Supplementary Material

Marin Scalbert[1, 2]
marin.scalbert@centralesupelec.fr
Maria Vakalopoulou[1]
maria.vakalopoulou@centralesupelec.fr
Florent Couzinié-Devy[2]
f.couzinie-devy@vitadx.com

[1] MICS
CentraleSupélec
Gif-sur-Yvette, France
[2] VitaDX International
Paris, France

1 Datasets details

DomainNet. DomainNet is the largest dataset for benchmarking domain adaptation methods. It has been introduced by [6] and contains more than 0.6 millions images for six distincts domains (*clipart, infograph, painting, quickdraw, real, sketch*) spread over 345 classes.

MiniDomainNet. MiniDomainNet [11] is a subset of DomainNet that uses less images ($\sim 140K$ images) with smaller size of 96×96 spread over 4 selected domains and 126 selected classes. MiniDomainNet was introduced to reduce the requirements for computing resources and to remove noisy domains/examples.

Office-Home. Office-home [8] has been widely used as a standard dataset for evaluating domain adaptation methods. It includes approximately 15500 images from 4 different domains: *Art*, *Clipart*, *Product* and *Real-World*. For each domain, it contains images of 65 object categories found typically in Office and Home settings.

2 Implementation details

Strong data augmentation. Strong data augmentation $t_s(.)$ is based on the same image transformations as RandAugment [1]. An image is augmented with $t_s(.)$ following these steps:

1. Two transformations are sampled randomly in the list of possible transformations (Table 1) and the parameters defining these transformations are drawn randomly inside their corresponding range (Table 1).

2. The image is horizontally and randomly flipped with a probability $p = 0.5$.

3. Random cropping is applied to the image with a crop size between $[90\%, 100\%]$ of the original image size. The crop is resized to $(96, 96)$ for MiniDomainNet, $(180, 180)$ for DomainNet and $(224, 224)$ for Office-Home.

Weak data augmentation. Weak data augmentation $t_w(.)$ is made only of random horizontal flips. The image is finally resized to $(96, 96)$ for MiniDomainNet, $(180, 180)$ for

Transformation	Hyperparameter	Range	Description
AutoContrast	C	$[0,1]$	Maximize (normalize) image contrast. It calculates a histogram of the input image, removes C percent of the lightest and darkest pixels from the histogram, and remaps the image so that the darkest pixel becomes black (0), and the lightest becomes white (255).
Brightness	B	$[0.1,1.9]$	Control image brightness by a factor B. $B=0$ gives a black image and $B=1$ gives the original image.
Color	C	$[0.1,1.9]$	Adjust the colour balance of an image given a enhancement factor C. $C=0$ gives a black and white image, $C=1$ gives the original image.
Contrast	C	$[0.1,1.9]$	Control the contrast of an image given an enhancement factor C. $C=0$ gives a solid grey image and $C=1$ gives the original image.
Equalize		$[0,1]$	Equalize the image histogram.
Identity		$[0,1]$	Returns the original image.
Invert		$[0,1]$	Invert the image.
Posterize	B	$[4,8]$	Reduce the number of bits to B bits for each color channel.
Rotate	θ	$[-30,30]$	Rotate the image counter clocksize with an angle θ.
Sharpness	S	$[0.1,1.9]$	Adjust the image sharpness given an enhancement factor S. $S=0$ gives a blurred image, $S=1$ the original image and $S=2$ an sharpened image.
ShearX	R	$[-0.3,0.3]$	Shear the image along the horizontal axis with rate R
ShearY	R	$[-0.3,0.3]$	Shear the image along the vertical axis with rate R
Solarize	S	$[0,256]$	Invert all pixel values above a threshold S.
TranslateX	t	$[-0.3,0.3]$	Translate an image with size (H,W) along the horizontal axis by $t\times W$ pixels.
TranslateY	t	$[-0.3,0.3]$	Translate an image with size (H,W) along the vertical axis by $t\times H$ pixels.

Table 1: Transformations used for strong data augmentation $t_s(.)$. Most of the transformations are described by some hyperparameters. When one transformation depending on parameters is drawn, its hyperparameters are randomly sampled within the specified Range.

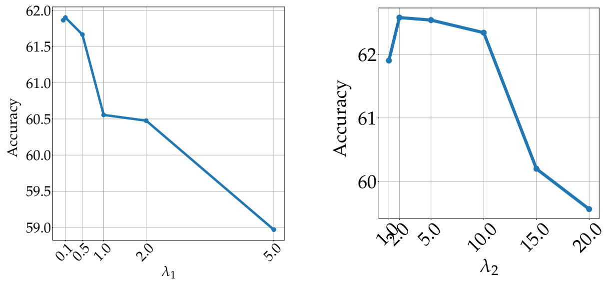
DomainNet and $(224,224)$ for Office-Home.

Architecture and hyperparameters. For DomainNet and Office-Home, the features extractor F corresponds to a ResNet50 [3] while for MiniDomainNet a ResNet18 is used. All networks are pretrained on the ImageNet dataset [2]. The embeddings $\boldsymbol{h}$ of the ResNet50 and the ResNet18 are respectively the 2048 dimensional features vector ($d_1=2048$) and the 512 dimensional features vector ($d_1=512$) obtained after the global average pooling layer. In all experiments, dimension of the projection head representation $\boldsymbol{z}$ is set to $d_2=256$, temperature T is set to 0.1 and probability threshold τ to 0.95. The α hyperparameter for MixUp is set to 0.4 for DomainNet, MiniDomainNet and 0.2 for Office-Home. λ_1 and λ_2 are respectively set to 0.1 and 1. All our experiments are conducted using the PyTorch library [5] with 2 Nvidia Tesla V100 GPUs. For optimization, Adam [4] method is used with an initial learning rate of 10^{-4} and a cosine decay learning rate with a minimum learning rate of 0. On DomainNet, MiniDomainNet and Office-Home, the models are trained respectively for $70K$, $60K$ and $20K$ iterations. For these datasets, source and target batch sizes (N_S,N_T) are respectively set to $(256,256)$, $(256,256)$ and $(128,128)$. For each raw source example, two strongly augmented examples are generated.

Target examples pseudo labeling.

Exponential moving average weights, as in the original FixMatch method [7], are used for predicting the pseudo labels on the weakly augmented target examples and for the final evaluation of the method. During all experiments, exponential decay parameter α_{EMA} is set to $\alpha_{EMA} = 0.999$.

3 Additional ablations



(*a*) (*b*)

Figure 1: Averaged accuracy on MiniDomainNet with respect to $\lambda_1(a)$ and $\lambda_2(b)$.

Performances with respect to λ_1 and λ_2. The hyperparameters λ_1 and λ_2 correspond to the weights of the $\mathcal{L}_{ISCL}$ and $\mathcal{L}_{unsup}$ loss respectively. Setting λ_1 and λ_2 is a compromise between the influence of $\mathcal{L}_{ISCL}$ to align source class conditional distributions enabling general and transferable features and $\mathcal{L}_{unsup}$ to adjust features for the target domain.

To find a reasonable starting combination of (λ_1, λ_2) for further experiments and assess their respective influence on the model performances, CMSDA has been trained for each target domain of MiniDomainNet with two different settings. In the first setting, CMSDA is trained with different λ_1 while setting $\lambda_2 = 1$. In the second setting, CMSDA is trained with different λ_2 while fixing $\lambda_1 = 0.1$. Averaged accuracies over MiniDomainNet target domains with respect to λ_1 or λ_2 are respectively reported on Figure 1a and Figure 1b. In Figure 1a, we can notice that performances remain stable when $\lambda_1 \in [0.05, 0.5]$ with the best accuracy achieved at $\lambda_1 = 0.1$. For $\lambda_1 > 0.5$, the performance starts to decrease. In Figure 1b, it seems that the performance remains stable for a large range of λ_2 ($\lambda_2 \in [2, 10]$). The best average accuracy is achieved for $\lambda_2 = 2$ while for $\lambda_2 < 2$ and $\lambda_2 > 10$ performances begin to deteriorate. For experiments on other datasets, we suggest starting CMSDA with $(\lambda_1, \lambda_2) = (0.1, 2)$.

Source examples mixing strategies. In this section, a comparison between different source examples mixing strategies is performed. More specifically, we compare MixUp [10] with CutMix [9] when used for combining source examples. For the two mixing strategies, accuracies for each target domain of MiniDomainNet are reported in Table 2. These re-

Target domain	*clipart*	*painting*	*real*	*sketch*	*average*
MixUp	71.38	53.76	66.23	56.24	61.90
CutMix	69.68	53.18	67.14	51.43	60.36

Table 2: Performances on MiniDomainNet when using different mixing strategies.

sults reveal that CutMix has comparable performances with MixUp except on *sketch* where MixUp outperforms CutMix. We believe that by fine-tuning CutMix hyperparameters, it would be possible to close the small performances gap between MixUp and CutMix. However, overall, there is no clear advantages to use one mixing method over another.

References

[1] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[6] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019.

[7] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[8] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

[9] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[10] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018.

[11] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *arXiv preprint arXiv:2003.07325*, 2020.