Unsupervised Domain Adaptation for Semantic Segmentation via Feature-space Density Matching

Tushar Kataria[1,2], Beatrice Knudsen[3], and Shireen Elhabian[1,2]

[1] Kahlert School of Computing, University Of Utah
[2] Scientific Computing and Imaging Institute, University of Utah
[3] Department of Pathology, University of Utah

Abstract. Semantic segmentation is a critical step in automated image interpretation and analysis where pixels are classified into one or more predefined semantically meaningful classes. Deep learning approaches for semantic segmentation rely on harnessing the power of annotated images to learn features indicative of these semantic classes. Nonetheless, they often fail to generalize when there is a significant domain (i.e., distributional) shift between the training (i.e., *source*) data and the dataset(s) encountered when deployed (i.e., *target*), necessitating manual annotations for the target data to achieve acceptable performance. This is especially important in medical imaging because different image modalities have significant intra- and inter-site variations due to protocol and vendor variability. Current techniques are sensitive to hyperparameter tuning and target dataset size. This paper presents an unsupervised domain adaptation approach for semantic segmentation that alleviates the need for annotating target data. Using kernel density estimation, we match the target data distribution to the source data in the feature space. We demonstrate that our results are comparable or superior on multiple-site prostate MRI and histopathology images, which mitigates the need for annotating target data.

Keywords: Domain Adaptation · Semantic Segmentation · Density and Matching.

1 Introduction

Semantic segmentation is one of the fundamental tasks in computer vision. Human visual systems classify and delineate every object present in their environment. This is especially important in medical imaging because of the highly specific domain knowledge required to outline the relevant objects (e.g., tumor, disease tissue, cancer). Accurately identifying the exact boundaries of these objects (or the size of the tumor) is necessary for reliable and interpretable automation of disease diagnosis, analysis, and treatment planning [1]. Wrong predictions can have disastrous consequences for the patient's health under test.

Deep learning models, when trained with a representative and sufficient amount of training data, do give consistently better predictions. However, because these models are pattern-seeking machines, they can focus on learning spurious signals [2] rather than features of actual disease pathology. Deep learning

models learn low-level texture features more than high-level shape/morphological features [10]. This impacts the performance of the learned model (trained on *source dataset*) when new data with different low-level data statistics (*target dataset*) is introduced during inference. This is called a distributional (or domain) shift in the input dataset. Such a shift results in a loss of precision and trust in the model's predictions based on the new data. Even minor distributional shifts where input images are sketches of the same objects have shown significant drops in performance [3]. This domain shift is problematic in medical imaging [14] because not every site has access to large amounts of training data, hence sites have to rely on models trained from other sites. If the model's prediction cannot be relied upon for a new site, then the applicability of the method is significantly limited.

Domain adaptation [4,5] has been proposed to curb the degradation in the performance of the model. Both supervised and unsupervised domain adaptation (UDA) techniques have been proposed [6,7], depending on whether the model has access to target domain annotations. Because pixel-wise annotation for segmentation tasks, especially in medical images, can be extremely expensive due to the need for specialized knowledge. UDA techniques tend to be more helpful.

Unsupervised domain adaptation approaches for semantic segmentation can be broadly categorized into three classes. First is *adversarial domain adaptation* [4,20], which aims to learn domain-independent backbone features by maximizing the domain classification loss using source and target features and passing a negative gradient to the feature extraction backbone via a gradient reverse layer. Second is *Fourier domain adaptation* [18,21], which uses Fourier domain transformations for domain adaptation. The assumption being that phase information between domains does not change, so adaptation of frequency amplitude can help alleviate the degradation. Third is *density matching*, where the source and target densities of either input space [8], output space [16], or feature space [17] are matched. [8] used conditional GANs (generative adversarial networks) to transform images of source dataset to look like target dataset. Whereas [16] and [17] only use discriminator for density matching between source and target features. Density matching with other penalties, such as Maximum Mean Discrepancy (MMD) [15,19] or Wasserstein GAN [19], has also been tried. Adverserial-based approaches are highly sensitive to hyperparameter selection [4,20]. Fourier domain adaptation frameworks are sensitive to frequency space selection and mixing ratio. But a similar phase assumption might not be true for all domain adaptation applications (e.g., MRI vs CT). Density-matching approaches are highly sensitive to hyperparameters [15,19], and are difficult to train because of minimax games. They also require large amounts of data to converge.

Here, we propose a novel technique for unsupervised domain adaptation for semantic segmentation, where we leverage nonparameteric density matching in the feature space induced by the segmentation networks. Kernel density estimation (KDE) [11,22] has been shown to perform better for generative modeling for smaller datasets [9]. Hence, KDE offers a more stable solution for matching source and target feature densities, compared to adversarial learning, in low-

sample size scenarios, which is typical in medical imaging. The nonparameteric nature of KDE provides a rich training signal for domain adaptation compared with MMD [15] where only moments are used to match density. Furthermore, KDE allows for batch-wise density matching during training, where the full density in the feature space is being matched through the batch samples. The kernel bandwidth is estimated by randomly drawing training samples and mapping them to the feature space. The estimated densities of the source and target datasets are matched using Jenson Shannon divergence (JSD). This regularization does not let the model wander very far from the features of the target dataset making the model learn more generic features which are domain independent.

We compare our results with density matching using MMD [15], but instead of using constant bandwidth as done in other proposed techniques, we estimate the bandwidth of MMD using the same proposed estimation for KDE for a fair comparison. We also compare our results with adversarial training [4,20] and density matching using discriminator in feature space [17] as well as output space [16]. Our method is closely related to feature space [17] and output space [16] density matching but instead of using a discriminator for density matching, we use JSD for divergence and KDE for the underlying probability distribution. We follow the methods listed in the respective papers to implement our own versions for comparison. Our results show that density matching using MMD and JSD performs statistically similarly but better than the other three on both the datasets. The contributions of this paper are as follows:-

- Proposed a novel approach for unsupervised domain adaptation for semantic segmentation that is based on rich (non-parameteric) representation of the underlying feature distribution.
- Demonstrate the efficacy of the proposed approach on different datasets (histopathology[12,13] and multi-site MRI[14]), supported by several ablation experiments to assess the impact of feature space choice, frequency of bandwidth estimation, and target data sample size.

2 Methodology

2.1 Problem Setup

Most deep learning architectures for semantic segmentation follow an encoder-decoder configuration as depicted in Figure 1. Let $f_\theta(.)$ be the encoder and $g_\phi(.)$ be the decoder. For an input image $\mathbf{I}$, the model does the following operations, $\mathbf{x} = f_\theta(\mathbf{I})$ and $\mathbf{y} = g_\phi(\mathbf{x})$, where $\mathbf{x}$ is the encoded features in the learned feature space. For segmentation networks, we can have multiple deep feature encoding and decoding spaces, but for the sake of simplicity, we assume the deepest feature space as $\mathbf{x}$ (one with the lowest spatial resolution and highest channel resolution).

Deep learning models fail to generalize when there is a domain shift in the input space. We hypothesize that this domain shift causes a density shift in the feature space of the learned model, causing it to fail for unseen data. We propose that if the model is regularized by a density-matching loss between feature space

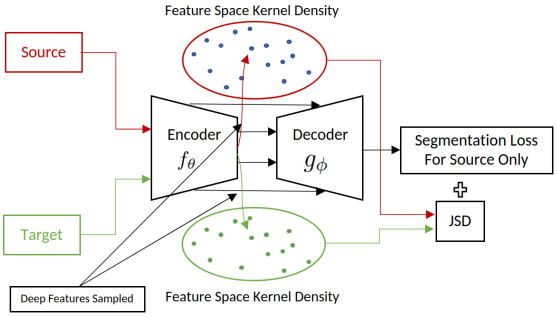

Fig. 1. Block Diagram. The model is assumed to be a standard encoder-decoder configuration with skip connections. It is trained using annotations only from the source dataset. Deep Features are extracted from both the source and the target, KDE is estimated, and JSD is subsequently utilized for matching.

distributions of the two domains, the feature space will not suffer from the same domain shift on seeing the new domain. There are two main aspects to address the feature space density matching (1) the representation of density and (2) the density matching loss.

Representation of density. Density in feature space can be represented by moments (mean, variance) where factorized Gaussian is assumed as the default distribution of the feature space. However, this implies a limiting assumption of a unimodal, distentangled distribution in the feature space. We can also assume parametric densities following certain characteristics of multivariate Gaussian or mixture of Gaussians. But both of these make strong assumptions on distribution of sample points. Non-parametric methods such as KDE, on the other hand, do not make such strong assumptions and are more suited to be learned from data. Hence, these methods have better chance of providing a rich and flexible description of the feature space density. These methods can also be scaled with large number of data points.

Density matching loss. KL divergence is asymmetric property so may not be suitable for domain adaptation application. JSD, on the other hand, is symmetric, which helps the model to learn the features on source dataset but also stay close to the target feature space. This loss acts as a regularizer to the network by not letting the model learn source dataset biases.

2.2 Unsupervised Domain Adaptation via KDE

Block diagram of our proposed methodology is shown in Figure 1. Segmentation Model is trained using annotations from only the source dataset. In our setting, no annotations are used from the target dataset. But this methodology can be used for semi-supervised domain adaptation as well, where we can have access to some annotated samples from the target dataset. Density matching loss acts as

a regularizer, making sure that the feature distribution of the source and target datasets do not diverge from each other. The network is thus trained with loss given by

$$\mathcal{L} = \mathcal{L}_{(seg,source)} + \lambda JSD[p_s, p_t] \tag{1}$$

where $\mathcal{L}_{(seg,source)}$ is the supervised segmentation loss on the source dataset, and λ is a hyper parameter that defines the contribution of the density matching loss, and p_s and p_t are density estimates for source and target dataset, respectively.

Kernal density estimation. Let $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N$ be the number of sampled points from the encoded feature space. The Kernel Density Estimate $p_{est}(\mathbf{x})$ can be written as :

$$p_{est}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} K \left(\frac{\|\mathbf{x} - \mathbf{x}_n\|_2}{\sigma} \right) \tag{2}$$

where K is assumed to be a Gaussian kernel in our experiments. The bandwidth parameter (σ) is estimated to be the mean of the distance between the nearest neighbors in the feature space.

$$\sigma = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x_n} - \gamma(\mathbf{x}_n)\|_2^2 \tag{3}$$

where $\gamma(\mathbf{x}_n)$ returns the nearest neighbour of $\mathbf{x}_n$. Using Eq. 2, we estimate the density of the source (p_s) and target (p_t) datasets using the same kernel but different bandwidth parameters obtained from their respective feature spaces. JSD density matching loss is calculated as :

$$JSD[p_s, p_t] = \frac{1}{2}\big\{KL[p_s, M] + KL[p_t, M]\big\}, M = \frac{p_s + p_t}{2} \tag{4}$$

where KL is the KL-divergence between the two distributions. We tested two scenarios for density matching loss (1) source and target densities are matched with each other and (2) source and target densities are matched to a standard normal. Matching source and target densities performed better than matching both to a standard normal distribution.

3 Results and Discussion

3.1 Experimental Setup

Datasets. We used datasets for gland segmentation in histopathology images and prostate segmentation in a multi-site MRI dataset. Two datasets CRAG [12] and GlaS [13] are used for gland segmentation in the colon histology dataset. CRAG has higher number of samples compared with GlaS, so we are able to test domain adaptation when the source has greater variability than the target and vice versa. A multi-site MRI dataset [14] from six different sites, with different field strengths (3 and 1.5 Tesla) and different vendors, was used with different

source and target configurations. This enabled us to test multi-source, multi-target, as well as held out target settings.

Training setup and hyperparameters. Networks are trained for 5 different train/validation data splits and respective performance(using dice scores [13]) mean and standard deviations are reported when trained from scratch with gaussian initialization. For density estimation, the number of samples for KDE is set to 20. KDE points are sampled every 5 epochs for bandwidth estimation. λ is set to 0.01 for the histopathology dataset and 0.001 for the MRI dataset based on performance on the validation set.

3.2 Results

We report the average result of 5 different splits of the dataset, eliminating the stochasticity that may be caused due to training and validation data used for source training and density estimations. Hyperparameters are optimized using validation metrics and the test set is only used at the end for evaluation.

Domain adaptation results. In Table 1, "No Adapt" refers to the generic case where a model is trained on the source dataset and tested on the target dataset without any finetuning. We treat this as a baseline because we are targeting UDA, where we cannot access target dataset annotations. For other baselines, here "Adver" refers to [4,20] and "DiscF" refers to discriminator in feature space [17] and "DiscO" refers to discriminator in output space [16].

We can observe from Table 1 that there is a significant drop in performance without any domain adaptation. Using adversarial domain adaptation and discriminator in output space, we do not see any benefit for the source or target datasets, but we observe an improvement in the target accuracy using discriminator in the feature space. Feature space density matching using the proposed bandwidth estimation and density matching loss outperforms other techniques. The performance gain is higher when the source dataset is CRAG compared to GlaS. This is possibly due to the difference in the number of samples. The CRAG dataset has more samples, which can result in the model getting more biased toward the source dataset. But the proposed method successfully helps overcome that bias resulting in a higher gain. Qualitative results are shown in Figure 2.

For the multi-site MRI data, prostate segmentation data is available for 6 different sites. Hence, we modified the testing methodology to have out-of-distribution (held-out) domain that is not shown to any network during training or UDA. This setup helps in gauging whether the proposed UDA methodology can improve the model's prediction for an unseen dataset. We club these 6 datasets in multiple training, domain adaptation, and held-out test sets. Tables 2,3(4,5 in supplementary) show results for different settings mentioned above. From Table 2, we can observe that the proposed technique outperforms or is competitive in all target datasets and performs better for the held-out dataset. Table 3 shows results of multi-source, multi-target, and held-out dataset settings. The proposed technique outperforms all others in 4 out of 6 datasets.

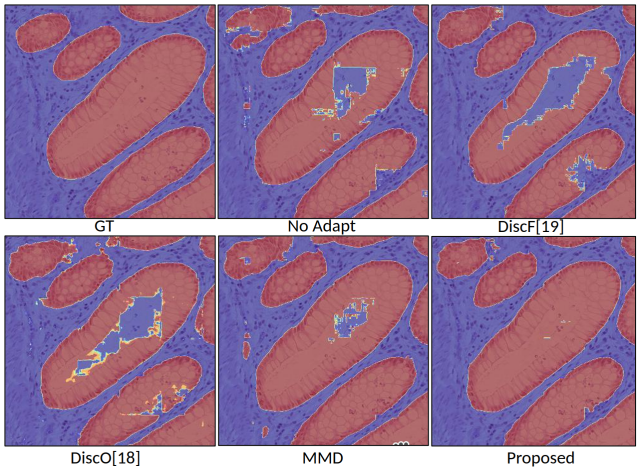

Fig. 2. Qualitative results for Gland Segmentation Results on Target dataset(CRAG) when model is trained using GlaS dataset as source.

Table 1. Mean and standard deviations of Dice Scores when GlaS the is source dataset and CRAG the is target dataset and vice-versa.

	GlaS Source		**CRAG Source**	
	Source	**Target**	**Source**	**Target**
No Adapt	0.874 ± 0.013	0.765 ± 0.043	0.863 ± 0.003	0.834 ± 0.043
Adver [20]	0.767 ±0.028	0.691 ±0.027	0.634 ±0.018	0.58 ±0.04
DiscF [17]	0.888 ± 0.002	0.805 ± 0.012	0.867 ±0.004	0.845 ± 0.038
DiscO [16]	0.88 ± 0.002	0.751 ±0.01	0.865 ±0.003	0.816 ±0.065
MMD	**0.89 ± 0.008**	**0.817 ±0.025**	**0.872 ±0.003**	**0.876 ±0.003**
JSD	**0.895 ±0.003**	**0.81 ±0.012**	**0.87 ±0.004**	**0.879 ±0.005**

Similar results are observed for vendor-grouped results reported in Table 4 and 5 in supplementary. Using more sites for training models results in better performance of target and held-out datasets. The proposed technique is not able to improve results on the "BIDMC" dataset when used as a target or held-out dataset. BIDMC is the only GE data cohort in the multi-site MRI dataset. So this approach works well for the multi-site same vendor, but may not work well for multi-vendor datasets.

Target dataset size ablation. Here, we assess the impact of changing the number of target samples for feature density matching. We trained with three different ratios of target datasets 3%, 30%, and 100% of the target data. The results are shown in Supplementary Table 6. We can observe that there is no significant change in domain adaptation results. This shows that once the model has converged, the distance between densities for each dataset may be constant and unaffected by the sample size used for density estimation and matching.

Table 2. Mean(standard deviations) of Dice Scores when source and target datasets are chosen from 6 sites MRI cohort.

	Source	**Target**				**Held Out**
	RUNMC	**BMC**	**I2CVB**	**UCL**	**BIDMC**	**HK**
No Adapt	0.87(0.022)	0.79(0.01)	0.63(0.046)	0.74(0.026)	0.53(0.007)	0.649(0.035)
Adver [20]	0.65(0.034)	0.55(0.02)	0.55(0.032)	0.50(0.007)	0.47(0.004)	0.491(0.002)
DiscF [17]	0.89(0.009)	**0.82(0.01)**	0.64(0.023)	**0.80(0.011)**	0.53(0.014)	0.63(0.036)
DiscO [16]	0.87(0.013)	0.78(0.02)	0.63(0.023)	0.75(0.034)	0.53(0.025)	0.64(0.048)
MMD	**0.90(0.007)**	0.80(0.01)	**0.65(0.01)**	0.78(0.03)	**0.53(0.02)**	**0.66(0.037)**
JSD	**0.90(0.007)**	0.81(0.01)	**0.67(0.01)**	0.78(0.02)	**0.54(0.012)**	**0.66(0.036)**

Table 3. Mean(standard deviations) of Dice Scores for multi-source and multi-target datasets are choosen from 6 site MRI cohort.

	Source		**Target**		**Held Out**	
	RUNMC	**I2CVB**	**BMC**	**BIDMC**	**UCL**	**HK**
No Adapt	0.85(0.022)	0.89(0.08)	0.79(0.032)	0.50(0.02)	0.66(0.08)	0.51(0.037)
Adver[20]	0.628(0.02)	0.706(0.021)	0.54(0.013)	0.48(0.001)	0.50(0.008)	0.494(0.019)
DiscF[17]	0.89(0.009)	0.90(0.005)	0.81(0.013)	0.51(0.008)	0.70(0.049)	0.51(0.008)
DiscO [16]	0.86(0.027)	0.89(0.01)	0.80(0.016)	0.50(0.006)	0.70(0.03)	**0.53(0.02)**
MMD	**0.89(0.007)**	**0.91(0.004)**	**0.81(0.025)**	0.50(0.008)	**0.72(0.039)**	0.52(0.019)
JSD	**0.90(0.007)**	**0.91(0.004)**	**0.83(0.019)**	0.50(0.01)	**0.72(0.038)**	0.52(0.017)

Feature space ablation. We also tried different feature spaces for the segmentation model. We observed the difference between the test metrics for source and target datasets are not statistically different. Results are shown in Supplementary Table 7.

Frequency of bandwidth estimation. Changing the frequency of bandwidth estimation from 1, 5, 25 , and 125 epochs does not show a significant change in test set performance metrics. Results are shown in Table 8 in supplementary.

4 Conclusion and Future Work

We proposed a technique for unsupervised domain adaptation based on density matching and non-parametric density estimate. We showed the efficacy of the proposed approach on 2 different modalities datasets, histopathology and multi-site MRI. The proposed technique not only improves results on target datasets but also showed consistent improvement in source and held-out results. Evaluating whether performing density matching in more than one feature space can help a model acquire a more accurate representation is a topic for future research. Although the proposed method is not sensitive to hyperparameters, it does require that number of KDE points and kernel bandwidth to be correctly chosen for the dataset. One future direction would be find these hyper parameters automatically dependent on feature space diversity.

References

1. Wang, Y.C., Hsieh, T.C., Yu, C.Y., Yen, K.Y., Chen, S.W., Yang, S.N., Chien, C.R., Hsu, S.M., Pan, T., Kao, C.H. and Liang, J.A., 2012. The clinical application of 4D 18F-FDG PET/CT on gross tumor volume delineation for radiotherapy planning in esophageal squamous cell cancer. Journal of radiation research, 53(4), pp.594-600.
2. DeGrave, Alex J., Joseph D. Janizek, and Su-In Lee. "AI for radiographic COVID-19 detection selects shortcuts over signal." Nature Machine Intelligence 3, no. 7 (2021): 610-619.
3. Wang, Haohan, Songwei Ge, Zachary Lipton, and Eric P. Xing. "Learning robust global representations by penalizing local predictive power." Advances in Neural Information Processing Systems 32 (2019).
4. Tzeng, Eric, Judy Hoffman, Kate Saenko, and Trevor Darrell. "Adversarial discriminative domain adaptation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7167-7176. 2017.
5. Vu, Tuan-Hung, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2517-2526. 2019.
6. Toldo, Marco, Andrea Maracani, Umberto Michieli, and Pietro Zanuttigh. "Unsupervised domain adaptation in semantic segmentation: a review." Technologies 8, no. 2 (2020): 35.
7. Wang, Zhonghao, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-Mei Hwu, Thomas S. Huang, and Honghui Shi. "Alleviating semantic-level shift: A semi-supervised domain adaptation method for semantic segmentation." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 936-937. 2020.
8. Bousmalis, Konstantinos, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. "Unsupervised pixel-level domain adaptation with generative adversarial networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3722-3731. 2017.
9. Saha, Surojit, Shireen Elhabian, and Ross Whitaker. "GENs: generative encoding networks." Machine Learning 111, no. 11 (2022): 4003-4038.
10. Hermann, Katherine, Ting Chen, and Simon Kornblith. "The origins and prevalence of texture bias in convolutional neural networks." Advances in Neural Information Processing Systems 33 (2020): 19000-19015.
11. Węglarczyk, Stanisław. "Kernel density estimation and its application." In ITM Web of Conferences, vol. 23, p. 00037. EDP Sciences, 2018.
12. Graham, Simon, Hao Chen, Jevgenij Gamper, Qi Dou, Pheng-Ann Heng, David Snead, Yee Wah Tsang, and Nasir Rajpoot. "MILD-Net: Minimal information loss dilated network for gland instance segmentation in colon histology images." Medical image analysis 52 (2019): 199-211.
13. Sirinukunwattana, Korsuk, Josien PW Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang et al. "Gland segmentation in colon histology images: The glas challenge contest." Medical image analysis 35 (2017): 489-502.
14. Liu, Quande, Qi Dou, Lequan Yu, and Pheng Ann Heng. "MS-Net: multi-site network for improving prostate segmentation with heterogeneous MRI data." IEEE transactions on medical imaging 39, no. 9 (2020): 2713-2724.
15. Li, Chun-Liang, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. "Mmd gan: Towards deeper understanding of moment matching network." Advances in neural information processing systems 30 (2017).

16. Tsai, Yi-Hsuan, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. "Learning to adapt structured output space for semantic segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7472-7481. 2018.
17. Long, Mingsheng, Yue Cao, Jianmin Wang, and Michael Jordan. "Learning transferable features with deep adaptation networks." In International conference on machine learning, pp. 97-105. PMLR, 2015.
18. Yang, Yanchao, and Stefano Soatto. "Fda: Fourier domain adaptation for semantic segmentation." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4085-4095. 2020.
19. Erkent, Özgür, and Christian Laugier. "Semantic segmentation with unsupervised domain adaptation under varying weather conditions for autonomous vehicles." IEEE Robotics and Automation Letters 5, no. 2 (2020): 3580-3587.
20. Bolte, Jan-Aike, Markus Kamp, Antonia Breuer, Silviu Homoceanu, Peter Schlicht, Fabian Huger, Daniel Lipinski, and Tim Fingscheidt. "Unsupervised domain adaptation to improve image segmentation quality both in the source and target domain." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 0-0. 2019.
21. Xu, Qinwei, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. "A fourier-based framework for domain generalization." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14383-14392. 2021.
22. Kim, JooSeuk, and Clayton D. Scott. "Robust kernel density estimation." The Journal of Machine Learning Research 13, no. 1 (2012): 2529-2565.

5 Supplementary

Table 4. Multi-site source and target: The means (standard deviations) of Dice Scores. For all experiments presented here and in the main paper, the learning rate, batch size, and a number of epochs are set to 1e-4, 10, and 1000, respectively. All models were trained using the PyTorch framework and the Adam optimizer with a 1e-4 weight decay on Nvidia A30(24 GB) GPUs. The optimal hyperparameters were selected based on the performance of the validation set. 20 % of the training set was fixed as the validation set. Testing is performed after hyperparameter optimization.

	Source			**Target**	**Held Out**	
	RUNMC	**I2CVB**	**BIDMC**	**UCL**	**BMC**	**HK**
No Adapt	0.86(0.021)	0.87(0.024)	0.70(0.048)	0.79(0.042)	0.83(0.007)	0.70(0.02)
DiscF [17]	0.9(0.004)	0.91(0.006)	0.71(0.021)	0.82(0.01)	0.86(0.007)	0.69(0.029)
DiscO[16]	0.87(0.027)	0.90(0.013)	0.71(0.015)	0.80(0.009)	0.84(0.007)	0.66(0.019)
MMD	**0.90(0.009)**	**0.91(0.008)**	**0.75(0.012)**	**0.83(0.021)**	**0.87(0.007)**	**0.72(0.019)**
JSD	**0.90(0.007)**	**0.91(0.005)**	**0.74(0.002)**	**0.84(0.001)**	**0.86(0.007)**	**0.72(0.018)**

Table 5. Multi-site Source Datasets for Single Vendor : Mean(standard deviations) of Dice Scores for multi-source same vendor and multi-target(same or other vendors) datasets are choosen from 6 sites MRI cohort. Overall there is a good performance gain, more than 5% relative increase in performance, except when tested on the BIDMC dataset (GE vendor).

	Source		**Target**	**Held Out**		
	HK	**UCL**	**RUNMC**	**BMC**	**BIDMC**	**I2CVB**
No Adapt	0.87(0.028)	0.84(0.036)	0.78(0.03)	0.77(0.007)	0.56(0.007)	0.64(0.034)
Adver[20]	0.57(0.065)	0.49(0.018)	0.51(0.028)	0.53(0.021)	0.48(0.007)	0.52(0.029)
DiscF [17]	0.87(0.007)	0.86(0.019)	0.80(0.007)	0.76(0.008)	0.54(0.007)	0.65(0.011)
DiscO[16]	0.81(0.05)	0.77(0.044)	0.74(0.038)	0.71(0.025)	0.53(0.007)	0.63(0.033)
MMD	**0.89(0.019)**	**0.88(0.021)**	**0.83(0.014)**	**0.81(0.009)**	**0.58(0.024)**	**0.66(0.017)**
JSD	**0.88(0.006)**	**0.89(0.021)**	**0.83(0.013)**	**0.81(0.021)**	**0.57(0.032)**	**0.67(0.027)**

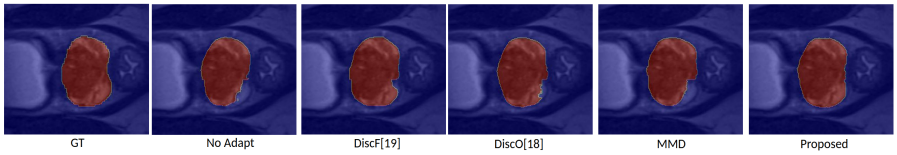

Fig. 3. Qualitative results for MRI cohort Results on BMC target corresponding to table 5

.

Table 6. Target Data size Ablation Mean Dice Score and standard deviations for test metrics with different percentages of the target dataset.

	GlaS Source		**CRAG Source**	
	Source	**Target**	**Source**	**Target**
No Adapt	0.874 ± 0.013	0.765 ± 0.043	0.863 ± 0.003	0.834 ± 0.043
JSD (100%)	**0.895 ±0.003**	**0.81 ±0.012**	**0.87 ±0.004**	**0.879 ±0.004**
JSD (30%)	**0.893 ±0.005**	**0.80 ±0.015**	**0.872 ±0.004**	**0.866 ±0.001**
JSD (3%)	**0.893 ±0.004**	**0.81 ±0.013**	**0.875 ±0.004**	**0.8725 ±0.001**

Table 7. Feature Space Ablation Mean and standard deviation of performance for density matching for different encoder and decoder feature spaces. We used a U-Net with skip connections with up to 5 decompositions for our experiments. The weights were initialized using Pytorch default initialization. Encoders are assumed to be feature spaces just before the skip connections in the encoding pipeline. Decoder feature spaces are just before skip connections in the decoding pipeline.

	CRAG Source	
Feature Space	**Source**	**Target**
Deepest	0.87 ±0.003	0.876 ±0.012
ENC1	0.871 ±0.005	0.877 ±0.015
ENC2	0.874 ±0.001	0.876 ±0.009
ENC3	0.872 ±0.003	0.874 ±0.009
ENC4	0.871 ±0.001	**0.877 ±0.004**
DEC4	0.869 ±0.014	0.874 ±0.009
DEC3	0.870 ±0.007	0.876 ±0.011
DEC2	**0.875 ±0.003**	0.875 ±0.013
DEC1	0.866 ±0.013	0.868 ±0.016

Table 8. Bandwidth Estimation Frequency Ablation Mean Dice Score and standard deviations for test metrics with different frequency/epochs for bandwidth estimation.

	GlaS Source		**CRAG Source**	
	Source	**Target**	**Source**	**Target**
5 epochs	**0.895 ±0.003**	0.81 ±0.012	0.87 ±0.004	**0.879 ±0.004**
1 epoch	0.894 ±0.007	0.816 ±0.017	**0.874 ±0.001**	0.877 ±0.003
25 epoch	0.894 ±0.004	0.805 ±0.023	**0.874 ±0.004**	0.875 ±0.004
125	0.894 ± 0.002	**0.821 ± 0.013**	0.866 ± 0.003	0.863 ± 0.043