Published in final edited form as:
Med Image Anal. 2021 August ; 72: 102136. doi:10.1016/j.media.2021.102136.

Autoencoder Based Self-Supervised Test-Time Adaptation for Medical Image Analysis

Yufan He[a,*], Aaron Carass[a], Lianrui Zuo[a,b], Blake E. Dewey[a], Jerry L. Prince[a]

[a]Dept. of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218, USA

[b]Laboratory of Behavioral Neuroscience, National Institute on Aging, National Institute of Health, Baltimore, MD 20892, USA

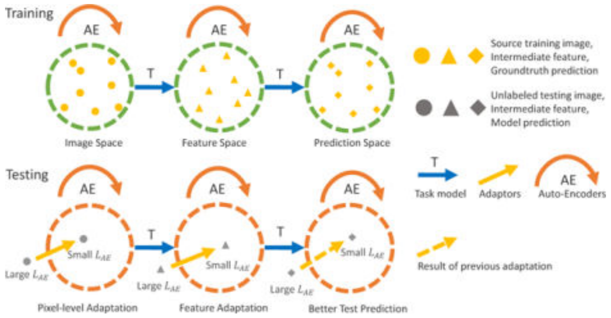
Abstract

Deep neural networks have been successfully applied to medical image analysis tasks like segmentation and synthesis. However, even if a network is trained on a large dataset from the source domain, its performance on unseen test domains is not guaranteed. The performance drop on data obtained differently from the network's training data is a major problem (known as domain shift) in deploying deep learning in clinical practice. Existing work focuses on retraining the model with data from the test domain, or harmonizing the test domain's data to the network training data. A common practice is to distribute a carefully-trained model to multiple users (e.g., clinical centers), and then each user uses the model to process their own data, which may have a domain shift (e.g., varying imaging parameters and machines). However, the lack of availability of the source training data and the cost of training a new model often prevents the use of known methods to solve user-specific domain shifts. Here, we ask whether we can design a model that, once distributed to users, can quickly adapt itself to each new site without expensive retraining or access to the source training data? In this paper, we propose a model that can adapt based on a single test subject during inference. The model consists of three parts, which are all neural networks: a task model (T) which performs the image analysis task like segmentation; a set of autoencoders (AEs); and a set of adaptors (As). The task model and autoencoders are trained on the source dataset and can be computationally expensive. In the deployment stage, the adaptors are trained to transform the test image and its features to minimize the domain shift as measured by the autoencoders' reconstruction loss. Only the adaptors are optimized during the testing stage with a single test subject thus is computationally efficient. The method was validated on both retinal optical coherence tomography (OCT) image segmentation and magnetic resonance imaging (MRI) T1-weighted to T2-weighted image synthesis. Our method, with its short optimization time for the adaptors (10 iterations on a single test subject) and its additional required disk space for the autoencoders (around 15 MB), can achieve significant performance improvement. Our code is publicly available at: https://github.com/YufanHe/self-domain-adapted-network.

[*]Corresponding author: yhe35@jhu.edu.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Graphical Abstract



2000 MSC

41A05; 41A10; 65D05; 65D17; 68T45

Keywords

Unsupervised Domain Adaptation; Self Supervised Learning; Medical Image Analysis; Test Time Adaptation

1. Introduction

Deep learning has been successfully applied in medical image analysis tasks like segmentation and synthesis. It has achieved the best performance in many major medical segmentation challenges (Carass et al., 2018; Simpson et al., 2019; Isensee et al., 2019; Kavur et al., 2020), and also prevailed in medical image synthesis tasks like CT to MRI (Zhao et al., 2017; Yang et al., 2018), multi-contrast MRI synthesis (Dar et al., 2019; Sharma and Hamarneh, 2019), and medical image super-resolution (Zhao et al., 2019b; Oktay et al., 2016). The high performance of deep networks comes from learning a "*good*" mapping between paired training input and output data, and implicitly assumes that the network input training and testing data are from the same distribution. However, this may not hold in real clinical practice. In multi-center studies, for example, each center may have scanners from different manufacturers and the imaging parameters can even vary from subject to subject. The deep network will have significant performance drop on those scans which are acquired differently to the training data. This domain shift problem is a major problem in deploying deep networks for multi-site studies.

To increase the robustness of deep networks to domain shifts, existing work can be separated into two categories: domain generalization and unsupervised domain adaptation (UDA). Domain generalization tries to train the model on a larger and more diverse source training set, such that the model can be more robust and generalize well to unseen domains. Dou et al. (2019) trains a more generalizable model on multi-site MRI images with Model-Agnostic Meta-Learning. Mahapatra et al. (2020), Zhao et al. (2019a), Billot et al. (2020), and Zhang et al. (2020) use data augmentation or simulated training data to train a more robust model to be used on the target test domain. With more diverse training data and a better training strategy, the space of input images that can be successfully processed by the model is enlarged. However, obtaining or simulating realistic paired training data for every scanner and imaging parameter is expensive or even infeasible, and there is no guarantee that the trained robust model can generalize well on an unseen test dataset.

Unsupervised domain adaptation (UDA) attempts to reduce the domain shift between a specific unlabeled test domain and source domain. The first type is pixel-level harmonization, where the test domain images are transformed to be similar to source domain images. It includes simple operations like histogram matching, deep learning based style transfer (Gatys et al., 2016) and unpaired image-to-image translation like CycleGAN (Zhu et al., 2017) and UNIT (Liu et al., 2017). Liu et al. (2020) and Ma et al. (2019) used style transfer to improve cardiovascular MR and ultrasound image segmentation, respectively. Seeböck et al. (2019) and Modanwal et al. (2020) applied a CycleGAN to harmonize OCT images and MRI images from different scanners, respectively. As shown in Fig. 1 (a), pixel-level harmonization does not require the subsequent analysis model, and thus can be more flexible when applied to non-deep learning based subsequent analysis methods. However, the similarity measurement between the harmonized target image and source image is not optimal because of ignorance of the subsequent task; thus, the information that is important for the subsequent task might be distorted (e.g., anatomy hallucination in CycleGAN (Cohen et al., 2018) for subsequent segmentation). Moreover, more complicated methods like CycleGAN must be trained for each test domain separately, which is computationally expensive and requires a large amount of both source and test data for training.

Alternatively, a second type of UDA retrains the model with labeled source training data and unlabeled test domain data. Although the test domain data has no groundtruth label, it can be used to regularize the learned features. The goal is to train the model to learn domain invariant (indistinguishable) features for source and test data. As labeled source data is used for model training, the features from the source distribution will produce correct predictions. Since the feature distribution for test and source is domain-invariant, the test data is assumed to be processed successfully. The similarity measurement for the test and source features can be maximum mean discrepancy (Long et al., 2015), domain classifier (Ganin and Lempitsky, 2015) and adversarial learning (Tzeng et al., 2017; Tsai et al., 2018; Dou et al., 2018; Hoffman et al., 2018), which can be used for model training, as shown in Fig. 1 (c). However, those methods must retrain the model for each test domain, and therefore require access to labeled source training and a large amount of test domain data.

In many scenarios, the test data is available only when the model is deployed by users, which means that the expensive training on the source dataset has already been completed. As shown in Fig. 1 (b), when a user deploys a pre-trained model on user specific test data, the user may not have access to the source dataset and if so may not have the computation resources to retrain a expensive model. If we only have the pre-trained model and the domain shifted test data, how can we improve the test results? Adapting a pre-trained model to the test data was first proposed by Jain and Learned-Miller (2011) using Gaussian process regression. Recently, Sun et al. (2020), Karani et al. (2020), Wang et al. (2020), and He et al. (2020c) have developed test-time adaptation for deep networks, none of which require heavy retraining or access to the source data. Sun et al. (2020) adapt part of the pre-trained classification model to a single unlabeled image using a self-supervised rotation prediction task. Wang et al. (2020) adapt the model to the test data by optimizing the normalization layer parameters to minimize the test data prediction entropy. Karani et al. (2020) optimize a shallow intensity normalization layer of the pre-trained model to minimize the denoising autoencoder reconstruction loss for medical image segmentation.

In this paper, we propose a unified self-supervised test-time adaptation model for medical image analysis, addressing both synthesis and segmentation tasks in different source image modalities. This work presents an extension to our previous conference paper (He et al., 2020c). We provide a more detailed interpretation of our method in the context of recent work in the area. We also provide extensive experiments including new comparisons to unsupervised domain adaptation methods Tsai et al. (2018), recently published test-time adaptation methods Karani et al. (2020); Sun et al. (2020); Wang et al. (2020) and multiple new ablation studies. Our model includes three parts: 1) a task model (T) which performs medical image analysis tasks of synthesis or segmentation, 2) a set of autoencoders (AEs) which are used as a similarity measurement, 3) a set of adaptors (As) which are used to transform test image and features to be similar to the source. The task model and autoencoders are trained on labeled source training data and kept fixed during deployment. Conventionally,a test subject is processed by the pre-trained task model through a single forward pass during deployment, but the result may not be satisfactory if the test subject exhibits domain shift. Instead of a single forward pass, our method optimizes the adaptors to alleviate domain shift by transforming the test image and its features (from the fixed pre-trained task model) to be close to the source. To do this, we use the idea of anomaly detection by using autoencoders in the following way: normal input will have lower autoencoder reconstruction error than abnormal input Gong et al. (2019), where source training data is considered to be normal while domain-shifted test data is considered to be abnormal. By minimizing the reconstruction error, we assume that the domain shift is alleviated by the adaptors. After the quick adaptation, we generate test image prediction through a single forward pass of the combination of the adaptors and the task model. Compared to conventional deployment of deep networks, the cost of our method is a short adaptation time for each test subject (about 15 s in our experiments) and space for autoencoders (about 15 MB disk size). Karani et al. (2020) also uses autoencoders to optimize an adaptor. However, they use a single denoising autoencoder to learn segmentation label priors and then optimize a pixel-level normalization module, which is the first three layers of the task prediction model; we use autoencoders as anomaly detectors to measure the similarity between the test data and the source data. Our autoencoders and adaptors are multi-level, performing both pixel-level and multi feature-level alignment. We show that our framework works for both synthesis and segmentation on two different medical imaging modalities.

2. Method

The overall framework of our method is shown in Fig. 2. We first train the task model (T) on the labeled source dataset using any network training strategy. The task model, which is fixed after training, maps the source image to the source feature space and final prediction space, which are then used to train the autoencoders. At the testing stage, the task model and autoencoders are fixed, and the adaptors are optimized on a single test subject such that its autoencoders' reconstruction loss $\left(\mathscr{L}^t_{AE}\right)$ is minimized.

2.1. Task Model (T)

The task model performs medical image analysis tasks and can be any state-of-the-art model for a given task. In this work, we are not focusing on designing an optimal task model for a certain task, so we use a residual U-Net (He et al., 2019a,2020b), a variant of the popular U-Net (Ronneberger et al., 2015) for both synthesis and segmentation (where both the activation and number of channels of the output layer are different). As shown in Fig. 3, the model has three 2 × 2 max-pooling operations and down-samples the feature tensors into four resolution levels. The number of feature channels is fixed to 64 for every level.

2.2. Autoencoders (AEs)

We use a set of multi-level fully convolutional autoencoders $\{AE^x, \{AE^i\}_{i=1,2,3}, AE^y\}$ to encode the image level, feature levels, and output prediction level distributions of the source domain. The autoencoders' architecture are based on the task model T, but use two max-pooling instead of three and use instance normalization (Ulyanov et al., 2016) instead of batch normalization. The long skip connections as used in U-Net are removed. As shown in Fig. 3, AE^x and AE^y are trained to reconstruct the source image x_s and the prediction y'_s from T, respectively, encoding the information at the highest resolution level. For the intermediate features $\left\{f^i_S\right\}_{i=1,\cdots,6}$ (all 64 channels) at the three lower resolution levels from T, we train $\{AE^i\}_{i=1,2,3}$ to reconstruct them. The input to AE^i is the concatenation of features f^i_S and f^{7-i}_s at the same resolution level i. The number of encoder feature channels of $\{AE^i\}_{i=1,2,3}$ are 64, 32, 16 (reverse for the decoder), and 32, 16, 8 for AE^x and AE^y.

2.3. Offiine training

The task model and autoencoders are trained with labeled source dataset x_s, y_s, where x_s is the image data and y_s is the ground truth label. For synthesis and segmentation, y_s has the same spatial resolution as x_s. We first train the task model T's parameters W_T using conventional gradient decent to minimize a segmentation or synthesis loss $\mathscr{L}_{\mathcal{T}}(T(x_s; W_T), y_s)$. After T is trained, we freeze W_T and generate the intermediate features $\left\{f^i_S\right\}_{i=1,\cdots,6}$ and output prediction y'_s (without softmax) of x_s from T. The autoencoders are trained by minimizing the reconstruction loss $\mathscr{L}^s_{AE}$ where $\{\cdot\}$ is the feature concatenation in the channel dimension.

$$\begin{aligned} & y'_s, \left\{f^i_s\right\}_{i=1,\cdots,6} = T(x_s; W_T) \\ & \mathscr{L}^s_{AE} = \sum_{i=1}^{3} \left|AE^i\left(\left\{f^i_s, f^{7-i}_s\right\}\right) - \left\{f^i_s, f^{7-i}_s\right\}\right|_2^2 + \left|AE^x(x_s) - x_s\right|_2^2 + \left|AE^y(y'_s) - y'_s\right|_2^2 \end{aligned} \tag{1}$$

The training strategies for a more robust and generalizable task model as mentioned in Sec. 1 can also be applied in this step to boost performance and have no conflicts with our proposed test-time adaptation. The offline training for T and AEs on all the source dataset can be computationally expensive, but we only need to train them once.

2.4. Domain Adaptors

As shown in Fig. 2, the adaptors adapt the test data to the source domain in both the pixel-level and feature-level to improve the final prediction. We use a pixel-level adaptor A^x and three feature-level adaptors $\{A^i\}_{i=1,2,3}$ for the intermediate features. As shown in Fig. 4, A_x transforms input test image x_t to minimize the reconstruction loss from AE^x, while $\{A^i\}_{i=1,2,3}$ transforms the intermediate features $\left\{f_t^i\right\}_{i\,=\,1,\,\cdots,\,3}$ (extracted from x_t using T's encoder) to minimize the reconstruction loss from $\{AE^i\}_{i=1,2,3}$, respectively. Note that the adaptors at the front will affect the features generated later. The features $\left\{f_t^i\right\}_{i\,=\,4,\,\cdots,\,6}$ of T's decoder will be subsequently changed by the adaptors. When deploying the trained T and AEs for a test subject x_t, we optimize the adaptors' parameters W_A by minimizing $\mathscr{L}^t_{AE}$:

$$\begin{aligned} & x_t', \left\{f_t^i\right\}_{i\,=\,1,\,\cdots,\,6}, y_t' = T\left(x_t, A^x, \left\{A^i\right\}_{i\,=\,1,\,2,\,3}; W_A, W_T\right) \\ & \mathscr{L}^t_{AE} = \sum_{i\,=\,1}^{3}\left|AE^i\left(\left\{f_t^i, f_t^{7\,-\,i}\right\}\right) - \left\{f_t^i, f_t^{7\,-\,i}\right\}\right|_2^2 + \left|AE^x(x_t') - x_t'\right|_2^2 + \left|AE^y(y_t') - y_t'\right|_2^2 \end{aligned} \tag{2}$$

However, the transformations that the adaptors can represent must be carefully designed. With only a single unlabeled test subject and pre-trained T and AEs, we must address the following challenges. 1) Without direct supervision, how can we avoid feature hallucination (Cohen et al., 2018) and geometry shift (or blurriness with directions due to convolutions) (Yang et al., 2020), which might be introduced by the adaptors transformation? 2) How can we make the adaptors trainable with a single test subject? 3) Autoencoders have strong generalization capability, especially when its training images are complicated (Pidhorskyi et al., 2018; Abati et al., 2019); thus, the transformed test features can have low reconstruction error even far from the source. Moreover, the adaptors trained by minimizing reconstruction loss may lead to mode collapse as shown in Fig. 5.

For the first two problems, we design our adaptors to be as simple as possible: our pixel level adaptor A^x is a pure histogram manipulator, and $\{A^i\}_{i=1,2,3}$ are 1×1 convolutions. A^x consists of three 1×1 convolutions (output channel numbers are 64, 64, 1) and each is followed by a LeakyReLU and Instance normalization (Ulyanov et al., 2016). Karani et al. (2020) also use a simple pixel level adaptation module, which forms the first three convolution layers (kernel size 3×3) of the task prediction model. They rely on this relatively simple pixel level adaptor to solve the domain shift; however, this design choice will cause blurriness on the pixel-level adapted image and can be detrimental to the synthesis task. The directional blur from convolutions can also cause anatomy boundary shift and affect boundary sensitive segmentation tasks like retinal layer segmentation. Trying to solve the domain shift problem only from the pixel domain requires the adaptor to extract high-level information from the raw image which requires stronger supervision and more training data. Instead, we only use a very simple pixel-level adaptor, which permits only low-level image statistics (like histogram) to be modified in the pixel domain. High-level information is extracted from the task model and separately matched using $\{A^i\}_{i=1,2,3}$.

For the third problem with the use of autoencoders, we limit our application to relatively minor shifts of domain and make the assumption that the task model can extract features from the test image that are close to the features extracted from source images. Given this assumption, to alleviate the feature mode collapse problem, as shown in Fig. 5, we further limit the transformation of the feature adaptors $\{A^i\}_{i=1,2,3}$. A^i is a 1 × 1 convolution with 64 input and 64 output channels, and its parameter W_{A^i} contains a 64 × 64 linear transformation matrix $W_{A^i}^k$ and a 1D bias vector $W_{A^i}^b$ of length 64. For a feature map f_t^i with 64 channels, each pixel on the feature map has an 1D feature with length 64. For two 1D features $f_t^{i,a}$, $f_t^{i,b}$ at pixel a,b, the transformations by A^i are $W_{A^i}^k f_t^{i,a} + W_{A^i}^b$, $W_{A^i}^k f_t^{i,b} + W_{A^i}^b$. The test feature is supposed to be transformed closer to the source, but the discriminative structure within the test feature should be preserved. We therefore want to keep the L_2 distance between $f_t^{i,a}$, $f_t^{i,b}$ for every pair of pixels a,b before and after transformation as in

$$\begin{aligned}&\left(f_t^{i,a} - f_t^{i,b}\right)^T\left(f_t^{i,a} - f_t^{i,b}\right) = \left(f_t^{i,a} - f_t^{i,b}\right)^T\left(W_{A^i}^k\right)^T W_{A^i}^k\left(f_t^{i,a} - f_t^{i,b}\right)\\ &\Rightarrow \quad \left(W_{A^i}^k\right)^T W_{A^i}^k = I\end{aligned} \tag{3}$$

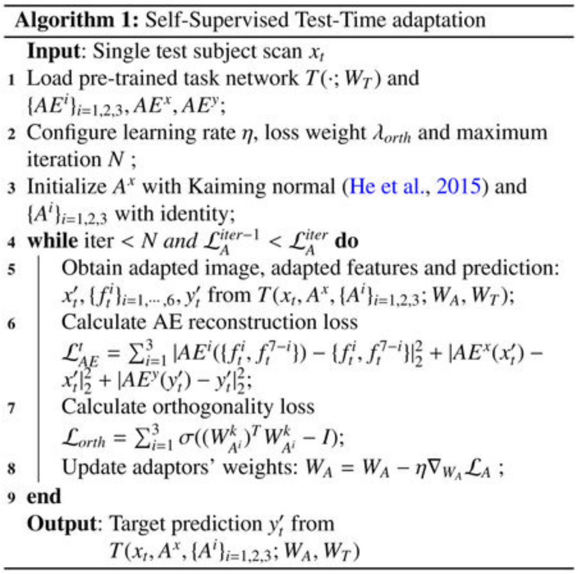
which requires the orthogonality of $W_{A^i}^k$. To do this, we apply Spectral Restricted Isometry Property Regularization (Bansal et al., 2018),which minimizes the spectral norm of $\left(W_{A^i}^k\right)^T W_{A^i}^k - I$ to impose the orthogonality of $W_{A^i}^k$. This is implemented using the orthogonal loss $\mathscr{L}_{orth}$ given by

$$\mathscr{L}_{orth} = \sum_{i=1}^{3} \sigma\left(\left(W_{A^i}^k\right)^T W_{A^i}^k - I\right) = \sum_{i=1}^{3} \underset{z \in R^n, z \neq 0}{sup} \left|\frac{\left\|W_{A^i}^k z\right\|^2}{\|z\|^2} - 1\right| \tag{4}$$

where the implementation details are in Bansal et al. (2018). We optimize the adaptors by minimizing $\mathscr{L}_A$ given by

$$\mathscr{L}_A = \mathscr{L}_{AE}^t + \lambda_{orth}\mathscr{L}_{orth} \tag{5}$$

Here, λ is used to balance the transformation complexity and the AE's reconstruction loss. After a quick optimization of the adaptors via minimizing $\mathscr{L}_A$, we perform a conventional forward pass using the combination of task model and adaptors. The overall algorithm is summarized in Alg. 1.



```
Algorithm 1: Self-Supervised Test-Time adaptation
Input: Single test subject scan $x_t$
Load pre-trained task network $T(\cdot; W_T)$ and $\{AE^i\}_{i=1,2,3}, AE^x, AE^y$;
Configure learning rate $\eta$, loss weight $\lambda_{orth}$ and maximum iteration $N$ ;
Initialize $A^x$ with Kaiming normal (He et al., 2015) and $\{A^i\}_{i=1,2,3}$ with identity;
while iter $< N$ and $\mathcal{L}_A^{iter-1} < \mathcal{L}_A^{iter}$ do
    Obtain adapted image, adapted features and prediction: $x_t', \{f_t^i\}_{i=1,\cdots,6}, y_t'$ from $T(x_t, A^x, \{A^i\}_{i=1,2,3}; W_A, W_T)$;
    Calculate AE reconstruction loss
    $\mathcal{L}_{AE}^t = \sum_{i=1}^3 |AE^i(\{f_t^i, f_t^{7-i}\}) - \{f_t^i, f_t^{7-i}\}|_2^2 + |AE^x(x_t') - x_t'|_2^2 + |AE^y(y_t') - y_t'|_2^2$;
    Calculate orthogonality loss
    $\mathcal{L}_{orth} = \sum_{i=1}^3 \sigma((W_{A^i}^k)^T W_{A^i}^k - I)$;
    Update adaptors' weights: $W_A = W_A - \eta \nabla_{W_A} \mathcal{L}_A$ ;
end
Output: Target prediction $y_t'$ from
        $T(x_t, A^x, \{A^i\}_{i=1,2,3}; W_A, W_T)$
```

3. Experiments

The proposed method was validated on two medical image analysis tasks: 1) retinal layer segmentation from OCT images. 2) T1-weighted (T1) to T2-weighted (T2) synthesis from MRI. Retinal layer segmentation is important for monitoring disease progression for both ophthalmological and neurological diseases (Saidha et al., 2011b). For example, retinal layer thicknesses are important biomarkers that change slowly over time in multiple sclerosis (MS) subjects (Saidha et al., 2011a). This means that a tiny geometry shift caused by pixel-level adaptation will be detrimental to the thickness measurement (He et al., 2020a) and a change in scanners or their protocols can interfere with the measurement of this change. For MRI, imaging the same anatomy with different imaging sequences can help in diagnosis or various image analysis tasks (Kabir et al., 2007). Due to the imaging time limitations or patient motion, some contrasts may not be available or may be corrupted by imaging artefacts and can be recovered by image synthesis.

3.1. OCT Dataset

We used a publicly available OCT dataset (He et al., 2019b) from a Heidelberg Spectralis scanner as our source training dataset. The dataset consists of 35 3D volumes, each of which contains 49 2D slices of size 496 × 1024. Since the physical distance (124 μm) between slices is 10 times larger than the within-slice resolution (3.9 μm, 5.8 μm), all our models were performed in 2D. Nine surfaces on each slice were manually delineated as segmentation ground-truth. The dataset was split into 588 slices for training, 147 slices for validation, and 980 for testing, yielding a fairly conventional split for deep network training (He et al., 2020b). A private dataset of OCT images obtained from a Cirrus scanner was used as the domain-shifted test dataset. It contains OCT scans from 6 subjects (each have 8 slices of size 1024×512, resolution 2.0×11.7μm, and with manual delineations of nine layer surfaces). For each subject, we ran Alg. 1 independently and report the segmentation results here. We first resampled the Cirrus 2D slices to the same physical within-slice resolution as the Spectralis scans. All the 2D slices from both scanners were flattened to the Bruch's membrane (He et al., 2020b) and cropped to size 128×1024. Examples of source Spectralis and test Cirrus images and manual segmentation are shown in Fig. 6.

3.2. MRI Dataset

We used the IXI dataset[1] as the test dataset. This dataset contains paired T1–T2 images from three clinical sites, Hammersmith Hospital (HH, T1 SPGR from Philips Intera 3T, T_R=9.6ms, T_E=4.6ms), Guys Hospital (GH, T1 SPGR from Philips Gyroscan Intera 1.5T, T_R=9.8ms, T_E=4.6ms), and the Institute of Psychiatry (IOP, GE 1.5T, unknown parameters). The first 30 3D scans from each site were used for testing. The source training dataset is a paired T1–T2 dataset acquired from Johns Hopkins Hospital (JHU, T1 MPRAGE from Philips Achieva 3T, T_R=3000ms, T_E=6ms). The JHU source training set was split into 30 training, 4 validation, and 15 testing 3D scans. All the source and test data were preprocessed with N4 inhomogeneity correction (Tustison et al., 2010), MNI space registration, and white matter peak normalization (Reinhold et al., 2019). We extracted 21 axial slices from each 3D scan (equally extracted from slice number 60 to 120, 3mm slice distance) and trained a 2D synthesis network on the source JHU dataset and tested each subject (21 slices from one scan) using Alg. 1 independently for the IXI dataset.

3.3. Baselines

The focus of the paper is on test-time adaptation, where only a model (with its trained weights from source) and user specific test data are given. Our method adapts to each single test subject without changing the model trained from source domain, so we compare our method to harmonization methods that do not require expensive retraining of the prediction model or need to use a specifically designed prediction model for domain-invariant feature learning (Ouyang et al., 2019; Dou et al., 2018). For OCT segmentation, we compare to (1) NA: Direct application of the task model without any adaptation, (2) Hist: 3 × 3 median filtering followed by histogram matching for OCT (Seeböck et al. , 2019) (3) ST[2]: Style transfer using pre-trained VGG19 (Gatys et al., 2016; Ma et al., 2019). (4) Cyc[3]: CycleGAN (Zhu et al., 2017); For (2) and (3), we use one slice from the source training set as the reference. CycleGAN needs enough training data from both source and test domains, and we used 588 slices from the Cirrus scanner (48 from the test set and 540 additional images) and the Spectralis training set (588 slices). For the T1 to T2 synthesis task, we compared to the same baseline methods as OCT, except that we removed the median filtering before histogram matching. When performing histogram matching and style transfer for the i-th slice in a test subject, we used the i-th slice from the first subject in the source dataset as a reference. For each test site in IXI, we trained a CycleGAN using all the test images (630 slices) and source training images (630 slices). Besides harmonization methods, we also compare to (5) DTTA: concurrent test-time adaptation method (Karani et al., 2020) which uses a denoising autoencoder on the segmentation maps to optimize a shallow normalization module (the first three convolution layers of the task prediction model). However, as the prediction model (Karani et al., 2020) has different architecture and training strategy, it is

[1] https://brain-development.org/ixi-dataset/
[2] https://pytorch.org/tutorials/advanced/neural_style_tutorial.html
[3] https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

hard to compare the adaptation improvements when the prediction model has different performance on the test dataset. To make sure the prediction results without adaptation on the test set are the same, we reimplemented their core idea on our pipeline. In particular, we changed our A^x to their normalization module and trained a 2D denoising AE^y on the groundtruth segmentation mask y_s with their noise simulation method and training losses. Compared to their original implementation in TensorFlow, where the segmentation backbone is a four level U-Net with feature dimension [16, 32, 64, 128] and has a three-layer convolution block (normalization module) with radial basis functions on the front, our U-Net as described in Sec. 2.1 is implemented in Pytorch and the normalization module is not included (not optimized during source training) and is only optimized during test-time adaptation. We also use a 2D AE instead of 3D in their original implementation for a fair comparison. We report the results using their source code[4] as DTTA-source for reference.

3.4. Training details

The hyperparameters for OCT segmentation and T1-T2 synthesis are the same except the orthogonality constraint λ for the feature adaptors. We set $\lambda = 5$ and $\lambda = 1$ for synthesis and segmentation, respectively, to add more constraints to the synthesis task. This is because with the synthesis task, the task model needs to map the features to infinite continuous values; thus, the subtle difference between features should be preserved. However for segmentation, the features from raw pixels are gradually clustered to a few classes (segmentation labels); thus, the feature distances are less important to keep. These values were determined experimentally. We trained the task model and AEs for 20 epochs with the Adam optimizer (learning rate = 0.001) using a batch size = 2 on the source dataset without data augmentation. For the task model, we used cross-entropy and mean squared error (MSE) as training losses for segmentation and synthesis, respectively. The task model is stopped early according to the validation set accuracy. After off-line training, we used Alg. 1 for online deployment. The adaptors were optimized with the Adam optimizer (learning rate = 0.001) and a batch size = 1 for at most 10 epochs. In our experiments, a subject scan is usually acquired as a 3D scan consisting of multiple 2D slices. For a test subject, one epoch means iterating over all the 2D slices (one slice per iteration). If the average AE reconstruction loss in an epoch was higher than the previous epoch, the optimization was stopped early. The test performance versus adaptation training loss in each adaptation epoch is shown in Fig. 7.

3.5. OCT segmentation results

The segmentation Dice scores (calculated for each slice) for the eight retinal layers from the test Cirrus dataset (48 slices from 6 subjects) are shown in Table 1. The direct deployment of the task model on the source Spectralis test dataset (980 slices from 20 subjects) are reported as the upperbound (NA in the right part of Table 1). We also applied our method to these 20 subjects, which are supposed to have no domain shift. The baseline DTTA method in Table 1 is the reimplementation for comparison, which is not the same as the original method (Karani et al., 2020). The results for DTTA using their source code (DTTA-source) are provided in Table 9 for reference. As shown in Table 1, our test-time domain adaptation

[4] https://github.com/neerakara/test-time-adaptable-neural-networks-fordomain-generalization

greatly improves the Cirrus segmentation results, while only taking 15 seconds for optimizing the adaptor, and 3 seconds for prediction (on a Nvidia Titan Xp GPU), all on a single test subject (8 slices). The results on the source test dataset show that the performance can be improved by test-time adaptation even with data from the source domain. Some qualitative results are shown in Fig. 8. All the baseline methods improve over direct deployment of the task model without adaptation. CycleGAN and Style Transfer achieved good visual results but worse quantitative results as compared to our method. Our adapted Cirrus image is not visually similar to the source because our adaptors are trained to minimize the auto-encoder reconstruction error and will not necessarily make them look similar. Meanwhile, our pixel-level adaptor A^x only adjusts the histogram of the original Cirrus image, which helps avoid tiny boundary shifts that may cause measurement bias, but the strong constraints over the adaptors may produce less appealing results. Thus feature level adaptation is used to further improve the results.

3.6. MRI T1 to T2 synthesis

We compare the synthesized T2 images (630 slices from 30 subjects for each site) from three IXI sites (HH, GH, IOP) to their groundtruth T2 images using mean squared error (MSE) and structural similarity index measure (SSIM) (Wang et al., 2004). The results are shown in Table 2. The test results of our method on the source JHU test dataset are also reported. The DTTA baseline does not support synthesis tasks so we changed their denoising autoencoder reconstruction loss from Dice to MSE for both our reimplementation and their source code (DTTA-source result provided in Table 10). We achieve the best results on HH and GH. Although slightly worse SSIM results than CycleGAN on IOP, our model was applied to all three test datasets without knowing the test domain before deployment. In contrast, the CycleGAN needs to be trained for each site separately and uses much more data from each test domain. For the test results on the source test dataset, unlike the improvement over the no adaptation in OCT, our results are slightly worse, which was expected. We assume that the task model performs optimally within the source domain. Adapting the source domain data point may cause the data point to shift from the optimality, thus we expect a performance drop. Since the source domain is not clearly defined and the imaging procedure can cause a difference even using the same scanner, the performance improvement on OCT may be due to the test data distribution difference with the training data. The experiments on source test data are trying to answer whether the adaptation will shift the source data and cause performance change, and the results show a relatively minor change, which is important for test data from unknown domains. Some qualitative results are shown in Fig. 9. The histogram matching and style transfer are all causing artefacts. The harmonized image using style transfer is much worse than in OCT because the brain anatomy is more complicated, so disentangling style from anatomy can fail. Our implementation of the DTTA method does not work well for the synthesis task, nor does the original DTTA (Karani et al., 2020) in Table 10. The worse results for our reimplementation may be because the pixel-level adaptor is not a part of the task model, which is different from the original DTTA method.

3.7. Ablation Study

3.7.1. Orthogonality Constraints—We added orthogonality constraints for the parameters of $A^i{}_{i = 1,2,3}$ to keep the feature distance after the adaptation, which should limit the transformation power of the adaptors. We performed ablation studies to see how the adaptor constraint λ affects the performance. As shown in Table 4, the synthesis performance with an orthogonality constraint improves all test datasets for both MSE and SSIM (from $\lambda = 0$ to $\lambda = 1$). The improvement saturates when increasing λ to 5. However, as shown in Table 3, a large λ will reduce the performance of the OCT segmentation. As illustrated in Sec. 3.4, a segmentation deep network maps features extracted from raw pixels to a fixed number of segmentation classes, so the clustering of features from different spatial locations is expected. The large adaptor constraints may reduce the adaptation ability but can help keep the discriminative structure of the features.

3.7.2. Multi-Level adaptors and autoencoders—We performed an ablation study using OCT segmentation. As in Karani et al. (2020), the test-time adaptation can be performed with a single pixel-level adaptor optimized by a single denoising autoencoder on the output segmentation. We ask if the feature level adaptation and feature level autoencoders are necessary. (1) Single Adaptor (SA): we only used a pixel level adaptor A^{x} while removing all the feature adaptors $A^i{}_{i=1,2,3}$. All the AEs' reconstruction loss are used for optimization. (2) Single AE (SAE): We only kept a single AE^y, which reconstructs the prediction, and used its reconstruction loss to optimize all the adaptors A^x, $A^i{}_{i=1,2,3}$. The results are shown in Table. 3. Comparing SA and Ours ($\lambda = 1$), the multi-level feature adaptor improves the results from 0.795 to 0.825. Comparing SAE and Ours ($\lambda = 1$), using multi-level feature AEs' reconstruction loss improves the results from 0.808 to 0.825.

3.7.3. Combination with data augmentation—As mentioned in Sec. 1, there are two ways to alleviate the domain shift problem: training a generalizable model and domain adaptation. Our method belongs to domain adaptation class and can be combined with methods for training a generalizable model. As illustrated in Zhang et al. (2020) and Zhao et al. (2019a), data augmentation can greatly improve the generalizability of the deep network. We used extensive data augmentation for both synthesis and segmentation (random gamma adjustment, random affine transformation, random flip and adding Gaussian noise) during offline training for the task model T (no augmentation is used for training autoencoder and Gaussian noise augmentation is not used for the synthesis task model). We report the test-time adaptation performance with a more robust task model in Table 5 and Table 6. Compared to Table 1 and Table 2, the task model achieves much better results on the test dataset with domain shift. By applying our test time adaptation, the test results are further improved. This shows that our method does not conflict with an effort to use a more robust and generalizable model.

3.7.4. Scalability Analysis—Our base task model is a U-Net with 64 feature channels as in He et al. (2019a). If the number of feature channel scales up, will it cause high-dimensional feature learning and computational difficulties for our auto-encoders? To test the scalability of our method, we replace our U-Net encoder with ResNet50 (ResNet50-UNet), with the number of features in the encoder being [64, 256, 512, 1024, 2048]. For the decoder, we concatenate the feature maps from the encoder and use a convolution block to output 64 channel feature maps after every up-sampling layer. For AEs, we have AE^x and AE^y for task model input and output, and $AE^i, i = \{1, \cdots, 5\}$ for intermediate features at five resolution levels. The largest number of input feature channels for the AE is (2048 + 64) and all AE^i's encoder feature map channels are [64, 32, 16] (those intermediate channel numbers do not change based on the input channel). In this setting our final AE model size (which includes seven AEs) is only 46MB, while ResNet50-UNet takes 411MB. The GPU memory usage for training 7 auto-encoders with 2D input images of size 128×1024, batch size 2 using Pytorch 1.6 is 2GB. We believe that this computational cost is acceptable given the task model size.

To analyze the adaptation performance on a high-dimensional feature space, we repeat the OCT segmentation adaptation experiments with adapters A^x on the input, $A^i, i = \{1, \cdots, 5\}$ for the intermediate features from ResNet50-UNet. The overall Dice results after adaptation for ResNet50-UNet and our original UNet (UNet-64) are shown in Table 7. By comparing the columns $\{A\} + AE^y$ & $\{A\} + \{AE\}$, and the columns $\{A^x, A^1\} + AE^y$ & $\{A^x, A^1\} + \{AE\}_,$ we can see that using the AEs on the high-dimensional feature space can improve the adaptation. The linear affine adaptors $A^i, i = \{1, \cdots, 5\}$ adapt features on 64, 128,50512, 1024, and 2048 channel space respectively for ResNet50-UNet, and on all 64 channels for UNet-64. The columns $\{A\}+\{AE\}$, $\{A^x, A^1, A^2\} + \{AE\}$, $\{A^x, A^1\} + \{AE\}$ show degraded with more adaptors for ResNet50-UNet while improved performances for UNet-64. The UNet-64 results show that adaptation on more feature levels may improve the results, but the ResNet50-UNet results expose a risk that the source feature and target feature in the high-dimensional feature space may be hard to align using a linear adaptor, and the performance can drop.

3.7.5. Comparison with other test-time adaptation and UDA methods—In this section, we provide a comparison with recent test-time adaptation methods in classification tasks based on rotation prediction (Sun et al. Sun et al. (2020), denoted as Rot) and entropy minimization (Wang et al. Wang et al. (2020), denoted as Tent). We also compare with unsupervised domain adaptation methods based on learning domain invariant feature space (UDAS) Tsai et al. (2018). We compare those methods on the OCT segmentation tasks. The network used in Sun et al. is for classification, and we modified our U-Net, where the U-Net encoder last output features are used for rotation prediction. During training, we randomly rotate the input image through [0, 90, 180, 270] degrees and train the U-Net to predict the segmentation mask and the rotation class (0, 1, 2, 3). During testing, we randomly rotate the input test image and use the rotation classification loss from the U-Net encoder to update the U-Net encoder parameters. Unfortunately, the results for Sun et al. Sun et al. (2020), as shown in Table 8 (Rot-NA and Rot), are disappointing. We believe that this is in part because, unlike natural images, retinal layers from OCT are basically flat layer structures and have a standard topology which makes it easy for the U-Net encoder to predict the orientation. While the convolution kernels should be more like “vertical gradient” filters when segmenting retinal layers. Rotation augmentation with 90, 180, and 270 degrees does not make sense for retinal layer segmentation and will hurt the performance of the segmentation network. For Tent Wang et al. (2020), the segmentation entropy is minimized by updating the normalization layer affine parameters, which we applied using our U-Net. We achieve similar results with Tent for segmentation. However, the entropy is not available for synthesis tasks as a single value is regressed for each pixel. The UDAS Tsai et al. (2018) uses adversarial loss to learn domain invariant features in both intermediate feature space and output space. We use their source code with the DeepLab backbone (multi-level setting). UDAS is trained using an additional 588 slices of Cirrus data (no overlap with test Cirrus data and no ground-truth segmentation) and 588 slices of source Spectralis images (the same training data as our U-Net). The results are shown in Table 8. We also provide the model performance trained only on the source training dataset (UDAS-Source).

4. Discussion and Conclusion

In this paper, we propose a test-time adaptation method for medical image analysis. By adding two additional components (autoencoders and adaptors) to a conventional deep network (task model), we equip the task model with a certain resistance to unknown test domain shifts.

Although we can develop a more generalizable deep network via more training data, novel augmentation, and training strategies like meta-learning, we do not know what test data will be used and whether our data augmentation covers the variation of the test data. The OCT segmentation accuracy is greatly improved by augmentation as shown in Table 5 and the improvement from test-time adaptation is relatively small, however, common augmentations do not improve the synthesis results much as shown in Table 6, but our test-time augmentation can substantially improve the results. Since we do not know all the variances in the test domain, it is hard to train a fully generalizable task model. Test-domain adaptation does not conflict with a more generalizable model but can compensate the already trained model on potential uncovered test domains. The method requires minimum changes to an existing training and deployment pipeline for a task model,which makes it widely applicable. The additional cost in the deployment stage is small. In particular, the test-time adaptation only takes 15 seconds (task model inference time is 3 seconds) and the additional disk space for saving the autoencoders' parameter is 15 MB (task model needs 10 MB). Our work shares similarities with the DTTA method. DTTA uses a label denoising auto-encoder to learn shape and topology priors and uses this prior to fine-tune the initial layers of the segmentation model. However, our work is developed from an unsupervised domain adaptation perspective. Our work tries to align source and target domain features including the network intermediate features and output results. The alignment measurement is usually performed by adversarial learning which requires sufficient source data. We overcome this problem by using auto-encoder reconstruction loss as an alignment measurement (not designed to learn shape priors). Compared to DTTA, our major strength is feature level adaptation, which means our adaptation does not completely depend on pixel level harmonization. As shown in the synthesis task, the DTTA pixel-level adaptor trained by AEs reconstruction loss will cause blurriness, which is detrimental to the synthesis task. If we further constrain the DTTA pixel-level adaptor, it will lose the ability for good adaptation. Our method limits the pixel-level adaptor to avoid blurriness and adds adaptation in the feature space, thus we can address both segmentation and synthesis tasks. However, our feature level adaptation is also a limitation of this work. We lack a theoretical guarantee that the target domain features can be aligned to the source feature via a single affine adaptor, especially in high-dimensional spaces. While DTTA fine-tunes the low-level image feature extractor to directly focus on minimizing the label space reconstruction error, which can be more stable.

Our method also has several other limitations which should be addressed in the future. First, the method is based on the idea of using autoencoder's reconstruction error for anomaly detection, which in our work, served as a similarity measure between the test features and source features. As pointed out in (Pidhorskyi et al., 2018; Abati et al., 2019; Perera et al., 2019; Gong et al., 2019), an autoencoder for anomaly detection is not always reliable. A low reconstruction error from the adapted test feature thus does not guarantee a good alignment with the source. At the initial step of adaptation, if the test feature is far from the source feature (ie. large domain shift), the test feature may adapt to an off-source domain local minimum (because of the strong generalizability of the AE) or the adaptors may not have an effective gradient to transform the target to the source (because it is too far away). So we assume that the test features from the pre-trained task model are close to the source feature, such that the quick optimization of the adaptors will have a better chance to successfully align test feature to the source. This assumption may not hold in cases of large domain shift, thus we limit our application to minor domain shifts. Karani et al. Karani et al. (2020) address the large domain shift problem by registering an atlas to the test image as pseudo-groundtruth and minimize the large shift as the first step to provide a good initialization for AE based optimization. Although this introduces new problem like registration stability and computation cost, using coarse labels for direct supervision is a promising direction. The test-source alignment measurements in most existing work in test-time adaptation are not satisfactory. Wang et al. (2020) use prediction entropy, Sun et al. (2020) use a pre-trained rotation angle classifier and Karani et al. (2020) use denoising autoencoders. None of them are directly measuring the distance between the source and test feature distribution and we do not know under what condition these measurements can work. Finding a robust similarity measurement with only a few test data samples is a challenging future direction.

A second limitation is that the transformation abilities of our adaptors are limited. The pixel level adaptor can only perform histogram manipulation, and the feature level adaptors are linear transformations (1 × 1 convolutions). We tried to relax the constraints for the pixel-level adaptor by using several 3 × 3 convolution layers, but the blurriness (due to the 3 × 3 convolutions) in the adapted image is detrimental to the synthesis result, and it also poses a risk of boundary shifting. Existing work also tries to limit the adaptation ability. Wang et al. (2020) only optimize the affine transformation parameters of the normalization layer. Karani et al. (2020) optimize the first three layers of the task model (pixel-level harmonization module). Altering the pre-trained task model parameters without supervision from training task labels may cause the task model to shift from the desired task prediction. How to correctly balance the adaptation ability with the risk of task prediction shift is another direction. Overall, we propose a novel self-supervised test-time adaptation method which shows promising results on both synthesis and segmentation. We hope the proposed work can help in developing a more clinically robust and easy to deploy deep network.

Acknowledgments

This work was supported by the NIH/NEI under grant R01-EY024655 and NIH/NINDS grant R01-NS082347, and in part by the Intramural Research Program of the NIH, National Institute on Aging.

References

Abati D, Porrello A, Calderara S, Cucchiara R, 2019. Latent space autoregression for novelty detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 481–490.

Bansal N, Chen X, Wang Z, 2018. Can we gain more from orthogonality regularizations in training deep networks?, in: Advances in Neural Information Processing Systems, pp. 4261–4271.

Billot B, Greve D, Van Leemput K, Fischl B, Iglesias JE, Dalca AV, 2020. A learning strategy for contrast-agnostic mri segmentation. arXiv preprint arXiv:2003.01995.

Carass A, Cuzzocreo JL, Han S, Hernandez-Castillo CR, Rasser PE, Ganz M, Beliveaux V, Dolz J, Ayed IB, Desrosiers C, Thyreau B, Romero JE, Coupé P, Manjón JV, Fonov VS, Collins DL, Ying SH, Crocetti D, Landman BA, Mostofsky SH, Thompson PA, Prince JL, 2018. Comparing fully automated state-of-the-art cerebellum parcellation from magnetic resonance images. NeuroImage 183, 150–172. [PubMed: 30099076]

Cohen JP, Luck M, Honari S, 2018. Distribution matching losses can hallucinate features in medical image translation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 529–536.

Dar SU, Yurt M, Karacan L, Erdem A, Erdem E, Çukur T, 2019. Image synthesis in multi-contrast mri with conditional generative adversarial networks. IEEE transactions on medical imaging 38, 2375–2388. [PubMed: 30835216]

Dou Q, de Castro DC, Kamnitsas K, Glocker B, 2019. Domain generalization via model-agnostic learning of semantic features, in: Advances in Neural Information Processing Systems, pp. 6450–6461.

Dou Q, Ouyang C, Chen C, Chen H, Heng PA, 2018. Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 691–697.

Ganin Y, Lempitsky V, 2015. Unsupervised domain adaptation by backpropagation, in: Bach F, Blei D (Eds.), Proceedings of the 32nd International Conference on Machine Learning, PMLR. pp. 1180–1189.

Gatys LA, Ecker AS, Bethge M, 2016. Image style transfer using convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2414–2423.

Gong D, Liu L, Le V, Saha B, Mansour MR, Venkatesh S, Hengel A.v.d., 2019. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection,in: Proceedings of the IEEE International Conference on Computer Vision, pp. 1705–1714.

He K, Zhang X, Ren S, Sun J, 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034.

He Y, Carass A, Liu Y, Filippatou A, Jedynak BM, Solomon SD, Saidha S, Calabresi PA, Prince JL, 2020a. Segmenting retinal oct images with inter-b-scan and longitudinal information, in: Medical Imaging 2020: Image Processing, International Society for Optics and Photonics. p. 113133C.

He Y, Carass A, Liu Y, Jedynak BM, Solomon SD, Calabresi PA, Prince JL, 2019a. Fully convolutional boundary regression for retina OCT segmentation, in: 22nd International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2019), Springer Berlin Heidelberg. pp. 120–128.

He Y, Carass A, Liu Y, Jedynak BM, Solomon SD, Saidha S, Calabresi PA, Prince JL, 2020b. Structured layer surface segmentation for retina oct using fully convolutional regression networks. Medical Image Analysis 68, 101856. [PubMed: 33260113]

He Y, Carass A, Solomon SD, Saidha S, Calabresi PA, Prince JL, 2019b. Retinal layer parcellation of optical coherence tomography images: Data resource for multiple sclerosis and healthy controls. Data in brief 22, 601–604. [PubMed: 30671506]

He Y,Carass A,Zuo L,Dewey BE,Prince JL,2020c. Self domain adapted network, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 437–446.

Hoffman J, Tzeng E, Park T, Zhu JY, Isola P, Saenko K, Efros A, Darrell T, 2018. Cycada: Cycle-consistent adversarial domain adaptation, in: International Conference on Machine Learning, pp. 1989–1998.

Isensee F, Jäger PF, Kohl SA, Petersen J, Maier-Hein KH, 2019. Auto-mated design of deep learning methods for biomedical image segmentation. arXiv preprint arXiv:1904.08128.

Jain V, Learned-Miller E, 2011. Online domain adaptation of a pre-trained cascade of classifiers, in: CVPR 2011, IEEE. pp. 577–584.

Kabir Y, Dojat M, Scherrer B, Forbes F, Garbay C, 2007. Multimodal mri segmentation of ischemic stroke lesions, in: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE. pp. 1595–1598.

Karani N, Erdil E, Chaitanya K, Konukoglu E, 2020. Test-time adaptable neural networks for robust medical image segmentation. Medical Image Analysis 68, 101907. [PubMed: 33341496]

Kavur AE, Gezer NS, Barıs M, Aslan S, Conze PH, Groza V, Pham DD, Chatterjee S, Ernst P, Özkan S, et al., 2020. Chaos challenge-combined (ct-mr) healthy abdominal organ segmentation. Medical Image Analysis , 101950.

Liu MY, Breuel T, Kautz J, 2017. Unsupervised image-to-image translation networks, in: Advances in neural information processing systems, pp. 700–708.

Liu Z, Yang X, Gao R, Liu S, Dou H, He S, Huang Y, Huang Y, Luo H, Zhang Y, et al., 2020. Remove appearance shift for ultrasound image segmentation via fast and universal style transfer, in: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), IEEE. pp. 1824–1828.

Long M, Cao Y, Wang J, Jordan MI, 2015. Learning transferable features with deep adaptation networks. arXiv preprint arXiv:1502.02791.

Ma C, Ji Z, Gao M, 2019. Neural style transfer improves 3d cardiovascular mr image segmentation on inconsistent data, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 128–136.

Mahapatra D, Bozorgtabar B, Shao L, 2020. Pathological retinal region segmentation from oct images using geometric relation based augmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9611–9620.

Modanwal G, Vellal A, Buda M, Mazurowski MA, 2020. Mri image harmonization using cycle-consistent generative adversarial network, in: Medical Imaging 2020: Computer-Aided Diagnosis, International Society for Optics and Photonics. p. 1131413.

Oktay O, Bai W, Lee M, Guerrero R, Kamnitsas K, Caballero J, de Marvao A, Cook S, ORegan D, Rueckert D, 2016. Multi-input cardiac image super-resolution using convolutional neural networks, in: International conference on medical image computing and computer-assisted intervention, Springer. pp. 246–254.

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

Perera P, Nallapati R, Xiang B, 2019. Ocgan: One-class novelty detection using gans with constrained latent representations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2898–2906.

Pidhorskyi S,Almohsen R,Doretto G,2018. Generative probabilistic novelty detection with adversarial autoencoders. Advances in neural information processing systems 31, 6822–6833.

Reinhold JC,Dewey BE,Carass A,Prince JL,2019. Evaluating the impact of intensity normalization on MR Image Synthesis, in: Proceedings of SPIE Medical Imaging (SPIE-MI 2019), San Diego, CA, February 16 – 21, 2019, p. 109493H.

Ronneberger O, Fischer P, Brox T, 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation, in: 18th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2015), Springer Berlin Heidelberg. pp. 234–241.

Saidha S, Syc SB, Durbin MK, Eckstein C, Oakley JD, Meyer SA, Conger A, Frohman TC, Newsome S, Ratchford JN, Frohman EM, Calabresi PA, 2011a. Visual dysfunction in multiple sclerosis correlates better with optical coherence tomography derived estimates of macular ganglion cell layer thickness than peripapillary retinal nerve fiber layer thickness. Mult. Scler 17, 1449–1463. [PubMed: 21865411]

Saidha S, Syc SB, Ibrahim MA, Eckstein C, Warner CV, Farrell SK, Oakley JD, Durbin MK, Meyer SA, Balcer LJ, Frohman EM, Rosenzweig JM, Newsome SD, Ratchford JN, Nguyen QD, , Calabresi PA, 2011b. Primary retinal pathology in multiple sclerosis as detected by optical coherence tomography. Brain 134, 518–533. [PubMed: 21252110]

Seeböck P, Romo-Bucheli D, Waldstein S, Bogunovic H, Orlando JI, Gerendas BS, Langs G, Schmidt-Erfurth U, 2019. Using CycleGANs for effectively reducing image variability across OCT devices and improving retinal fluid segmentation, in: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), IEEE. pp. 605–609.

Sharma A, Hamarneh G, 2019. Missing MRI pulse sequence synthesis using multi-modal generative adversarial network. IEEE Trans. Med. Imag 39, 1170–1183.

Simpson AL, Antonelli M, Bakas S, Bilello M, Farahani K, Van Ginneken B, Kopp-Schneider A, Landman BA, Litjens G, Menze B, et al., 2019. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063.

Sun Y, Wang X, Liu Z, Miller J, Efros A, Hardt M, 2020. Test-time training with self-supervision for generalization under distribution shifts, in: International Conference on Machine Learning, PMLR. pp. 9229–9248.

Tsai YH, Hung WC, Schulter S, Sohn K, Yang MH, 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.

Tustison NJ, Avants BB, Cook PA, Zheng Y, Egan A, Yushkevich PA, Gee JC, 2010. N4ITK: Improved N3 bias correction. IEEE Trans. Med. Imag 29, 1310–1320.

Tzeng E, Hoffman J, Saenko K, Darrell T, 2017. Adversarial discriminative domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7167–7176.

Ulyanov D, Vedaldi A, Lempitsky V, 2016. Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022.

Wang D, Shelhamer E, Liu S, Olshausen B, Darrell T, 2020. Fully test-time adaptation by entropy minimization. arXiv preprint arXiv:2006.10726.

Wang Z, Bovik AC, Sheikh HR, Simoncelli EP, 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Trans. Imag. Proc 13, 600–612.

Yang H, Sun J, Carass A, Zhao C, Lee J, Prince JL, Xu Z, 2020. Unsupervised MR-to-CT Synthesis Using Structure-Constrained CycleGAN. IEEE Trans. Med. Imag 39, 4249–4261.

Yang H, Sun J, Carass A, Zhao C, Lee J, Xu Z, Prince J, 2018. Unpaired brain mr-to-ct synthesis using a structure-constrained cyclegan, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, pp. 174–182.

Zhang L, Wang X, Yang D, Sanford T, Harmon S, Turkbey B, Wood BJ, Roth H, Myronenko A, Xu D, et al., 2020. Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. IEEE Trans. Med. Imag 39, 2531–2540.

Zhao A, Balakrishnan G, Durand F, Guttag JV, Dalca AV, 2019a. Data augmentation using learned transformations for one-shot medical image segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8543–8553.

Zhao C, Carass A, Lee J, He Y, Prince JL, 2017. Whole brain segmentation and labeling from ct using synthetic mr images, in: International Workshop on Machine Learning in Medical Imaging, Springer. pp. 291–298.

Zhao C, Shao M, Carass A, Li H, Dewey BE, Ellingsen LM, Woo J, Guttman MA, Blitz AM, Stone M, et al., 2019b. Applications of a deep learning method for anti-aliasing and super-resolution in MRI. Mag. Reson. Im 64, 132–141.

Zhu JY, Park T, Isola P, Efros AA, 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE international conference on computer vision, pp. 2223–2232.

Highlights

- A novel formulation for domain adaptation without retraining the model nor using source training data.
- Quick test-time adaptation to a single test subject at inference stage.
- Applicable to both medical image synthesis and segmentation

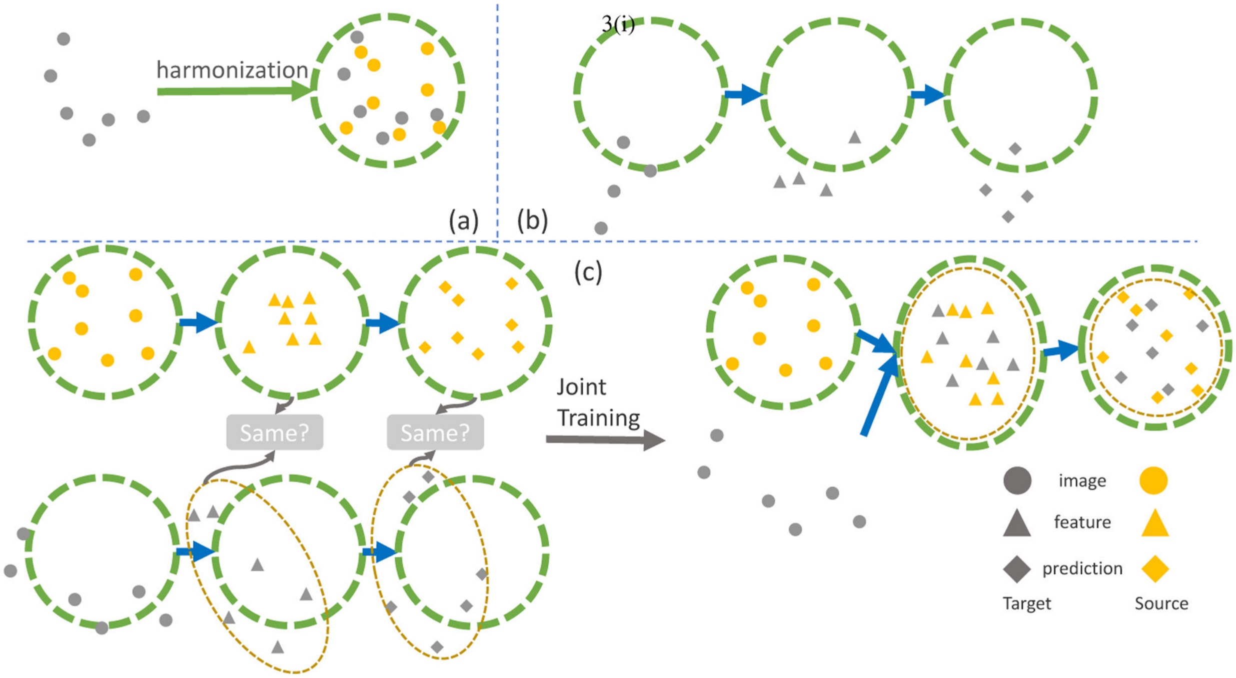
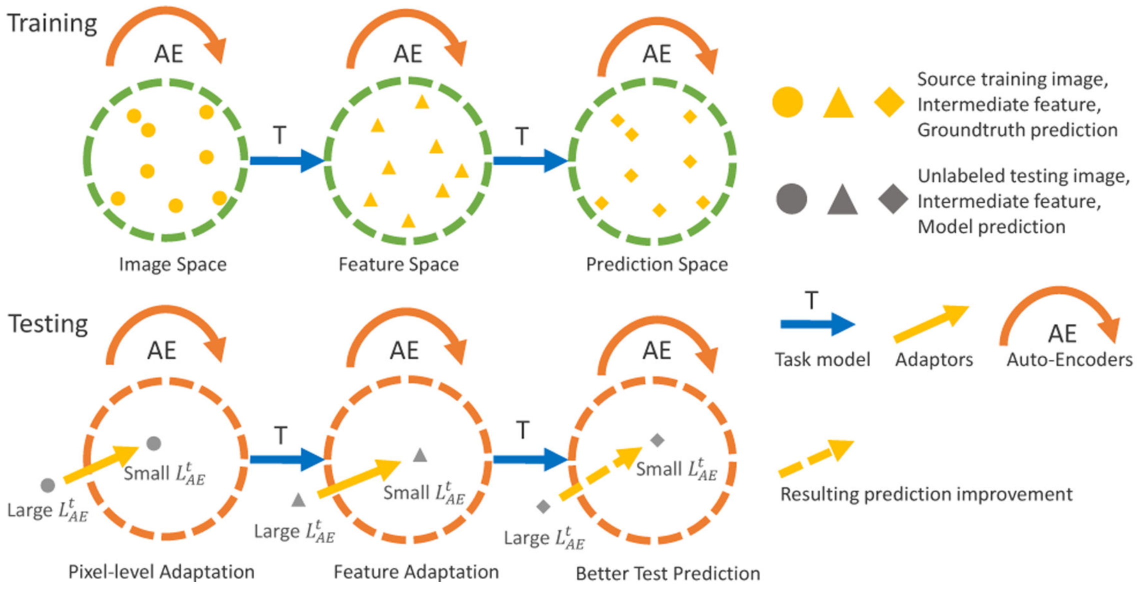


Fig. 1.

Existing methods for domain adaptation. The samples within the green circles can be processed successfully by the trained model. (a) Harmonize the test image to the source, which requires abundant source and test images. (b) In the deployment stage, where we only have unknown domain test data and a trained model, how can we solve the domain shift problem? (c) The model is trained to produce domain-invariant features and predictions which requires abundant unlabeled test images and labeled source images.



3(a)

Fig. 2.

T and AEs are trained on labeled source dataset. The area within green circle is the source domain. During testing stage, only a test subject and the trained models are given. The source domain is unknown since source data are unavailable. We use the AEs' reconstruction loss $\left(\mathscr{L}_{AE}^{t}\right)$ as a source domain similarity measurement (orange circle) and optimize adaptors to transform the domain-shifted test image and its features towards the source domain.

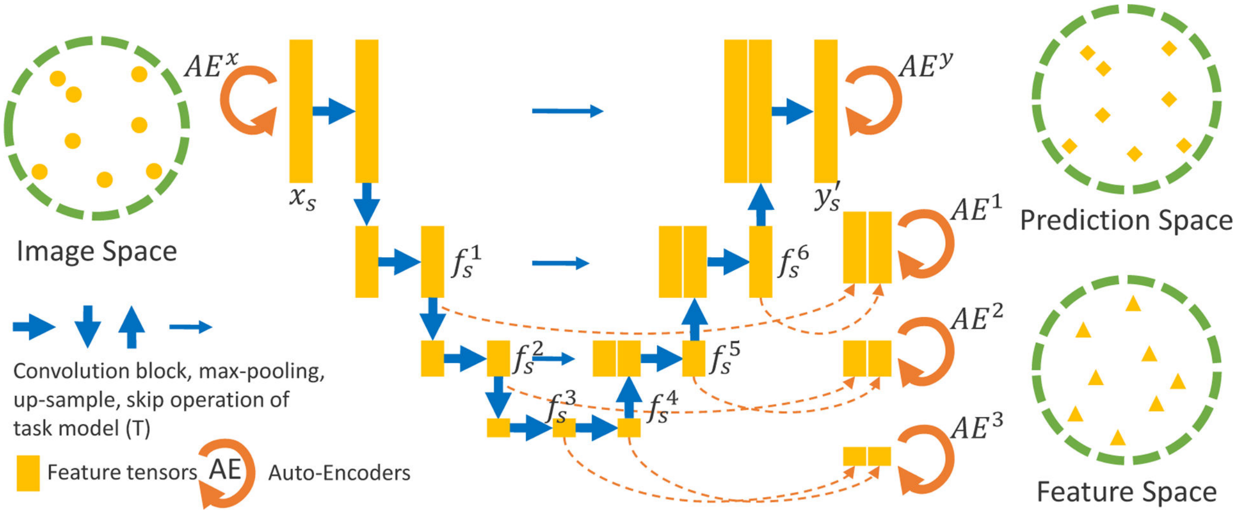


Fig. 3.

The task model (T) and autoencoders (AEs) in offline training. T is trained on the labeled source dataset using conventional gradient descent. The parameters of T are then fixed. AEs are trained by minimizing the reconstruction loss of the source input image x_s, intermediate features from the pre-trained T and the prediction $y's$.

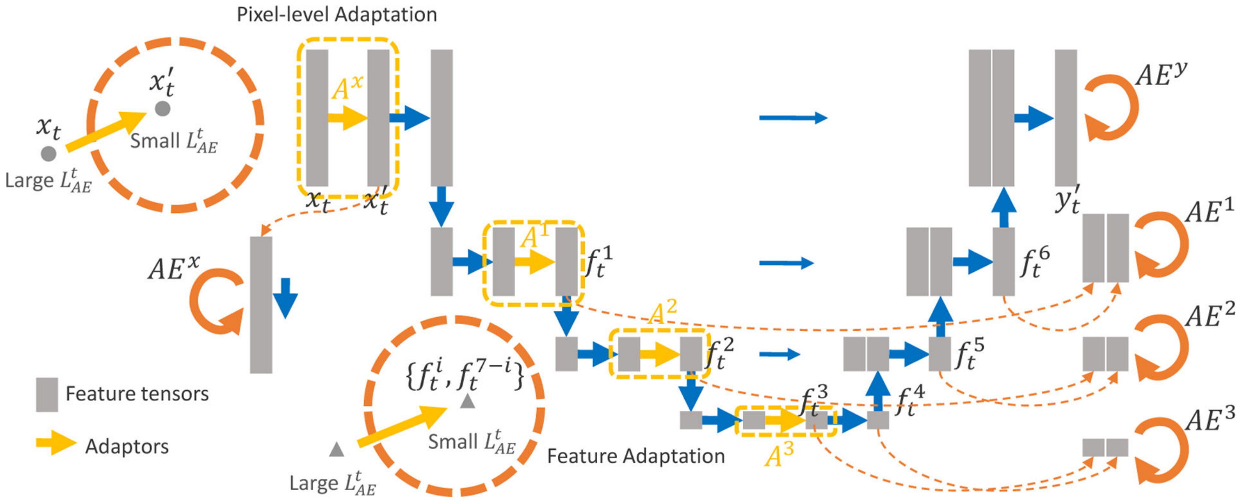


Fig. 4.

In the testing stage, the parameters of T and AEs are fixed. The adaptors are added into the task model to perform the prediction together. The adaptors are optimized to transform the input test image and features extracted using T's encoder to achieve a lower AEs' reconstruction loss.

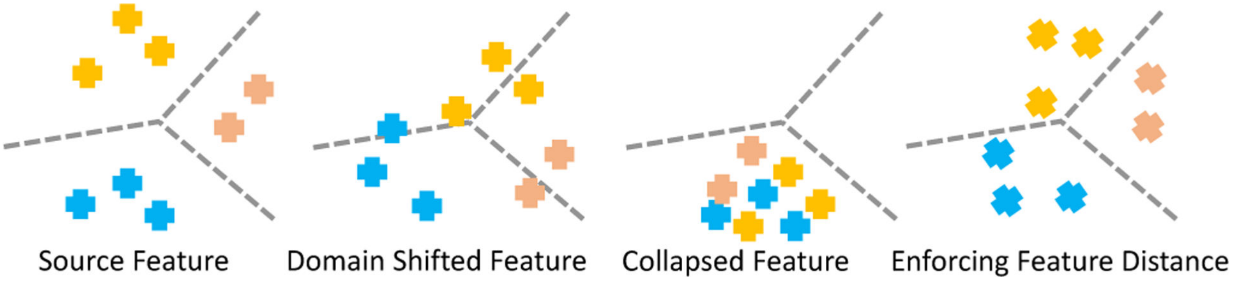


Fig. 5.

The potential feature collapse problem. The dashed lines are T’s classifier. The domain shifted features will not generate a good prediction (each cross is a feature vector at a spatial location on one input image, however, each dot/triangle in Fig. 2 represents the whole image/image features). The discriminative structure of the features from one test image may be collapsed due to the adaptors transformation and we enforce the same L_2 distance between the features after adaptation.

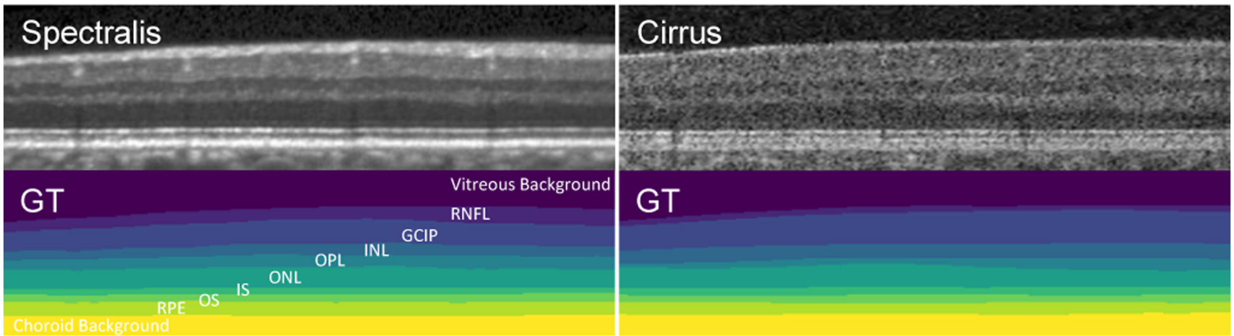


Fig. 6.

Top: examples of Spectralis and Cirrus images. Bottom: corresponding manual segmentation and retinal layer names for Spectralis (left) and Cirrus (right).

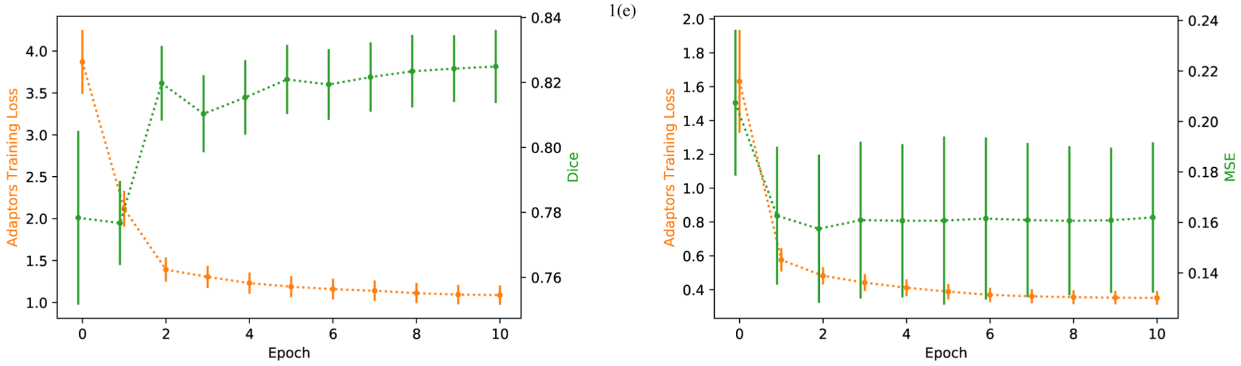


Fig. 7.

Test performance versus adaptation training loss in each adaptation epoch. The green dotted line and vertical bars are the mean and standard deviation for the performance on each test subject (the left figure is the overall Dice score on the test Cirrus subjects and the right figure is the mean squared error (MSE) for the synthesised T2 from HH site). The orange dotted line and vertical bars are the mean and standard deviation of training loss $\mathscr{L}_A$ during adaptation.

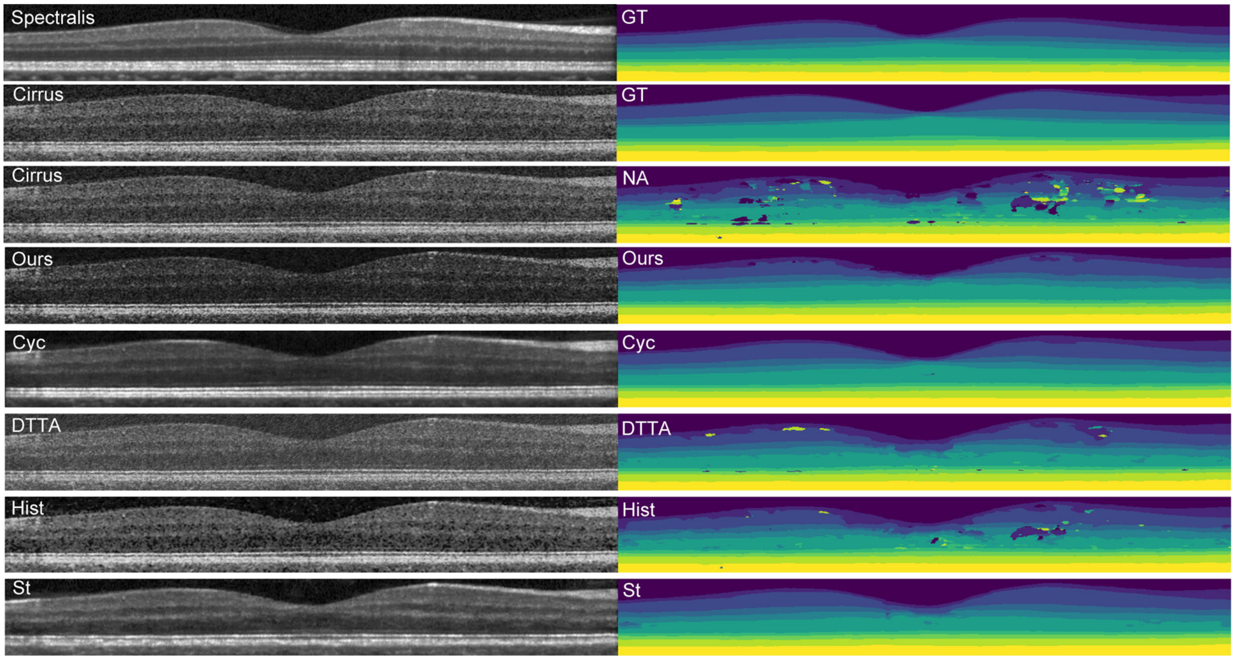


Fig. 8.

Examples of the segmentation results. Left (from top to bottom): An example source Spectralis slice, an example Cirrus slice (not paired with the Spectralis slice), the same Cirrus slice, our pixel-level adapted image (Ours, the features are also adapted to generate the results), CycleGAN harmonized image (Cyc), DTTA (single pixel-level adaptor based on a three-layer convolution block with a segmentation label denoising auto-encoder), median filter followed by histogram matching (Hist) and style transfer (St). Right: The ground truth segmentation for Spectralis and Cirrus (GT) respectively and the corresponding segmentation results for each method.

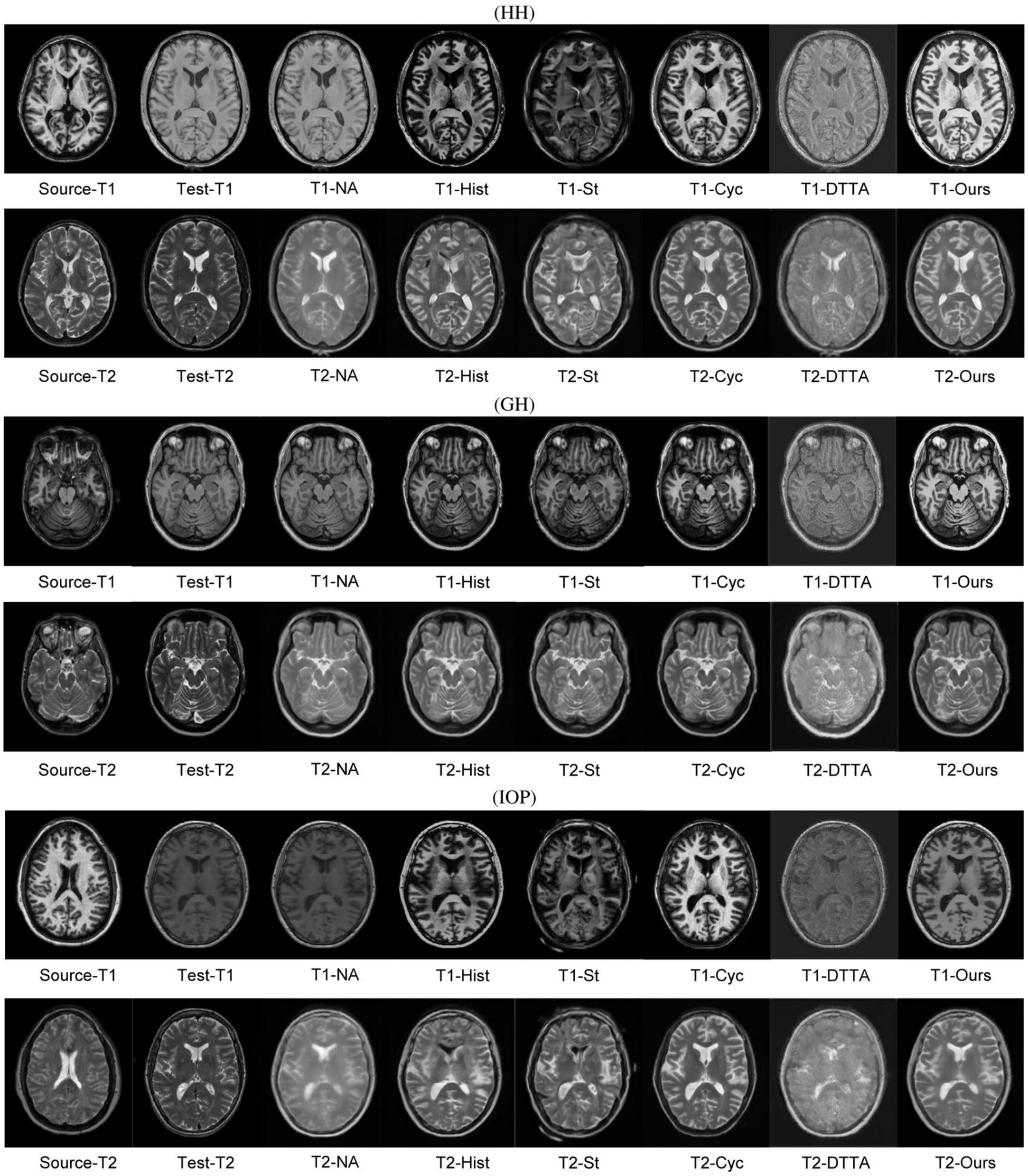


Fig. 9.

Examples of the T1 to T2 synthesis results for three IXI clinical sites (HH, GH, IOP). The first column are three examples of registered T1-T2 pairs from the source JHU dataset. The second column are the paired T1-T2 examples from HH, GH, and IOP, respectively. The rest are examples of the harmonized T1 image and corresponding synthesis results for each method: No Adapt (NA), Histogram matching (Hist), Style transfer (St), CycleGAN (Cyc), Test-time adaptation based on our inplementation (DTTA (Karani et al., 2020)) and our results (Ours).

Table 1.

Mean Dice scores (Std. Dev.) of retinal layer segmentation.

	Target Test Results						Source Test Results	
Layer	**NA**	**Hist**	**St**	**Cyc**	**DTTA**	**Ours**	**Ours**	**NA**
RNFL	0.660(0.171)	0.739(0.135)	**0.773(0.086)**	0.767(0.100)	0.701(0.151)	0.732(0.130)	**0.933(0.032)**	0.931(0.038)
GCIP	0.764(0.089)	0.848(0.055)	**0.866(0.046)**	0.859(0.031)	0.802(0.074)	0.851(0.042)	**0.947(0.026)**	0.947(0.028)
INL	0.722(0.064)	0.764(0.050)	0.782(0.050)	**0.797(0.030)**	0.767(0.036)	0.791(0.033)	**0.880(0.038)**	0.880(0.037)
OPL	0.633(0.076)	0.642(0.067)	0.644(0.058)	0.680(0.054)	0.650(0.067)	**0.692(0.061)**	0.901(0.030)	**0.902(0.030)**
ONL	0.862(0.034)	0.872(0.028)	0.876(0.027)	0.886(0.027)	0.873(0.028)	**0.904(0.024)**	**0.949(0.018)**	0.948(0.020)
IS	0.871(0.035)	0.871(0.035)	0.883(0.029)	0.854(0.043)	0.874(0.030)	**0.884(0.031)**	**0.877(0.035)**	0.868(0.041)
OS	0.887(0.030)	0.892(0.024)	0.869(0.048)	0.880(0.035)	0.897(0.025)	**0.899(0.019)**	**0.887(0.039)**	0.878(0.044)
RPE	0.828(0.037)	0.843(0.033)	0.826(0.045)	**0.854(0.039)**	0.836(0.037)	0.849(0.033)	**0.921(0.041)**	0.919(0.042)
Overall	0.778(0.067)	0.809(0.053)	0.815(0.049)	0.822(0.045)	0.800(0.056)	**0.825(0.047)**	**0.912(0.032)**	0.909(0.035)

Left: results from 48 Cirrus slices (from 6 subjects) using NA: no adaptation, Hist: median filter followed by histogram matching, St: style transfer, Cyc: CycleGAN, DTTA: another test-time adaptation method (Karani et al., 2020) which is reimplemented with our task model. Right: results from 980 Source Spectralis slices (20 subjects) using NA: direct application of the task model trained from the source Spectralis training set. Ours: applied Alg. 1 on the source Spectralis test data even though there is no domain shift.

Table 2.

MSE and SSIM (Std. Dev.) for synthesized T2 evaluated on four datasets (best result is in bold for each row).

		NA	Hist	St	Cyc	DTTA	Ours
MSE	HH	0.207(0.033)	0.197(0.055)	0.315(0.295)	0.196(0.041)	0.317(0.040)	**0.162(0.040)**
	GH	0.259(0.037)	0.252(0.095)	0.326(0.220)	0.255(0.064)	0.408(0.061)	**0.231(0.052)**
	IOP	0.331(0.052)	0.303(0.075)	0.406(0.164)	0.281(0.070)	0.417(0.081)	**0.277(0.056)**
	JHU (Source)	0.090(0.042)	-	-	-	-	0.101(0.052)
SSIM	HH	0.613(0.051)	0.675(0.060)	0.564(0.172)	0.690(0.047)	0.449(0.061)	**0.702(0.044)**
	GH	0.599(0.045)	**0.665(0.067)**	0.592(0.136)	0.648(0.043)	0.433(0.056)	0.664(0.048)
	IOP	0.498(0.066)	0.576(0.074)	0.459(0.101)	**0.621(0.060)**	0.376(0.068)	0.578(0.063)
	JHU (Source)	0.776(0.053)	-	-	-	-	0.767(0.056)

HH, GH, and IOP compares domain adaptation performance while Source illustrates the performance of task model (NA) and our method on source testing data.

Table 3.

The Dice scores of the Cirrus dataset.

Target Cirrus Test Results				
Layer	**SA**	**SAE**	**Ours (λ = 5)**	**Ours (λ = 1)**
RNFL	0.682(0.160)	0.703(0.148)	0.721(0.138)	**0.732(0.130)**
GCIP	0.799(0.068)	0.824(0.056)	0.842(0.048)	**0.851(0.042)**
INL	0.749(0.042)	0.762(0.035)	0.784(0.033)	**0.791(0.033)**
OPL	0.641(0.068)	0.661(0.066)	0.678(0.061)	**0.692(0.061)**
ONL	0.875(0.028)	0.887(0.029)	0.898(0.026)	**0.904(0.024)**
IS	0.882(0.027)	0.875(0.031)	0.882(0.030)	**0.884(0.031)**
OS	0.894(0.020)	**0.900(0.020)**	0.898(0.019)	0.899(0.019)
RPE	0.839(0.035)	**0.851(0.033)**	0.849(0.033)	0.849(0.033)
Overall	0.795(0.056)	0.808(0.052)	0.819(0.049)	**0.825(0.047)**

SA: only used a single pixel-level adaptor, A^{X}, for adaptation. SAE: only used the segmentation autoencoder, AE^{Y}, for reconstruction loss. Ours(λ = 5) and Ours(λ = 1): our method in Alg. 1 with λ = 5 and λ = 1, respectively.

Table 4.

MSE and SSIM (Std. Dev.) for synthesized T2 evaluated on HH, GH, and IOP.

		Ours (λ= 0)	**Ours (λ = 1)**	**Ours (λ = 5)**
MSE	HH	0.180(0.038)	0.165(0.041)	**0.162(0.040)**
	GH	0.240(0.046)	0.232(0.050)	**0.231(0.052)**
	IOP	0.281(0.048)	**0.274(0.054)**	0.277(0.056)
SSIM	HH	0.666(0.043)	0.696(0.043)	**0.702(0.044)**
	GH	0.641(0.046)	0.658(0.045)	**0.664(0.048)**
	IOP	0.543(0.058)	0.572(0.060)	**0.578(0.063)**

The ablation study compares the effects of different values for the orthogonality constraint λ.

Table 5.

Mean Dice scores (Std. Dev.) of retinal layer segmentation results from task model which is trained with extensive augmentation and has better generalizability.

	Target Test Results						Source Test Results	
Layer	**NA**	**Hist**	**St**	**Cyc**	**DTTA**	**Ours**	**Ours**	**NA**
RNFL	0.766(0.103)	0.771(0.103)	0.764(0.086)	0.766(0.097)	0.781(0.092)	**0.787(0.094)**	0.931(0.033)	**0.932(0.035)**
GCIP	0.868(0.034)	0.872(0.033)	0.856(0.055)	0.855(0.033)	0.873(0.034)	**0.876(0.034)**	0.945(0.027)	**0.947(0.027)**
INL	0.810(0.033)	0.803(0.035)	0.790(0.071)	0.808(0.028)	0.819(0.033)	**0.820(0.033)**	0.877(0.038)	**0.880(0.037)**
OPL	0.693(0.053)	0.682(0.053)	0.660(0.056)	0.682(0.052)	0.702(0.053)	**0.706(0.052)**	0.895(0.031)	**0.900(0.029)**
ONL	0.896(0.024)	0.890(0.024)	0.884(0.024)	0.881(0.027)	**0.902(0.022)**	0.902(0.022)	0.946(0.019)	**0.948(0.018)**
IS	**0.895(0.027)**	0.883(0.035)	0.892(0.028)	0.859(0.040)	0.893(0.029)	0.894(0.030)	**0.876(0.035)**	0.872(0.037)
OS	0.904(0.025)	0.898(0.030)	0.875(0.047)	0.879(0.032)	**0.905(0.023)**	0.905(0.024)	**0.883(0.036)**	0.879(0.042)
RPE	0.831(0.035)	0.834(0.034)	0.819(0.045)	**0.846(0.042)**	0.831(0.035)	0.836(0.035)	**0.917(0.043)**	0.917(0.044)
Overall	0.833(0.042)	0.829(0.043)	0.818(0.052)	0.822(0.044)	0.838(0.040)	**0.841(0.041)**	0.909(0.033)	**0.909(0.034)**

Left: results from 48 Cirrus slices (from 6 subjects) using NA: no adaptation, Hist: median filter followed by histogram matching, St: style transfer, Cyc: CycleGAN, DTTA: another test-time adaptation method (Karani et al., 2020) which is reimplemented with our task model. Right: results from 980 Source Spectralis slices (20 subjects) using NA: direct application of the task model trained from the source Spectralis training set. Ours: apply our Alg. 1 on the source Spectralis test data even there is no domain shift.

Table 6.

MSE and SSIM (Std. Dev.) for synthesized T2 evaluated on four datasets (best result is in bold for each row). The task model is trained with extensive augmentation and has better generalizability.

		NA	Hist	St	Cyc	DTTA	Ours
MSE	HH	0.210(0.039)	0.191(0.044)	0.330(0.253)	0.177(0.040)	0.321(0.057)	**0.170(0.033)**
	GH	0.233(0.042)	0.224(0.067)	0.313(0.201)	**0.210(0.050)**	0.417(0.103)	0.216(0.046)
	IOP	0.264(0.056)	0.259(0.068)	0.430(0.161)	0.268(0.071)	0.323(0.061)	**0.239(0.051)**
	JHU (Source)	0.083(0.041)	-	-	-	-	0.093(0.049)
SSIM	HH	0.635(0.046)	0.696(0.057)	0.538(0.177)	**0.707(0.047)**	0.442(0.072)	0.704(0.044)
	GH	0.632(0.040)	**0.685(0.056)**	0.592(0.141)	0.672(0.043)	0.429(0.072)	0.681(0.047)
	IOP	0.556(0.072)	0.608(0.073)	0.449(0.107)	**0.626(0.063)**	0.410(0.071)	0.601(0.064)
	JHU (Source)	0.789(0.053)	-	-	-	-	0.776(0.057)

HH, GH, and IOP compares domain adaptation performance while Source illustrates the performance of task model (NA) and our method on source testing data.

Table 7.

Overall Dice on test Cirrus data for ResNet50-UNet and our UNet-64.

	NA	$\{A\}+AE^y$	$\{A\}+\{AE\}$	$\{A^x, A^1, A^2\}+\{AE\}$	$\{A^x, A^1\}+\{AE\}$	$\{A^x, A^1\}+AE^y$
ResNet50-UNet	0.8148	0.8061	0.8181	0.8237	0.8271	0.8206
UNet-64	0.7783	0.8080	0.8252	0.8233	0.8178	0.8083

NA: no test time adaptation, $\{A\}+\{AE^y\}$: using all adpators for adaptation and only AE^y for reconstruction loss, $\{A\}+\{AE\}$: using all adpators ans AEs, $\{A^X, A^1, A^2\}+\{AE\}$: using A^X, A^1, A^2 and all AEs, $\{A^X, A^1\}+\{AE\}$: using A^X, A^1 and all AEs, $\{A^X, A^1\}+AE^y$: using A^X, A^1 and only AE^y

Table 8.

Mean Dice scores (Std. Dev.) of retinal layer segmentation results from 48 Cirrus slices (6 subjects).

Layer	NA	Rot-NA	Rot	Tent	UDAS-Source	UDAS	Ours
RNFL	0.660(0.171)	0.551(0.128)	0.319(0.111)	0.769(0.103)	**0.776(0.108)**	0.760(0.117)	0.732(0.130)
GCIP	0.764(0.089)	0.745(0.086)	0.746(0.101)	0.868(0.032)	0.873(0.036)	**0.877(0.029)**	0.851(0.042)
INL	0.722(0.064)	0.706(0.068)	0.715(0.119)	0.790(0.033)	**0.811(0.036)**	0.806(0.032)	0.791(0.033)
OPL	0.633(0.076)	0.426(0.097)	0.510(0.137)	0.686(0.057)	0.667(0.067)	**0.695(0.053)**	0.692(0.061)
ONL	0.862(0.034)	0.610(0.096)	0.640(0.159)	0.899(0.026)	0.877(0.035)	0.899(0.029)	**0.904(0.024)**
IS	0.871(0.035)	0.774(0.058)	0.372(0.241)	0.880(0.032)	0.872(0.047)	0.867(0.053)	**0.884(0.031)**
OS	0.887(0.030)	0.824(0.044)	0.336(0.275)	**0.901(0.025)**	0.869(0.046)	0.875(0.045)	0.899(0.019)
RPE	0.828(0.037)	0.734(0.045)	0.152(0.139)	0.832(0.035)	0.833(0.039)	0.846(0.038)	**0.849(0.033)**
Overall	0.778(0.067)	0.671(0.078)	0.474(0.160)	**0.828(0.043)**	0.822(0.052)	0.828(0.050)	0.825(0.047)

NA: Our U-Net trained with no adaptation; Rot-NA: Our U-Net with rotation prediction using U-Net encoder output (modified from Sun et al. (2020)) with no test-time adaptation; Rot: test-time adaptation for U-Net encoder as in Sun et al. (2020); Tent: Test-time adaptation using Wang et al. (2020), and the segmentation model is our U-Net (NA column); UDAS-Source: segmentation model from Tsai et al. (2018) and trained on source; UDAS: joint training using labeled source and unlabeled target domain images as in Tsai et al. (2018).

Table 9.

Test-time adaptation Cirrus segmentation Dice results using (Karani et al., 2020) source code.

Target test results using DTTA-source		
Layer	**DTTA-NA**	**DTTA-adapt**
RNFL	0.755(0.099)	**0.787(0.082)**
GCIP	0.813(0.062)	**0.857(0.041)**
INL	0.701(0.074)	**0.757(0.037)**
OPL	0.606(0.071)	**0.637(0.063)**
ONL	0.792(0.123)	**0.869(0.028)**
IS	0.870(0.033)	**0.883(0.031)**
OS	0.889(0.027)	**0.901(0.022)**
RPE	0.838(0.036)	**0.840(0.035)**
Overall	0.783(0.066)	**0.816(0.042)**

DTTA-NA and DTTA-Adapt are the results without and with adaptation.

Table 10.

Test-time adaptation T1 to T2 synthesis results using (Karani et al., 2020) source code, we modified the denoising autoencoder's reconstruction loss from Dice to MSE. DTTA-NA and DTTA-Adapt are the results without and with adaptation.

		DTTA-NA	**DTTA-Adapt**
MSE	HH	0.254(0.035)	**0.234(0.043)**
	GH	0.283(0.042)	**0.258(0.045)**
	IOP	**0.346(0.048)**	0.360(0.048)
SSIM	HH	0.561(0.057)	**0.579(0.050)**
	GH	0.572(0.052)	**0.591(0.044)**
	IOP	**0.457(0.069)**	0.452(0.063)