Generalizable Medical Image Segmentation via Random Amplitude Mixup and Domain-Specific Image Restoration

Ziqi Zhou[1,2], Lei Qi[3*], and Yinghuan Shi[1,2*]

[1] State Key Laboratory for Novel Software Technology, Nanjing University
[2] National Institute of Healthcare Data Science, Nanjing University
[3] School of Computer Science and Engineering, Southeast University
zhouzq@smail.nju.edu.cn, qilei@seu.edu.cn, syh@nju.edu.cn

Abstract. For medical image analysis, segmentation models trained on one or several domains lack generalization ability to unseen domains due to discrepancies between different data acquisition policies. We argue that the degeneration in segmentation performance is mainly attributed to overfitting to source domains and domain shift. To this end, we present a novel generalizable medical image segmentation method. To be specific, we design our approach as a multi-task paradigm by combining the segmentation model with a self-supervision *domain-specific image restoration* (DSIR) module for model regularization. We also design a *random amplitude mixup* (RAM) module, which incorporates low-level frequency information of different domain images to synthesize new images. To guide our model be resistant to domain shift, we introduce a semantic consistency loss. We demonstrate the performance of our method on two public generalizable segmentation benchmarks in medical images, which validates our method could achieve the state-of-the-art performance. [‡]

Keywords: Medical image segmentation, domain generalization, self-supervision

1 Introduction

Recently, deep convolution neural networks (DCNNs) have progressed remarkably in computer vision tasks (*e.g.*, image classification, semantic segmentation, object detection, *etc.*). Especially in medical image segmentation tasks, deep learning based methods have taken over the dominant position [33,29]. Usually, DCNNs require large numbers of annotated training images to alleviate the risk of overfitting. However, datasets in medical image segmentation tasks are often relatively small in amount than those in natural image segmentation tasks. Moreover, it is notoriously time-consuming to acquire segmentation annotations of medical images. Accurate annotations also requires specific expertise in radiodiagnosis. Except for the data amounts and annotations problem, basic deep

[*] Corresponding Authors: Yinghuan Shi and Lei Qi.
[‡] Code is available at https://github.com/zzzqzhou/RAM-DSIR.

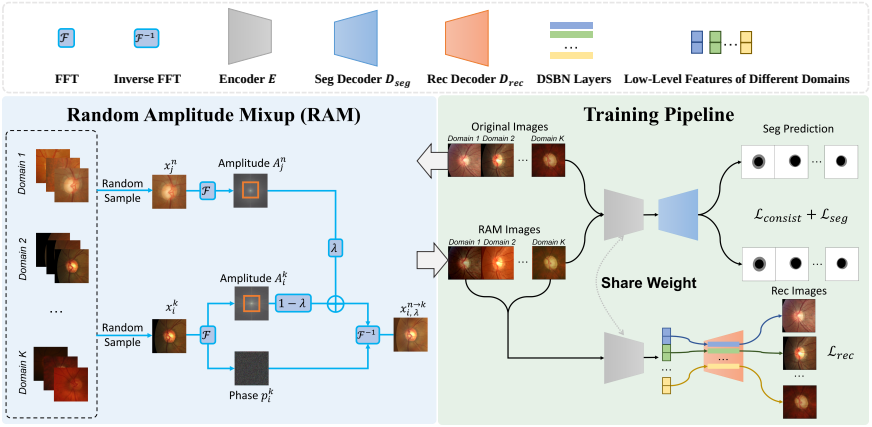

Fig. 1. The overall architecture of our method. (a) Random Amplitude Mixup (RAM): We extract the amplitude maps from two random sampled images of different domains and incorporate the amplitude maps of them. Then we can synthesize new images that have different domain styles and preserve original semantic information. (b) The synthesized images from RAM module are utilized to train the segmentation model and DSIR decoder. Basic segmentation loss combined with semantic consistency loss and image recovering loss are employed to train our network.

learning methods assume that training data and test data share same distribution information. This assumption requires that training and test data are collected from the same distribution, which is a strong assumption. Due to data distribution shifts, this assumption usually becomes invalid in the real clinical setting. It is known that the quality of medical images varies greatly due to many factors, such as different scanners, imaging protocols, and operators. As a result, the segmentation model directly trained on a set of training images may lack generalization ability on test images drawn from another hospital or medical center, which follows a different distribution.

To fight against distribution shift, tremendous researchers have investigated several practical settings, such as unsupervised domain adaptation (UDA), domain generalization (DG), *etc.* UDA-based segmentation methods have gained much popularity in medical image segmentation [18,9,44]. To be specific, UDA attempts to learn a segmentation network on single or multiple source domain images including the annotations along with unlabeled target domain images. UDA-based methods intend to narrow the domain gap between source and target domains. However, this prerequisite is sometimes impractical or infeasible in real-world application. Since data privacy protection is rigorous in medical image scenarios, we may sometimes have no chance to access target domain images from some medical centers.

Recently, domain generalization (DG) is proposed to alleviate the application limitation in UDA. DG is a more feasible yet challenging setting requiring

only source domains for training. After training on source domain images, we can directly deploy the segmentation model to new unseen target domains. Recently, several literature have developed domain generalization methods to improve model generalization ability with multiple source domains. Among these previous methods, most of them attempt to learn a domain-irrelevant feature representation among multi-source domains for generalization [11,24,22,23]. Some data augmentation based methods have emerged to tackle the problem of lack of prior information from target domains [40,43] by synthesizing newly stylized images to expand diversities of source domain images. Some pioneers have proposed self-supervised tasks (*i.e.*, solving a Jigsaw Puzzles) to help regularize the model [3,37]. These methods indicate that an auxiliary self-supervised task can better help the model learn domain-invariant knowledge, thus improving model regularization. However, solving a Jigsaw Puzzles may not be a sufficient self-supervision for DG segmentation tasks. To this end, we aim to design a more complex self-supervision to better learn domain-invariant semantic representation for medical image segmentation.

In our work, we present a new framework based on vanilla generalizable medical image segmentation model. To be specific, we first introduce a *random amplitude mixup* (RAM) module by utilizing the Fourier transform to capture frequency space signals from different source domain images and incorporating low-level frequency information of different source domain images to generate new images with different styles. We then use these synthetic images as data augmentation to train the segmentation model and improve robustness. To further regularize our model and combat domain shifts, we employ a semantic consistency training loss to minimize the discrepancy between predictions of real source domain images and synthetic images. To learn more robust feature representation, we introduce a *domain-specific image restoration* (DSIR) decoder to recover low-level features from synthetic images to original source domain images. We demonstrate the effectiveness of our approach on two DG medical image segmentation benchmarks. Our method achieves the state-of-the-art performance compared with competitive methods. We display the overall architecture of our method in Fig. 1.

2 Related Work

Unsupervised Domain Adaptation. Unsupervised Domain Adaptation (UDA) is a particular branch of Domain Adaptation (DA) that leverages labeled data from one or multiple source domains along with unlabeled data from the target domain to learn a classifier for the target domain [4,5,9,18,27,36,44]. Under such a problem setting, data from the target domain can be utilized to guide the optimization procedure. The general motivation of UDA is to align the source domain and target domain distributions. Some methods adopted a generative model to narrow the pixel-level distribution gap between source and target domains [7,9,18,44]. Dou *et al*. [12] aligned the feature distribution between source and target domains by adversarial training to keep semantic features consistent

in different domains. Differently, some methods attempted to narrow the distribution gap between source and target domains in output space level [8,34,35,36]. However, due to data privacy protection, some unlabeled target domain data can not be accessed in some cases. The target domain is not available in the training process, making UDA methods impractical in some real-world applications.

Domain Generalization. In contrast to UDA, Domain Generalization (DG) purely trains a model on one or more related source domains and directly generalizes to target domains. A large amount of DG methods have been proposed recently [6,13,14,17,32,40,41]. Some methods tried to minimize the domain discrepancy across multiple source domains to learn domain-invariant representations [15,19,24]. With the recent advance of the episodic training strategy for domain generalization [1,11,22,23], some meta-learning-based methods have been developed to generalize models to unseen domains. Li *et al*. [23] proposed an episodic training procedure to simulate domain shift at runtime to improve the robustness of the network. Unlike previous meta-learning-based methods, our method is based on vanilla training policy by aggregating different source domain images. We apply a self-supervised image-level-recovering task and semantic consistency training policy to improve the generalization performance on unseen target domains. In medical image segmentation, several prior literature have studied DG segmentation. For instance, Zhang *et al*. [43] proposed a deep-stacked transformation approach that utilized a stack of image transformations to simulate domain shift in medical imaging. Liu *et al*. [26] introduced a shape-aware supervision combined with meta-learning to help generalizable prostate image segmentation. Wang *et al*. [38] stored domain-specific prior knowledge in a pool as domain attributes for domain aggregation. Liu *et al*. [25] proposed a continuous frequency space augmentation with episodic training policy to improve the generalization ability across different domains. Similar to Liu *et al*. [25], we apply frequency space information for image augmentation in our method. However, we utilize augmented images for image segmentation and the auxiliary image-level-recovering task. This will help our model be more robust to domain shifts and alleviate overfitting.

Self-supervisied Regularization. Self-supervised learning have gained much attention in computer vision, natural language processing *etc.* [10,16,2], recently. It utilizes annotation-free tasks to learn feature representations of data for the downstream tasks. In DG scenario, some methods have also introduced self-supervision tasks to regularize the semantic feature learning [3,37]. We also develop an image-level-recovering self-supervision task to help regularize the model. Different from [3,37] solving a Jigsaw Puzzles, our image-level-recovering task is more complicated, which can better regularize the model.

3 Our Method

3.1 Definition and Overview

We denote a set of K source domains as $D_s = \{(x_i^k, y_i^k)_{i=1}^{N_k}\}_{k=1}^{K}$, where x_i^k is the i-th image from k-th source domain; y_i^k is the segmentation label of x_i^k; N_k is

the number of samples in k-th source domain. We aim to learn a generalizable medical image segmentation model F_θ on D_s. The model F_θ is expected to show a satisfactory generalization performance on unseen target domain $D_t = \{x_i\}_{i=1}^{N_t}$, where x_i represent the i-th image in target domain, and N_t is the number of image samples in target domain.

Our proposed method contains an encoder-decoder segmentation model with an auxiliary *domain-specific image restoration* (DSIR) decoder. In front of our training pipeline, we introduce a data augmentation and corruption module named as *random amplitude mixup* (RAM). The workflow of our method contains three steps. First, in the RAM module, we apply the Fourier transform on two source domain images that share different domain labels to obtain their frequency space signals; then, we incorporate their low-frequency signals and utilize inverse Fourier transform to generate new images. Secondly, in our DSIR module, the encoder of the segmentation model obtains low-level features of images generated by RAM. A decoder with domain-specific batch normalization is trained to recover original images in a specific source domain from the low-level features. Finally, the encoder-decoder segmentation model is trained by the segmentation loss of source domain images and augmented images; also we adopt a consistency loss between the outputs of source domain images and augmented images to help the segmentation model better resist domain shifts. We discuss all of these components next in detail.

3.2 Random Amplitude Mixup

To address the restriction of domain discrepancy between source and target domains, a reasonable idea is to apply data augmentation on source domains to diversify source domain data. In this case, we can regularize the model and alleviate overfitting to source domains. Among plenty of data augmentation methods, Mixup [42] has been widely used in image recognition tasks. Image-level-Mixup (IM) incorporates two different images from the training dataset. However, IM will also disturb the semantic information of images, which may negatively influence semantic segmentation tasks. Inspired by prior literature [39,25], we propose to exploit the inherent information of source domains in the frequency space and incorporate distribution information (*i.e.*, style) in the amplitude spectrum of different images. We name our module as *random amplitude mixup* (RAM).

To be specific, we randomly take a sample image $x_i^k \in \mathbb{R}^{H\times W\times C}$ (C represents the number of image channels; H and W are height and width of the image) from source domain k. Then, we perform the Fourier transform [31] $\mathcal{F}$ to obtain the frequency space signal of image x_i^k, which can be written as:

$$\mathcal{F}(x_i^k)(u,v,c) = \sum_{h=0}^{H-1}\sum_{w=0}^{W-1} x_i^k(h,w,c)e^{-j2\pi(\frac{h}{H}u+\frac{w}{W}v)}, j^2 = -1. \tag{1}$$

After the Fourier transform, we can decompose the frequency signal $\mathcal{F}(x_i^k)$ into an amplitude spectrum $\mathcal{A}_i^k \in \mathbb{R}^{H\times W\times C}$ and a phase image $\mathcal{P}_i^k \in \mathbb{R}^{H\times W\times C}$,

where the amplitude spectrum contains low-level statistics (*e.g.*, style) while the phase image includes high-level (*e.g.*, object) semantics of the original image. We incorporate the amplitude spectrum of different images from multiple source domains. To this end, we randomly select another sample image $x_j^n (n \neq k)$ from source domain n and perform the Fourier transform on it as well. So that, we obtain another amplitude $\mathcal{A}_j^n$ of image x_j^n. To incorporate the low-frequency component within amplitude $\mathcal{A}_i^k$ and $\mathcal{A}_j^n$, we introduce a binary mask $\mathcal{M}$ which can control the scale of low-frequency component in amplitude spectrum to be incorporated. After that, we incorporate the amplitude information of image x_i^k and image x_j^n by:

$$\mathcal{A}_{i,\lambda}^{n \to k} = \mathcal{A}_i^k * (1 - \mathcal{M}) + ((1 - \lambda)\mathcal{A}_i^k + \lambda \mathcal{A}_j^n) * \mathcal{M}, \tag{2}$$

where $\mathcal{A}_{i,\lambda}^{n \to k}$ is the newly interpolated amplitude spectrum; λ is a parameter that used to adjust the ratio between $\mathcal{A}_i^k$ and $\mathcal{A}_j^n$. Finally, we can transform the merged amplitude $\mathcal{A}_{i,\lambda}^{n \to k}$ into a newly stylized image through inverse Fourier transform $\mathcal{F}^{-1}$ as follows:

$$x_{i,\lambda}^{n \to k} = \mathcal{F}^{-1}(\mathcal{A}_{i,\lambda}^{n \to k}, \mathcal{P}_i^k), \tag{3}$$

where the generated image $x_{i,\lambda}^{n \to k}$ contains the semantic information of x_i^k and its low-level information (*e.g.*, style) is a mixture of low-level information of x_i^k and x_j^n. In our implementation, we follow [25] to dynamically sample λ from $[0.0, 1.0]$ to generate images. Fig. 1 (a) illustrates the overall architecture of RAM.

To further indicates that RAM can increase the diversity of source domain and narrow the domain discrepancy. We show the t-SNE [28] visualization of image features in **Fundus** dataset in Fig. 2. Fig. 2 (a) shows the original distribution information of different domains in the **Fundus** dataset. From the visualization, we can observe that the image features from different domains are clearly separated. This leads to the problem that training the model on original source domains make the model easily overfit to specific source domains, which might degrade generalization performance on target domains. However, in Fig. 2 (b), we discover that, by applying RAM on original source domains, we can narrow domain gaps significantly, showing domain invariant representation. The distribution of different domains is more compacted and diversified.

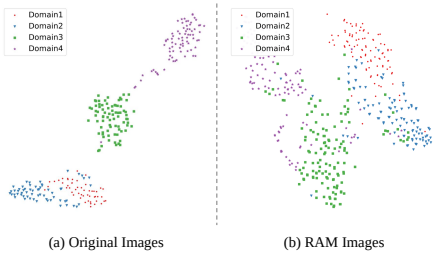

Fig. 2. t-SNE visualization of features of original images and RAM augmented images from **Fundus** dataset. We use different colors and markers to denote different domains.

3.3 Semantic Consistency Training

To segment images from the target domain, one straightforward method is to train a vanilla segmentation model in a unified fashion by directly feeding multi-source domain images into the model. We name training such a vanilla segmentation model as "DeepAll". Although the "DeepAll" method might have good generalization performance on multi-source domains, it may not preserve satisfactory segmentation performance on target domain images. Training a vanilla segmentation model on multi-source domains does not introduce supervision to combat domain shifts. Also, we have mentioned that original multi-source domain images lack sufficient diversity in feature distribution, which may lead to overfitting to a specific source domain.

We design a semantic consistency training strategy to tackle problems of the "DeepAll" method. To be specific, we introduce an encoder-decoder structure [33] as our segmentation model. The encoder E will extract low-level semantic features from images while the segmentation decoder D_{seg} is used to predict segmentation masks. We formulate the forward propagation of the segmentation model on source domain image x_i^k as:

$$\hat{y}_i^k = D_{seg}(E(x_i^k)), \tag{4}$$

where $\hat{y}_i^k$ is the predicting segmentation mask. Since we utilize RAM to generate newly stylized images from original source domains, we can use these augmented images to help train the segmentation model. This can also regularize the segmentation model and improve its generalization performance on target domains. Similar to Eq. (4), the forward propagation on $x_{i,\lambda}^{n\to k}$ can be written as:

$$\hat{y}_{i,\lambda}^{n\to k} = D_{seg}(E(x_{i,\lambda}^{n\to k})), \tag{5}$$

where $\hat{y}_{i,\lambda}^{n\to k}$ represents the prediction. Then, we utilize the unified cross-entropy (CE) loss [30] and Dice loss [29] as our segmentation loss to optimize the model. The CE and dice loss on original source domain k are formulated as:

$$\mathcal{L}_{ce}^k = -\frac{1}{N}\sum_{i=0}^{N-1}\Big(y_i^k \log \hat{y}_i^k + (1-y_i^k)\log(1-\hat{y}_i^k)\Big), \tag{6}$$

$$\mathcal{L}_{dice}^k = 1 - \frac{2\sum_{i=0}^{N-1}\hat{y}_i^k y_i^k}{\sum_{i=0}^{N-1}(\hat{y}_i^k + y_i^k + \epsilon)}, \tag{7}$$

where y_i^k is the shared ground truth of x_i^k and $x_{i,\lambda}^{n\to k}$; N represents the number of samples from domain k; ϵ is a smooth factor to avoid dividing by 0. The CE loss $\mathcal{L}_{ce}^{n\to k}$ and dice loss $\mathcal{L}_{dice}^{n\to k}$ on the generated images are similar as above. So, segmentation losses on x_i^k and $x_{i,\lambda}^{n\to k}$ can be written as:

$$\mathcal{L}_{seg}^k = \mathcal{L}_{dice}^k + \mathcal{L}_{ce}^k, \quad \mathcal{L}_{seg}^{n\to k} = \mathcal{L}_{dice}^{n\to k} + \mathcal{L}_{ce}^{n\to k}. \tag{8}$$

To combat domain shifts, we propose a novel semantic consistency loss in our method. Specifically, we regard the generated image $x_{i,\lambda}^{n\to k}$ as a style augmentation of x_i^k. We intend to force the segmentation model to predict consistent segmentation results from x_i^k and $x_{i,\lambda}^{n\to k}$. So that the segmentation model can be less sensitive to domain shift. We design a loss term to minimize the Kullback-Leibler (KL) divergence [21] between soft predictions $\hat{y}_i^k$ and $\hat{y}_{i,\lambda}^{n\to k}$. Our semantic consistency loss is as follows:

$$\mathcal{L}_{consist}^k = \frac{1}{N}\sum_{i=0}^{N-1}\Big(\mathrm{KL}(\hat{y}_i^k \| \hat{y}_{i,\lambda}^{n\to k}) + \mathrm{KL}(\hat{y}_{i,\lambda}^{n\to k} \| \hat{y}_i^k)\Big), \tag{9}$$

where KL represents the KL-divergence [21]. We compute a symmetric version of KL-divergence between $\hat{y}_i^k$ and $\hat{y}_{i,\lambda}^{n\to k}$. By explicitly enhancing the consistency of results, the segmentation model can extract semantic features more robust to domain shift, thus improving performance on unseen target domains.

3.4 Domain-Specific Image Restoration

To further regularize the segmentation model and reduce overfitting on source domains, we propose a self-supervised auxiliary task to help train a more robust segmentation model. To be specific, we introduce an image restoration decoder with domain-specific batch normalization (DSBN) layers [5]. The image restoration decoder is utilized to recover image from the low-level features extracted by the segmentation encoder E from the RAM image $x_{i,\lambda}^{n\to k}$.

To better recover images of different source domains, we add DSBN in our image restoration decoder. Let our image restoration decoder denote as $D_{rec} = \{D_{rec}^1, D_{rec}^2, \cdots, D_{rec}^K\}$, where K represents the number of source domain, D_{rec}^k is used to recover images from low-level features of RAM images generated by k-th source domain images. All of the decoders in D_{rec} share the same model parameters but have different batch normalization layers [20]. Since distribution information of multi-source domains is quite different, using different batch normalization layers in different domains can better preserve domain intrinsic features for image restoration. The forward propagation of the image restoration module on source domain k are as follows:

$$\hat{x}_i^k = D_{rec}^k(E(x_{i,\lambda}^{n\to k})), \tag{10}$$

where E is the encoder in our segmentation model; $\hat{x}_i^k$ is the recovering image from $x_{i,\lambda}^{n\to k}$. We utilize this image restoration decoder as a regularization of the segmentation encoder E. We show detailed information of our image restoration module in Fig. 1 (b).

To train the image restoration module, we employ L2 distance as recovering loss to optimize D_{rec} and E. The recovering loss on k-th source domain are:

$$\mathcal{L}_{rec}^k = \frac{1}{NHWC}\sum_{i=0}^{N-1}\sum_{h=0}^{H-1}\sum_{w=0}^{W-1}\sum_{c=0}^{C-1}\Big(x_i^k(h,w,c) - \hat{x}_i^k(h,w,c)\Big)^2, \tag{11}$$

where N represents the number of samples from domain K; H, W, C are width, height and channel of the image.

Overall, we can formulate our whole framework as a multi-task learning paradigm. The total training loss are as follows:

$$\mathcal{L}_{total} = \frac{1}{K}\sum_{k=1}^{K}\Big(\lambda_1\mathcal{L}_{seg}^{k} + \lambda_2\mathcal{L}_{seg}^{n\to k} + \lambda_3\mathcal{L}_{rec}^{k} + \lambda_4\mathcal{L}_{consist}^{k}\Big), \tag{12}$$

where K represents the number of source domains; λ_1, λ_2, λ_3, and λ_4 are hyper-parameters to balance the weights of basic segmentation loss, consistency loss, and image restoration loss respectively.

4 Experiments

4.1 Datasets

We evaluate our method on two public DG medical image segmentation datasets as popular used in [26,38,25]: **Fundus** [38] and **Prostate** [26]. The **Fundus** dataset contains retinal fundus images from 4 different medical centers for optic cup and disc segmentation. Each domain has been split into training and testing sets. For pre-processing, we follow the prior literature [38] and center-crop disc regions with a 800 × 800 bounding-box for all of images in **Fundus** dataset. After that, we randomly resize and crop a 256 × 256 region on each cropped images as network input. The **Prostate** dataset collected T2-weighted MRI prostate images from 6 different data sources for prostate segmentation. All of the images have been cropped to 3D prostate region and 2D slices in axial plane have been resized to 384×384. For model training, we feed 2D slices of prostate images into our model. We normalize the data individually to [-1, 1] in intensity values on both datasets.

4.2 Implementation Details

We employ a UNet-based [33] encoder-decoder structure as our segmentation model. The DISR decoder is similar to our segmentation decoder by replacing batch normalization layers with DSBN layers. We implement our experiment with the PyTorch framework on 1 Nvidia RTX 2080Ti GPU with 11 GB memory. We train our model for 400 epochs on **Fundus** dataset and 200 epochs on **Prostate** dataset. For each dataset, we set 8 as training batch size. We also employ the Adam optimizer with an initial learning rate of 0.001 to optimize our model. To stabilize the training process, the learning rate is decayed by the polynomial rule. Last but not least, we set λ_1, λ_2, λ_3, and λ_4 as 1, 1, 0.1 and 0.5 empirically in Eq. (12).

Since the **Fundus** dataset has already split each domain into training and testing sets, we train our model on training sets of source domains and evaluate on testing sets of target domains. During testing, we first resize 800 × 800 test

images to size of 256×256 and get 256×256 segmentation masks. We then resize segmentation masks to 800×800 and compute evaluation metrics on them. For **Prostate** dataset, we directly train segmentation model on source domains and test on target domains. Since original images of **Prostate** dataset are all 3D volumes, we first get 2D predictions and concatenate all 2D predictions of each 3D sample, then compute evaluation metrics on 3D predictions. When testing on **Prostate** dataset, we also skip those 2D slices that not contain any prostate region. All of implementations on datasets follow previous methods [26,38]. For evaluation, we adopt commonly-used metric of Dice coefficient (Dice) and Average Surface Distance (ASD) to quantitatively evaluate the segmentation results of whole region and the surface shape respectively. Higher Dice coefficient represents better performance and ASD is the opposite. To avoid randomness, we repeat our experiments for 3 times and report the average performance.

4.3 Comparison with Other DG methods

Experiment setting. In our experiments, we follow the practice in prior literature of domain generalization and employ the leave-one-domain-out strategy, *i.e.*, training on K source domains and test on the left one target domain (total $K+1$ domains). So that, for **Fundus** and **Prostate** datasets, we have four and six distinguished tasks, respectively.

We choose five recent state-of-the-art domain generalization methods to compare with ours and reproduce their results. First of all, the JiGen [3] is an effective self-supervised based DG methods for model regularization by solving jigsaw puzzles. The BigAug [43] is an augmentateion based DG method. SAML [26] and FedDG [25] are two meta-learning based generalizable medical image segmentation methods. Finally, the DoFE [38] is a domain-invariant feature representation learning approach. We further train a vanilla segmentation model by simply aggregating all source domain images as our baseline model.

In Tables 1 and 2, we show Dice coefficient and ASD results of different domains in **Fundus** dataset. All of the methods successfully outperform our baseline method except BigAug [43] (Dice coefficient 85.49% *vs.* 85.63%; ASD 14.18 voxel *vs.* 13.98 voxel). We assume that this is because BigAug [43] was first designed to augment grey-scale medical images (*e.g.*, CT, MRI, *etc.*) for domain generalization segmentation tasks. Images in **Fundus** dataset are all RGB images which have quite different image properties compared with other medical images. So that the generalization performance of BigAug [43] could be degraded. Other methods gain improvements above baseline more or less and prove that different regularization and generalization strategies can help the model to learn more robust feature representation. Compared with these methods, we achieve higher average Dice coefficient and better average ASD on **Fundus** dataset. This thanks to our RAM and DSIR module. The RAM helps to diversify our source domain images to alleviate overfitting. Also, the image restoration tasks can regularize our model to learn more robust feature representation. Last but not least, we adopt a semantic consistency training policy to resist to domain shift. All of these key components contribute to success

Table 1. Dice coefficient of different methods on **Fundus** segmentation task (%). We mark the top results in **bold**.

Task	Optic Cup/Disc Segmentation				Avg.
Unseen Site	Domain 1	Domain 2	Domain 3	Domain 4	
JiGen [3]	82.45/95.03	77.05/87.25	87.01/**94.94**	80.88/91.34	86.99
BigAug [43]	77.68/93.32	75.56/87.54	83.33/92.68	81.63/92.20	85.49
SAML [26]	83.72/95.03	77.68/87.57	84.20/94.49	82.08/92.78	87.19
FedDG [25]	81.72/95.62	77.87/88.71	83.96/94.83	81.90/93.37	87.25
DoFE [38]	84.17/94.96	**81.03**/89.29	86.54/91.67	**87.28**/93.04	88.50
Baseline	81.44/95.52	77.20/87.96	85.11/94.56	72.30/90.97	85.63
Ours	**85.48**/**95.75**	78.82/**89.43**	**87.44**/94.67	85.84/**94.10**	**88.94**

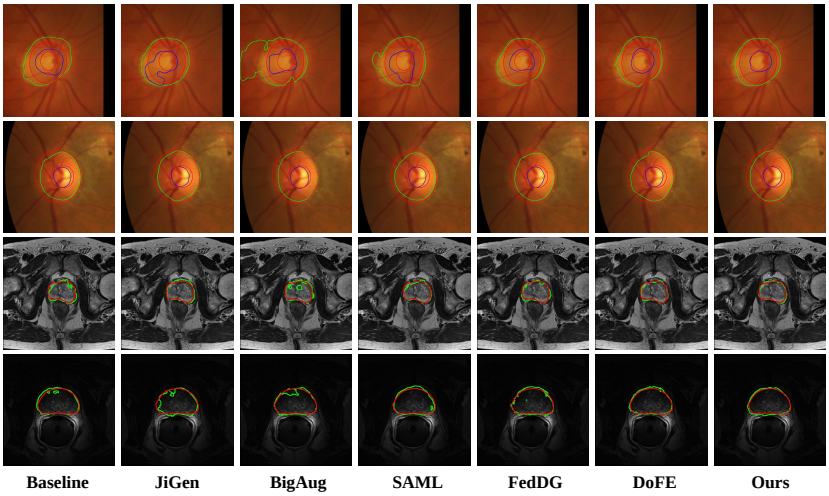

Fig. 3. Visualization on segmentation results of different methods on **Fundus** (top two rows) and **Prostate** datasets (bottom two rows). The red contours indicate the boundaries of ground truths while the green and blue contours are predictions.

of our method on **Fundus** dataset. Compared with baseline, our method achieves consistent improvements over baseline across all unseen domain settings, with the average performance increase of 3.31% in Dice coefficient and 3.66 voxel average improvement in ASD.

To further indicate effectiveness of our method, we provide experiment results on **Prostate** dataset in Tables 3 and 4. For prostate segmentation task, all of the comparison DG method outperform baseline. Our method also obtains the highest Dice coefficient and ASD across most unseen domains. The average Dice coefficient 88.08% and ASD 1.37 voxel are the best compared with other DG methods. Specially, compared with baseline, the increase in overall Dice coefficient of our method is 4.04% and ASD decreases 1.55 voxel. In Fig. 3, we show the visualization results of two sample images from target domains of **Fundus** and **Prostate** datasets. It is explicit that our method can accurately

Table 2. Average Surface Distance (ASD) of different methods on **Fundus** segmentation task (voxel). We mark the top results in **bold**.

Task	Optic Cup/Disc Segmentation				Avg.
Unseen Site	Domain 1	Domain 2	Domain 3	Domain 4	
JiGen [3]	18.57/9.43	17.29/19.53	9.15/**6.99**	15.84/12.14	13.62
BigAug [43]	22.61/12.53	17.95/17.64	11.48/10.33	11.57/9.36	14.18
SAML [26]	17.08/9.01	16.72/18.63	10.87/7.87	16.28/8.64	13.14
FedDG [25]	18.57/7.69	15.87/16.93	11.09/7.28	10.23/7.51	11.90
DoFE [38]	16.07/7.18	**13.44**/17.06	10.12/10.75	**8.14**/7.29	11.26
Baseline	18.16/8.99	15.67/17.95	11.96/9.42	20.03/9.64	13.98
Ours	**16.05**/**7.12**	14.01/**13.86**	**9.02**/7.11	8.29/**7.06**	**10.32**

Table 3. Dice coefficient of different methods on **Prostate** segmentation task (%). We mark the top results in **bold**.

Task	Prostate Segmentation						Avg.
Unseen Site	Domain 1	Domain 2	Domain 3	Domain 4	Domain 5	Domain 6	
JiGen [3]	85.45	89.26	85.92	87.45	86.18	83.08	86.22
BigAug [43]	85.73	89.34	84.49	88.02	81.95	87.63	86.19
SAML [26]	86.35	90.18	85.03	88.20	86.97	87.69	87.40
FedDG [25]	86.43	89.59	85.30	88.95	85.93	87.39	87.27
DoFE [38]	**89.64**	87.56	85.08	**89.06**	86.15	87.03	87.42
Baseline	85.30	87.56	82.33	87.37	80.49	81.40	84.04
Ours	87.56	**90.20**	**86.92**	88.72	**87.17**	**87.93**	**88.08**

segment the objective structure of unseen domain images and the boundary of the structure is smoother while other methods may fail to do so.

4.4 Analysis of Our Method

We conduct extensive ablation studies on our method. Firstly,s we investigate the effectiveness of our random amplitude mixup for data augmentation and DSIR module on **Fundus** and **Prostate** dataset. We need to note that, without RAM, our DISR module cannot be implemented. Since our RAM module is utilized to conduct style augmentation and image corruption at the same time, here we discuss the style augmentation and image corruption separately. The experimental results are illustrated in Tables 5 and 6. The RAM_{Aug} indicates that the RAM style augmentation is employed in our method and DSIR represents the domain-specific image restoration module with image corruption. The method without these two components (*i.e.*, the first row in Tables 5 and 6) is the baseline method, which is the same with the baseline results in Tables 1 and 3. From Tables 5 and 6, we observe that each component plays a significant role in our method. By adding RAM style augmentation in our method, the overall segmentation performance on fundus segmentation task can increase 2.12% in Dice coefficient and on prostate segmentation tasks the improvements of Dice coefficient is 3.23%. Besides, when equipping with domain-specific image restoration module, our model can gain 1.58% and 1.03% overall improvements

Table 4. Average Surface Distance (ASD) of different methods on **Prostate** segmentation task (voxel). We mark the top results in **bold**.

Task	Prostate Segmentation						Avg.
Unseen Site	Domain 1	Domain 2	Domain 3	Domain 4	Domain 5	Domain 6	
JiGen [3]	1.11	1.81	2.61	1.66	**1.71**	2.43	1.89
BigAug [43]	1.13	1.78	4.01	1.25	1.92	1.89	2.00
SAML [26]	1.09	1.54	2.52	1.41	2.01	1.77	1.72
FedDG [25]	1.30	1.67	2.36	1.37	2.19	1.94	1.81
DoFE [38]	**0.92**	1.49	2.74	1.46	1.89	1.53	1.68
Baseline	1.22	1.95	4.68	1.51	3.95	4.23	2.92
Ours	1.04	**0.81**	**2.23**	**1.16**	1.81	**1.15**	**1.37**

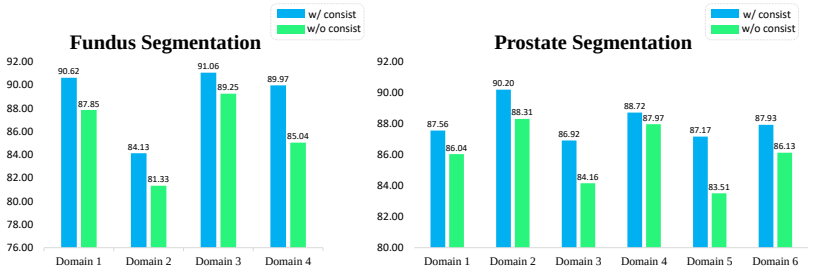

Fig. 4. Ablation study of our semantic consistency training policy. Green and blue bars represent average Dice coefficient of our complete method and the method without consistency loss respectively. We show results on different domains from **Fundus** and **Prostate** datasets.

in Dice coefficient on **Fundus** and **Prostate** datasets respectively. Based on these results, we justify that, our RAM and DSIR module can help regularize our segmentation model and improve the generalization ability. Last rows in Tables 5 and 6 display the results by adding all of the components in our method, which are the same as results of our method in Tables 1 and 3.

As aforementioned, during the training process, we employed a semantic consistency loss as a supervision signal to make the model resistant to domain shift. In Fig. 4, we investigate the effectiveness of our semantic consistency loss on **Fundus** and **Prostate** datasets. We observe that without the semantic consistency loss, all of the results degenerate on both datasets. This indicates that the semantic consistency loss do help improve the generalization performance of our model which means our model can be more robust to domain shift.

Moreover, we experiment different types of consistency loss on **Fundus** dataset. In Table 7, we show the results of different kinds of consistency loss. Except for KL-divergence (KL-Div), we also employ mean squared error (MSE) and Jensen–Shannon divergence (JS-Div). We observe that using different consistency loss will not affect the overall results of our method much, which means our method is robust to different types of consistency loss.

Table 5. Ablation Study of key components in our method on **Fundus** Segmentation Task (%). We mark the top results in **bold**.

Task		Optic Cup/Disc Segmentation				Avg.
$\mathbf{RAM}_{\mathrm{Aug}}$	**DSIR**	Domain 1	Domain 2	Domain 3	Domain 4	
-	-	81.44/95.52	77.20/87.96	85.11/94.56	72.30/90.97	85.63
✓	-	83.06/94.86	78.09/89.04	86.73/**95.01**	82.28/92.89	87.75
-	✓	83.76/95.31	77.43/88.07	85.84/94.19	81.58/91.48	87.21
✓	✓	**85.48/95.75**	**78.82/89.43**	**87.44**/94.67	**85.84/94.10**	**88.94**

Table 6. Ablation Study of key components in our method on **Prostate** Segmentation Task (%). We mark the top results in **bold**.

Task		Prostate Segmentation						Avg.
$\mathbf{RAM}_{\mathrm{Aug}}$	**DSIR**	Domain 1	Domain 2	Domain 3	Domain 4	Domain 5	Domain 6	
-	-	85.30	87.56	82.33	87.37	80.49	81.40	84.04
✓	-	87.28	89.94	85.45	87.86	86.17	86.94	87.27
-	✓	86.57	88.04	83.19	87.42	82.08	83.14	85.07
✓	✓	**87.56**	**90.20**	**86.92**	**88.72**	**87.17**	**87.93**	**88.08**

Table 7. Dice coefficient of different consistency loss on **Fundus** segmentation task (%). We mark the top results in **bold**.

Task	Optic Cup/Disc Segmentation				Avg.
Unseen Site	Domain 1	Domain 2	Domain 3	Domain 4	
MSE	85.45/95.13	77.96/89.14	86.73/**94.76**	**85.93/94.16**	88.65
JS-Div	85.04/94.91	78.02/88.27	86.32/93.91	85.14/93.87	88.19
KL-Div	**85.48/95.75**	**78.82/89.43**	**87.44**/94.67	85.84/94.10	**88.94**

5 Conclusion

We present a novel generalizable medical image segmentation method for fundus and prostate image segmentation. To combat with overfitting in DG segmentation, we introduce random amplitude mixup (RAM) module to synthesize images with different domain style. We utilize the synthetic images as data augmentation to train the segmentation model and propose a self-supervised domain-specific image restoration (DSIR) module to recover the original images from synthetic images. Moreover, to further make the model resistant to domain shift and learn more domain invariant feature representation, we employ a semantic consistency loss in our training process. Our experimental results and ablation analysis indicate that all of the proposed components can help regularize the model and improve generalization performance on unseen target domains.

Acknowledgements. This work was supported by NSFC Major Program (62192783), CAAI-Huawei MindSpore Project (CAAIXSJLJJ-2021-042A), China Postdoctoral Science Foundation Project (2021M690609), Jiangsu Natural Science Foundation Project (BK20210224), and CCF-Lenovo Bule Ocean Research Fund.

References

1. Balaji, Y., Sankaranarayanan, S., Chellappa, R.: Metareg: Towards domain generalization using meta-regularization (2018)
2. Bao, H., Dong, L., Piao, S., Wei, F.: BEit: BERT pre-training of image transformers. In: ICLR (2022)
3. Carlucci, F.M., D'Innocente, A., Bucci, S., Caputo, B., Tommasi, T.: Domain generalization by solving jigsaw puzzles. In: CVPR (2019)
4. Chang, W.L., Wang, H.P., Peng, W.H., Chiu, W.C.: All about structure: Adapting structural information across domains for boosting semantic segmentation. In: CVPR (2019)
5. Chang, W.G., You, T., Seo, S., Kwak, S., Han, B.: Domain-specific batch normalization for unsupervised domain adaptation. In: CVPR (2019)
6. Chattopadhyay, P., Balaji, Y., Hoffman, J.: Learning to balance specificity and invariance for in and out of domain generalization. In: ECCV (2020)
7. Chen, C., Dou, Q., Chen, H., Qin, J., Heng, P.A.: Synergistic image and feature adaptation: Towards cross-modality domain adaptation for medical image segmentation. In: AAAI (2019)
8. Chen, Y., Li, W., Van Gool, L.: Road: Reality oriented adaptation for semantic segmentation of urban scenes. In: CVPR (2018)
9. Chen, Y.C., Lin, Y.Y., Yang, M.H., Huang, J.B.: Crdoco: Pixel-level domain transfer with cross-domain consistency. In: CVPR (2019)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
11. Dou, Q., Coelho de Castro, D., Kamnitsas, K., Glocker, B.: Domain generalization via model-agnostic learning of semantic features. NeurIPS (2019)
12. Dou, Q., Ouyang, C., Chen, C., Chen, H., Heng, P.A.: Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss. In: IJCAI (2018)
13. Du, Y., Xu, J., Xiong, H., Qiu, Q., Zhen, X., Snoek, C.G., Shao, L.: Learning to learn with variational information bottleneck for domain generalization. In: ECCV (2020)
14. Gong, R., Li, W., Chen, Y., Gool, L.V.: Dlow: Domain flow for adaptation and generalization. In: CVPR (2019)
15. Gong, Y., Lin, X., Yao, Y., Dietterich, T.G., Divakaran, A., Gervasio, M.: Confidence calibration for domain generalization under covariate shift. In: ICCV (2021)
16. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
17. Hoffer, E., Ben-Nun, T., Hubara, I., Giladi, N., Hoefler, T., Soudry, D.: Augment your batch: Improving generalization through instance repetition. In: CVPR (2020)
18. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: ICML (2018)
19. Hsu, Y.C., Lv, Z., Kira, Z.: Learning to cluster in order to transfer across domains and tasks. In: ICLR (2018)
20. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
21. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22**(1), 79–86 (1951)
22. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Learning to generalize: Meta-learning for domain generalization. In: AAAI (2018)

23. Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.Z., Hospedales, T.M.: Episodic training for domain generalization. In: ICCV (2019)
24. Li, H., Pan, S.J., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: CVPR (2018)
25. Liu, Q., Chen, C., Qin, J., Dou, Q., Heng, P.A.: Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: CVPR (2021)
26. Liu, Q., Dou, Q., Heng, P.A.: Shape-aware meta-learning for generalizing prostate mri segmentation to unseen domains. In: MICCAI (2020)
27. Liu, Z., Miao, Z., Pan, X., Zhan, X., Lin, D., Yu, S.X., Gong, B.: Open compound domain adaptation. In: CVPR (2020)
28. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. JMLR (2008)
29. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3DV (2016)
30. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press (2012)
31. Nussbaumer, H.J.: The fast fourier transform. In: Fast Fourier Transform and Convolution Algorithms, pp. 80–111. Springer (1981)
32. Qiao, F., Zhao, L., Peng, X.: Learning to learn single domain generalization. In: CVPR (2020)
33. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
34. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: CVPR (2018)
35. Tsai, Y.H., Sohn, K., Schulter, S., Chandraker, M.: Domain adaptation for structured output via discriminative patch representations. In: ICCV (2019)
36. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: CVPR (2019)
37. Wang, S., Yu, L., Li, C., Fu, C.W., Heng, P.A.: Learning from extrinsic and intrinsic supervisions for domain generalization. In: ECCV (2020)
38. Wang, S., Yu, L., Li, K., Yang, X., Fu, C.W., Heng, P.A.: Dofe: Domain-oriented feature embedding for generalizable fundus image segmentation on unseen datasets. IEEE TMI (2020)
39. Yang, Y., Soatto, S.: Fda: Fourier domain adaptation for semantic segmentation. In: CVPR (2020)
40. Yue, X., Zhang, Y., Zhao, S., Sangiovanni-Vincentelli, A., Keutzer, K., Gong, B.: Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In: ICCV (2019)
41. Zakharov, S., Kehl, W., Ilic, S.: Deceptionnet: Network-driven domain randomization. In: ICCV (2019)
42. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: ICLR (2018)
43. Zhang, L., Wang, X., Yang, D., Sanford, T., Harmon, S., Turkbey, B., Wood, B.J., Roth, H., Myronenko, A., Xu, D., et al.: Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. IEEE TMI (2020)
44. Zhang, Y., Qiu, Z., Yao, T., Liu, D., Mei, T.: Fully convolutional adaptation networks for semantic segmentation. In: CVPR (2018)