Unsupervised Domain Adaption via Similarity-Based Prototypes for Cross-Modality Segmentation

Ziyu Ye, Chen Ju, Chaofan Ma, and Xiaoyun Zhang(✉)

Cooperative Medianet Innovation Center, Shanghai Jiao Tong University,
Shanghai, China
{ziyu_ye,ju_chen,chaofanma,xiaoyun.zhang}@sjtu.edu.cn

Abstract. Deep learning models have achieved great success on various vision challenges, but a well-trained model would face drastic performance degradation when applied to unseen data. Since the model is sensitive to domain shift, unsupervised domain adaption attempts to reduce the domain gap and avoid costly annotation of unseen domains. This paper proposes a novel framework for cross-modality segmentation via similarity-based prototypes. In specific, we learn class-wise prototypes within an embedding space, then introduce a similarity constraint to make these prototypes representative for each semantic class while separable from different classes. Moreover, we use dictionaries to store prototypes extracted from different images, which prevents the class-missing problem and enables the contrastive learning of prototypes, and further improves performance. Extensive experiments show that our method achieves better results than other state-of-the-art methods.

Keywords: Unsupervised domain adaption · Medical image segmentation · Prototype · Contrastive learning

1 Introduction

Medical image segmentation is a pixel-wise classification task, which is the basis of many clinical applications [2]. Though deep neural networks have made significant progress in medical image analysis [13,24], most supervised works have the assumption that enough annotated data is collected, which is prohibitively difficult in reality. In clinical scenarios, data collection is time-consuming and laborious, and pixel-wise annotations require expert knowledge of doctors. Hence, unsupervised domain adaption (UDA) is introduced as an annotation-efficient method to help cross-modality medical image segmentation [1].

UDA transfers the knowledge learned in a label-rich domain to a label-lacking domain, bridging the domain gap. Currently, there are two main streams for UDA. One is image-level adaption [4,10,21], which aims to make the images of different domains appear similar, so that the label-lacking target domain can

S. Albarqouni et al. (Eds.): DART 2021/FAIR 2021, LNCS 12968, pp. 133–143, 2021.
https://doi.org/10.1007/978-3-030-87722-4_13

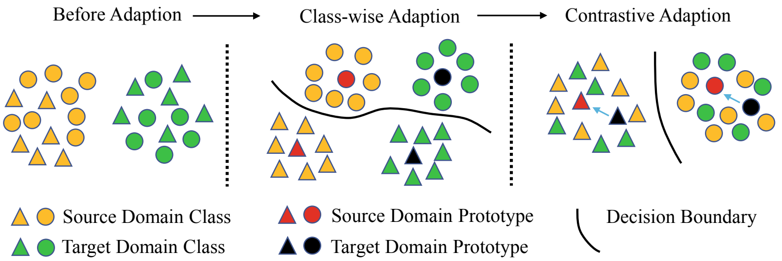

Fig. 1. Illustration of our adaption procedure. On the one hand, our method performs class-wise adaption to align semantic features to their prototypes, on the other hand, we align class-wise prototypes across domains using the contrastive loss.

learn from the transferred source domain. The other stream focuses on feature-level adaption [16,22], which aims to match the feature distributions with adversarial learning or contrastive learning. Besides, DualHierNet [1] also uses edges as self-supervision for the target domain, and EntMin [5] uses entropy minimization to narrow domain gaps.

For methods based on image adaption, most works only conduct the two-direction images translation between source and target domains separately, which may be insufficient to eliminate the domain gap. To this end, we propose a unified framework to fully exploit the two-direction translation results, and our network can be trained end-to-end. For methods based on feature adaption, most works employ adversarial learning to make the semantic features indistinguishable to discriminators, which aligns features in an implicit way. In this paper, we explicitly align features to their prototypes using a class-wise similarity loss, which aims to minimize intra-class and maximize inter-class feature distribution difference. Then, with the help of feature dictionaries, we use the contrastive loss to align class-wise prototypes across domains, which further alleviates the domain shift problem. Figure 1 shows the illustration of our adaption procedure.

2 Methodology

Given a labeled source dataset $\mathbb{D}_s = \left\{x_s^i, y_s^i\right\}_{i=1}^{N_s}$ and an unlabeled target dataset $\mathbb{D}_t = \left\{x_t^i\right\}_{i=1}^{N_t}$, unsupervised domain adaption (UDA) for semantic segmentation aims to train a model with supervision from $\mathbb{D}_s$ and information from $\mathbb{D}_t$ to narrow domain gap and improve segmentation performance on $\mathbb{D}_t$.

2.1 Motivation

In our method, we utilize class-wise feature prototypes to perform explicit feature alignment. Firstly, we use a similarity-based loss to regularize the embedded space, and the purpose is to boost feature consistency. Features of the same class are encouraged to be closer to the prototype, and prototypes of different classes are encouraged to be separable. Secondly, we use dictionary to store prototypes from various images, and then contrastive learning is used to improve feature adaption across domains. We expect to adapt prototypes from target domain to source domain, so features of both domains are explicitly aligned.

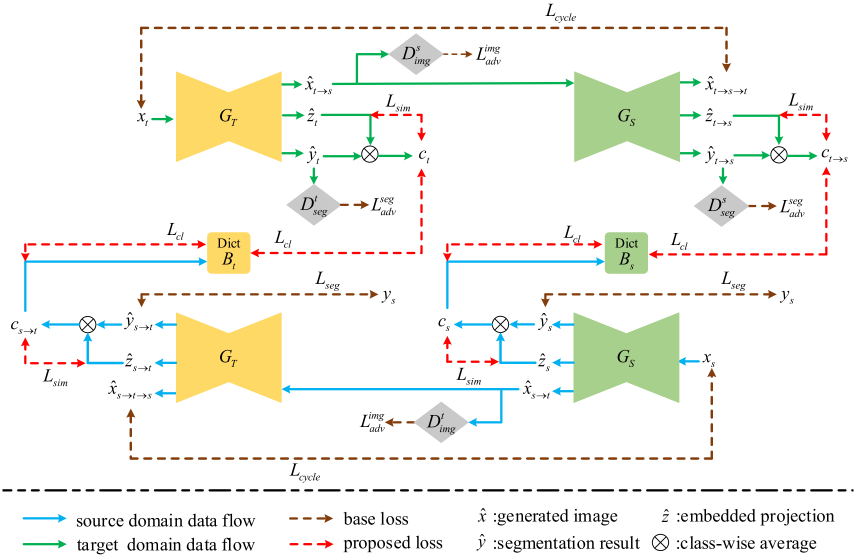

Fig. 2. Our framework has a cycle structure, and mainly consists of G_S and G_T. These modules have the same structure and output the translated image $\hat{x}$, segmentation result $\hat{y}$ and embedded projection $\hat{z}$. Prototypes c are obtained by performing a class-wise average operation on $\hat{z}$ under supervision from $\hat{y}$. During training, only c_s and $c_{s\to t}$ are stored into feature dictionaries B_s and B_t. Besides the widely used circle consistency loss L_{cycle}, segmentation loss L_{seg}, adversarial loss L^{img}_{adv} and L^{seg}_{adv}, we additionally use the proposed loss L_{sim} and L_{cl} to perform explicit feature alignment.

2.2 Proposed Framework

The overall framework is shown in Fig. 2. It has a cycle structure inspired from Cycle-GAN [12], and consists of two modules G_S and G_T, which have the same structures, but process images of different domains. Concretely, G_S processes source domains images, while G_T processes target domain images. Structurally, we input an image for G_S or G_T, and it will output the translated image $\hat{x}$, the segmentation result $\hat{y}$ and the embedded projection $\hat{z}$. During training, our framework is trained in a cycle manner. At each iteration, we calculate the class-wise prototypes c from $\hat{z}$ under the supervision of $\hat{y}$. Note that we only store c_s and $c_{s\to t}$ into feature dictionaries, since $\hat{y}_s$ and $\hat{y}_{s\to t}$ are trained under supervision of ground truth y_s, and we expect to adapt features from $c_{t\to s}$, c_t to c_s, $c_{s\to t}$. During inference, we get the final output by directly averaging the target segmentation result $\hat{y}_t$ and the target-to-source segmentation result $\hat{y}_{t\to s}$.

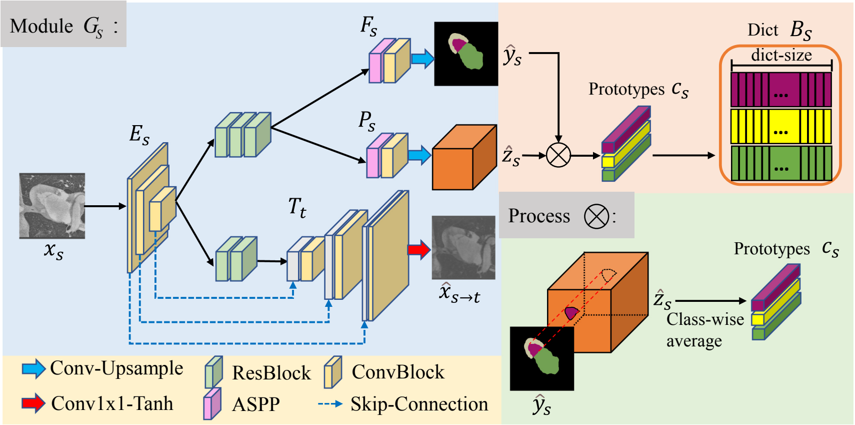

Fig. 3. Details of G_S and prototypes c_s. G_S consists of a feature encoder E_s, an image generator T_t, and symmetric heads F_s and P_s. These heads have same structures but different output channels. F_s outputs the segmentation results $\hat{y}_s$, while P_s outputs the embedded representation $\hat{z}_s$. The prototypes c_s are obtained by performing class-wise average on $\hat{z}_s$ under supervision of $\hat{y}_s$. And we store these prototypes in Dictionary B_s.

To make it clear, we show the detailed structure of G_S and the process to get prototypes c_s in Fig. 3. We introduce skip-connections to the image translation branch to help model convergence and image structure preservation [13]. The segmentation head C_s and projection head P_s have the same structure but different output channels. This symmetrical design is proved effective for semantic feature extraction and regularization [7]. We obtain prototypes c_s from projection z_s with the supervision from $\hat{y}_s$ by performing a class-wise average operation, and then we store c_s into feature dictionary B_s.

In Fig. 2, we denote the loss using the brown and red dash lines. Following GAN-based UDA methods [2,4,9], we use the cycle consistency loss L_{cycle}, segmentation loss L_{seg} and adversarial loss L_{adv}^{img}, L_{adv}^{seg} (see brown dash lines). Additionally, we design a class-wise similarity loss L_{sim} to promote intra-class consistency and inter-class discrepancy at feature level. L_{sim} is calculated between the projection $\hat{z}$ and the prototype c, which is calculated using $\hat{z}$ and $\hat{y}$. Besides, the contrastive loss L_{cl} is used to align prototypes across domains, further reducing the domain gap and improving the model performance. The calculation of L_{cl} needs the prototypes from feature dictionaries. At each iteration, prototypes c_s and $c_{s\to t}$ are first used to calculate L_{sim} and L_{cl}, then stored into Dict B_s and B_t, respectively.

2.3 Feature Prototypes and Class-Wise Similarity Loss

It is observed that the features of the same category tend to be clustered together [19], but the features across different domains have significant discrepancies.

To solve this issue, we regard class-wise prototypes as centers, and explicitly align features to their prototypes. As a result, prototypes of target domain are aligned to those of source domain.

Feature Prototypes. Following [6,7], we calculate the class-wise prototypes in a similar way, and the difference is that we use network segmentation result $\hat{y}$ instead of ground truth as supervision. As shown in Fig. 3, we get prototypes c_s by performing a class-wise average operation on $\hat{z}_s$ under supervision of $\hat{y}_s$. This procedure can be formulated as:

$$c_s^m = \frac{1}{N_m} \sum_{i=1}^{H \times W} \delta\left(\hat{y}_s[i], m\right) \hat{z}_{s,i}, \tag{1}$$

where c_s^m denotes the prototype of the m-th category from x_s, N_m denotes the total pixels of the m-th category, $\delta\left(\hat{y}_s[i], m\right) = 1$ if the i-th pixel of $\hat{y}_s$ belongs to category m, and $\hat{z}_s$ is the embedded representation.

Class-Wise Similarity Loss. We propose a cosine similarity based loss L_{sim} to explicitly regularize features in the embedding space, and we impose the constraint to $\hat{z}$. Taking $\hat{z}_s$ for example, the proposed loss L_{sim} is the summation of the following L_{sc} and L_{dc}.

$$L_{sc} = \frac{1}{C} \sum_{m=1}^{C} \frac{1}{N_m} \sum_{i=1}^{H \times W} \delta(\hat{y}_s[i], m) \left(1 - \cos sim\left(c_s^m, \hat{z}_{s,i}\right)\right), \tag{2}$$

$$L_{dc} = \frac{1}{N_c} \sum_{m=1}^{C} \sum_{n=m+1}^{C} \left(1 + \cos sim(c_s^m, c_s^n)\right), \tag{3}$$

where $\cos sim(u, v) = \frac{u^T v}{\|u\| \|v\|}$ denotes the cosine similarity, C denotes the number of categories in the image, $N_c = \frac{C!}{2!(C-2)!}$ denotes the number of combinations. L_{sc} becomes minimal when the similarity between category prototype c_s^m and representation $\hat{z}_{s,i}$ is maximal. This aims to minimize the intra-class feature discrepancy. In L_{dc}, the similarity is calculated between prototypes of different classes, and it becomes minimal when these prototypes are as dissimilar as possible. This aims to maximize inter-class variance.

For target domain images, since the $\hat{y}_t$ and $\hat{y}_{t \to s}$ are trained without ground truth supervision, we use the pixel-wise predictions of high confidence to supervise $\hat{z}_t$ and $\hat{z}_{t \to s}$, and get the prototypes c_t and $c_{t \to s}$. The similar idea has already been proven successful in pseudo-labeling [20].

2.4 Contrastive Loss via Feature Dictionaries

To align features across domains and boost representative embedded projection, we use dictionaries to store class-wise prototypes from various images, which avoids the category missing problem and enables contrastive learning.

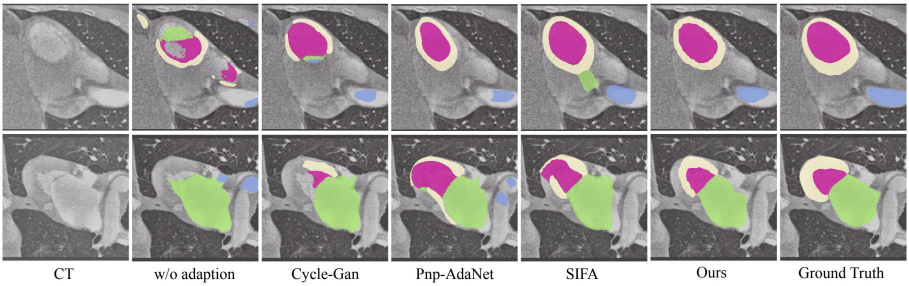

Fig. 4. Visual comparison of representative methods. The structures of MYO, LAC, LVC, AA are denoted by yellow, green, red, blue colors, respectively. (Color figure online)

Feature Dictionaries. In our framework, Dict B_s and B_t are used to store prototypes from x_s and $x_{s\rightarrow t}$, respectively. Following [8], each dictionary has category labels as the keys and the values of each key are prototypes. We denote B_s^c as the source domain dictionary accessed with category key c, and its shape is $[depth \times dict\_size]$. B_s and B_t are updated at every iteration, and old prototypes will be de-queued if the dictionary is full.

Contrastive Loss of Prototypes. Taking c_s^m (category m from source image x_s) as an example, we first calculate the cosine similarity between c_s^m and all prototypes stored in the dictionary B_s. Then for each category, we calculate the average of the highest k similarity values. And the contrastive loss can be formulated as:

$$[v_1^{m,c}, v_2^{m,c}, ..., v_L^{m,c}] = \cos sim(c_s^m, [d_{s,1}^c, d_{s,2}^c, ..., d_{s,L}^c]), \tag{4}$$

$$v^{m,c} = \frac{1}{k}\sum_{i=1}^{k} topk(v_1^{m,c}, v_2^{m,c}, ..., v_L^{m,c}), \tag{5}$$

$$L_{cl} = -\frac{1}{C}\sum_{m=1}^{C} \log \frac{\exp\left(v^{m,m}/\tau\right)}{\sum\limits_{i=1,i\neq m}^{C} \exp\left(v^{m,i}/\tau\right) + \exp\left(v^{m,m}/\tau\right)}, \tag{6}$$

where $v_i^{m,c}$ denotes the cosine similarity between c_s^m and $d_{s,i}^c$, $d_{s,i}^c$ denotes i-th value from category c of dictionary B_s, and τ is the temperature factor.

The contrastive loss not only makes the representation discriminative in embedding space, but also pulls target features closer to the source. Thus both domains are explicitly aligned at the feature level.

Overall Objectives. Following [2,4,9], the widely used cycle consistency loss L_{cycle}, segmentation loss L_{seg} and adversarial loss L_{adv}^{img}, L_{adv}^{seg} are also employed in our training process, we denote them as L_{base}. By adding our proposed similarity-base losses, and the overall objectives can be formulated as:

$$L_{all} = L_{base} + \lambda_1 L_{sim} + \lambda_2 L_{cl}, \tag{7}$$

where λ_1, λ_2 are balance parameters.

Table 1. Results of the MRI → CT task for four cardiac structures on *MMWHS*.

Methods	Volumetric dice↑					Volumetric ASD↓				
	AA	LAC	LVC	MYO	Avg.	AA	LAC	LVC	MYO	Avg.
Supervised training	92.7	91.1	91.9	87.7	90.9	1.5	3.5	1.7	2.1	2.2
W/o adaptation	28.4	27.7	4.0	8.7	17.2	20.6	16.2	-	48.4	-
PnP-AdaNet [16]	74.0	68.9	61.9	50.8	63.9	12.8	6.3	17.4	14.7	12.8
SynSeg-Net [23]	71.6	69.0	51.6	40.8	58.2	11.7	7.8	7.0	9.2	8.9
AdaOutput [22]	65.2	76.6	54.4	43.6	59.9	17.9	5.5	5.9	8.9	9.6
CycleGAN [12]	73.8	75.7	52.3	28.7	57.6	11.5	13.6	9.2	8.8	10.8
CyCADA [21]	72.9	77.0	62.4	45.3	64.4	9.6	8.0	9.6	10.5	9.4
EntMin [5]	83.0	**81.3**	67.2	58.4	72.5	**2.9**	**2.7**	6.3	6.4	**4.6**
SIFAv1 [3]	81.1	76.4	75.7	58.7	73.0	10.6	7.4	6.7	7.8	8.1
SIFAv2 [2]	81.3	79.5	73.8	61.6	74.1	7.9	6.2	5.5	8.5	7.0
DSFN [4]	**84.7**	76.9	79.1	62.4	75.8	-	-	-	-	-
Ours	82.6	**81.3**	**81.7**	**64.3**	**77.5**	6.1	6.0	**3.6**	**4.8**	5.1

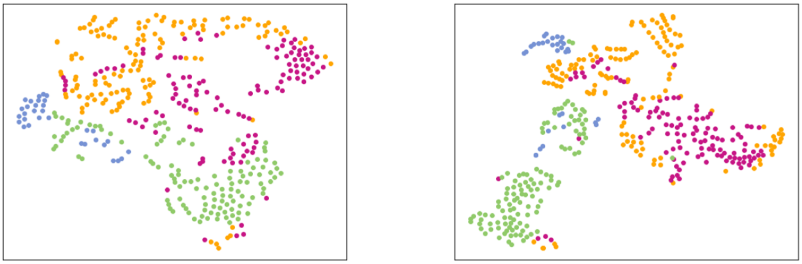

Fig. 5. t-SNE visualization of foreground features in Fig. 4. **Left:** The results without the proposed losses, **Right:** The results with the proposed losses.

3 Experiments

3.1 Datasets and Details

The proposed method is validated on the *Multi-Modality Heart Segmentation Challenge 2017* (MMWHS) dataset [15], which consists 20 unpaired MR and CT volumes data with their pixel-level ground truth of heart structures. The left ventricle blood cavity (LVC), the left atrium blood cavity (LAC), the myocardium of the left ventricle (MYO) and the ascending aorta (AA) are usually selected to evaluate the model segmentation performance. For a fair comparison, we use

Table 2. Ablation study of the proposed losses

Components		Dice↑					ASD↓				
L_{sim}	L_{cl}	AA	LAC	LVC	MYO	Avg.	AA	LAC	LVC	MYO	Avg.
×	×	79.7	82.9	77.5	58.1	74.5	7.4	**4.3**	4.1	**4.3**	**5.0**
✓	×	77.7	**83.0**	78.0	61.8	75.2	9.5	5.0	4.8	4.7	6.0
✓	✓	**82.6**	81.3	**81.7**	**64.3**	**77.5**	**6.1**	6.0	**3.6**	4.8	5.1

Table 3. Ablation study of three methods to utilize feature dictionaries

Methods	Dice↑				
	AA	LAC	LVC	MYO	Avg.
Max Similarity	79.1	84.1	78.3	61.2	75.7
Mean All	80.5	**82.1**	79.1	62.7	76.1
Mean Top-k	**82.6**	81.3	**81.7**	**64.3**	**77.5**

Table 4. Ablation study of dict sizes S

S	Dice↑				
	AA	LAC	LVC	MYO	Avg.
w/o	77.7	83.0	78.0	61.8	75.2
200	81.1	**84.8**	81.6	58.4	76.5
400	**82.6**	81.3	**81.7**	**64.3**	**77.5**
600	80.3	82.8	80.3	63.6	76.8

the preprocessed data released by [2,16], which contains randomly selected 16 volumes for training and 4 volumes for testing for both modalities. All data were first normalized to zero-mean and unit standard deviation, and then switched to $[-1, 1]$. Each slice was cropped and resized to the size of 256×256. These data were also augmented by rotation, scaling, and affine transformations.

Implementations. The discriminators follow patchGAN [17], except that we replace log objective with least-squares loss for a stable training [14]. We empirically set $\lambda_1 = 0.05$, $\lambda_2 = 0.02$, and the dictionary size S is set to 400, temperature τ is set to 1, while top-k is set to 20. Batch size and training epoch are set to 4 and 35, respectively. Besides, we use Adam optimizer [18] with weight decay of 1×10^{-4}, and the learning rate for discriminators is set to 2×10^{-4}, while 3×10^{-4} for G_S and G_T. To warm up training, we apply our proposed loss after the first epoch, and our model is trained on a NVIDIA Tesla V100 with PyTorch.

3.2 Results and Analysis

Quantitative and Qualitative Analysis. Table 1 shows the MRI→CT adaption performance comparison with other methods. Since our experiment is conducted under the same setting as [2,16], we directly refer to their paper results. As shown in Table 1, the model without adaption gets a poor performance on the unseen target domain. Methods based on image-alignment [12,21,23] and methods based on feature-alignment [5,16,22] can significantly improve the model results by narrowing the domain gap. [2,4] further improve the performance by taking both perspectives into account. Our proposed method outperforms these methods in terms of dice, and achieves an average result of 77.5%, besides, we achieve an average ASD of 5.1, which is slightly worse than EntMin [5]. This indicates that our generated results may be not smooth on the boundary regions, while EntMin [5] conduct the entropy minimization to deal with high uncertainty of the boundary. Figure 4 shows the visual comparison, and we choose [2,12,16] as the representative methods from different alignment perspectives. We visualize the feature distribution of Fig. 4 using t-SNE [25] in Fig. 5.

Ablation Study. Firstly, we evaluate the effectiveness of the proposed losses. As shown in Table 2, when neither of the proposed losses were used, our method can be seen as a variant of [4], except that we redesign network structure to utilize low-level features to help image-translation and we do not use auxiliary task for feature adaption. In this case, our method achieves an average Dice of 74.5%. When only the class-wise similarity loss is used, the result gets a gain of 0.7%. When both losses are applied, our model achieves an average dice of 77.5%, surpassing other methods by a large margin.

Secondly, we test on several strategies to utilize prototypes in dictionaries. **Mean Top-k** means taking average of the top-k similarity values. **Mean All** means taking average of all similarity values. **Max Similarity** means only use the largest similarity value. Table 3 shows the results, and we can find that **Mean Top-k** achieves the best performance. This may due to the fact that sampling average can improve the robustness of similarity calculation, and the similarity will not be generalized to much.

Thirdly, we consider different dictionary sizes S. Table 4 shows the results, we can achieve the best performance when $S = 400$, which indicates that an appropriate dictionary size is necessary. A small dictionary may not have sufficient feature diversity, while a big dictionary may induce a slow updating of L_{cl}, as we calculate the average similarity using the top-k strategy.

4 Conclusion

This paper proposes a novel unsupervised domain adaption framework for medical image segmentation. The framework is a unified network that can be trained end-to-end. We propose an innovative class-wise loss (calculated within a single sample) to boost feature consistency and learn representative prototype. Moreover, we conduct contrastive learning of prototypes (calculated with prototypes of multiple samples) to further improve feature adaption across domains. Compared with existing adversarial learning based methods, we explicitly align features. Extensive experiments prove the effectiveness of our method, and show the superiority of the class-wise similarity loss and prototype contrastive learning via dictionary. In the future, we will test our method with different datasets and explore to apply it to domain generalization task.

References

1. Xue, Y., Feng, S., Zhang, Y., Zhang, X., Wang, Y.: Dual-task self-supervision for cross-modality domain adaptation. In: Martel, A.L., et al. (eds.) MICCAI 2020. LNCS, vol. 12261, pp. 408–417. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59710-8_40
2. Chen, C., Dou, Q., Chen, H., et al.: Unsupervised bidirectional cross-modality adaptation via deeply synergistic image and feature alignment for medical image segmentation. IEEE Trans. Med. Imag. **39**(7), 2494–2505 (2020)
3. Chen, C., Dou, Q., Chen, H., et al.: Synergistic image and feature adaptation: towards cross-modality domain adaptation for medical image segmentation. In: AAAI, vol. 33 no. 01, pp. 865–872 (2019)
4. Zou, D., Zhu, Q., Yan, P.: Unsupervised domain adaptation with dual-scheme fusion network for medical image segmentation. In: IJCAI, pp. 3291–3298 (2020)
5. Vesal, S., Gu, M., Kosti, R., et al.: Adapt everywhere: unsupervised adaptation of point-clouds and entropy minimisation for multi-modal cardiac image segmentation. IEEE Trans. Med. Imag. (2021). https://ieeexplore.ieee.org/document/9380742
6. Liu, Z., Zhu, Z., Zheng, S., et al.: Margin Preserving Self-paced Contrastive Learning Towards Domain Adaptation for Medical Image Segmentation. arXiv preprint arXiv:2103.08454 (2021)
7. Marsden, R.A., Bartler, A., Döbler, M., et al.: Contrastive Learning and Self-Training for Unsupervised Domain Adaptation in Semantic Segmentation. arXiv preprint arXiv:2105.02001 (2021)
8. Chung, I., Kim, D., Kwak, N.: Maximizing Cosine Similarity Between Spatial Features for Unsupervised Domain Adaptation in Semantic Segmentation. arXiv preprint arXiv:2102.13002 (2021)
9. Tomar, D., Lortkipanidze, M., Vray, G., et al.: Self-attentive spatial adaptive normalization for cross-modality domain adaptation. IEEE Trans. Med. Imag. (2021). https://ieeexplore.ieee.org/document/9354186
10. Chen, Y.C., Lin, Y.Y., Yang, M.H., et al.: Crdoco: pixel-level domain transfer with cross-domain consistency. In: CVPR, pp. 1791–1800 (2019)
11. Wang, J., Huang, H., Chen, C., Ma, W., Huang, Y., Ding, X.: Multi-sequence cardiac MR segmentation with adversarial domain adaptation network. In: Pop, M., et al. (eds.) STACOM 2019. LNCS, vol. 12009, pp. 254–262. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-39074-7_27
12. Zhu, J.Y., Park, T., Isola, P., et al.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV., pp. 2223–2232 (2017)
13. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
14. Mao, X., Li, Q., Xie, H., et al.: Least squares generative adversarial networks. In: CVPR, pp. 2794–2802 (2017)
15. Zhuang, X., Shen, J.: Multi-scale patch and multi-modality atlases for whole heart segmentation of MRI. Med. Image Anal. **31**, 77–87 (2016)
16. Dou, Q., Ouyang, C., Chen, C., et al.: PnP-AdaNet: plug-and-play adversarial domain adaptation network at unpaired cross-modality cardiac segmentation. IEEE Access **7**, 99065–99076 (2019)

17. Isola, P., Zhu, J.Y., Zhou, T., et al.: Image-to-image translation with conditional adversarial networks. In: CVPR, pp. 1125–1134 (2017)
18. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Zhang, Q., Zhang, J., Liu, W., et al.: Category anchor-guided unsupervised domain adaptation for semantic segmentation. arXiv preprint arXiv:1910.13049 (2019)
20. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. In: CVPR, pp. 6936–6945 (2019)
21. Cycada: cycle-consistent adversarial domain adaptation. In: ICML, pp. 1994–2003 (2018)
22. Tsai, Y.H., Hung, W.C., Schulter, S., et al.: Learning to adapt structured output space for semantic segmentation. In: CVPR, pp. 7472–7481 (2018)
23. Huo, Y., Xu, Z., Moon, H., et al.: Synseg-Net: synthetic segmentation without target modality ground truth. IEEE Trans. Med. Imag. **38**(4), 1016–1025 (2018)
24. Shen, D., Wu, G., Suk, H.I.: Deep learning in medical image analysis. Ann. Rev. Biomed. Eng. **19**, 221–248 (2017)
25. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(11) (2008). https://www.jmlr.org/papers/v9/vandermaaten08a.html