HyDA: Hypernetworks for Unsupervised Domain Adaptation in Medical Imaging

Anonymized Authors

Anonymized Affiliations
email@anonymized.com

Abstract. Domain shift is a significant challenge in adapting medical imaging analysis models for practical use. Medical imaging datasets often exhibit variations due to differences in acquisition protocols, patient demographics, and imaging devices. Current approaches often struggle with generalization across diverse datasets, leading to decreased performance when models trained on data from specific sources are applied to others. These limitations hinder the widespread deployment of models in clinical settings, where data variability is inevitable.
In this work, we introduce HyDA, a novel hypernetwork-based framework that leverages domain information instead of ignoring it, enabling dynamic adaptation at inference time. Unlike prior UDA methods that only utilize domain priors during training, HyDA learns implicit domain representations and uses them to adjust model parameters on-the-fly, making it the first approach to provide real-time adaptation to unseen medical domains. We validate HyDA on two clinically relevant tasks — MRI brain age prediction and chest X-ray pathology classification — demonstrating its ability to generalize across tasks and modalities. Our code is available at https://github.com/****/hyda (anonymized for submission, see supplementary material for an early version of the code).

Keywords: Domain Adaptation · Hypernetworks · MRI · Xray

1 Introduction

1.1 Unsupervised Domain Adaptation

One of the main challenges of medical image analysis is the domain shift observed when real world test cases differ in distribution to the available training set. Unsupervised domain adaptation (UDA) is an approach that aims to minimize the domain shift by transferring knowledge from a labeled training set to an unlabeled test set, referred to as the source and target domain respectively. There are several common approaches to UDA:

Domain adversarial learning introduced by Ganin et. at. [6], this method proposes to train a domain discriminator on top of the existing feature extractor, and utilize a gradient reversal layer (GRL) to encourage the shared latent

space to be domain-agnostic. This method, coined Domain-Adversarial Neural Networks (DANN), has since been improved and expanded upon ([22], [23] [25]), but the core principle of learning to ignore domain features remains. While this method has proven successful, it is not without its limitations - it requires access to the target domain data during training and its main focus is learning to abstract away the domain information in favor a domain agnostic representation. Instead, we propose to learn a domain specific representation and use it to dynamically adapt the model based on domain features. This enables adapting models on-the-fly to unseen target domains.

Feature alignment methods utilize metrics and measures from the field of information theory to disentangle task-specific features from domain-specific features or to reduce dissimilarity between source domain and target domain embeddings. Common metrics for this are Maximum Mean Discrepancy (MMD) [13], and more recently, variants of mutual information (MI) [14].

Test-time adaptation (TTA) is a subset of methods which aim to adapt pretrained models at test-time, after already being trained on source domains only. These methods mostly rely on some form of entropy minimization as a form of ensuring confident predictions on the target domain [19] [11] [24]. These methods are mostly tailored for multi-class classification tasks, which means they cannot be applied to common medical tasks such as regression or multi-label classification. Additionally, these methods often require specific model architecture and multiple forward passes per sample at inference time. Our approach is compatible with any task and architecture and uses a uses single-pass forward at training and inference time.

We propose to extract domain features directly from the input and utilize this learned domain embedding space in order to condition the model's predictions on the domain characteristics. This allows for adaptation at inference time and generalization towards unseen domains, so long as they are well represented in the domain feature space. Additionally, we propose a general framework that can be easily implemented alongside existing UDA methods.

1.2 Hypernetworks

First introduced by Ha et al.[7], hypernetworks are neural networks trained to generate weights for other networks, typically called primary networks. This meta-learning approach enables dynamically creating a unique set of weights for each input. Hypernetworks are trained using gradients from the primary network, and usually require specialized initialization and regularization methods [2].

Previous works have shown hypernetworks are useful in a variety of tasks including 3D shape reconstruction [12], federated learning [18] and medical image segmentation [15] to name a few. Notably, Aharon et. al. [1] have demonstrated

hypernetworks are capable of interpolation by conditioning an image denoising model on the expected noise variance and testing on previously unseen variances. Additionally, Duenias et. al. [4] have shown hypernetworks can be used to condition medical images features on some tabular metadata. Building on these ideas, we demonstrate that hypernetworks can be used for UDA of medical images by generating weights from domain features, effectively interpolating across a domain space.

2 Method

2.1 overview

The proposed method builds upon the traditional supervised learning setup which consists of a feature encoder and a task head, by adding a domain classification branch with its own encoder and head. The embedding obtained by the domain encoder is then used to condition the task branch using a hypernetwork. The main motivation for adding this branch is to learn implicit domain features and use them to adapt the network for unseen domains.

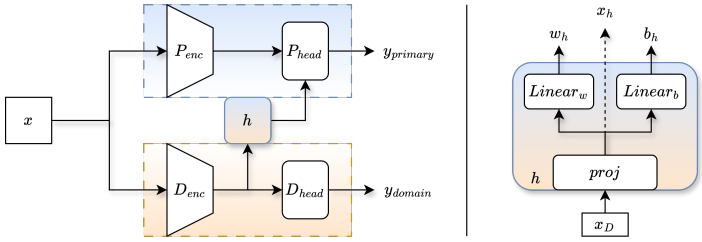


Fig. 1. Overview of the proposed method (left), with a detailed view of the hypernetwork (right). Input image x is passed to a network with 2 branches. The primary branch (top, denoted P) which performs the main task, and the domain branch (bottom, denoted D) which is trained to extract implicit domain features. Both branches are comprised of an encoder $_{enc}$ and a head $_{head}$. The domain features x_D are used as input to hypernetwork h, which generates weights and biases (w_h, b_h) for the primary head P_{head}. Optionally, the hypernetwork outputs embeddings x_h to be used for auxiliary multi-similarity loss.

Fig 1 illustrates the main building blocks and data flow of the proposed method. Given an input image $X \in \mathbb{R}^{W \times H \times L}$, a traditional supervised learning setup is comprised of the task branch only - the input is mapped to some lower dimension $X_P \in \mathbb{R}^{f_P}$ by the task encoder, and the task head maps the low-dimensional embedding to the output space y_P.

We propose to add an additional branch which allows incorporating domain information in a dynamic manner - changing the weights of the network during

inference - depending on the input's domain characteristics. The domain encoder maps the input x to a domain embedding $x_D \in \mathbb{R}^{f_D}$, which is used as input to a hypernetwork h. The hypernetwork generates weights for layers in the primary head P_{head}, effectively conditioning its output on the domain embedding. x_D is also passed to a domain head which is used during training only to allow learning of a meaningful domain embedding space.

The overall architecture can be described by the following equation:

$$y_P = P_{head}(P_{enc}(X), h(D_{enc}(x))$$

$$y_D = D_{head}(D_{enc}(X))$$

where y_P is the primary output and y_D is an auxiliary output used to train the domain brach. Training the primary branch is done using a loss suitable for the primary task (i.e cross entropy for classification, MSE/MAE for regression, etc.), where the gradients are also applied to the hypernetwork. We do not propagate the gradients from the primary branch to the domain encoder in order to ensure that is x_D is a domain embedding and does not represent any task specific information. This design choice essentially enforces x_D to represent domain-specific features while freeing x_P to represent task-specific features.

2.2 Domain Conditioning Hypernetwork

The hypernetwork plays a key role in our suggested framework, as it serves as the connection between the primary branch and the domain branch. The hypernetwork maps a given input x_D to weights and biases (w_h, b_h) to be used in certain layers in the primary network. This property of the hypernetwork allows it to be used in any layer of any given primary branch - making it applicable in almost any setting.

Given a mini batch of data, the hypernetwork generates a different set of weights and biases for each element in the batch. This allows for conditioning the primary network's output on domain features, while enabling domain-heterogeneous batches. In a standard linear layer, the output is computed as:

$$y = x * w, \quad w \in \mathbb{R}^{(p,n)}$$

where $x \in \mathbb{R}^{(b,m,p)}$ is the input batch, $*$ is matrix multiplication, and $y \in \mathbb{R}^{(b,m,n)}$ is the produced output.

A hyper-linear layer instead assigns a unique weight matrix to each batch element:

$$y_i = x_i * w_i, \quad w_i \in \mathbb{R}^{(p,n)} \quad i \in (0, b)$$

Other than the output layer, the hypernetwork is not limited to a specific architecture and can be implemented in any way that contributes to the primary task. For simplicity, we use a single linear layer followed by a Rectified Linear Unit (ReLU) activation. We show that even such a simple implementation is enough to successfully generate domain-aware weights for the primary network.

In order to ensure stable convergence, We follow Chang et. al. [2] hypernetwork weight initialization method such that its output will maintain the input variance to the primary network. Additionally, we constrain the generated weights by adding their $L2$ norm as a regularization factor to the loss.

2.3 Losses

The primary branch and the hypernetwork are trained using the following loss:

$$L_P = L_{task} + \lambda_P |W_P| + \lambda_h |W_h|$$

where L_{task} is the loss used for the primary task the model is trained on (e.g. cross-entropy for classification, MSE for regression), W_P are the primary network's weights, W_h are the hypernetwork generated weights, and λ_P, λ_h are their corresponding coefficients.

The domain branch is trained on the dual task of domain classification and metric learning using the following loss:

$$L_D = L_{CE} + \alpha L_{MSim} + \lambda_D |W_D| \tag{1}$$

Where L_{CE} is cross-entropy loss, L_{MSim} is multi-similarity loss as suggested by Wang et. al. [21], W_D are the domain network's weights and α, λ_D are coefficients. L_{MSim} can be calculated on the domain embeddings x_D and, depending on implementation, any internal representation within the hypernetwork x_h. Calculating multi-similarity loss on auxiliary outputs of the hypernetwork allows it to maintain domain-specific representation while also learning to generate weights for the primary network.

This optimization scheme ensures each branch is optimized separately, and that the hypernetwork is optimized for utilizing the learned domain representation as a context for the primary task.

3 Experiments & Results

We demonstrate our proposed framework on two different tasks using medical imaging data - chest X-ray pathology classification, and MRI brain age prediction.

3.1 Chest X-ray Pathology Classification

For this experiment we train a model to perform multi-label classification on chest X-ray scans from 3 publicly available datasets. We compare HyDA to a baseline model with no adaptation at all, and to 2 adaptation methods - one for UDA and one for TTA. We compare to the soft version of MDAN [25] as a UDA comparison and to TENT [19] as a TTA method.

The Data We used three publicly available datasets: **NIH** [20], **CheXpert** [9] and **VinDr** [16]. A subset of 5 classes were chosen for the classification task as they were the only classes that were present in all 3 datasets: Atelectasis, Cardiomegaly, Consolidation, Effusion and Pneumothorax. The combined dataset includes 90,570 X-ray scans. Following pervious works on this data, we resize the images to 224×224 and scale pixel values to $[-1024, 1024]$. Additionally, we employ data augmentation in the form of a random affine transformation of 45°, translation of 15% and a scale factor in range $(0.9, 1.1)$.

Implementation Details Following previous work on chest X-ray classification [17] [3], we fine-tune a DenseNet121 model [8] pre-trained on ImageNet. The model's input and output layers were replaced to handle single channel images and output 5 classes. The domain branch is a simple CNN with 4 convolution blocks and a single linear layer for classification. Since DenseNet has a single classification layer, the hypernetwork is an MLP that generates a single set of weights and biases. Both baseline and HyDA models were trained for 150 epochs with AdamW optimizer with learning rate of $1e-3$, weight decay of 0.05 and a cosine annealing learning rate scheduler with minimum learning rate of $1e-6$.

Results Table 1 shows the results of the chest X-ray experiment, measured in area under ROC curve (AUC). HyDA provides the best results in both fully-supervised and in leave-one-out settings. It is worth noting that the improvement over the baseline correlates with the separability of the domain features as seen in fig. 2. For target domains that were well-clustered such as CheXpert and VinDr, HyDA improves over the baseline by a larger margin than compared to target domains that were ambiguously clustered such as NIH. This result emphasizes the importance of quality domain features for domain adaptation.

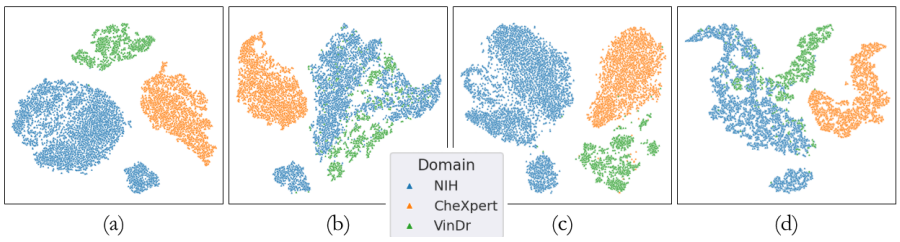


Fig. 2. t-SNE projections of domain features obtained from the HyDA domain branch trained in a leave-one-out setting. The plot illustrate how well the learned domain features handle previously unseen domains: No target domain (a), NIH (b), CheXpert (c) and VinDr (d).

Model	Target Domain	Atel.	Cardio.	Cons.	Eff.	Pneu.	Avg.
Baseline	-	0.85	0.95	0.86	0.94	0.87	0.89
MDAN	-	0.86	0.96	0.86	0.94	0.88	0.90
HyDA	-	**0.87**	**0.97**	**0.86**	**0.94**	**0.89**	**0.91**
Baseline	NIH	**0.70**	0.81	**0.76**	0.86	0.77	0.78
MDAN	NIH	0.67	0.89	0.76	0.86	0.77	0.79
TENT	NIH	0.61	0.70	0.64	0.81	0.67	0.69
HyDA	NIH	0.68	**0.89**	0.75	**0.88**	**0.79**	**0.80**
Baseline	CheXpert	0.81	**0.86**	0.73	0.87	0.74	0.80
MDAN	CheXpert	0.77	0.76	0.71	0.84	0.72	0.76
TENT	CheXpert	0.76	0.86	0.77	0.89	0.76	0.81
HyDA	CheXpert	**0.82**	0.85	**0.82**	**0.89**	**0.74**	**0.82**
Baseline	VinDr	0.60	0.76	0.85	0.88	0.91	0.80
MDAN	VinDr	**0.68**	0.82	0.88	0.87	0.89	0.83
TENT	VinDr	0.51	0.72	0.80	0.74	0.86	0.73
HyDA	VinDr	0.66	**0.87**	**0.93**	**0.89**	**0.92**	**0.85**

Table 1. Chest X-ray classification results measured by AUC on each class. Disease abbreviations: Atel (Atelectasis), Cardio (Cardiomegaly), Cons (Consolidation), Eff (Effusion), Pneu (Pneumothorax). Each group of rows compares a baseline model to MDAN, TENT, and HyDA models on the same target domain. Best results in bold.

Ablation Study In order to quantify the contribution of the different losses, we ablated the different losses used to train the domain branch. Results are shown in table 2 and illustrate the importance of each loss component to achieving the best possible performance. This incremental improvement in the primary task is achieved using task-agnostic losses, which points to the fact that quality domain features are indeed helpful for domain adaptation.

CE	Msim	Aux Msim	NIH	CheXpert	VinDr
✓			0.72 (0.11)	0.79 (0.05)	0.81 (0.13)
✓	✓		0.76 (0.09)	0.81 (0.06)	0.83 (0.10)
✓	✓	✓	**0.80 (0.08)**	**0.82 (0.05)**	**0.85 (0.10)**

Table 2. Loss ablation study results. Each row represents an incremental combination of losses used in the domain branch starting with cross-entropy (CE), adding multi-similarity (MSim) and auxiliary multi-similarity (Aux MSim). Results are average AUC (std in brackets) on the target domain for each of the three datasets.

3.2 Brain Age Prediction

For this experiment, we utilize a diverse dataset of brain MRI scans to train a brain age prediction model. The task of brain age prediction serves as an important biomarker - a large gap between a patient's chronological age and predicted age might indicate a neurodegenerative disease or cognitive decline (i.e. Alzheimer) [5].

The Data We adopted the same dataset and preprocessing workflow suggested by Levakov et. at. [10]. The dataset includes 26,691 brain MRI scans from 19 different studies.

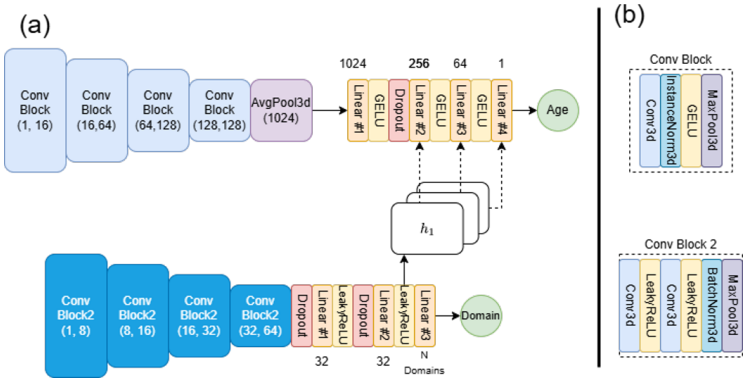


Fig. 3. Overview of the brain age prediction experiment. The primary network is comprised of a CNN encoder with 4 conv blocks and a 4 layer MLP regressor. The hypernetwork generates weights to any of the last 3 layers of the regressor based on domain implicit features. The domain branch follows a similar design to the primary network but has less parameters overall. The domain features are obtained from the output of the last layer before domain classification.

Implementation Details In this experiment, both the task branch and the domain branch are comprised of a CNN encoder and an MLP head. The task branch solves the brain age prediction problem as a regression task. The domain branch is trained to distinguish between different MRI datasets. The hypernetwork generates weights for the age classifier given the domain embedding obtained from the final layer of the domain branch. Architecture details are illustrated in fig. 3

The model was trained using AdamW optimizer with a learning rate of $1e-4$, weight decay of 0.05 and a cosine annealing learning rate scheduler with a minimum learning rate of $1e-6$ for 150 epochs.

Results Tables 3 and 4 show the results of the brain age experiment. Notably, HyDA improves over the baseline both in a fully supervised scenario and in a leave-one-out scenario, which is more representative of a UDA setting. These results demonstrate HyDA's ability to learn meaningful domain representations, interpolate to unseen domains and utilize the domain features to adapt the model on-the-fly. This is seen in fig. 4(b) where samples from a previously unseen domain are projected in between existing domains, effectively interpolating a new domain representation. The effectiveness of this representation is shown in fig 4(a) where target domain features are better aligned with source domain features in our HyDA model compared to the baseline model.

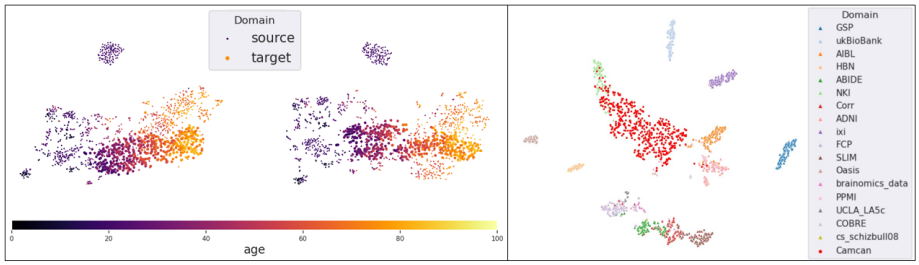
Model	CNP	NKI	ixi	Oasis	ABIDE	ADNI	AIBL	PPMI	Camcan	SLIM	Avg. (std)
Baseline	3.11	3.01	3.54	**3.29**	2.09	**2.8**	**2.74**	4.23	3.35	0.47	2.86 (0.96)
HyDA (ours)	**2.39**	**2.92**	**3.22**	**3.29**	**1.74**	3.04	2.94	**3.94**	**3.21**	**0.37**	**2.71 (0.95)**

Table 3. Brain age prediction experiment results in a fully supervised setting. Since all domains were seen during training, results are measured as mean average error (MAE) on the validation set.

Model	CNP	NKI	ixi	Oasis	ABIDE	ADNI	AIBL	PPMI	Camcan	SLIM	Avg. (std)
Baseline	3.36	3.90	4.41	5.40	3.25	**4.31**	3.56	**4.15**	3.50	1.44	3.73 (0.97)
HyDA (ours)	**2.86**	**3.44**	**4.14**	**5.20**	**3.16**	4.48	**3.45**	4.24	**3.35**	**1.34**	**3.57 (1.00)**

Table 4. Brain age prediction experiment results in a leave-one-out setting (i.e., single target domain). Each entry represents the test MAE of a Baseline/Hypernet model trained on that particular target domain.

Ablation Study In order to asses the robustness of the proposed framework, we ablate the number of layers for which the hypernetwork generates weights. The primary network has an MLP head with 4 layers, out of which 3 are suitable for weight generation. Table 5 shows that utilizing the domain features to generate weights improves over the baseline for most layer combinations, showing robustness to layer choice. One can interpret these results via the lens of task-specific and domain-specific signals: the hypernetwork generated weights are domain-specific while static weights are task-specific. Combining task and domain specific weights provides optimal performance. Using only task specific weights prevents generalization to unseen domains and using only domain specific weights erodes the task related information and yields the worst results.



(a) (b)

Fig. 4. Results for Camcan as target domain. (a) t-SNE projections of age features from the primary branch of a baseline model (left) and our proposed method (right). Target domain projections are enlarged for better distinction. (b) t-SNE projections of domain features obtained from the HyDA domain branch. The target domain projections (red) are located in between the source domain projections, effectively demonstrating the model's interpolation capabilities to unseen domains.

Layer 1	Layer 2	Layer 3	Average (std)
			4.57 (0.62)
✓			4.36 (0.55)
	✓		4.43 (0.70)
		✓	4.28 (0.72)
	✓	✓	4.26 (0.72)
✓	✓	✓	4.68 (0.73)

Table 5. Hypernetwork layer ablation. We ablate the layers in the primary network for which the hypernetwork generates weights. Each layer configuration was trained on 3 target domains (NKI, ixi, Oasis). Reported results are mean (std) of target domain MAE.

4 Conclusion

In this work, we introduced HyDA, a novel hypernetwork-based framework for unsupervised domain adaptation in medical imaging. Unlike existing UDA methods that focus on learning domain-invariant features, HyDA leverages domain information explicitly to dynamically adapt model parameters at inference time. This allows for on-the-fly adaptation to unseen medical domains, allowing for a robust deployment of models in clinical applications.

We demonstrated HyDA's effectiveness on two distinct medical imaging tasks: MRI brain age prediction and chest X-ray pathology classification. In both tasks, HyDA outperformed both the naive approach and traditional domain invariant learning methods, highlighting its superiority in handling complex medical domain shifts. Furthermore, we show that HyDA's learned domain representations enable meaningful interpolation between domains, thus improving model generalization.

Beyond its strong empirical performance, HyDA is a task-agnostic and modality-agnostic framework that can be seamlessly integrated into various medical imaging pipelines. Instead of tailoring the adaptation method to the primary task or to a specific modality or model architecture, HyDA aims to be a general purpose framework that can be utilized in any setting.

By addressing the critical challenge of domain shift in medical imaging, HyDA provides a framework for adapting models to provide consistent results in clinical settings, ensuring robustness to unseen domains.

References

1. Aharon, S., Ben-Artzi, G.: Hypernetwork-based adaptive image restoration. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)
2. Chang, O., Flokas, L., Lipson, H.: Principled weight initialization for hypernetworks. In: International Conference on Learning Representations (2019)
3. Cohen, J.P., Hashir, M., Brooks, R., Bertrand, H.: On the limits of cross-domain generalization in automated x-ray prediction. In: Medical Imaging with Deep Learning. pp. 136–155. PMLR (2020)
4. Duenias, D., Nichyporuk, B., Arbel, T., Raviv, T.R.: Hyperfusion: A hypernetwork approach to multimodal integration of tabular and medical imaging data for predictive modeling. arXiv preprint arXiv:2403.13319 (2024)
5. Elliott, M.L., Belsky, D.W., Knodt, A.R., Ireland, D., Melzer, T.R., Poulton, R., Ramrakha, S., Caspi, A., Moffitt, T.E., Hariri, A.R.: Brain-age in midlife is associated with accelerated biological aging and cognitive decline in a longitudinal birth cohort. Molecular psychiatry **26**, 3829–3838 (2021)
6. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of machine learning research **17**(59), 1–35 (2016)
7. Ha, D., Dai, A.M., Le, Q.V.: Hypernetworks. In: International Conference on Learning Representations (2017), https://openreview.net/forum?id=rkpACe1lx

8. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
9. Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., et al.: Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 590–597 (2019)
10. Levakov, G., Rosenthal, G., Shelef, I., Raviv, T.R., Avidan, G.: From a deep learning model back to the brain—Identifying regional predictors and their relation to aging. Human Brain Mapping **41**(12), 3235–3252 (2020). https://doi.org/10.1002/hbm.25011, https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.25011
11. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: International conference on machine learning. pp. 6028–6039. PMLR (2020)
12. Littwin, G., Wolf, L.: Deep meta functionals for shape representation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1824–1833 (2019)
13. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015. pp. 97–105 (2015)
14. Lu, C., Zheng, S., Gupta, G.: Unsupervised domain adaptation for cardiac segmentation: Towards structure mutual information maximization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 2588–2597 (June 2022)
15. Ma, T., Dalca, A.V., Sabuncu, M.R.: Hyper-convolution networks for biomedical image segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1933–1942 (2022)
16. Nguyen, H.Q., Lam, K., Le, L.T., Pham, H.H., Tran, D.Q., Nguyen, D.B., Le, D.D., Pham, C.M., Tong, H.T., Dinh, D.H., et al.: Vindr-cxr: An open dataset of chest x-rays with radiologist's annotations. Scientific Data **9**(1), 429 (2022)
17. Rajpurkar, P.: Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. ArXiv preprint (2017)
18. Shamsian, A., Navon, A., Fetaya, E., Chechik, G.: Personalized federated learning using hypernetworks. In: International Conference on Machine Learning. pp. 9489–9502. PMLR (2021)
19. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. arXiv preprint arXiv:2006.10726 (2020)
20. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2097–2106 (2017)
21. Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5022–5030 (2019)
22. Wei, G., Lan, C., Zeng, W., Zhang, Z., Chen, Z.: Toalign: Task-oriented alignment for unsupervised domain adaptation. Advances in Neural Information Processing Systems **34**, 13834–13846 (2021)
23. Wolleb, J., Sandkühler, R., Bieder, F., Barakovic, M., Hadjikhani, N., Papadopoulou, A., Yaldizli, O., Kuhle, J., Granziera, C., Cattin, P.C.: Learn to

Ignore: Domain Adaptation for Multi-site MRI Analysis. In: Medical Image Computing and Computer Assisted Intervention - MICCAI 2022. pp. 725–735. Lecture Notes in Computer Science, Springer Nature Switzerland (2022)
24. Zhang, M., Levine, S., Finn, C.: Memo: Test time robustness via adaptation and augmentation. Advances in neural information processing systems **35**, 38629–38642 (2022)
25. Zhao, H., Zhang, S., Wu, G., Moura, J.M., Costeira, J.P., Gordon, G.J.: Adversarial multiple source domain adaptation. Advances in neural information processing systems **31** (2018)