Supplementary Material: Domain Attention Consistency for Multi-Source Domain Adaptation

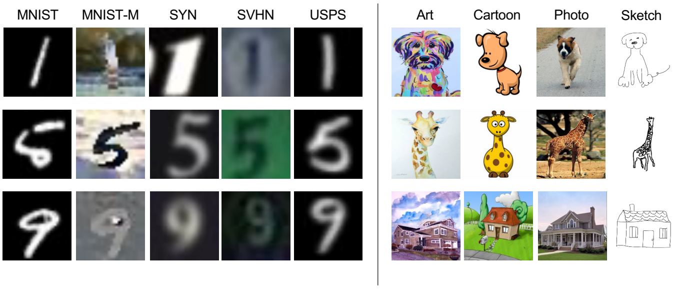
Zhongying Deng[1, 2]
z.deng@surrey.ac.uk
Kaiyang Zhou[3]
k.zhou.vision@gmail.com
Yongxin Yang[1]
yongxin.yang@surrey.ac.uk
Tao Xiang[1, 2]
t.xiang@surrey.ac.uk

[1] University of Surrey
Guildford, UK
[2] iFlyTek-Surrey Joint Research Center on Artificial Intelligence
[3] Nanyang Technological University
Singapore

The supplementary material is organized as follows: Section A gives the experimental setting of our experiments; Section B evaluates the class compactness loss; Section C provides more experiments and analyses for the comparison between domain attention consistency (DAC) and feature distribution alignment (FDA); Section D evaluates our DAC-Net on single-source to single-target domain adaptation task; Section E investigates how the number of source domains influences the performance of DAC-Net; Section F further explores the influence of the amount of target domain data on our DAC-Net; Section G shows visualizations on feature distribution.

A Experimental Setting

Datasets and protocols. (1) **DomainNet** is by far the largest and most challenging MSDA dataset. It has around 0.6 million images and 345 categories. There are six distinct domains (Clipart, Infograph, Painting, Quickdraw, Real and Sketch) with dramatic domain shift in image style, color, background, strokes, etc. See Figure 1 in the main paper for example images. It can be observed that some domains, e.g., Quickdraw and Sketch where shape information is particularly important, are closer than others like Real and Quickdraw where color/texture information is essential for the former but not required at all for the latter. Therefore, to succeed in DomainNet one needs to identify features that are transferable between the target and source domains. (2) **Digit-Five** consists of five digit datasets including MNIST [7], MNIST-M [3], Synthetic Digits [3], SVHN [11], and USPS. Following [12], all 9,298 images in USPS are used as source domain for model training only; on each of the other four datasets, 25,000 images are used for training and 9,000 images for testing. The domain shift mainly takes place in font color/style and image background. See Figure A(a) for example images. (3) **PACS** [8] includes 9,991 images of seven categories across four domains, i.e. Cartoon, Photo, Sketch and Art Painting. The domain shift mainly corresponds



(a) Digit-Five (b) PACS

Figure A: Example images from Digit-Five and PACS.

to image style changes. Example images are provided in Figure A(b). We follow the official train-val splits provided in [8].

For evaluation, one domain is chosen in turn as the target domain while the rest are regarded as source domains. A model is trained using the labeled source data and the unlabeled target data training split, and tested on the test split of the target domain.

Training details. On Digit-Five and DomainNet, we use SGD with momentum to train the model. The learning rate is updated with the cosine annealing strategy [9]; on PACS, we use Adam [6] as the optimizer.

On DomainNet, ResNet-101 [5] is used as the CNN backbone, following [12]. We sample 6 images from each domain at each iteration. The model is trained for 40 epochs, with an initial learning rate of 0.002.

On Digit-Five, we follow [12] to construct the CNN model with three convolution layers followed by two fully connected layers. The batch size B is set to 64 for each domain. We train the model for 30 epochs, with an initial learning rate of 0.05.

On PACS, we follow the setting in [14] and use ResNet-18 [5] as the CNN backbone. The model is trained for 100 epochs with an initial learning rate of 5e-4 and the batch size of 16.

B Ablation Study on class compactness Loss

In this section, we first compare our class compactness loss with other related work, and then discuss whether this loss should be applied to source domain data.

JDDA [1] and HoMM [2] are the most related works to our class compactness loss. Both of them use parameterized class centers for discriminative feature learning. Instead, our class compactness loss takes advantage of classification weight vectors, which shows superior performance to class-center-based methods (see Table 2 in the main paper).

In practice, we can apply L_c to both source and target domains and denote it as L'_c, i.e.

$$L'_c = \frac{1}{B}\sum_{i=1}^{B} \mathbb{1}(q(\hat{y}_i^{\mathcal{T}}) \geq \tau)||f_i^{\mathcal{T}} - W_{\hat{y}_i^{\mathcal{T}}}||_2^2 + \frac{1}{KB}\sum_{k=1}^{K}\sum_{i=1}^{B}||f_i^{\mathcal{S}_k} - W_{y_i^{\mathcal{S}_k}}||_2^2. \tag{A}$$

We then compare the L_c and L'_c in Table A. We can see that applying the compactness loss to source domains decreases the accuracy by 2.03%. The degradation implies that the compactness for source domain data could even harm the learning of the discriminative features for the target data.

Methods	**Avg**
L_c (only target data)	91.42
L'_c (both source and target data)	89.39
ℓ_2 penalty on the weight W_{y_i}	90.91

Table A: Ablation study on whether applying the compactness loss to source domain data on PACS.

Methods	Apply domain alignment loss after	**Avg**
DAC	L_d after last two residual blocks	**90.79**
FDA	L_d after final features	89.27
	L_d after last two residual blocks	89.75
	L_d after last three residual blocks	88.75
	L_d after all four residual blocks	88.78
	MMD loss after final features	89.65
	MMD loss after last two residual blocks	89.23
	MMD loss after last three residual blocks	89.80
	MMD loss after all four residual blocks	89.74

Table B: Ablation study on where to apply the domain alignment loss in the ResNet architecture on PACS. FDA: Feature Distribution Alignment.

Furthermore, we compare our class compactness loss with another loss which is a ℓ_2 penalty on the weight matrix W_{y_i}. We can see from Table A that such a loss works worse than our class compactness loss. Our class compactness works better because it pulls the features of target data to the corresponding classification weight vectors, thus possibly away from the decision boundary.

C Attention Alignment vs. Feature Alignment

In Table B, we compare our DAC loss with different variants of the FDA loss. For a fair comparison, the experimental setup for all methods is kept the same. For the FDA-based methods, we impose the domain alignment loss on the average-pooled features to directly align feature distributions. Two different domain alignment losses are explored here. The first is L_d (defined in Eq. (2)), which is applied to features instead of attribute attention weights. The second is maximum mean discrepancy (MMD) [4]. From the results, we can see that DAC surpasses all FDA variants by about 1%. This suggests that aligning attribute attention weights is more effective than aligning feature distributions for MSDA.

D Single-Source to Single-Target Domain Adaptation

We further evaluate our DAC on popular single-source domain adaptation datasets, namely Digits datasets. Following the setting in [15], we conduct experiments on three transfer tasks: SVHN → MNIST, USPS → MNIST, MNIST → USPS. As in [13, 15], a three-layer CNN is used for the first task while a two-layer CNN for the latter two tasks. We then apply channel attention to these CNNs to facilitate DAC. Other training details are kept the same as [13, 15]: Adam [6] optimizer with learning rate of 2e-4, mini-batch size of 128. The results are shown in Table C. We can see that our DAC effectively improves the accuracy by

Methods	**SVHN→ MNIST**	**USPS→ MNIST**	**MNIST→ USPS**	**Avg**
w/ DAC	77.82	92.25	89.91	86.66
w/o DAC	68.79	94.44	80.55	81.26

Table C: Accuracy on Digits dataset for single-source domain adaption.

Methods	**ArtPainting**	**Cartoon**	**Sketch**	**Photo**	**Avg**
DAC-Net (Single best)	86.67	78.00	68.30	97.54	82.63
DAC-Net (Source combined)	**91.62**	89.53	82.07	**98.48**	90.43
DAC-Net (MSDA)	91.39	**91.39**	**84.97**	97.93	**91.42**
Oracle	99.53	99.84	99.53	99.92	99.71

Table D: Accuracy on PACS. Best results for each target domain (except oracle) are in bold.

5.4% over the model without DAC.

In Table D, we also compare the MSDA task with other single-source domain adaptation tasks on PACS, including single best and source combine. Single best is the best results of adapting each of three source domains to the target, and source combine means combining all source domains as a single source domain. It is obvious that the our DAC-Net works best under the setting of MSDA. This is because the transferable attributes learning of our DAC-Net can effectively alleviate the negative transfer, and take full advantage of multiple source domains to improve the adaptation performance.

E Influence of the Number of Source Domains

In this section, we gradually increase the number of source domains to see how it influences the accuracy of our DAC-Net. We show the results in Table E. The findings can be summarized as follows: (1) Generally, more source domains increase the performance on the target, possibly because more source domains contribute to the learning of transferable semantic attributes. (2) When the target domain is Photo, the Sketch domain brings a minor negative impact on the performance (-0.09%, i.e. 98.02% vs. 97.93%). This may be caused by that the attributes learned from Sketch domain are too abstract to transfer to the concrete attributes of Photo. Overall, our DAC-Net can benefit from more source domains.

F Influence of the Amount of Target Domain Data

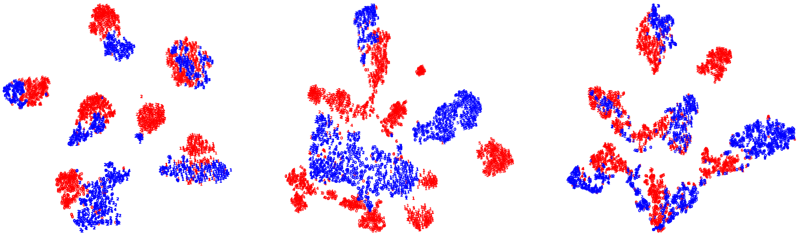
To investigate how many target data are required for our DAC-Net (under MSDA setting), we gradually reduce the amount of target domain data. From Table F, we can see that the less

Tasks	**Avg**	**Tasks**	**Avg**	**Tasks**	**Avg**	**Tasks**	**Avg**
C→A	87.55	**A→C**	78.00	**C→S**	68.30	**C→P**	94.19
C+S→A	88.67	**A+S→C**	90.19	**C+A→S**	78.88	**C+A→P**	98.02
C+S+P→A	91.39	**A+S+P→C**	91.39	**C+A+P→S**	84.97	**C+A+S→P**	97.93

Table E: Ablation study on how the number of source domains influence the accuracy of DAC-Net on PACS. 'C': Cartoon, 'A': ArtPainting, 'S': Sketch, 'P': Photo.

Amount of Target Data	ArtPainting	Cartoon	Sketch	Photo	Avg
100%	91.39	**91.39**	**84.97**	97.93	**91.42**
70%	**92.11**	90.46	80.36	97.69	90.16
50%	91.26	90.57	76.78	**98.22**	89.21
30%	91.04	88.72	80.05	97.54	89.34
10%	86.34	85.57	77.25	97.43	86.65

Table F: Accuracy of DAC-Net on PACS.



(a) DAC-Net (b) M^{3}SDA (c) MCD

Figure B: Visualization of feature distributions of different methods on PACS using t-SNE [10]. Each number stands for the class label of a feature (7 classes in total; better viewed with zoom-in). Red/blue denotes the source/target domain.

target data leads to worse performance. But overall, our DAC-Net can beat the source-only baseline even when only 10% amount of target data are used, e.g. 86.65% vs. 81.94%.

G Visualization

In this section, we provide visualizations of feature distribution to help understand why our DAC-Net works.

Figure B depicts the feature distributions of our DAC-Net, M^{3}SDA [12] and MCD [13]. MCD performs class-level feature distribution alignment across domains while M^{3}SDA only does alignment at the domain level. Figure B shows that both of them cannot achieve either class-level or domain-level feature alignment. As a result, the target domain data of 7 classes are not well separable. In contrast, our DAC-Net does not enforce any feature alignment, but the source and target features are much better aligned. More importantly, the 7 classes of all domains are easily separable using the same set of decision boundaries—after all, all domains share the classifier. This suggests that: (1) explicitly enforcing feature alignment is counter-productive since it even *harms* the discriminative feature learning for classification task; (2) learning latent attributes by enforcing attention weight consistency is more effective for MSDA because it properly reduces the domain shift, which further *contributes to* discriminative features learning on the target domain.

References

[1] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *AAAI*, 2019.

[2] Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. Homm: Higher-order moment matching for unsupervised domain adaptation. In *AAAI*, 2020.

[3] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

[4] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *JMLR*, 2012.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Y Lecun and L Bottou. Gradient-based learning applied to document recognition. *IEEE*, 1998.

[8] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *CVPR*, 2017.

[9] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008.

[11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS-W*, 2011.

[12] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

[13] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

[14] Hang Wang, Minghao Xu, Bingbing Ni, and Wenjun Zhang. Learning to combine: Knowledge aggregation for multi-source domain adaptation. In *ECCV*, 2020.

[15] Ni Xiao and Lei Zhang. Dynamic weighted learning for unsupervised domain adaptation. In *CVPR*, 2021.