergistically train the main segmentation network $Seg_{main}^{stage1}$ with the supervision by ground truth labels and class-specific knowledge learning from a pair of class-specific decoders $Seg_{cs}$, a class-specific decoder $Seg_{cs}^{i}$ for category $i$ has the same architecture as the main segmentation network decoder $Seg_{main-dec}^{stage1}$. The class specific decoders serve as an independent logits predictor for each category, which could learn the class-specific knowledge compared with the main segmentation network $Seg_{main}^{stage1}$. As we know, dice loss [29] is the region metric

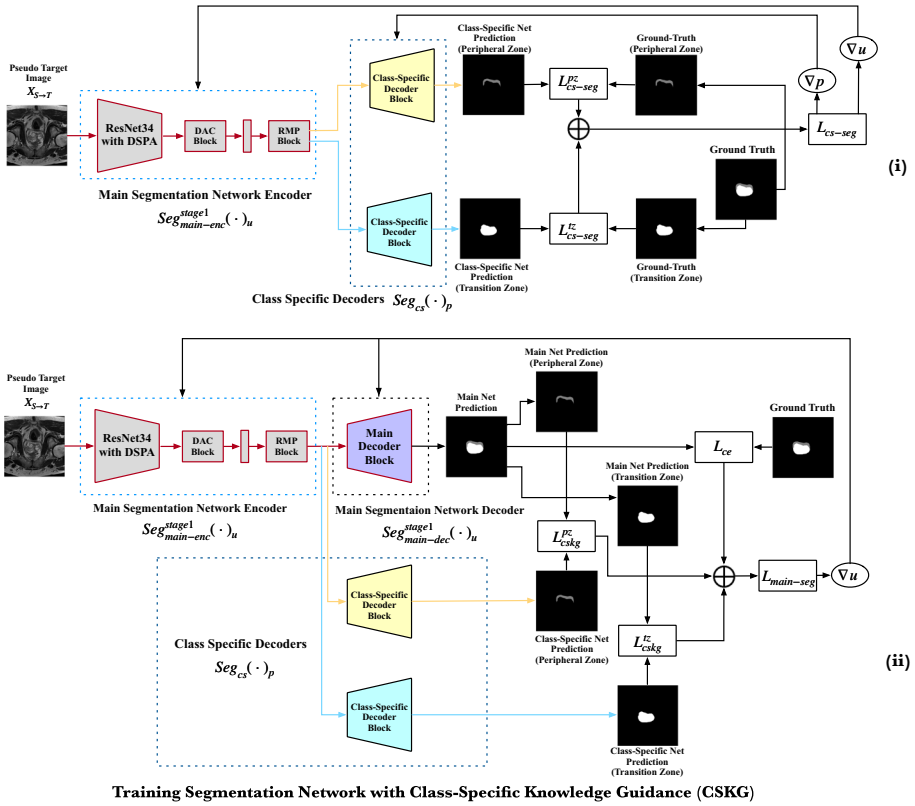



**Fig. 4** Illustration of segmentation network training procedure with in GSKG strategy stage 1. Note that the line in red denote the DSPA module for processing pseudo target images, the line in green denote the DSPA module for processing real target images

based loss that is aware to the region, so we trained each $Seg^{i}_{cs}$ with dice loss for distilling the class-specific information.

The training process in alternating in each iteration, instead of transferring the class-specific knowledge from $Seg^{i}_{cs}$ sequentially into $Seg^{stage1}_{main}$, we conduct knowledge transfer from all class-specific decoders $Seg_{cs}$ into the $Seg^{stage1}_{main-enc}$, encouraging the CNN kernels which are weight-shared in $Seg^{stage1}_{main}$ to capture class-agnostic and class-specific representations concurrently.

Specifically, the class-specific knowledge guidance loss $\mathcal{L}^{i}_{cskg}$ for each category $i$ is defined as follows:

$$\mathcal{L}^{i}_{cskg}(P^{i}, Q) = 1 - \frac{2\sum_{(\mathbf{p^i},\mathbf{q})\in(P^{i},Q)}\sum_{k=1}^{K}\mathbf{p^i}_{k}\mathbf{q}_{k}}{\sum_{(\mathbf{p^i},\mathbf{q})\in(P^{i},Q)}\sum_{k=1}^{K}\left((\mathbf{p^i})_{k}^{2}+\mathbf{q}_{k}^{2}\right)} \tag{8}$$

where $\mathbf{p^i} \in P^{i}, \mathbf{q} \in Q$ , $K$ denotes the total number of pixels per batch of data. Note that we transform the groud truth masks into one-hot format to keep the dimensions consistent with the probability maps. Denote the activation value following the sigmoid function of $Seg^{i}_{cs}$ as $Q = Seg^{i}_{cs-dec}(Seg^{stage1}_{main-enc}(x)) \in \mathbb{R}^{b\times h\times w\times 1}$, the activation values following the softmax function of $Seg^{stage1}_{main}$ as $P = Seg^{stage1}_{main-dec}(Seg^{stage1}_{main-enc}(x)) \in \mathbb{R}^{b\times h\times w\times c}$, $P^{i} \in \mathbb{R}^{b\times h\times w\times 1}$ is each category of $P$, where $b$ is the batch size, $h$ and $w$ are the spatial dimensions of feature map, $c$ is the category number.

**Segmentation network training** For training the segmentation network in each iteration, we first train it on the pseudo target images $x_{S\to T}$ with its corresponding labels $y_S$. In this step, we update the parameters of main segmentation network encoder $Seg^{stage1}_{main-enc}$ and class-specific decoders $Seg_{cs-dec}$ by the class-specific dice loss $\mathcal{L}_{cs-seg}$, as shown in Fig. 4(a-i) top part. Given the segmentation sigmoid output $Q_{S\to T} = Seg^{i}_{cs-dec}(Seg^{stage1}_{main-enc}(x_{S\to T})) \in \mathbb{R}^{b\times h\times w\times 1}$, the class-specific dice loss $\mathcal{L}^{i}_{cs-seg}$ for each category $i$ is defined as:

$$\mathcal{L}^{i}_{cs-seg}(Q_{S\to T}, {y_S}^{i}) = 1 - \frac{2\sum_{(\mathbf{q},\mathbf{y^i})\in(Q_{S\to T},{y_S}^{i})}\sum_{k=1}^{K}\mathbf{q}_{k}(\mathbf{y^i})_{k}}{\sum_{(\mathbf{q},\mathbf{y})\in(Q_{S\to T},{y_S}^{i})}\sum_{k=1}^{K}\left(\mathbf{q}_{k}^{2}+(\mathbf{y^i})_{k}^{2}\right)} \tag{9}$$

In the next step, we update the parameters of $Seg^{stage1}_{main-enc}$ and decoder $Seg^{stage1}_{main-dec}$ for learning the class-specific knowledge by class-specific knowledge guidance loss $\mathcal{L}_{cskg}$ and universal representation by cross entropy loss $\mathcal{L}_{ce}$, as shown in Fig. 4(a-i) bottom part. Given the segmentation softmax output $P_{S\to T} = Seg^{stage1}_{main-dec}(Seg^{stage1}_{main-enc}(x_{S\to T})) \in \mathbb{R}^{b\times h\times w\times c}$:

$$\begin{aligned}\mathcal{L}_{ce}(P_{S\to T}, x_{S\to T}, y_S) = \\ -\sum_{b}\sum_{h,w}\sum_{c\in C}\mathbb{E}_{x_{S\to T}\sim X_{S\to T}, y_S\sim Y_S}[{y_S}^{(b,h,w,c)}\log\left({P_{S\to T}}^{(b,h,w,c)}\right)]\end{aligned} \tag{10}$$

the class-specific knowledge guidance loss $\mathcal{L}^{i}_{cskg}$ for each category $i$ is defined as (8).

As shown in Fig. 4(a-ii), after forwarding the pseudo images $X_{S\to T}$ with its corresponding labels $Y_S$ to the main segmentation network encoder and decoder, we forward target images to $Seg^{stage1}_{main}$, obtain the prediction $P_T = Seg^{stage1}_{main}(x_T)$. An adversarial loss to make the

distribution of $P_T$ to $P_S$:

$$\mathcal{L}_{adv}(P_T, x_T) = -\sum_b \sum_{h,w} \mathbb{E}_{x_T \sim X_T}[\log(\mathrm{D_{dp}}(P_T)^{(b,h,w,1)})] \tag{11}$$

We use this loss to train the segmentation network and fool the domain predictor by maximizing the probability of the target prediction being considered as the source prediction.

**Pixel-level domain predictor training** As shown in Fig. 5, we forward $P$ to a fully-convolutional domain predictor $D_{dp}$ using a cross-entropy loss $\mathcal{L}_d$ for the two classes (i.e., source and target). The loss can be written as:

$$\mathcal{L}_d(P, D_{dp}, x) = -\sum_b \sum_{h,w} (1-z)\log\left(D_{dp}(P)^{(b,h,w,0)}\right) + z\log\left(D_{dp}(P)^{(b,h,w,1)}\right) \tag{12}$$

where z = 0 if the sample is drawn from the target domain, and z = 1 for the sample from the source domain.

**Multi-level adaptation** For retaining the low-level feature, we utilize additional adversarial module on the layer before last feature map layer to further enhance the adaptation capacity. Then the object loss paradigm can be extended from (6) as:

$$\sum_j \lambda_{seg}^j \mathcal{L}_{seg}^j + \sum_j \lambda_{adv}^j \mathcal{L}_{adv}^j \tag{13}$$

where $j$ indicates how many layers of feature map in was used to adapt in the output space adaptation. Note that the segmentation outputs were predicted on each layer individually, $\mathcal{L}_{seg}^j$ and $\mathcal{L}_{adv}^j$ keep the same form as (7) and (11) respectively.

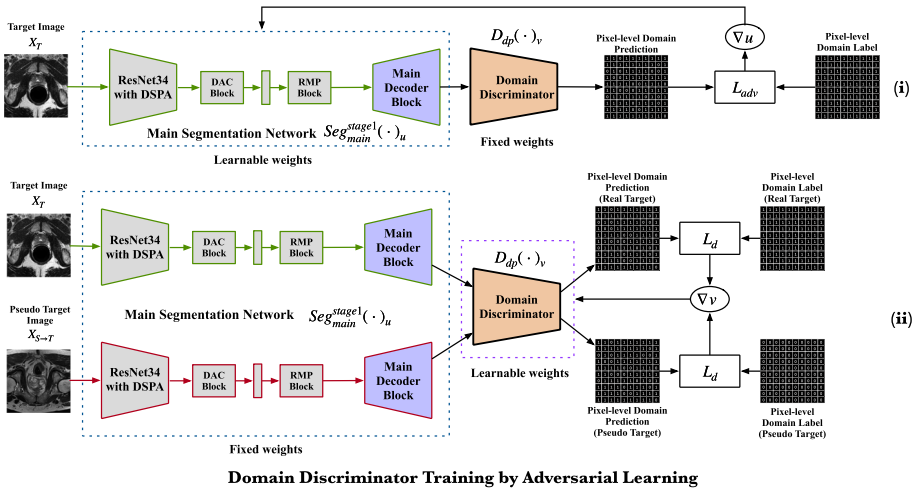



**Fig. 5** Illustration of domain discriminator training procedure in stage 1. The weight of domain discriminator was fixed while the weight of segmentation network was learnable in step (i), The weight of segmentation network was fixed while the weight of domain discriminator in step (ii). After adversarial training, we obtain a segmentation network that was adapted on unlabeled target domain, and a pixel-level domain predictor

### 3.4 Stage 2: pseudo labels correction and final segmentation network training

We reduce the noise of pseudo labels by adjusting the loss of all training examples before updating the objective DNN [30], it estimates the noise transition matrix(NTM) to correct the forward or backward loss. Concretely, loss correction assume that pseudo labels can be shifted to ground truth labels via NTM $T \in [0,1]^{C\times C}$, which specifies the probability of ground truth label $j$ flipping to pseudo label $k$ by $T_{jk} = p(\widehat{y}_t = k \mid y_t = j)$, it encourages the similarity between noise adapted class probabilities and the noisy pseudo labels. The self-training loss is defined as:

$$\mathcal{L}_{ST} = -\sum_{t\in\mathcal{T}} \widehat{y}_t \log\left[f\left(x_t, \mathbf{w}\right) T\right]. \tag{14}$$

where $\widehat{y}_t$ denote the pseudo labels of target domain which were generated from stage 1, . Another importance thing is the construction and optimization of NTM. Rather than construct it by inherent noise type [14, 15] or by cleaned and labeled target domain data [17], we utilize a meta learning based NTM construction strategy [5, 16], it tackles the problem of lacking annotated and clean target domain data by just using source data to make clean data, enhancing the generalization capability of NTM.

As shown in Fig. 2 bottom part, there are two steps in stage 2: i) Noisy Labels Generation and Meta Set Construction; ii) Noisy Labels Correction and Self-Training. In step i), we construct the meta data set $\{X_{\mathcal{M}}, M_{\mathcal{M}}\} = \{x_m, y_m\}_{m\in\mathcal{M}}$ from labeled source data, then in step ii), we optimize the NTM(T) to $T^*$ via:

$$T^* = \underset{T\in[0,1]^{c\times c}}{\arg\min} -\sum_{m\in\mathcal{M}} y_m \log f\left(x_m, \mathbf{w}(T)^*\right), \tag{15}$$

where

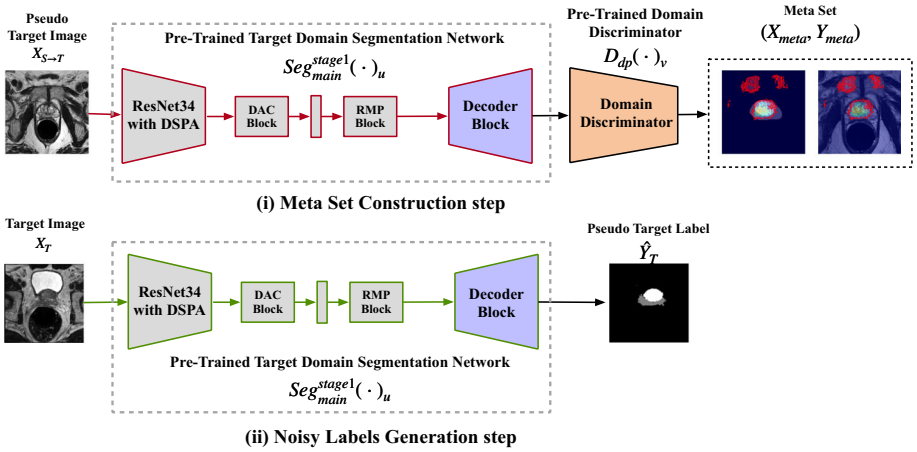
$$\mathbf{w}(T)^* = \underset{\mathbf{w}}{\arg\min} -\sum_{t\in\mathcal{T}} \widehat{y}_t \log\left[f\left(x_t, \mathbf{w}\right) T\right], \tag{16}$$

$w(T)^*$ represents the optimal segmentation net with the minimal corrected loss on the noisy pseudo-labeled target data, the updated $T^*$ should minimize the empirical risk loss on meta data. The training objective of the final segmentation network for target domain ($\text{Seg}_{\text{main}}^{\text{stage2}}$) can be formulated as:

$$\mathcal{L}_{ST} = -\sum_{t\in\mathcal{T}} \widehat{y}_t \log\left[f\left(x_t, \mathbf{w}\right) T^*\right] \tag{17}$$

#### 3.4.1 Noisy labels generation and meta set construction

Model agnostic gradient based meta-learning methods [31, 32] can learn more invariant feature via second-order back-propagation. MAML [31] can make fast adaptation on few shot classification tasks, which use the large annotated classes to execute the first-order approximation of the gradient, and evaluate the error on few shot classes, then calculate the second-order gradient to update the parameters in each iteration. MLDG [32] extends meta-learning to domain generalization problem by randomly split the train set to "meta-train", which was used to train the model at each iteration, and "meta-test", which was used to validate the model at each iteration, can well simulate the potential distribution shift for solving the non-IID or OOD problems of unseen domain generalization. Meta-learning could help constructing a more robust NTM to against the noisy of pseudo labels from target domain, without using any additional annotations in target domain.




**(a) Noisy Labels Generation and Meta Set Construction**

**Fig. 6** Illustration of training process in stage 2. First, we generate the pseudo labels for target domain and meta dataset for optimizing the Noisy Transition Matrix(NTM), as shown in (a)

In this step, the train set of target image was fed to the pre-trained target domain segmentation network $Seg^{stage1}_{main}$ to generate initial target pseudo labels, then a pre-trained pixel-level domain discriminator $D_{dp}(\cdot)_{\mathbf{v}}$, which was obtained from stage 1, was used to select the target-like pixels to construct a meta set $\{X_{\mathcal{M}}, M_{\mathcal{M}}\} = \{x_m, y_m\}_{m\in\mathcal{M}}$ from source data. The illustration of this step was shown in Fig. 6. Samples with predictions $D_{dp}(x_s, \mathbf{v})$ larger than the pre-defined threshold will be meta set to the following meta-learning based NTM construction procedure.

### 3.4.2 Noisy labels correction and self-training

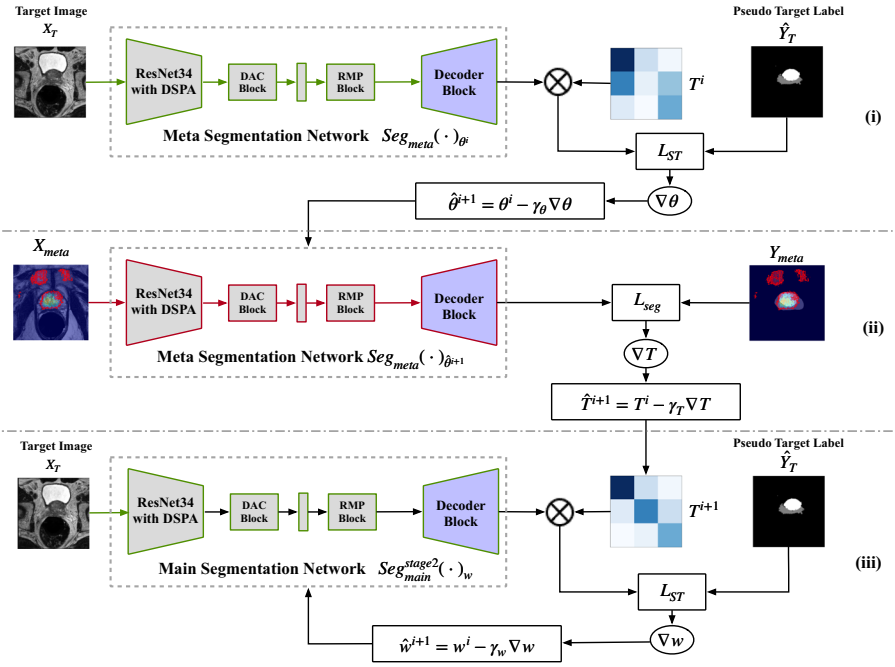
The training procedure alternatively optimize the $T$ and final target segmentation network $Seg^{stage2}_{main}$ within each iteration including three sub-steps: *meta net virtual optimization, meta net meta optimization*, and *segmentation net actual optimization*, as shown in Fig. 7. The first two sub-steps aim to optimize $T$. The third sub-step aims to optimize $Seg^{stage2}_{main}$ with optimized $T^*$. Note that while whole three sub-steps were completed, $Seg^{stage2}_{main}$ finished optimization in one iteration.

In the sub-step of *meta net virtual optimization*, as shown in Fig. 7(i), given a NTM $T^i$ in $i$ th iteration, a meta net $Seg_{meta}$ is copied from $Seg^{stage2}_{main}$ with its parameters $w^i$. $Seg_{meta}$ with its current parameters $w^i$ was updated to $w^{i+1}$ along the gradient descent direction of corrected loss by $T^i$ with learning rate $\gamma_v$:

$$\hat{\mathbf{w}}^{i+1}\left(T^i\right) = \mathbf{w}^i + \gamma_v \nabla_{\mathbf{w}} \sum_{t\in\mathcal{T}} \widehat{y}_t \log\left[f\left(x_t, \mathbf{w}^i\right) T^i\right] \tag{18}$$

Then in the sub-step of *meta net meta optimization*, as shown in Fig. 7(ii), following the (4), $T^i$ was updated to $\tilde{T}^{i+1}$ by minimizing the cross entropy loss on meta dataset with :

$$\tilde{T}^{i+1} = T^i + \gamma_m \nabla_T \sum_{m\in\mathcal{M}} y_m \log f\left(x_m, \hat{\mathbf{w}}^{i+1}\left(T^i\right)\right) \tag{19}$$




**(b) Noisy Labels Correction and Self-Training**

**Fig. 7** Illustration of training process in stage 2. After acquiring the refined pseudo labels, we retrain a new segmentation network by the these labels, which was rectified by the meta-learning based NTM correction strategy, the process of the noisy labels correction and self-training were shown in (b)

where $\gamma_m$ is the learning rate for NTM optimization. The intuition of the meta net meta optimization is to obtain an optima of $\widetilde{T}^{i+1}$ with low empirical risk and high generalization via second-order back-propagation [31]. For preventing the negative values of $\widetilde{T}^{i+1}$ after updating parameters, [5] set the lower bound constraint $\widetilde{T}^{i+1} = \max\left(\widetilde{T}^{i+1}, 0\right)$ to enable the non-negative matrix and then perform normalization along the row direction $T^{i+1}_{jk} = \tilde{T}^{i+1}_{jk} / \sum \tilde{T}^{i+1}_{j}$, which ensure the transition probabilities of class $j$ summed to 1.

In the sub-step of *segmentation net actual optimization*, as shown in Fig. 7(iii), the noisy pseudo labels of target domain which was generated by $Seg^{stage1}_{main}$ was forwarded in $Seg^{stage2}_{main}$. The optimized $\widetilde{T}^{i+1}$ was used to optimize $Seg^{stage2}_{main}$:

$$\mathbf{w}^{i+1} = \mathbf{w}^i + \gamma_a \nabla_{\mathbf{w}} \sum_{t \in \mathcal{T}} \widehat{y}_t \log\left[f\left(x_t, \mathbf{w}\right) T^{i+1}\right] \tag{20}$$

where $\gamma_a$ is the learning rate for $Seg^{stage2}_{main}$ optimization. Through the gradient based meta-learning optimization strategy, both the $T$ and $Seg^{stage2}_{main}(\cdot)_w$ can be gradually ameliorated based on the optimal solution computed in the last step.

### 3.5 Domain-Specific Parallel Adapter (DSPA)

Considering the heterogeneity still exist between real target domain and pseudo target domain, it is not an effective way to share batch normalization layer due to the statistical difference of heterogeneous domains might bring difficulty for learning generic representations, since the shared kernels would bother with the nonessential domain-specific variations, and the shared BN layers may result in inaccurate estimation of global mean and variance in the training phase given inter-site statistical differences, it would lead to performance degradation on the target domain in the validation and testing phase. To overcome the limitations mentioned above, we introduce domain specific batch normalization(DSBN) to replace plain batch normalization(BN) layer first, by allocating domain specific affine parameters $\gamma_d$, $\beta_d$ for domain $d \in \{S, T\}$. Let $\mathbf{x}_d \in \mathbb{R}^{H\times W\times N}$ denote activations of each channel belong to domain $d$, the DSBN layer is defined as follows:

$$\mathrm{DSBN}_d\left(\mathbf{x}_d; \gamma_d, \beta_d\right) = \gamma^d \cdot \hat{x}_d + \beta^d \tag{21}$$

where

$$\hat{\mathbf{x}}_d = \frac{\mathbf{x}_d - \mu_d}{\sqrt{\sigma_d^2 + \epsilon}} \tag{22}$$

and

$$\mu_d = \frac{\sum_n \sum_{i,j} \mathbf{x}_d}{N \cdot H \cdot W} \tag{23}$$

$$\sigma_d^2 = \frac{\sum_n \sum_{i,j} \left(\mathbf{x}_d - \mu_d\right)^2}{N \cdot H \cdot W} \tag{24}$$

To further retain for statistical differences of source and target domain, we introduce a domain-specific parallel adapter (DSPA) [33] module to each residual block [34] of the main segmentation network. Specifically, let $\phi_l$ be the convolutional layer in the main segmentation network and $\mathbf{F}^l \in \mathbb{R}^{k\times k\times C_i\times C_o}$ be corresponding filters for the layer, where $k \times k$ denotes the kernel size and $C_i$, $C_o$ are the number of input and output feature channels respectively. $\mathbb{Z}_d^l \in \mathbb{R}^{1\times 1\times C_i\times C_o}$ is a set of DSPA filters of domain $d$, it is introduced in a parallel way. Given an input $\mathbf{x}_l \in \mathbb{R}^{H\times W\times C_i}$, the output $\mathbf{y}_l \in \mathbb{R}^{H\times W\times C_o}$ of layer $l$ is defined as follows:

$$\mathbf{y}_d^l = \mathbf{F}^l * \mathbf{x}_d[i, j, c_i] + \mathbf{Z}_d^l * \mathbf{x}_d[i, j, c_i] \tag{25}$$

Figure 8 shows the DSPA module with residual block. In each block, an additional convolution operation was installed in parallel, and DSBN for each domain individually. In stage 1, the module shifted through source and target data in each iteration, while in stage 2, only the target domain part was activated to train the final segmentation model for the target domain.

### 3.6 Network and training details

**Image translation network** Following the configuration of [6], The adaptive instance normalization (AdaIN) layers was applied both in content encoders, style encoders and decoders to adjust the style of the output image. The one different thing compare with [6] is we remove the cycle consistency part to keep the image translation network to have the capacity of undeterministic mapping.

**Segmentation network** We adopt CE-Net [35] with ResNet-34 [34] encoder pre-trained on ImageNet as our segmentation network. In the main segmentation network, all residual blocks was introduce a 1x1 convolution operation parallel with the original 3x3 convolution,

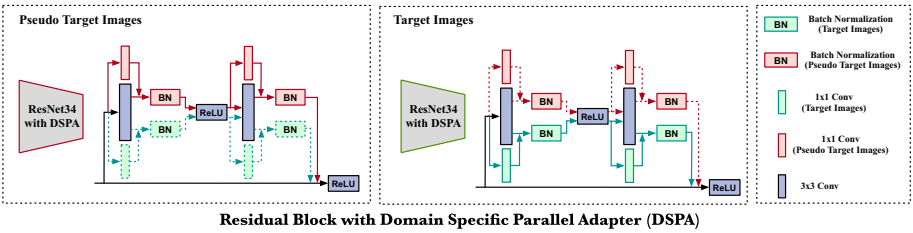



**Fig. 8** The DSPA module for UDA. The input feature map choose one of the branches according to its domain. Note that all parameters except those of DSPA are shared across the real target domain and pseudo target domain

and all normalization layers was replaced to DSBN layers for solving the problem of the inter-site discrepancy. The class-specific branch has the same architecture as the decoder of the main segmentation network, but substitutes the DSBN layers with BN layers, and no 1x1 convolution operation parallel with 3x3 convolution in each residual block. For multi-level output space adaptation, we use the original segmentation logits output, and apply an up-sampling layer for the feature map layer before the original segmentation output to match the size of the input image.

**Pixel-level domain predictor** Following the configuration of the AdaptSegNet [11], the feature maps used for multi-level output space adaptation are also utilized for pixel-level domain prediction. We adopt an architecture similar to DCGAN [36], while instead of utilizing fully-convolutional layers. It consists of five cascade convolution layers with kernel size 4x4 and the stride of 2, with output channel number s {64, 128, 256, 512, 1}, respectively. Each convolution layer is followed by a Leaky ReLU expect for the last layer. Because we jointly train the segmentation network with pixel-level domain predictor using a small batch, hence no batch normalization layers was used in the domain predictor.

**Training details** Our method is implemented using Pytorch toolbox on Nvidia RTX 3090. For training the image translation network, with a batch size of 16 and the learning rate of $10^{-4}$. To train the segmentation network, the Stochastic Gradient Descent (SGD) is utilized as the optimizer, the momentum is 0.9 and the weight decay is $5 \times 10^{-4}$. The initial learning rate of the segmentation network is set as $2.5 \times 10^{-4}$, with the polynomial learning rate scheduling with power of 0.9. To train the pixel-level domain discriminator, we adopt the Adam optimizer with the learning rate as $10^{-4}$ and the same polynomial learning rate scheduling as the segmentation network. The momentum is set as 0.9 and 0.99. For training the meta net, the inner learning rate for meta net virtual optimization was set as $10^{-4}$, and the outer learning rate for meta net meta optimization was set as $10^{-2}$. In each fold, we totally trained 9000 iterations and the batch size was set as 8.

## 4 Experiment

### 4.1 Datasets and evaluation metric

We use two public datasets: NCI-ISBI 2013 [2] and Decathlon [39] for evaluating our proposed approach, collected from two medical institutions: Radboud University Medical Cente (Site A), Boston Medical Centre (Site B), as summarized in Table 2. Note that Site B is the subset of NCI-ISBI 2013, which encompassing all 1.5T MRI data. Following the UDA problem setting, let 3.0T T2 MRI from Site A as the source domain, 1.5T T2 MRI from Site

**Table 2** Details of the scanning protocols for two sites

| Dataset | Case num | Field strength (T) | Resolution (in-/through-(plane)(mm) | Coil | Manufactor |
|---|---|---|---|---|---|
| Site A | 32 | 3 | 0.4/3 | Endorectal | Siemens |
| Site B | 40 | 1.5 | 0.6-0.625/3.6-4 | Surface | Philips |

B as the target domain, we utilized 30 training samples from Site B as unlabeled target data to perform UDA training and 10 remain samples for evaluation. Based on the experiments from [40, 41], We conduct pre-processing as following steps before inputting to our framework: normalized data to have zero mean and unit variance value, bias field correction and noise filter to reduce the intensity variance among source and target data. We conduct four-fold cross validation, using mean value of Dice coefficient(%) as evaluation metric for evaluating the segmentation performance in terms of prostate peripheral zone(PZ) and transition zone(TZ), respectively.

## 4.2 Effectiveness of our framework

In this section, we first compare our approach with baseline and the state-of-the-arts approaches, and then conduct comparison with other self-training based methods in different pseudo labels generation settings for validating the robustness of our framework. In addition, we make comparisons of mean Dice gap between the results of adapted (i.e. unsupervised) and oracle (i.e. fully supervised) setting.

### 4.2.1 Comparison with baseline and state-of-the-art methods

The results are listed in Table 3 with first and second best results highlighted in bold and underline. We first make comparison with baseline setting, the source-only approach means we train the model just on labeled source data, and directly make inference on target data, our

**Table 3** Comparison with baseline and state-of-the-arts results

| Methods | Mechni. | DSC | | |
|---|---|---|---|---|
| | | PZ | TZ | Overall |
| Source-only | – | 28.48 ± 3.51 | 52.57 ± 2.93 | 40.53 ± 3.22 |
| AdaptSegNet [11] | AL | 39.37 ± 3.80 | 69.13 ± 3.06 | 54.25 ± 3.43 |
| PatchAlign [37] | AL | 39.12 ± 4.25 | 72.52 ±3.24 | 55.82 ± 3.75 |
| LTIR [10] | AL | 40.71 ±4.96 | 75.35 ±3.83 | 58.03 ± 4.40 |
| CBST [13] | ST | 38.22 ±3.54 | 70.14 ±3.04 | 54.18 ± 3.29 |
| MRENT [38] | ST | 40.82 ±3.62 | 72.39 ±3.08 | 56.61 ± 3.35 |
| MaxSquare [9] | ST | 37.45 ±3.27 | 69.61 ±2.76 | 53.53 ± 3.02 |
| MetaCorrection [5] | ST | <u>43.25 ± 5.22</u> | <u>74.31 ± 3.92</u> | <u>58.78 ± 4.57</u> |
| Ours | ST | **45.81** ± 5.35 | **80.25** ± 4.59 | **63.03** ± 4.97 |

"AL" and "ST" denote adversarial learning and self-training respectively. The first and second best results highlighted in bold and underline

approach outperforms the source-only approach by a significant increment of 22.5% in Overall Dice. Then we conduct comparison with the state-of-the-art UDA semantic segmentation approaches in prostate zonal segmentation area [5] and natural image area [9, 10, 13, 37, 38], including adversarial based approaches [10, 37] and self-training based approaches [9, 13, 38]. Our proposed approach shows better performance compare with LTIR [10], the best performance in adversarial learning based approach, or MetaCorrection [5], the best performance in self-training based approach. Concretely, it outperforms LTIR and MetaCorrection with increments of 5% and 4.25% in overall Dice, 4.1% and 2.56% in Peripheral Zone Dice, 4.9% and 5.94% in Transition Zone Dice, respectively.

Figure 9 presents some example quantitative segmentation results of target data on three benchmark methods. We note that the self-training based method [5] could obviously promote the performance in comparison to the source-only method. Besides, in contrast to the state-of-the-art self-training method on non-IID problem of prostate multi-zonal segmentation, our proposed framework has better delineation of small-scale and irregularly shaped objects.

### 4.2.2 Robustness to various types of noisy labels

We evaluate the robustness of our proposed approach under various type of noise. Concretely, we make a comparison under four different methods of pseudo labels generation:

- **AdaptSegNet** [11]: An adversarial learning based UDA semantic segmentation method that utilize multi-level adaptation in output space, tackling the unsupervised domain adaptation for semantic segmentation.
- **Source**: Standard reference approach in UDA, directly training a model via source data, then infer on target domain to generate pseudo labels.
- **MUNIT-MRI** [6]: An input space UDA approach for semantic segmentation, which could translate the source data to target domain, transfer the labels from source domain

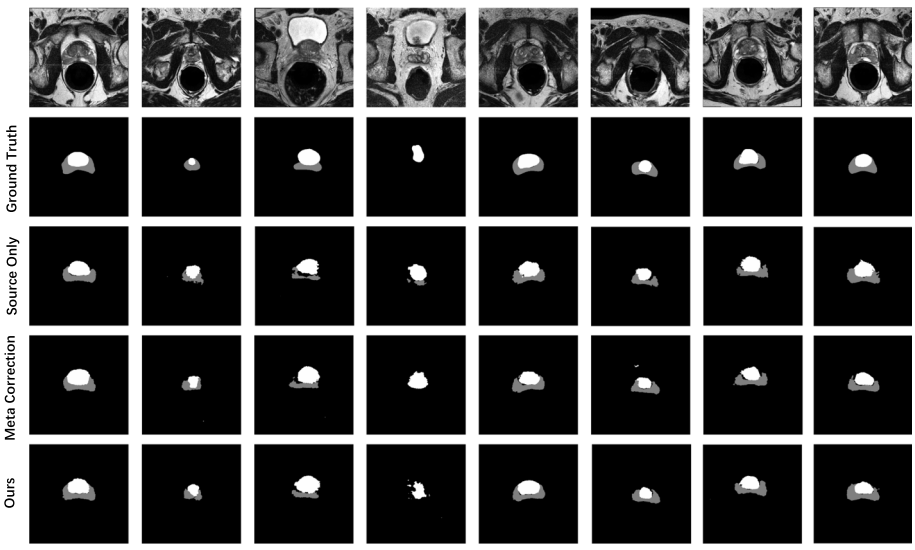



**Fig. 9** Qualitative segmentation results on each site of the target domain. From top to bottom are the results of Baseline approach, state-of-the-art approach and our approach, respectively

**Table 4** Comparison with different pseudo label generation models

| Methods | Pseudo Label Type | Average DSC | Δ |
|---|---|---|---|
| AdaptSegNet [11] | AdaptSegNet | 54.25 | – |
| Self-Training(MRENT [38]) | – | 56.61 | 2.28 |
| Self-Training(Threshold [13]) | – | 55.22 | 0.97 |
| MetaCorrection [5] | – | 58.78 | 4.53 |
| Ours(+DSPA, +CSKG) | – | **60.49** | **6.24** |
| MUNIT-MRI | MUNIT-MRI | 52.32 | – |
| Self-Training(MRENT [38]) | – | 54.17 | 1.85 |
| MetaCorrection [5] | – | 56.99 | 4.67 |
| Ours(+DSPA, +CSKG) | – | **58.3** | **5.98** |
| Source-only | Source Model | 40.53 | – |
| Self-Training(MRENT [38]) | – | 44.73 | 4.2 |
| MetaCorrection [5] | – | 49.43 | 8.9 |
| Ours(+DSPA, +CSKG) | – | **51.65** | **11.12** |
| AdaptSegNet+MUNIT-MRI | AdaptSegNet+MUNIT-MRI | 56.08 | – |
| Self-Training(MRENT [38]) | – | 57.39 | 1.31 |
| MetaCorrection [5] | – | 59.49 | 3.41 |
| Ours(+DSPA) | – | 60.88 | 4.8 |
| Ours(+DSPA, +CSKG) | – | **63.03** | **6.95** |

"Δ" denote the promotion of performance compare with current category of pseudo label generation method. The best result highlighted in bold

to target domain, solving the challenge of adapting to a more informative target domain where multiple target samples can emerge from a single source sample.

- **AdaptSegNet + MUNIT-MRI**: Incorporate input and output space adaptation methods in a tandem way to mitigate the discrepancy between source and target domain.

As listed in Table 4, our proposed approach(row 6) in "AdaptSegNet" setting have the superior result than other self-training based methods, including entropy minimization(row 3), threshold rule(row 4), loss correction(row 5), yielding increments of 3.88%, 5.27%, 1.71% overall Dice respectively. Another observation is different type pseudo labels generation model could obviously improved performance via our proposed approach. For example, in "Source" setting, our approach(row 14) could gain extra 2.22% increment in overall Dice than MetaCorrection(row 13), which reported the best performance on Decathlon adapt to NCI-ISBI13 dataset. We further incorporate input and output space adaptation to generate pseudo labels in our approach and achieve the best result(row 18), in this setting, all approach could have an increment around 1%-2% than other pseudo labels generation models.

### 4.2.3 Comparison of performance gap between adapted setting and oracle setting

To evaluate the adaptation performance, we make the measurement that how much gap is narrowed between the UDA model and fully-supervised model. Hence, we train the model using ground-truths in the target domain as oracle results. We show the performance gap under baseline, state-of-the-art and our proposed methods in Table 7, the characteristics of

methods including three part: i) adaptation module, it denotes which methods were used for domain adaptation; ii) segmentation module, it denotes which module were used for segmentation; iii) pseudo label module, it denotes which methods was used for self-training. We make a comparison between our proposed approach, a model train without using target ground truth labels, i.e.Adapt, and a model train on target domain with target ground truth labels, i.e.Oracle. Our approach achieves sub-optimal dice gap compare with oracle setting, and just lower than MetaCorrection 0.36%.

### 4.3 Ablation study of our framework

#### 4.3.1 CSKG with different loss ratio

We evaluate the effect of different hyper-parameter settings for $\lambda_{cskg}$ in CSKG, as shown in Table 6. We can found that CSKG strategy could consistently improve the segmentation performance when the loss weight range around 0.4, it is also obversed that $\lambda_{cskg}$ can not be set too high, when $\lambda_{cskg} = 1$, i.e. just training universal segmentation network part with class specific knowledge lead to performance degradation compare with $\lambda_{cskg} = 0$, i.e. without class specific knowledge during universal network training. In addition, we conduct extra study as shown in Table 5 last two row, we can see that after introduce the CSKG strategy in Plain method(i.e. AdaptSegNet + MUNIT-MRI), the result achieve a higher performance, increase the overall DSC by 1.17%. These results show that the CSKG strategy can indeed perform as class aware feature regularization to the universal network by jointly training the auxiliary branches and universal network (Table 6).

#### 4.3.2 Experiments with different segmentation, feature adaptation and feature extraction modules

Our approach is flexible that could be easily incorporate with different segmentation networks, feature extraction backbones and output space adaptation modules. Note that the single-level adaptation denotes the output space adaptation module was only used in the last feature map, the dual-level adaptation denotes the output space adaptation module was jointly used in last feature map and the feature map before it. To find an optimal combination of these modules for our framework, we compare with different combination of different segmentation networks, including DeepLab-v2 [42], U-Net [43] and CE-Net [35], with feature extraction backbones including ResNet-34 and ResNet-101 [34]. And two type of feature adaptation modes, including single-level and dual-level adversarial output space adaptation [11]. The results of various modules combination are present in Table 7. The best result of our proposed approach is CE-Net with ResNet-34 backbone, with dual-level adaptation, it achieves 63.03% in overall Dice. Moreover, compare with the corresponding results in Oracle setting(i.e. target domain fully supervised), it achieves the sub-optimal Dice gap. By the way, compare with others, CE-Net with ResNet-34 backbone(row 3,4) also achieves the best result in oracle setting.

#### 4.3.3 Experiments with different FOVs and image size

The size of FOV would obviously effect the performance of prostate segmentation task [44], because the region of prostate is too small compare with whole slice, especially for the PZ segmentation. There are two dominant FOV in source and target domain, here we choose a

**Table 5** The performance gap between the unsupervised setting (adapt) and the fully-supervised setting (oracle), with a baseline method, an adversarial learning based method, a self-training based method, and then show our result in the last row

| Methods | Characteristics | | | Adapt | Oracle | Dice Gap |
|---|---|---|---|---|---|---|
| | Adaptation module | Segmentation module | Pseduo label module | | | |
| AdaptSegNet [11] | multi-level adaptation | DeepLabv2, ResNet101 | – | 54.25 | 74.21 | −19.96 |
| LTIR [10] | feature adaptation, CyCleGAN | DeepLabv2, ResNet101 | – | 58.03 | 74.21 | −16.18 |
| MetaCorrection [5] | multi-level adaptation | DeepLabv2, ResNet101 | loss correction | <u>58.78</u> | 74.21 | **-15.43** |
| Ours | multi-level adaptation, MUNIT, DSPA | CE-Net, ResNet34, CSKG | loss correction | **63.03** | 78.82 | <u>−15.79</u> |

The first and second best results highlighted in bold and underline

**Table 6** Comparison of proposed method with different class specific knowledge guidance ratio, where "Plain" method denotes we incorporate AdaptSegtNet and MUNIT to generate the pseudo labels for target domain

| Methods | $\lambda_{cskg}$ | DSC | | |
|---|---|---|---|---|
| | | PZ | TZ | Overall |
| Ours | 0 | 44.21 | 79.55 | 61.88 |
| – | 0.1 | 45.14 | 79.52 | 62.33 |
| – | 0.2 | 45.49 | 78.83 | 62.16 |
| – | 0.3 | **45.97** | 79.65 | 62.81 |
| – | 0.4 | 45.81 | 80.25 | **63.03** |
| – | 0.5 | 44.23 | **80.63** | 62.43 |
| – | 0.6 | 45.02 | 79.52 | 62.27 |
| – | 0.7 | 45.36 | 79.68 | 62.52 |
| – | 0.8 | 44.96 | 79.24 | 62.1 |
| – | 0.9 | 44.19 | 79.33 | 61.76 |
| – | 1 | 42.8 | 79.06 | 60.93 |
| Plain | – | 39.53 | 72.63 | 56.08 |
| Plain + CSKG | 0.6 | 40.35 | 74.15 | 57.25 |

The best results highlighted in bold

smaller FOV, i.e. 160. Because of the difference of the intra-plane resolution, the pixel size of 2D slice varying in 256, 320, and 400. We first keep the same FOV, i.e. 160, and crop or interpolate three size images to a same size. As shown in Table 8, the result shows that when the FOV size is 160, and the image size is 256, the performance is the best. It may because when the size is set to 256, both images just need to crop or keep its original size, without introducing any pseudo pixels via interpolation. Moreover, we zoom the FOV into 140, the result shows the dice of PZ achieves best when FOV is 140, whereas TZ dice is degrade, the reason might be: PZ is considerable small than TZ, so it is good for PZ segmentation when FOV is small, the region of TZ could not be too small because it may need larger region context information. Note that after cropping the original 2D slice to a smaller region around prostate zone, the segmentation results improved, whereas the prostate zone in these slices

**Table 7** Optimal segmentation module and adaptation module selection

| Methods | Adapt type | Backbone | Adapt | Oracle | Dice Gap |
|---|---|---|---|---|---|
| U-Net | Single-Level | ResNet34 | 56.31 | 73.63 | −17.32 |
| – | Multi-level | – | 57.25 | – | −16.38 |
| CE-Net | Single-level | – | <u>62.31</u> | 78.82 | −16.51 |
| – | Multi-level | – | **63.03** | – | <u>−15.79</u> |
| Deeplabv2 | Single-level | – | 51.03 | 70.25 | −19.22 |
| – | Multi-level | – | 52.56 | – | −17.69 |
| U-Net | Single-Level | ResNet101 | 50.13 | 70.08 | −19.95 |
| – | Multi-level | – | 50.91 | – | −19.17 |
| CE-Net | Single-level | – | 52.69 | 70.58 | −17.89 |
| – | Multi-level | – | 54.28 | – | −16.3 |
| Deeplabv2 | Single-level | – | 57.44 | 74.21 | −16.77 |
| – | Multi-level | – | 59.49 | – | **−14.72** |

The first and second best results highlighted in bold and underline

**Table 8** Optimal FOV selection

| Origin FOV (mm) | Target FOV (mm) | Origin size (pixels) | Target size (pixels) | DSC | | |
|---|---|---|---|---|---|---|
| | | | | PZ | TZ | Overall |
| 160,200 | 140 | 256,320,400 | 224 | 46.06 | 78.76 | 62.41 |
| 160,200 | 160 | 256,320,400 | 256 | 45.81 | 80.25 | 63.03 |
| 160,200 | 160 | 256,320,400 | 320 | 45.12 | 78.38 | 61.75 |
| 160,200 | 160 | 256,320,400 | 400 | 44.21 | 75.13 | 59.67 |

were located in the center, it may expire when the scans are not standard, i.e. prostate zone close to the margin of the slice. It may need to predict a bounding-box first, rather than simply make a center crop operation.

## 5 Conclusion

In this paper, we propose a class-aware multi-stage unsupervised domain adaptation framework to tackle model performance degradation when the train and test datasets are non-identity distributed and there is no available annotations for model fine-tuning or retraining. The proposed framework takes the heterogeneous and unlabeled image as input and outputs central gland and peripheral zone masks, which makes the high availability of the pre-trained deep-learning-based segmentation method to the heterogeneous data without introducing any extra annotation. Our multiple-step adaptation strategy between heterogeneous domains and class-specific knowledge guidance strategy for the class-imbalanced problem is the key to a better result. We also introduce DSPA module for heterogeneous image segmentation to encourage model to learn domain-specific representation. It not only does our framework deliver good results, but also bridges the gap of different data distribution between train and test scenarios. The experimental results demonstrate that our framework achieve superior results to state-of-the-art UDA semantic segmentation approaches in prostate multi-zonal segmentation task. In future work, we are going to extend our approach to more complex medical images, such as 3D and multi-view prostate MRIs.

**Acknowledgements** This work was supported in part by the National Natural Science Foundation of China under Grants 61972046, 62002025 and 62002020, the Beijing Natural Science Foundation-Haidian Original Innovation Joint Fund Project (No. L182034), the Scientific Research Seed Fund of Peking University First Hospital (No.2021SF45), and the Fundamental Research Funds for the Central Universities (No.2019XD-A12).

**Author Contributions** People who have helped with acquisition of funding, Wendong Wang, Bo Zhang and Zheng Zhang; General supervision of your research group or general administrative support, Wendong Wang and Bo Zhang; writing assistance and language editing, Yu Bai, Yue Mi, Zheng Zhang and Bo Zhang; Technical editing, Yue Mi, Jingyun Wu and Haiwen Huang; Proofreading, Wendong Wang and Bo Zhang

**Data Availability** The first dataset("Decathlon", denoted by "Site A") analysed during the current study are available in the "Medical Segmentation Decathlon" repository, [http://medicaldecathlon.com/].
The second dataset("NCI-ISBI13", denoted by "Site B") analysed during the current study are available in the "The Cancer Imaging Archive(TCIA)" repository, [https://wiki.cancerimagingarchive.net/display/Public/NCI-ISBI+2013+Challenge+-+Automated+Segmentation+of+Prostate+Structures].

## Declarations

**Competing of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References


1. Litjens G, Toth R, van de Ven W, Hoeks C, Kerkstra S, van Ginneken B, Vincent G, Guillard G, Birbeck N, Zhang J et al (2014) Evaluation of prostate segmentation algorithms for mri: the promise12 challenge. Med Image Anal 18(2):359–373
2. Antonelli M, Reinke A, Bakas S, Farahani K, Kopp-Schneider A, Landman BA, Litjens G, Menze B, Ronneberger O, Summers RM et al (2022) The medical segmentation decathlon. Nat Commun 13(1):1–13
3. Mårtensson G, Ferreira D, Granberg T, Cavallin L, Oppedal K, Padovani A, Rektorova I, Bonanni L, Pardini M, Kramberger MG et al (2020) The reliability of a deep learning model in clinical out-of-distribution mri data: a multicohort study. Med Image Anal 66:101714
4. Gibson E, Hu Y, Ghavami N, Ahmed HU, Moore C, Emberton M, Huisman HJ, Barratt DC (2018) Inter-site variability in prostate segmentation accuracy using deep learning. In: International conference on medical image computing and computer-assisted intervention, pp 506–514. Springer
5. Guo X, Yang C, Li B, Yuan Y Metacorrection: domain-aware meta loss correction for unsupervised domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3927–3936 (2021)
6. Chiou E, Giganti F, Punwani S, Kokkinos I, Panagiotaki E Harnessing uncertainty in domain adaptation for mri prostate lesion segmentation. In: International conference on medical image computing and computer-assisted intervention, pp 510–520 (2020). Springer
7. Zheng Z, Yang Y (2021) Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. Int J Comput Vis 129(4):1106–1120
8. Chen Y-C, Lin Y-Y, Yang M.-H, Huang J-B (2019) Crdoco: pixel-level domain transfer with cross-domain consistency. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1791–1800
9. Chen C, Dou Q, Chen H, Qin J, Heng P-A (2019) Synergistic image and feature adaptation: towards cross-modality domain adaptation for medical image segmentation. Proceedings of the AAAI conference on artificial intelligence 33:865–872
10. Kim M, Byun H (2020) Learning texture invariant representation for domain adaptation of semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12975–12984
11. Tsai Y-H, Hung W-C, Schulter S, Sohn K, Yang M-H, Chandraker M (2018) Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7472–7481
12. Liu Q, Dou Q, Yu L, Heng PA (2020) Ms-net: multi-site network for improving prostate segmentation with heterogeneous mri data. IEEE Trans Med Imaging 39(9):2713–2724
13. Zou Y, Yu Z, Kumar B, Wang J (2018) Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European conference on computer vision (ECCV), pp 289–305
14. Hendrycks D, Mazeika M, Wilson D, Gimpel K (2018) Using trusted data to train deep networks on labels corrupted by severe noise. Advances in Neural Information Processing Systems 31
15. Patrini G, Rozza A, Krishna Menon A, Nock R, Qu L (2017) Making deep neural networks robust to label noise: A loss correction approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1944–1952
16. Shu J, Xie Q, Yi L, Zhao Q, Zhou S, Xu Z, Meng D (2019) Meta-weight-net: learning an explicit mapping for sample weighting. Advances in Neural Information Processing Systems 32
17. Wang Z, Hu G, Hu Q (2020) Training noise-robust deep neural networks via meta-learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4524–4533
18. Li H, Gong M (2017) Self-paced convolutional neural networks. In: IJCAI, pp 2110–2116
19. Arazo E, Ortego D, Albert P, O'Connor N, McGuinness K (2019) Unsupervised label noise modeling and loss correction. In: International conference on machine learning, pp 312–321. PMLR
20. Gao Y, Huang R, Chen M, Wang Z, Deng J, Chen Y, Yang Y, Zhang J, Tao C, Li H (2019) Focusnet: imbalanced large and small organ segmentation with an end-to-end deep neural network for head and neck

ct images. In: International conference on medical image computing and computer-assisted intervention, pp 829–838.Springer
21. Feng S, Zhao H, Shi F, Cheng X, Wang M, Ma Y, Xiang D, Zhu W (2020) Chen X Cpfnet: context pyramid fusion network for medical image segmentation. IEEE Trans Med Imaging 39(10):3008–3018
22. Akil M, Saouli R, Kachouri R et al (2020) Fully automatic brain tumor segmentation with deep learning-based selective attention using overlapping patches and multi-class weighted cross-entropy. Med Image Anal 63:101692
23. Yeung M, Sala E, Schönlieb C-B, Rundo L (2022) Unified focal loss: generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. Comput Med Imaging Graph 95:102026
24. Sugino T, Kawase T, Onogi S, Kin T, Saito N, Nakajima Y (2021) Loss weightings for improving imbalanced brain structure segmentation using fully convolutional networks. In: Healthcare, vol 9, pp 938. MDPI
25. Fujii H, Tanaka H, Ikeuchi M, Hotta K (2021) X-net with different loss functions for cell image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3793–3800
26. Huang X, Liu M-Y, Belongie S, Kautz J (2018) Multimodal unsupervised image-to-image translation. In: Proceedings of the european conference on computer vision (ECCV), pp 172–189
27. Ba J, Caruana R (2014) Do deep nets really need to be deep. Advances in Neural Information Processing Systems27
28. Chaudhari P, Choromanska A, Soatto S, LeCun Y, Baldassi C, Borgs C, Chayes J, Sagun L (2019) Zecchina R (2019) Entropy-sgd: Biasing gradient descent into wide valleys. J Stat Mech Theory Exp 12:124018
29. Milletari F, Navab N, Ahmadi S-A (2016) V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth international conference on 3D vision (3DV), pp 565–571. IEEE
30. Song H, Kim M, Park D, Shin Y, Lee J-G (2022) Learning from noisy labels with deep neural networks: a survey. IEEE Transactions on Neural Networks and Learning Systems
31. Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning, pp 1126–1135. PMLR
32. Li D, Yang Y, Song Y-Z, Hospedales T (2018) Learning to generalize: meta-learning for domain generalization. In: Proceedings of the AAAI conference on artificial intelligence, vol 32, pp 865–872
33. Rebuffi S-A, Bilen H, Vedaldi A (2018) Efficient parametrization of multi-domain deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8119–8127
34. He K, Zhang X, Ren S, Sun J (2016)Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
35. Gu Z, Cheng J, Fu H, Zhou K, Hao H, Zhao Y, Zhang T, Gao S, Liu J (2019) Ce-net: context encoder network for 2d medical image segmentation. IEEE Trans Med Imaging 38(10):2281–2292
36. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training gans. Advances in Neural Information Processing Systems 29
37. Tsai Y.-H, Sohn K, Schulter S, Chandraker M (2019) Domain adaptation for structured output via discriminative patch representations. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1456–1465
38. Zou Y, Yu Z, Liu X, Kumar B, Wang J (2019) Confidence regularized self-training. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 5982–5991
39. Keyvan F, Carl J, Anant M, Henkjan H, John F, Justin K, Andinet E, Larry C (2015) NCI-ISBI 2013 challenge: automated segmentation of prostate structures (2013)
40. Rundo L, Han C, Zhang J, Hataya R, Nagano Y, Militello C, Ferretti C, Nobile MS, Tangherloni A, Gilardi MC, et al (2020) Cnn-based prostate zonal segmentation on t2-weighted mr images: a cross-dataset study. Neural Approaches to Dynamics of Signal Exchanges, pp 269–280
41. Palumbo D, Yee B, O'Dea P, Leedy S, Viswanath S, Madabhushi A (2011) Interplay between bias field correction, intensity standardization, and noise filtering for t2-weighted mri. In: 2011 Annual international conference of the IEEE engineering in medicine and biology society, pp 5080–5083. IEEE
42. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
43. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention, pp 234–241. Springer

44. Liu Q, Dou Q, Heng P-A (2020) Shape-aware meta-learning for generalizing prostate mri segmentation to unseen domains. In: International conference on medical image computing and computer-assisted intervention, pp 475–485. Springer