SFHarmony: Source Free Domain Adaptation for Distributed Neuroimaging Analysis - Supplementary Material

Nicola K Dinsdale[1] Mark Jenkinson[2,3,4] Ana IL Namburete[1,2]

1. Oxford Machine Learning in NeuroImaging (OMNI) Lab, Department of Computer Science University of Oxford, UK
2. Wellcome centre for Integrative Neuroimaging, FMRIB, University of Oxford, Oxford, UK
3. Australian Institute for Machine Learning (AIML), Department of Computer Science, University of Adelaide, Adelaide, Australia
4. South Australian Health and Medical Research Institute (SAHMRI), North Terrace, Adelaide, Australia

nicola.dinsdale@cs.ox.ac.uk

1. Bhattacharyya Distance

The Bhattacharyya Distance is a general distance between two distributions, which can be used to measure the degree of similarity between two probability distributions. For two probability distributions P and Q on the same domain, the Bhattacharyya distance is defined as:

$$D_B(P,Q) = -ln(BC(P,Q)) \tag{1}$$

where

$$BC(P,Q) = \sum \sqrt{P(x)Q(x)} \tag{2}$$

is the Bhattacharyya coefficient for discrete probability distributions or

$$BC(p,q) = \int_x \sqrt{p(x)q(x)}dx. \tag{3}$$

for two continuous probability distributions $P(dx) \sim p(x)dx$ and $Q(dx) \sim q(x)dx$.

Thus if $p(x) = \mathcal{N}(\mu_p, \sigma_p)$ and $q(x) = \mathcal{N}(\mu_q, \sigma_q)$ then:

$$D_B(P,Q) = -ln\Big(\int_x \sqrt{p(x)q(x)}dx\Big) \tag{4}$$

$$D_B(P,Q) = -ln\Big(\int_x \frac{1}{\sqrt{2\pi}\sigma_p}\frac{1}{\sqrt{2\pi}\sigma_q} \exp\Big(-\frac{1}{2}(\frac{x-\mu_p}{\sigma_p})^2\Big)\exp\Big(-\frac{1}{2}(\frac{x-\mu_q}{\sigma_q})^2\Big)\Big)dx \tag{5}$$

and thus it can be shown that:

$$D_B(P,Q) = \frac{1}{4}\frac{(\mu_p-\mu_q)^2}{\sigma_p^2+\sigma_q^2} + \frac{1}{2}\ln\Big(\frac{\sigma_p^2+\sigma_q^2}{2\sigma_p\sigma_q}\Big). \tag{6}$$

This distance has several desirable properties for training neural networks. Firstly, it is zero when the two distributions are identical. Secondly, it is clearly differentiable.

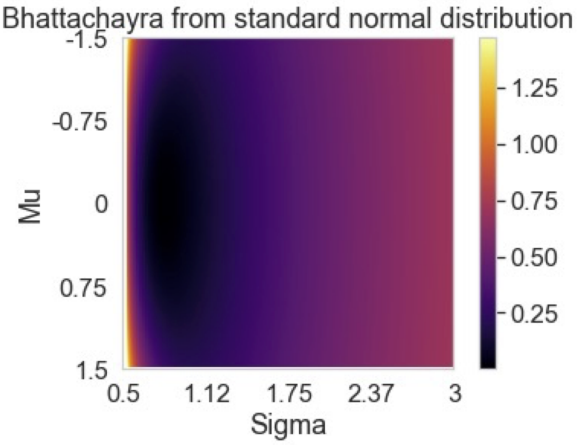

Figure 1. Bhattacharyya Distance for two GMMs: increasing loss as the second Gaussian increases in difference from the original reference Gaussian with $\mu = 0$ and $\sigma = 1$.

Figure 1 shows the Bhattacharyya Distance as the second distribution is increasingly removed from the standard normal distribution reference. The loss is clearly smoothly varying and also penalises the collapse of distribution - as σ tends to 0, the loss rapidly increases in magnitude. It can also be observed that in the case where the σ values are equal, the Bhattacharyya distance becomes the Mahalanobis distance between the two distributions. Thus, the Bhattacharyya distance is clearly the better choice, as it characterises differences between the distributions both in terms of the mean and also the standard deviation.

However, in this work we wish to consider the similarity between two Gaussian mixture models (GMM) with an arbitrary number of components. Note that for the Bhattacharyya distance, the two distributions being considered must be of the same type, and thus the reference and target

GMM distributions must contain the same number of components. If we consider two GMMs defined as:

$$p(x) \sim \sum_{k=1}^{M} \pi_{pk} \mathcal{N}(x; \mu_{pk}, \sigma_{pk}^2) \tag{7}$$

and similarly for $q(x)$, where k is the number of components in the GMM and π_p are the mixing weights, then if we calculated D_B for the simplest case when $k = 2$:

$$D_B = -ln\Big(\int_x \sqrt{PQ}\Big)\Big)dx \tag{8}$$

where

$$P = \pi_p \frac{1}{\sqrt{2\pi}\sigma_{p1}} \exp\Big(-\frac{1}{2}(\frac{x-\mu_{p1}}{\sigma_{p1}})^2\Big) + (1-\pi_p)\frac{1}{\sqrt{2\pi}\sigma_{p2}} \exp\Big(-\frac{1}{2}(\frac{x-\mu_{p2}}{\sigma_{p2}})^2\Big) \tag{9}$$

and

$$Q = \pi_q \frac{1}{\sqrt{2\pi}\sigma_{q1}} \exp\Big(-\frac{1}{2}(\frac{x-\mu_{q1}}{\sigma_{q1}})^2\Big) + (1-\pi_q)\frac{1}{\sqrt{2\pi}\sigma_{q2}} \exp\Big(-\frac{1}{2}(\frac{x-\mu_{q2}}{\sigma_{q2}})^2\Big). \tag{10}$$

This integral is thus the square root of a sum of exponential functions and, thus, has no closed form solution.

Thus, we utilise the approximation:

$$D_{GMM}(P,Q) = \sum_{k=1}^{M} \pi_{pk}\pi_{qk}\Big(\frac{1}{4}\frac{(\mu_{pk}-\mu_{qk})^2}{\sigma_{pk}^2+\sigma_{qk}^2} + \frac{1}{2}\ln\Big(\frac{\sigma_{pk}^2+\sigma_{qk}^2}{2\sigma_{pk}\sigma_{qk}}\Big)\Big) \tag{11}$$

and clearly when P=Q $D_{GMM} = 0$.

2. Classification

2.1. Dataset

The OrganAMNIST [10] dataset was used for the classification experiments, which was collated as part of the MedMNIST [11] dataset. The task is the classification of images as one of 11 organs with 58,850 samples available in total. As we wish to explore the effect of domain shift, we split the data into five 'sites', each representing a different domain shift:

- Site 1: no shift (source site)
- Site 2: reduced intensity range, max intensity clipped to 0.4 of the original
- Site 3: increased intensity range, max intensity increased to 1.1 of the original
- Site 4: applied Gaussian blurring, $0.25 < \sigma < 0.75$
- Site 5: applied salt and pepper noise, maximum intensity 0.1

Below are figures summarising the dataset. Figure 2 shows the distribution of training labels within the dataset. Figure 3 shows example images for each new site. Figure 4 shows the confusion matrix from training on each site in turn and testing on all sites: there is clearly a performance drop across sites, clearly demonstrating the domain shift. Finally, Figure 5 shows the result from training on all sites, in a centralised manner, showing that it is possible to train a single model which is able to span the variability across the sites. The code used to generate this data is supplied.

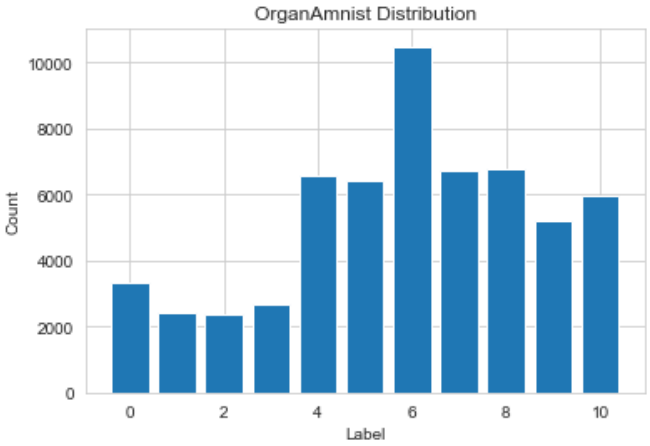

Figure 2. Distribution of labels in the OrganAMNIST dataset.

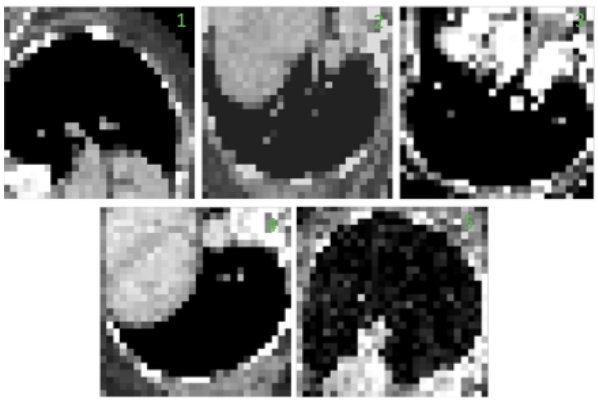

Figure 3. Example training image for each site for the same class label, all images are displayed in the same intensity range, using the minimum and maximum values for the original data.

2.2. Model Architecture

The model architecture was a simple CNN, with 4 convolutional layers and 2 maxpooling layers. The code for the

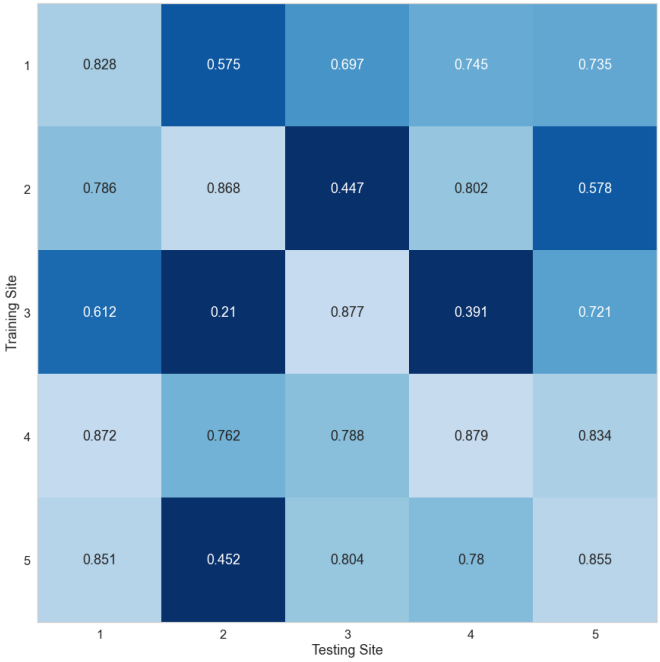

Figure 4. Confusion matrix of test accuracies from training on each individual site in turn with normal training and comparing to the other sites.

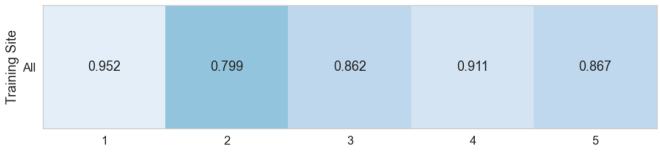

Figure 5. Accuracy results from training on all sites in a centralised manner.

architecture can be seen in figure 6. The key architecture choice is that the embeddings are returned before the ReLU function, so that they are better represented by a GMM and do not have a sharp cut off.

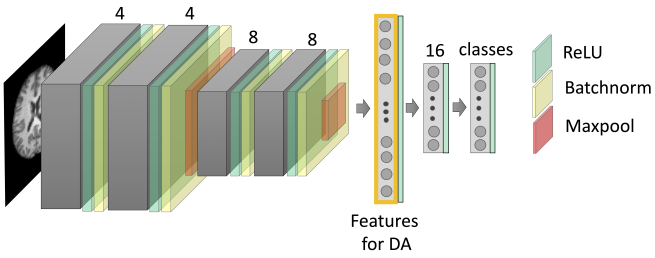

Figure 6. Classifier used for OrganAMNIST data. Convolutions are 2D.

2.3. Comparison Methods

We here detail the implementation details for the comparison methods used in this work for the classification task. As far as possible we used the recommended settings from the relevant papers, and kept the other implementation aspects as close to our proposed approach as possible. All models were trained with 5 fold cross validation, and the best fold used for test evaluation.

Source Model: Learning rate = 1×10^{-4} with AdamW optimiser. Trained with early stopping with a patience of 25 epochs, batchsize of 50 was used.

Centralised Data: Learning rate = 1×10^{-4} with AdamW optimiser. Trained with early stopping with a patience of 25 epochs.

Target finetune: Learning rate = 1×10^{-5} with AdamW optimiser. Trained with early stopping with a patience of 10 epochs. The label predictor was frozen such that it was shared across imaging sites.

DeepCORAL [9]: The loss function was implemented using code from the following repository: `https://github.com/SSARCandy/DeepCORAL/blob/master/models.py`. The model was trained with a learning rate 1×10^{-5} with AdamW optimiser. Trained with early stopping with a patience of 10 epochs.

FADA [7]: Implemented using the supplied code: `https://drive.google.com/file/d/1OekTpqB6qLfjlE2XUjQPm3F110KDMFc0/view`. The model was trained at the local site until convergence with a patience of 10 epochs. The model was trained with a learning rate 1×10^{-4} with AdamW optimiser.

FedHarmony [4]: Implemented using the supplied code: `https://github.com/nkdinsdale/FedHarmony`. The model was trained at the local site until convergence with a patience of 10 epochs. The model was trained with a learning rate 1×10^{-4} with AdamW optimiser. The recommended hyperparameters were used apart from for the proximal term which was tuned to the task: $\mu = 0.0001, \alpha = 1, \beta = 100$.

Minimise Entropy: Vanilla entropy minimisation. The model was trained with a learning rate 1×10^{-5} with AdamW optimiser. The model was trained at the local site until convergence with a patience of 10 epochs.

SHOT [6]: Three versions were implemented: 1) no smoothing, using the original source model without the training modifications, 2) with smoothing cross entropy used for the source model with a batchsize of 5 and 3) with smoothing cross entropy used for the source model with a batchsize of 500. Implemented using the supplied code: `https://github.com/tim-learn/SHOT`. The model was trained with a learning rate 1×10^{-5} with AdamW optimiser. The model was trained at the local site until convergence with a patience of 10 epochs. The entropy loss function weight was set to 1 and the class loss function weight was set to 0.3.

gSFDA [12]: Implemented using the supplied code: `https://github.com/Albert0147/G-SFDA`.

The model was trained using the learning rates specified in the original work, which vary across model components. The model was trained at the local site until convergence with a patience of 10 epochs.

USFAN [8]: Implemented using the supplied code: `https://github.com/roysubhankar/uncertainty-sfda`. The model was trained at the local site until convergence with a patience of 10 epochs. The model was trained using their learning rate scheduler.

2.4. Learning Rate

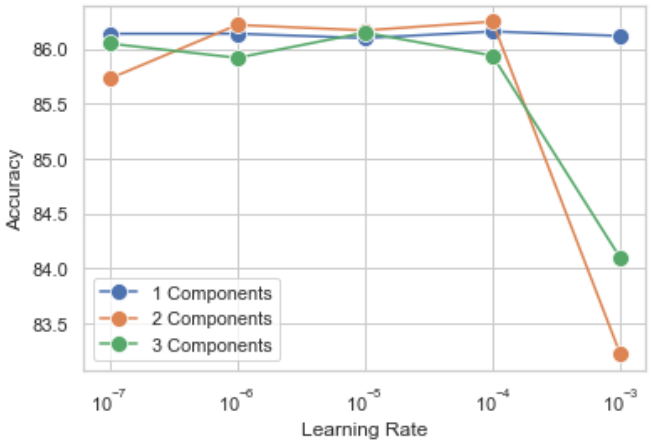

Figure 7. Effect of varying learning rate with varying numbers of components in the GMM.

Figure 7 shows the effect of changing the learning rate for differing numbers of components in the GMM fit. It can be seen that the performance was consistent across a wide range of learning rate values.

2.5. Results by Site

Results broken down by site are presented in the attached spreadsheet.

2.6. Features

The features learned by the model for the classification task can be seen in Fig. 8. It can clearly be seen that many of the features suggest at least two components, and that the EM fit is a good representation of the feature distributions.

2.7. Clock Times

Clock times (averaged over 5 runs) for the compared SFDA methods can be seen in Table 1.

Method	Entropy	SHOT	gSFDAN	USFAN	SFH1C	SFH2C	SFH3C
Clock Time	40.9	50.4	81.2	38.2	**5.4**	**7.0**	**12.0**

Table 1. Clock times (s) averaged over 5 runs. SFHXC = SFHarmony with X components.

3. Segmentation - CC359

3.1. Dataset

The CC359 dataset contains 359 T1 MRI scans of adult brains and corresponding brain masks. Details of this dataset can be found at: `https://sites.google.com/view/calgary-campinas-dataset/home`. The data were collected across six scanners, one each of a Phillips (P), GE and Siemens (S) scanner of 1.5 and 3T respectively. We trained the source model on each site in turn and compared the generalisability of the model to the other imaging sites, the result of which can be seen in Fig. 9. It is clear that the P15 model generalised least well, and thus this was chosen as the source site. All other sites were used as target sites. A PCA of the images showed P15 clustering separately to the other sites, and so this result was unsuprising. Figure 10 shows an example MRI image and corresponding brain mask.

3.2. Model Architecture

A 2D UNet was trained on slices of the MRI volumes, with 4 max pooling layers and 2 convolutional blocks per depth. An initial feature number of 4 was used. The code for the architecture can be seen in figure 11. Again, the key architecture choice is that the embeddings are returned before the ReLU function, so that they are better represented by a GMM and do not have a sharp cut off.

3.3. Comparison Methods

We here detail the implementation details for the comparison methods used in this work for the CC359 segmentation task. As far as possible we used the recommended settings from the relevant papers, and kept the other implementation aspects as close to our proposed approach as possible. All models were trained with 5 fold cross validation, and the best fold used for test evaluation.

Source Model: Learning rate = 1×10^{-4} with AdamW optimiser. Trained with early stopping with a patience of 25 epochs, batchsize of 50 was used.

Centralised Data: Learning rate = 1×10^{-4} with AdamW optimiser. Trained with early stopping with a patience of 25 epochs.

Target finetune: Learning rate = 1×10^{-5} with AdamW optimiser. Trained with early stopping with a patience of 10 epochs. The label predictor was frozen such that it was shared across imaging sites.

DeepCORAL [9]: The loss function was implemented using code from the following repository: `https://github.com/SSARCandy/DeepCORAL/blob/master/models.py`. The model was trained with a learning rate 1×10^{-5} with AdamW optimiser. Trained with early stopping with a patience of 10 epochs.

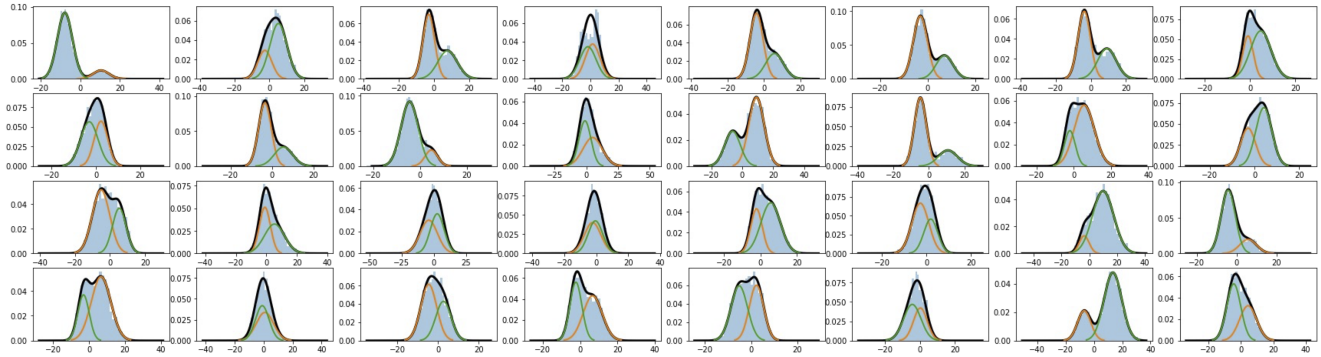

Figure 8. The feature distributions for the source test data, learned for the classification task, showing the GMM fit for 2 components. Single components are represented by the orange and green curves, and black shows the overall GMM fit.

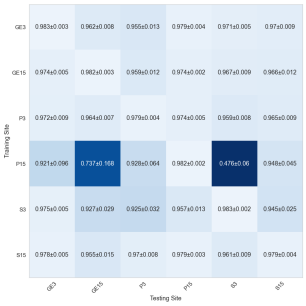

Figure 9. Confusion matrix showing Dice scores from CC359 brain extraction task, training on each site separately and then testing on all sites. P = Phillips, S = Siemens, 15= 1.5T, 3 = 3T.

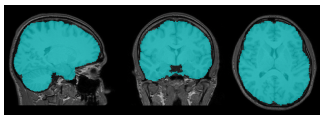

Figure 10. Example T1 scan and brain mask from CC359 dataset.

Only a batchsize of 5 could be used due to memory constraints.

FADA [7]: Implemented using the supplied code: `https://drive.google.com/file/d/1OekTpqB6qLfjlE2XUjQPm3F110KDMFc0/view`. The model was trained at the local site until convergence with a patience of 10 epochs. The model was trained with a learning rate 1×10^{-4} with AdamW optimiser.

FedHarmony [4]: Implemented using the supplied code: `https://github.com/nkdinsdale/FedHarmony`. The model was trained at the local site until convergence with a patience of 10 epochs. The model was trained with a learning rate 1×10^{-4} with AdamW optimiser. The recommended hyperparameters were used apart from for the proximal term which was tuned to the task: $\mu = 0.001, \alpha = 1, \beta = 100$.

Minimise Entropy: Vanilla entropy minimisation. The model was trained with a learning rate 1×10^{-5} with AdamW optimiser. The model was trained at the local site until convergence with a patience of 10 epochs.

AdaEnt [1]: Implemented using the supplied code: `https://github.com/mathilde-b/SFDA`. The hyperparameter weighting loss function contributions was set to 100 following the paper. The model was trained with a learning rate 1×10^{-5} with AdamW optimiser. The model was trained at the local site until convergence with a patience of 10 epochs.

AdaMI [2]: Implemented using the supplied code: `https://github.com/mathilde-b/SFDA`. The hyperparameter weighting loss function contributions was set to 100 following the paper. The tissue prior was estimated by averaging the source site labels and then the tissue ratio was used to create a slice depth dependent tissue prior. The model was trained with a learning rate 1×10^{-5} with AdamW optimiser. The model was trained at the local site until convergence with a patience of 10 epochs.

3.4. Results

Here we present the same results from the main paper with standard deviations included, in Table 2. These were originally omitted from the main script for clarity.

4. Segmentation - ABIDE

4.1. Dataset

We use data from the ABIDE dataset [3] for tissue segmentation (white matter, grey matter, CSF). The MR images from each site were processed using `FSL anat` (`https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/fsl_anat`) and the labels used were those generated by this process. Figure 12 shows an example brain and tissue segmentation. Four sites (Trinity, NYU, UCLA, Yale) were chosen for our experiments, so as to span both age distributions and subject number. NYU was used as the source site as it had the largest number of samples available and spanned the age range represented. Trinity: 49 subjects, NYU: 182 subjects, UCLA: 99 subjects, Yale:

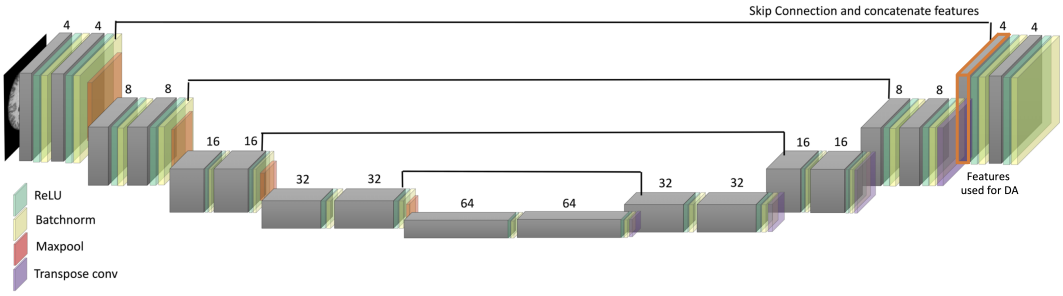

Figure 11. UNet model architecture.

Method	S	T	C	Information Communicated	Average Dice		
					Batchsize 5	Batchsize 50	Batchsize 500
Source Model	✓	x	x	-		0.832 ± 0.177	
Centralised Data	✓	✓	✓	All Data	0.983 ± 0.001	0.985 ± 0.001	0.983 ± 0.002
Target Finetune	x	✓	✓	Model Weights	0.981 ± 0.001	0.982 ± 0.001	0.982 ± 0.001
DeepCORAL [9]	✓	x	✓	All Data	0.768 ± 0.183	-	-
FADA [7]	✓	x	x	Model Weights + Features	0.967 ± 0.009	0.964 ± 0.010	0.959 ± 0.011
FedHarmony [5]	✓	x	x	Model Weights + Statistics	0.965 ± 0.009	0.962 ± 0.011	0.950 ± 0.012
Minimise Entropy	x	x	x	Model Weights	0.767 ± 0.176	0.849 ± 0.159	0.951 ± 0.007
AdaENT [1]	x	x	x	Model Weights	0.827 ± 0.166	0.817 ± 0.165	0.962 ± 0.007
AdaMI [2]	x	x	x	Model Weights	0.820 ± 0.162	0.835 ± 0.176	0.965 ± 0.007
Direct Fit [2]	x	x	x	Model Weights	0.648 ± 0.203	0.696 ± 0.171	0.873 ± 0.144
`SFHarmony` 1 GMM Component	x	x	x	Model Weights + Statistics	0.950 ± 0.008	0.949 ± 0.010	0.959 ± 0.009
`SFHarmony` 2 GMM Components	x	x	x	Model Weights + Statistics	0.970 ± 0.010	**0.970 ± 0.008**	**0.970 ± 0.009**
`SFHarmony` 3 GMM Components	x	x	x	Model Weights + Statistics	**0.972 ± 0.006**	0.968 ± 0.012	**0.970 ± 0.008**

Table 2. Results on the CC359 segmentation task. S = Source data required, T = Target labels required, C = Centralised data. The average accuracy is the performance across all 5 sites, weighted equally, and is reported for training batchsizes of 5, 50 and 500. Best SFDA method for each batchsize is in bold.

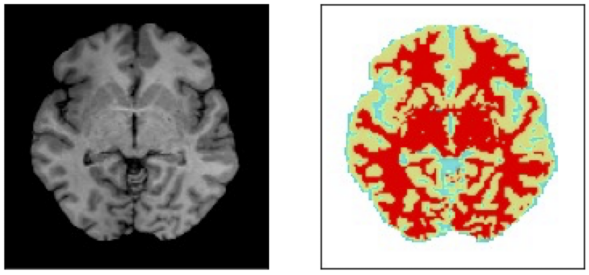

Figure 12. Example T1 scan and tissue segmentation mask from the ABIDE dataset. Blue = Cerebrospinal fluid, Yellow = White matter, Red = Grey matter.

56 subjects.

4.2. Network Architecture

The same network architecture was used as for the CC359 data, with the initial number of features set to 4.

4.3. Comparison Methods

The same comparison methods were used with the same settings. For AdaMI we again created a depth dependent tissue prior.

4.4. Results

Here we present the same results from the main paper with standard deviations included, in Table 3. These were originally omitted from the main script for clarity.

5. Regression - ABIDE

5.1. Dataset

We use data from the ABIDE dataset [3] for the age prediction task, following the setup presented in [5]. Four sites (Trinity, NYU, UCLA, Yale) were chosen for our experiments, so as to span both age distributions and subject number. Trinity: 49 subjects, 16.7±3.6 years (range: 12-25), NYU: 182 subjects, 14.7±6.6 (6-39), UCLA: 99 subjects, 12.5±2.2 (8-17), Yale: 56 subjects, 12.2±2.8 (7-17). The age distribution can be seen in Fig. 13. The images were resized to (128, 240, 160) and normalised to have zero mean and unit standard deviation.

5.2. Network architecture

The network architecture was a 3D VGG style networks with a single output node, following [5]. Again, the key architecture choice is that the embeddings are returned be-

Method	S	T	C	Information Communicated	Average Dice		
					Batchsize 5	Batchsize 50	Batchsize 500
Source Model	✓	x	x	-		0.775 ± 0.074	
Centralised Data	✓	✓	✓	All Data	0.884 ± 0.010	0.885 ± 0.012	0.875 ± 0.006
Target Finetune	x	✓	✓	Model Weights	0.883 ± 0.016	0.884 ± 0.016	0.885 ± 0.016
DeepCORAL [9]	✓	x	✓	All Data	0.523 ± 0.210	-	-
FADA [7]	✓	x	x	Model Weights + Features	0.830 ± 0.042	0.827 ± 0.044	0.825 ± 0.048
FedHarmony [5]	✓	x	x	Model Weights + Statistics	0.825 ± 0.043	0.810 ± 0.049	0.822 ± 0.044
Minimise Entropy	x	x	x	Model Weights	0.570 ± 0.200	0.542 ± 0.211	0.659 ± 0.176
AdaENT [1]	x	x	x	Model Weights	0.625 ± 0.168	0.656 ± 0.157	0.682 ± 0.141
AdaMI [2]	x	x	x	Model Weights	0.606 ± 0.178	0.657 ± 0.150	0.660 ± 0.154
Direct Fit [2]	x	x	x	Model Weights	0.615 ± 0.207	0.803 ± 0.066	0.830 ± 0.047
`SFHarmony` 1 GMM Component	x	x	x	Model Weights + Statistics	0.831 ± 0.045	$\mathbf{0.832 \pm 0.045}$	0.831 ± 0.047
`SFHarmony` 2 GMM Components	x	x	x	Model Weights + Statistics	0.832 ± 0.046	$\mathbf{0.832 \pm 0.046}$	$\mathbf{0.832 \pm 0.046}$
`SFHarmony` 3 GMM Components	x	x	x	Model Weights + Statistics	$\mathbf{0.833 \pm 0.044}$	$\mathbf{0.832 \pm 0.046}$	$\mathbf{0.832 \pm 0.046}$

Table 3. Results on the ABIDE segmentation task. S = Source data required, T = Target labels required, C = Centralised data. The average accuracy is the performance across all 5 sites, weighted equally, and is reported for training batchsizes of 5, 50 and 500. Best SFDA method for each batchsize is in bold.

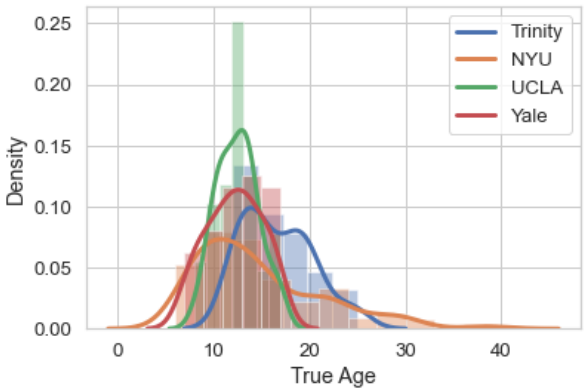

Figure 13. Normalised age distributions for the 4 sites from the ABIDE dataset.

fore the ReLU function, so that they are better represented by a GMM and do not have a sharp cut off. The network architecture can be seen in Fig. 14.

5.3. Comparison Methods

We could not identify any suitable SFDA comparison methods for regression. All methods were trained with 3 fold cross validation.

Source Model: Learning rate = 1×10^{-4} with AdamW optimiser. Trained with early stopping with a patience of 25 epochs, batchsize of 16 was used.

Centralised Data: Learning rate = 1×10^{-4} with AdamW optimiser. Trained with early stopping with a patience of 25 epochs.

Target finetune: Learning rate = 1×10^{-5} with AdamW optimiser. Trained with early stopping with a patience of 10 epochs. The label predictor was frozen such that it was shared across imaging sites.

DeepCORAL [9]: The loss function was implemented using code from the following repository: `https://github.com/SSARCandy/DeepCORAL/blob/master/models.py`. The model was trained with a learning rate 1×10^{-5} with AdamW optimiser. Trained with early stopping with a patience of 10 epochs.

FADA [7]: Implemented using the supplied code: `https://drive.google.com/file/d/1OekTpqB6qLfjlE2XUjQPm3F110KDMFc0/view`. The model was trained at the local site until convergence with a patience of 10 epochs. The model was trained with a learning rate 1×10^{-4} with AdamW optimiser.

FedHarmony [4]: Implemented using the supplied code: `https://github.com/nkdinsdale/FedHarmony`. The model was trained at the local site until convergence with a patience of 10 epochs. The model was trained with a learning rate 1×10^{-4} with AdamW optimiser. The recommended hyperparameters were used apart from for the proximal term which was tuned to the task: $\mu = 0.01, \alpha = 1, \beta = 100$.

5.4. Results

Here we present the same results from the main paper with standard deviations included, in Table 4. These were omitted from the main script for clarity.

References

[1] Mathilde Bateson, Hoel Kervadec, Jose Dolz, Hervé Lombaert, and Ismail Ben Ayed. Source-relaxed domain adaptation for image segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, pages 490–499, 2020.

[2] Mathilde Bateson, Hoel Kervadec, Jose Dolz, Hervé Lombaert, and Ismail Ben Ayed. Source-free domain adaptation for image segmentation. *Medical Image Analysis*, 82:102617, 2022.

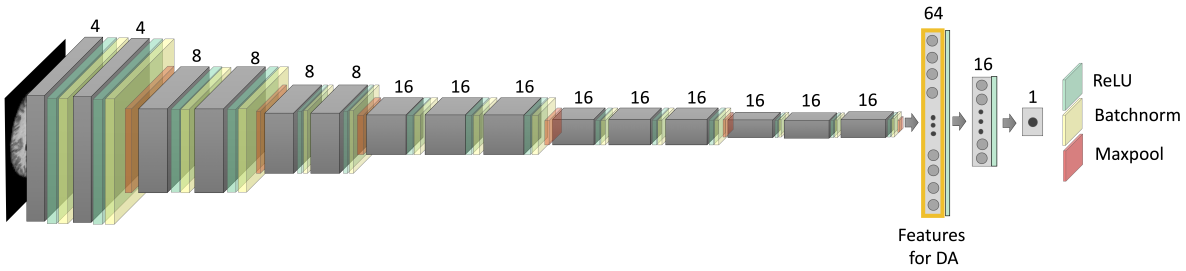

Figure 14. Regressor used for the age prediction task. Convolutions are 3D.

Method	S	T	C	Information Communicated	Average MAE		
					Bs 4	Bs 8	Bs 16
Source Model	✓	x	x	-	4.38 ± 1.55		
Centralised Training	✓	✓	✓	All Data	3.52 ± 1.75	3.38 ± 1.50	3.36 ± 1.48
Target Finetune	x	✓	x	Model Weights	3.57 ± 1.77	3.60 ± 1.75	3.58 ± 1.72
DeepCORAL [9]	✓	x	✓	All Data	4.58 ± 2.42	4.41 ± 2.10	4.12 ± 2.01
FADA [7]	✓	x	x	Model Weights + Features	3.55 ± 1.77	3.42 ± 1.62	3.78 ± 1.90
FedHarmony [5]	✓	x	x	Model Weights + Statistics	3.61 ± 1.88	3.50 ± 1.71	3.79 ± 1.89
Direct Fit	x	x	x	Model Weights + Statistics	4.70 ± 1.03	4.31 ± 1.00	4.05 ± 1.03
`SFHarmony` 1 GMM Component	x	x	x	Model Weights + Statistics	4.21 ± 1.01	4.13 ± 1.26	3.71 ± 1.47
`SFHarmony` 2 GMM Components	x	x	x	Model Weights + Statistics	3.87 ± 1.42	**3.72±1.40**	**3.69±1.42**
`SFHarmony` 3 GMM Components	x	x	x	Model Weights + Statistics	**3.64±1.46**	**3.72±1.39**	3.73 ± 1.70

Table 4. Results on the ABIDE dataset for the age prediction task. S = Source data required, T = Target labels required, C = Centralised data, Bs = Batchsize. The average MAE is the performance across all 4 sites, weighted equally, and is reported for training batchsizes of 4, 8 and 16: 16 was the largest batch achievable. The best SFDA method for each batchsize is in bold.

[3] Adriana di Martino, Chaogan Yan, Qingyang Li, Erin B. Denio, Francisco Xavier Castellanos, Kaat Alaerts, Jeffrey S. Anderson, Michal Assaf, Susan Y. Bookheimer, Mirella Dapretto, Ben Deen, Sonja Delmonte, Ilan Dinstein, Birgit B. Ertl-Wagner, Damien A. Fair, Louise Gallagher, Daniel P. Kennedy, Christopher Lee Keown, Christian Keysers, Janet E. Lainhart, Catherine Lord, Beatriz Luna, V. Menon, Nancy J. Minshew, Christopher S. Monk, Sophia Mueller, Ralph-Axel Müller, Mary Beth Nebel, Joel T. Nigg, Kirsten O'Hearn, Kevin A. Pelphrey, Scott J. Peltier, Jeffrey D. Rudie, Stefan Sunaert, Marc Thioux, Julian Michael Tyszka, Lucina Q. Uddin, Judith S. Verhoeven, Nicole Wenderoth, Jillian Lee Wiggins, Stewart H. Mostofsky, and Michael Peter Milham. The autism brain imaging data exchange: Towards large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19:659 – 667, 2013.

[4] Nicola K. Dinsdale, Mark Jenkinson, and Ana I.L. Namburete. Deep learning-based unlearning of dataset bias for mri harmonisation and confound removal. *NeuroImage*, 228:117689, 2021.

[5] Nicola K. Dinsdale, Mark Jenkinson, and Ana I. L. Namburete. Fedharmony: Unlearning scanner bias with distributed data. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pages 695–704, 2022.

[6] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6028–6039. PMLR, 13–18 Jul 2020.

[7] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. In *2020 International Conference on Learning Representations*, 2020.

[8] Subhankar Roy, Martin Trapp, Andrea Pilzer, Juho Kannala, Nicu Sebe, Elisa Ricci, and Arno Solin. Uncertainty-guided source-free domain adaptation. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, page 537–555, 2022.

[9] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision – ECCV 2016 Workshops*, pages 443–450, 2016.

[10] Xuanang Xu, Fugen Zhou, Bo Liu, Dongshan Fu, and Xiangzhi Bai. Efficient multiple organ localization in ct image using 3d region proposal network. *IEEE Transactions on Medical Imaging*, 38:1885–1898, 01 2019.

[11] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2 - a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10, 01 2023.

[12] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui. Generalized source-free domain adaptation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8958–8967, oct 2021.