Semi-supervised Domain Adaptation via Sample-to-Sample Self-Distillation
-Supplementary Materials-

Jeongbeen Yoon Dahyun Kang Minsu Cho
Pohang University of Science and Technology (POSTECH), South Korea
{jeongbeen,dahyun.kang,mscho}@postech.ac.kr

In this supplementary material, we provide our method's additional details, analyses, and experimental results.

A. Implementation details

The effectiveness of S^{3}D are shown in different experimental settings in our main paper. In this section, we provide our experimental details.

Some results of Tables 1, 2, and 3 of our main paper are borrowed from the paper of MME [7]: the results of S+T, DANN, ADR, CDAN, ENT, and MME in Table 1, their results with the AlexNet base network in Table 2, and their accuracies of unsupervised domain adaptation in Table 3.

Datasets. In Figure s.2, we visualize the examples of DomainNet and Office-Home datasets. In both datasets, all of four domains are distinct from each other, while Real and Product domains in Office-Home are quite similar.

Baselines. For a fair comparison, we reproduce S+T, MME [7], and APE [4] if the accuracy is not stated in their papers. To reproduce MME, we follow the official implementation [1] and set λ (from MME) to 0.1. For APE, we follow the official implementation [2] and set α, β, and γ to 10, 1, and 10, respectively.

Many-shot semi-supervised domain adaptation experiments. We use ResNet [3] for the many-shot experiments on DomainNet dataset in Figure 4. The accuracy of the S+T and MME baselines for one-shot and three-shot settings are borrowed from the MME paper.

Unsupervised domain adaptation experiments. In Table 3, we borrow the accuracy of AlexNet [5] from MME. We reproduce the accuracy of ResNet for unsupervised domain adaptation experiments under controlled settings.

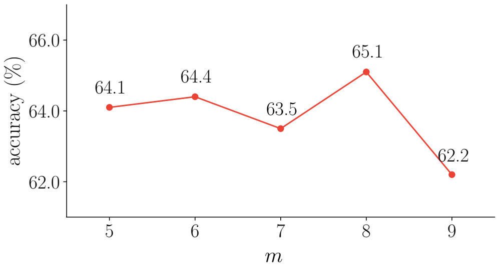


Figure s.1: Accuracy (%) in Real to Sketch one-shot scenario on the DomainNet using ResNet with various m.

Ablation studies. We conduct our experiments on DomainNet using ResNet34 for ablation experiments shown in Tables 4 and s.5. Table s.5 indicates the full ablation study of proposed components. In Figure 6, we use AlexNet for our base network and DomainNet for our dataset.

Inter-domain and intra-domain discrepancy histograms. We use ResNet for Office-Home [8] on one-shot setting to plot inter-domain and intra-domain discrepancy histograms shown in Figure 5. We choose the Clipart domain for the source domain and the Product domain for the target domain. For Figures 5a and 5c, we plot histograms of cosine similarity after the pre-training stage, specifically from 10,000th iteration. We plot the histogram every 3,000 iterations until the model converges. For APE, we plot the converged model for a comparison.

The balancing hyper-parameter λ. We use λ to balance $\mathcal{L}_{\text{pair}}$ in the overall loss and make up for the incompleteness of pseudo-labels. We set the hyper-parameter λ using a ramp-up function like in [2]:

$$\lambda = \frac{2}{1+e^{-mt}} - 1, \tag{s.1}$$

where $t \in [0,1]$ increases over iterations. The increasing t makes λ increases so that $\mathcal{L}_{\text{pair}}$ influences more on the learn-

[1] https://github.com/VisionLearningGroup/SSDA_MME
[2] https://github.com/TKKim93/APE

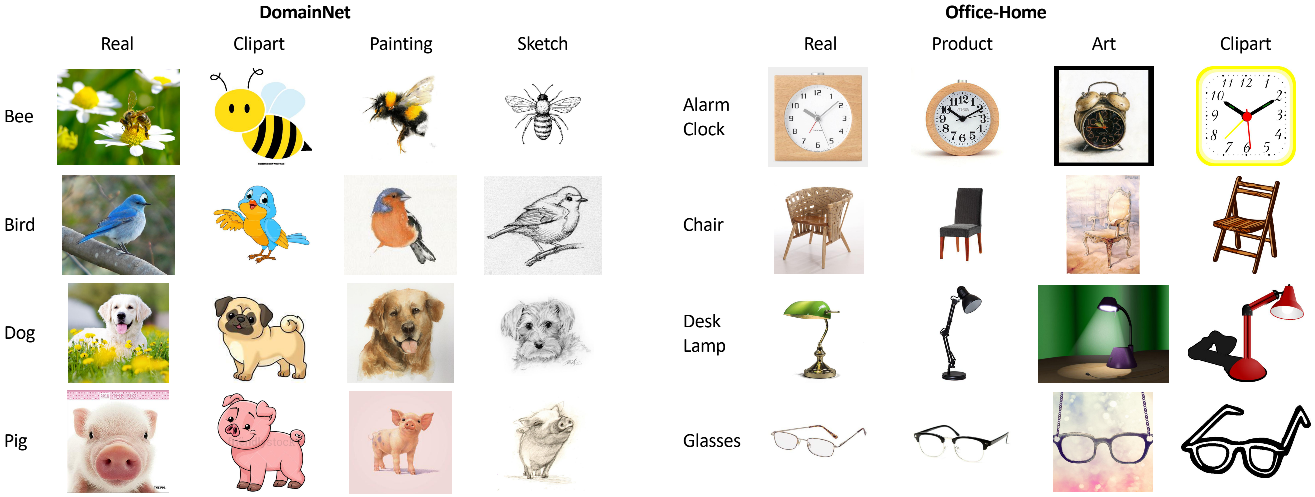
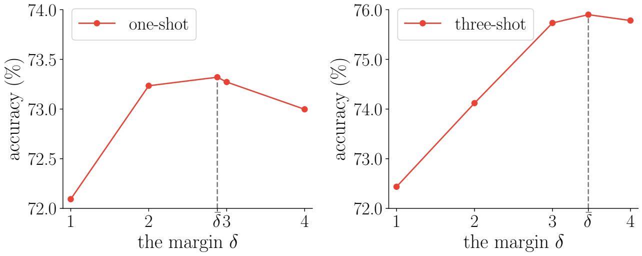


Figure s.2: DomainNet and Office-Home datasets. We visualize four domains of four classes on each dataset.



(a) One-shot setting. $\bar{\delta}$=2.88. (b) Three-shot setting. $\bar{\delta}$=3.46.

Figure s.3: Accuracy (%) on Real to Clipart (DomainNet) scenario using ResNet34 with various δ.

ing process. This incremental weighting technique is adequate since pseudo-labels are likely to be incorrect at the beginning of the sample-to-sample training stage.

To find a proper ramp-up function, we vary m to examine the effects of weighting $\mathcal{L}_{\text{pair}}$. Changing m controls the slope of the ramp-up function. We choose Real to Sketch one-shot domain scenario, and select m at the best validation accuracy. In DomainNet and Office-Home experiment, we set m to 8 on both AlexNet and ResNet. Figure s.1 shows the accuracy of our model when varying m.

The class logit margin δ. In Eq. (4), the margin δ is used to filter out unreliable target samples. Here, δ determines a trade-off between a number of pseudo-labels used and how reliable the pseudo-labels are. A small δ makes the pseudo-labels of the student set inaccurate. On the contrary, a large δ makes RSS filter many unlabeled target samples so that few student samples are used for training. Therefore, proper δ is critical for a student-set to contain precise and various student samples.

Method	δ				
	1	2	3	$\bar{\delta}$	4
S^{3}D ($\alpha = 0.7$)	72.6	72.9	73.1	73.5	73.4
S^{3}D ($\alpha = 0.8$)	72.5	74.4	75.0	**75.9**	75.7
S^{3}D ($\alpha = 0.95$)	72.4	74.1	75.7	**75.9**	75.8
S^{3}D w/o α	72.2	72.9	75.2	75.6	75.5

Table s.1: Accuracy (%) on Real to Clipart (DomainNet) three-shot scenario using ResNet34 by varying δ and α.

We investigate whether the average margin $\bar{\delta}$ of all unlabeled target samples is appropriate for δ or not. Figure s.3 shows the experiment, which is conducted on DomainNet Real to Clipart scenario using ResNet34. We compare the result of the average margin $\bar{\delta}$ to those of the margin δ from 1 to 4. $\bar{\delta}$ is initially calculated from the pre-trained model and is fixed afterward. In Figure s.3, the model calculates $\bar{\delta}$ as 2.88 and 3.46 in one-shot and three-shot setting, respectively. The model shows the best accuracy when the student-set is generated using $\bar{\delta}$. This result describes that $\bar{\delta}$ is an appropriate margin for RSS to make student-set abundant and precise.

Performance varying both δ and α. Unlike [9], we preset the margin δ by averaging all unlabeled target's margin so that we can obtain target adaptive δ. This is because S^{3}D deals with various target domains different from [9], which considers only Cityscapes [1] as a target dataset. Also, the threshold α is designed to avoid the situation that CAG [9] might exclude the sample with high confidence because of its low margin. To search the best values of δ and α, we jointly vary the values of them. The details are the same as the setting in Figure s.3b. The results are shown in Ta-

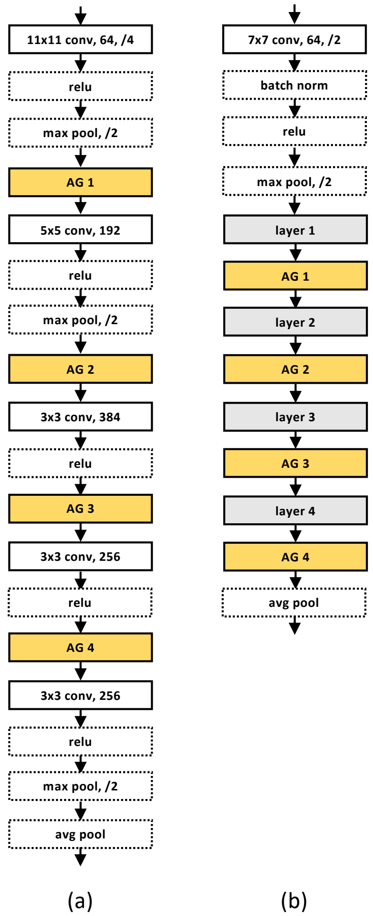


Figure s.4: The locations to apply AG operation. The yellow blocks represent the place to apply AG. (a) The feature extractor of AlexNet. (b) The feature extractor of ResNet34.

ble s.1, where S^{3}D ($\alpha = 0.95$, $\delta = \bar{\delta}$) performs the best. It also shows that S^{3}D is not sensitive to the margin parameter δ and the threshold α.

The locations to apply AG. To generate assistant features, AG operation is applied to different layers. In AlexNet, the operation can be placed at any yellow blocks shown in Figure s.4 (a). In ResNet34, the operation can be located at the end of every four residual blocks shown in Figure s.4 (b). We conduct experiments on various combinations of the location in Table s.2. We conjecture that the optimal combination of the operations is different between ResNet and AlexNet. As the overall accuracy of ResNet is higher than that of AlexNet, the pairs from ResNet are more reliable and thus the semantic meanings from teachers are more effective for students in ResNet than AlexNet. We select all locations for ResNet in DomainNet and Office-Home. For AlexNet, we select AG 1 and AG 1,2 in DomainNet and Office-Home, respectively.

Network	AG			
	1	1,2	1,2,3	1,2,3,4
AlexNet	**39.9**	38.5	31.8	32.6
ResNet34	60.5	61.2	62.5	**65.1**

Table s.2: Accuracy (%) in two networks on the DomainNet Real to Sketch one-shot scenario with various location combinations of AG operation.

Net	Method	DomainNet	Office-Home
AlexNet	S+T	40.0	44.1
	MixStyle	39.3	43.5
	S^{3}D ($\epsilon = 1$)	46.7	46.8
	S^{3}D (ours)	48.7	49.5
ResNet34	S+T	56.9	62.3
	MixStyle	66.6	60.2
	S^{3}D ($\epsilon = 1$)	69.1	69.8
	S^{3}D (ours)	69.9	70.3

Table s.3: Average classification accuracy (%) on the DomainNet and Office-Home datasets for one-shot on all domain scenarios that we cover.

Method	1-shot	3-shot
CDAN	62.9±1.5	65.3±0.1
ENT	59.5±1.5	63.6±1.3
MME	64.3±0.8	66.8±0.4
APE	65.2±0.9	67.3±0.9
S^{3}D	67.7±0.6	69.7±0.7

Table s.4: Classification accuracy (%) and standard deviation (%) on the Sketch to Painting scenario in the DomainNet averaged over three runs.

B. Additional experimental results

Comparison with MixStyle [10]. The main difference between S^{3}D and MixStyle is that we introduce the assistant (intermediate style feature) as a guidance for the student. The assistant is designed to transfer its knowledge to the student using knowledge distillation; for this reason, we do not back-propagate gradients through the path of assistant features (see the second dotted branch in Figure 3). This strategy has not been explored before. MixStyle, which is introduced for domain generalization, directly trains the model with stylized features; the features' predictions and given labels are used for calculating the cross-entropy loss, and the gradients are back-propagated through the features. This scheme is not adequate for SSDA for the reason that the goal of SSDA is to adapt the learner to the target domain. For comparison, we conduct experiments where we directly apply the scheme of [10] to SSDA; we only change the $\mathcal{L}_{\text{pair}}$ loss to the cross-entropy loss between assistants' predictions and pseudo-labels. We also searched the best hyper-parameters for this model as we did for S^{3}D. We

Method	$\mathcal{L}_{\text{unl}}$	$\mathcal{L}_{\text{pair}}$	RSS	R to C	R to P	P to C	C to S	S to P	R to S	P to R	MEAN
DANN				58.2	61.4	56.3	52.8	57.4	52.2	70.3	58.4
MME				70.0	67.7	69.0	56.3	64.8	61.0	76.1	66.4
APE				70.4	70.8	72.9	56.7	64.5	63.0	76.6	67.6
S^3D	✗	✗	✗	56.8	60.5	55.4	51.7	55.5	47.5	72.0	57.1
	✓	✗	✗	68.7	65.6	68.8	59.2	64.1	61.6	78.4	66.6
	✓	✗	✓	71.6	69.1	70.7	58.7	65.4	62.0	79.6	68.2
	✗	✓	✗	67.4	65.0	67.1	61.2	64.9	62.7	77.5	66.5
	✗	✓	✓	73.1	67.1	70.6	57.7	65.8	62.4	73.6	67.2
	✓	✓	✗	69.4	65.7	69.7	61.3	65.5	61.7	78.6	67.4
	✓	✓	✓	73.3	68.9	73.4	60.8	68.2	65.1	79.5	69.9

Table s.5: Comprehensive ablation study of S^3D on DomainNet dataset (%) for one-shot setting.

Method	0-shot	1-shot	3-shot	5-shot	10-shot	20-shot
S+T	54.5	55.6	60.0	64.6	67.6	71.5
MME	67.6	70.0	72.2	74.8	76.8	77.9
APE	65.4	67.0	72.2	72.8	76.9	77.0
S^3D	**72.7**	**73.3**	**75.9**	**77.5**	**78.0**	**79.1**

(a) Real to Clipart.

Method	0-shot	1-shot	3-shot	5-shot	10-shot	20-shot
S+T	55.9	56.8	59.4	64.4	68.4	71.1
MME	67.1	69.0	71.7	73.0	76.4	78.0
APE	63.8	67.7	71.3	72.6	76.6	78.1
S^3D	**66.5**	**73.4**	**75.1**	**76.9**	**77.2**	**79.6**

(b) Painting to Clipart.

Table s.6: Classification accuracy (%) on DomainNet with a varying number of target labels. (a) corresponds to Figure 4a, and (b) corresponds to Figure 4b.

set m to 9 for all experiments. For ResNet, we select AG 1,2,3 for DomainNet and AG 1,2,3,4 for Office-Home. For AlexNet, we select AG 1 for DomainNet and AG 1,2 for Office-Home. In Table s.3, it is obvious that MixStyle is not effective except for the experiment of ResNet in DomainNet. The model even shows low accuracy than the simple baseline S+T in several experiments.

The effect of intermediate styles. In Table s.3, we examine the effectiveness of transferring an intermediate style rather than a teacher's individual style. In Eq. (5), by controlling the value of ϵ, we can manipulate the style of the assistant feature. As the value of ϵ is close to 1, the style of the assistant approaches to the teacher's one. S^3D ($\epsilon = 1$) is the experiment that the assistant feature follows only the style of the teacher. When we compare S^3D ($\epsilon = 1$) and S^3D, the results show that the intermediate styles are more effective than the teacher's style to reduce the domain discrepancy.

Multiple runs. For a fair comparison, we report the average accuracy and its standard deviation of three independent runs in Table s.4. Our method less deviates than most of previous methods do, showing that our method is adequately stable and effective.

Many-shot experiments. In Table s.6, we attach exact values plotted in Figure 4 of the main paper.

Extra t-SNE visualization Figure s.5 visualizes how S^3D embeds instances from two domains over iterations. The embeddings are obtained using ResNet34 from examples of Office-Home dataset in the one-shot setting, and we visualize the first 30 classes for simplicity. We adopt t-SNE [6] with the perplexity of 30.0 and 1000 iterations. We observe that the sample-to-sample self-distillation stage clearly enhances the embedding quality from the pre-training stage. Two main points of the results are: (1) target samples gradually align with source samples over iterations. (2) samples from the same class pull each other over iterations.

Comprehensive ablation study. We evaluate different combinations of the proposed component in Table s.5 and compare them with previous work of [7, 2, 4]. The performance consistently increases as more components are used, indicating that each proposed component is effective for SSDA. Note that our method with all the components sets a new state of the art, outperforming APE [4].

References

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2

[2] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 1, 4

[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. 1

[4] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *European conference on computer vision*, 2020. 1, 4

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

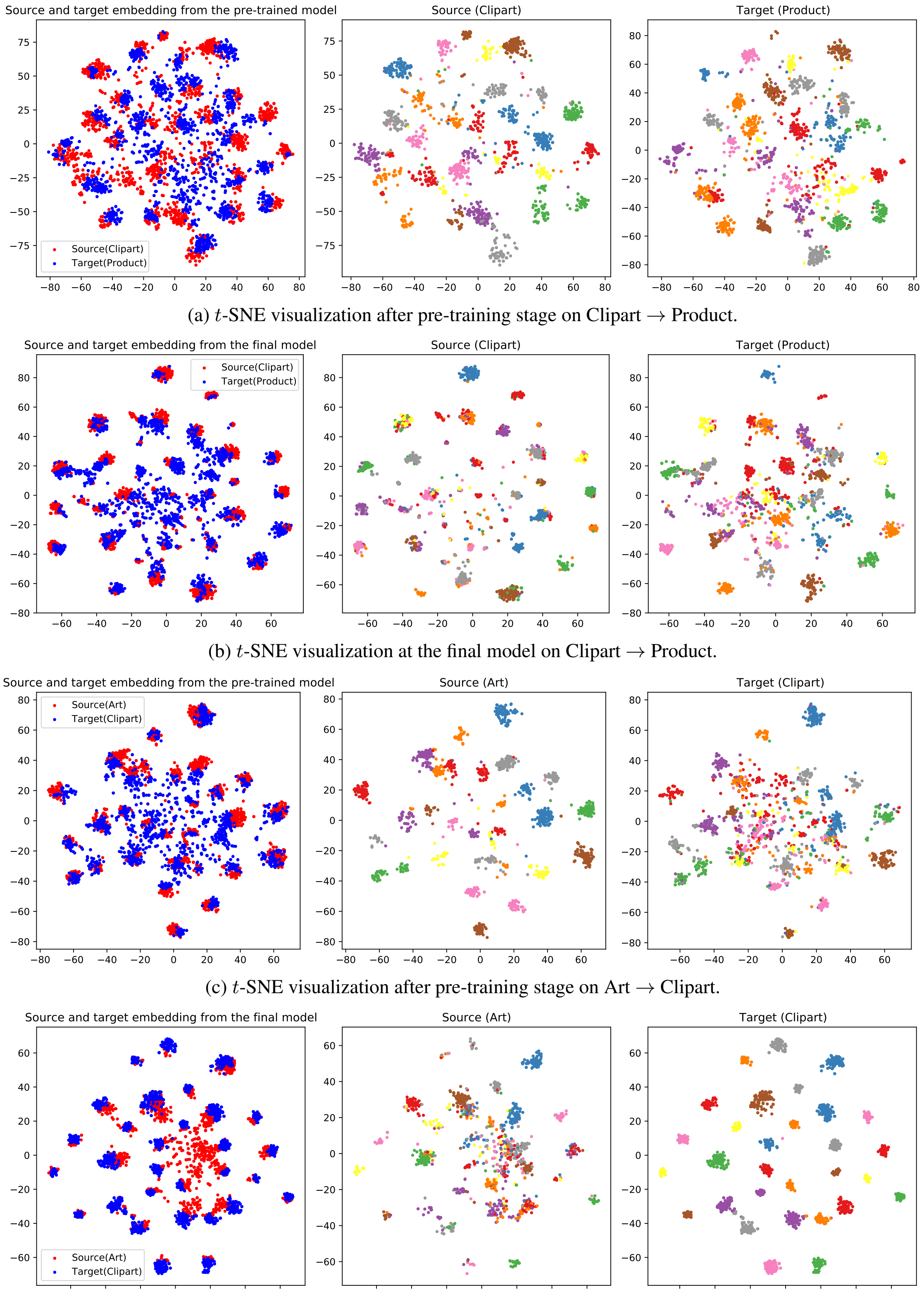
[6] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 4

[7] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8050–8058, 2019. 1, 4

[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. 1

[9] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. In *Advances in Neural Information Processing Systems*, pages 435–445, 2019. 2

[10] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2021. 3



(a) t-SNE visualization after pre-training stage on Clipart → Product.

(b) t-SNE visualization at the final model on Clipart → Product.

(c) t-SNE visualization after pre-training stage on Art → Clipart.

(d) t-SNE visualization at the final model on Art → Clipart.

Figure s.5: t-SNE visualization on the Office-Home. Left column: source and target embedding spaces. Middle column: source embedding spaces. Right column: target embedding spaces.