Combating Label Distribution Shift for Active Domain Adaptation

—Supplementary Material—

Sehyun Hwang[1] Sohyun Lee[2] Sungyeon Kim[1]
Jungseul Ok[1,2⋆] Suha Kwak[1,2⋆]

[1]Department of Computer Science and Engineering, POSTECH, Korea
[2]Graduate School of Artificial Intelligence, POSTECH, Korea

This material provides additional analysis and results that have been omitted due to the page limit. Section A presents in-depth analysis of LAMDA, including detailed component analysis (Sec. A.1), computational complexity (Sec. A.2), LAMDA with various UDA methods (Sec. A.3), t-SNE visualization (Sec. A.4), and the component analysis of our sampling strategy (Sec. A.5). Section B demonstrates more results, including comparison with SSDA methods (Sec. B.1), comparison with cosine classifier (Sec. B.2), and more results of DomainNet (Sec. B.3). Finally, the configuration of our DANN [2] (Sec. C.1) and implementation details of ADA methods (Sec. C.2) are provided in Section C.

A Further analysis of LAMDA

A.1 Component analysis for each adaptation scenarios

Table a1-a2 quantifies the impact of each component of LAMDA, in every domain adaptation scenario of the two datasets. Every component in LAMDA generally improves the performance of each domain adaptation scenario. Comparing the second and the third rows of Table a2, one can see the remarkable performance gain by our label distribution matching strategy since it alleviates the significant label distribution shift of OfficeHome-RSUT. Additionally, Our random sampling baseline incorporating DANN achieves 86.7% on VisDA-2017 and 51.3% on DomainNet.

A.2 Computational complexity of LAMDA

In Table a3, we compare the computational complexity and computation time per round of LAMDA and previous ADA methods. The computation time is measured on Intel(R) Xeon(R) Gold 6240 CPU. While the computational complexity of LAMDA exhibits quadratic growth regarding the target dataset's size, it is significantly faster than CLUE [9] and S^{3}VAADA [10], even on the large scale Clipart dataset of DomainNet consists of 30K images. This is because the calculation for our sampling consists of simple matrix multiplication, which can be computed efficiently in parallel. LAMDA achieves the best performance among the previous arts with a reasonable sampling time.

⋆ Co-corresponding authors

Table a1. Component analysis of LAMDA measured by accuracy (%) using 10%-budget for each source-target domain pair of four domains of OfficeHome: **A**rt, **C**lipart, **P**roduct, and **R**eal. We evaluate from ablation baseline at the last row to LAMDA at the first row by sequentially adding three components: (i) Prototype: prototype set sampling (o/w, sampling uniformly at random); (ii) Matching: label distribution matching (o/w, replacing p_i in Eq. (9)-(10) with uniform distribution); and (iii) Cosine: cosine classifier (o/w, linear classifier). The ablation baseline is equipped with DANN [2]. The accuracy is a mean of three runs, and the subscript denotes standard deviation.

Prototype	Matching	Cosine	OfficeHome												
			A → C	A → P	A → R	C → A	C → P	C → R	P → A	P → C	P → R	R → A	R → C	R → P	Avg
✓	✓	✓	75.4	88.5	85.9	73.3	88.7	83.8	75.2	75.3	87.1	80.9	77.8	91.8	$82.0_{\pm 0.1}$
✓	✓	✗	74.0	87.8	85.4	69.4	85.6	81.7	71.5	74.2	86.0	78.1	76.3	91.8	$80.2_{\pm 0.4}$
✓	✗	✗	69.6	82.7	82.8	64.6	84.2	78.4	66.3	72.1	84.9	74.8	74.1	91.3	$77.1_{\pm 0.0}$
✗	✗	✗	65.2	77.4	79.1	61.3	77.5	73.9	65.6	68.0	81.2	73.7	69.4	84.6	$73.1_{\pm 0.2}$

Table a2. Component analysis of LAMDA measured by accuracy (%) using 10%-budget for each source-target domain pair of three domains of OfficeHome-RSUT: **C**lipart, **P**roduct, and **R**eal. We evaluate from ablation baseline at the last row to LAMDA at the first row by sequentially adding three components: (i) Prototype: prototype set sampling (o/w, sampling uniformly at random); (ii) Matching: label distribution matching (o/w, replacing p_i in Eq. (9)-(10) with uniform distribution); and (iii) Cosine: cosine classifier (o/w, linear classifier). The ablation baseline is equipped with DANN [2]. The accuracy is a mean of three runs, and the subscript denotes standard deviation.

Prototype	Matching	Cosine	OfficeHome-RSUT						
			C → P	C → R	P → C	P → R	R → C	R → P	Avg
✓	✓	✓	82.0	74.9	60.8	81.5	65.4	85.9	$75.1_{\pm 0.8}$
✓	✓	✗	79.2	72.6	61.3	81.0	62.7	86.2	$73.8_{\pm 0.4}$
✓	✗	✗	70.5	64.6	52.8	75.7	53.8	79.6	$66.2_{\pm 1.0}$
✗	✗	✗	64.2	60.1	52.0	73.7	54.6	73.6	$63.1_{\pm 0.9}$

Table a3. Computational complexity and computation time (seconds and minutes) per round on two source-target domain pairs: Art to Clipart of OfficeHome and Clipart to Sketch of DomainNet, where n_{T} is the size of the target dataset, t is the number of clustering iterations, B is the budget, d is the dimension of embedding, and n_{P} is the size of the prototype set. The query time is measured at the first round when 2% of the target dataset is actively sampled. (†): Sampling of S^{3}VAADA[10] requires a large computation cost of network forward and backward pass proportional to n_{T}, since it adversarially perturbs each sample to measure uncertainty.

	AL strategy	Computational complexity	Computation time	
			OfficeHome (A → C)	DomainNet (C → S)
forward + sampling	TQS[1]	$O(n_{\mathrm{T}} \log n_{\mathrm{T}})$	5.41s	0.88m
	CLUE[9]	$O(tn_{\mathrm{T}}Bd)$	35.98s	43.94m
	S^{3}VAADA[10]	$O(n_{\mathrm{T}}^2(d+B^2))^{\dagger}$	132.20s	487.11m
	Ours	$O(n_{\mathrm{T}}^2(d+n_{\mathrm{P}}^2))$	7.11s	2.37m

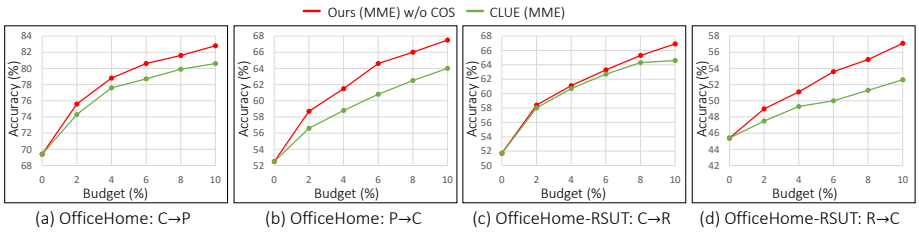

Fig. a1. Accuracy versus the percent of labeled target instances as budget, when each method is equipped with MME [11]. The accuracies are measured on the source-target domain pair of two domains of OfficeHome: **C**lipart and **P**roduct; and OfficeHome-RSUT: **C**lipart and **R**eal). w/o COS: Ours without cosine classifier.

A.3 Combine LAMDA with other DA methods

In Fig. a1, we compare the performance of LAMDA with with CLUE [9] varying budget, when equipped with MME [11], on source-target domain pair of OfficeHome: **C**lipart and **P**roduct; and OfficeHome-RSUT: **C**lipart and **R**eal. We follow the training protocol of CLUE, where at each round, the model is provided with the budget and is trained for 20 epochs with MME loss. While CLUE is specialized with MME, the result shows that LAMDA constantly outperforms CLUE in all four domain adaptation scenarios. The performance gap between LAMDA and CLUE increases as the budget increases. In Table a4, we compare the performance of ADA methods coupled with MCC [4], the state-of-the art UDA method, when using 10%-budget; the cosine classifier is not used in this comparison to verify the generalization of our sampling strategy. While MCC increases the overall performance of ADA methods, LAMDA still shows the best performance among them. This result demonstrates that LAMDA can significantly improve the performance regardless of adaptation technique by using the budget effectively for label distribution matching and supervised learning.

Table a4. Accuracy (%) of ADA methods coupled with MMC [4] on OfficeHome using 10%-budget for each source-target pair of four domains: **A**rt, **C**lipart, **P**roduct, and **R**eal. w/o COS: Ours without cosine classifier

DA method	AL method	OfficeHome												
		A → C	A → P	A → R	C → A	C → P	C → R	P → A	P → C	P → R	R → A	R → C	R → P	Avg
MCC [4]	TQS[1]	70.9	**91.4**	87.8	74.2	90.1	84.6	74.4	72.8	**87.3**	78.7	75.6	92.7	81.7
	CLUE[9]	**77.3**	91.1	87.7	74.3	90.6	85.4	74.7	77.2	87.2	81.3	78.4	93.1	83.2
	S^3VAADA[10]	71.3	89.1	87.7	74.2	87.5	85.5	74.9	71.8	86.3	81.3	76.2	92.4	81.5
	Ours w/o COS	76.9	90.2	**89.0**	**76.7**	**91.5**	**86.8**	**77.8**	**77.4**	88.5	**83.5**	**78.5**	**93.5**	**84.2**

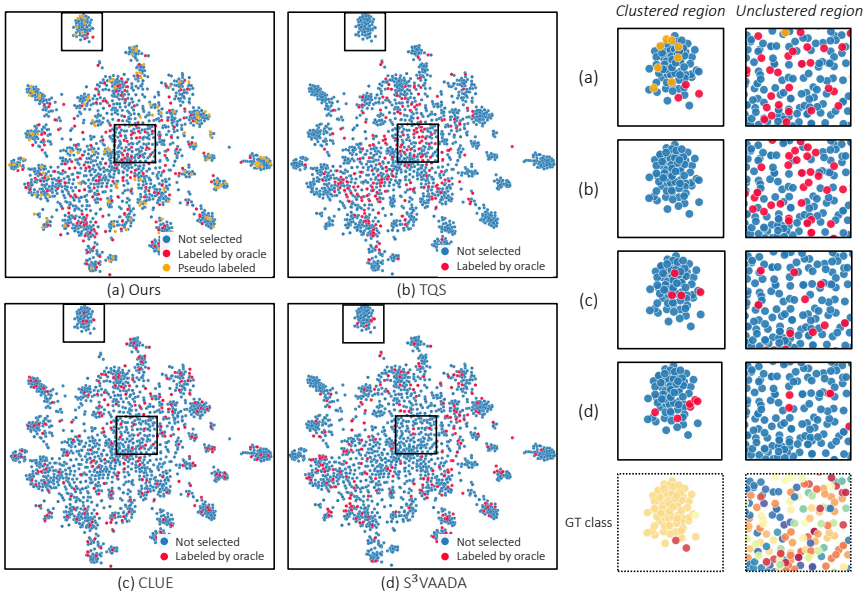

Fig. a2. *t*-SNE [8] visualization of target feature vectors from source pre-trained model on OfficeHome Real to Art scenario. The feature vector of the selected samples from each method are colored red. GT class: target feature vectors colored by ground-truth class labels.

A.4 t-SNE visualization

In Fig. a2, we visualize the target feature vectors selected by LAMDA and the previous ADA methods from the source pre-trained model. When the feature vectors are clustered, LAMDA selects a comparably large number of prototypes within the cluster to reflect its density (Fig. a2a), which provides a better estimation of target statistics as in Fig. 4. In the clustered region, most prototypes are pseudo labeled since there tend to include easy samples (GT class in Fig. a2). While for the unclustered region, the prototypes are labeled by an oracle (Fig. a2a), providing ground-truth supervision to the uncertain prototypes. This allows us to mainly invest a budget on uncertain samples while utilizing density-aware samples to estimate the target label distribution.

On the other hand, the selected samples of the previous methods less preserve the density of the target data as they avoid selecting easy samples within the clustered regions. Even though CLUE [9] and S^{3}VAADA [10] select part of the samples within the clustered region, they do not reflect the density of the cluster. Since CLUE and S^{3}VAADA spend part of their budgets in the clustered regions, they cannot select samples as many as LAMDA in the unclustered regions (Fig. a2c and Fig. a2d). Oppositely, while TQS [1] selects samples as many as LAMDA within the unclustered region, it ignores samples in the clustered region (Fig. a2b).

A.5 Detailed component analysis of prototype set sampling

Table a5 quantifies the contribution of each components of our sampling strategy: (i) Subset sampling using MMD measure; (ii) Preventing labeling of the easy prototypes; and (iii) Utilizing the pseudo-labeled prototypes for the label distribution estimation. Every component of our sampling strategy improves the performance for every adaptation scenario in OfficeHome. The performance gap between the last (random sampling with DANN [2]) and the second last rows verifies that our distribution-aware sampling boosts the effect of supervised learning. Comparing the second and the third rows, one can see the performance gain by considering uncertainty for the sampling. The performance gap between the first and the second rows shows utilizing the pseudo-labeled prototypes improves the quality of the label distribution estimation.

Table a5. Component analysis of prototype set sampling measured by accuracy (%) using 10%-budget for each source-target domain pair of four domains of OfficeHome: **A**rt, **C**lipart, **P**roduct, and **R**eal. We evaluate from ablation baseline at the last row to our sampling strategy at the first row by sequentially adding three components: (i) Density: Sampling with MMD (o/w, sampling uniformly at random); (ii) Uncertainty: only label uncertain prototypes using the top-2 margin (o/w, using $\Delta = 1$ in Algorithm 1; and (iii) Pseudo label: utilizing the pseudo-labeled prototypes for the label distribution estimation (o/w, ignoring the pseudo labeled prototypes in Eq. (7)). The ablation baseline is equipped with DANN [2].

Density	Uncertainty	Pseudo label	OfficeHome												
			A → C	A → P	A → R	C → A	C → P	C → R	P → A	P → C	P → R	R → A	R → C	R → P	Avg
✓	✓	✓	75.4	88.5	85.9	73.3	88.7	83.8	75.2	75.3	87.1	80.9	77.8	91.8	82.0
✓	✓	✗	72.5	87.3	84.1	73.2	87.6	81.3	74.0	73.6	84.8	78.7	76.0	90.6	80.3
✓	✗	✗	69.4	84.1	82.6	71.0	83.5	79.6	71.6	70.4	83.9	77.2	73.9	88.0	77.9
✗	✗	✗	66.1	76.7	79.2	62.2	76.1	73.6	66.1	70.3	82.1	72.9	71.3	83.1	73.3

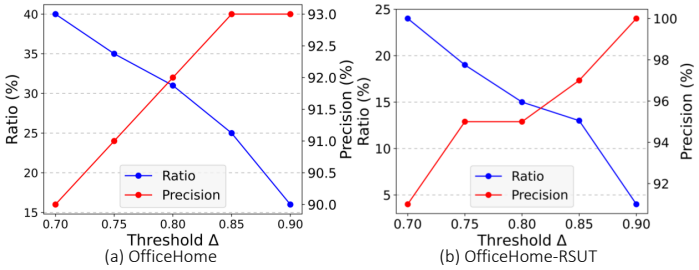

Fig. a3. The ratio (%) of the pseudo-labeled prototypes among all the prototypes and the precision (%) of the pseudo-labeled prototypes, along with different threshold Δ. The score is averaged over *all* scenarios of OfficeHome and OfficeHome-RSUT.

A.6 The ratio of the pseudo-labeled prototypes

In Fig. a3, we measure the ratio of the pseudo-labeled prototypes among all the prototypes with different threshold values. As shown in Fig. a3, a considerable amount of prototypes is pseudo-labeled. When Δ increases, the portion of pseudo-labeled prototypes decreases, while their precision increases.

B More results

B.1 Comparison with SOTA semi-supervised DA method

Semi-Supervised Domain Adaptation (SSDA) is a label-efficient domain adaptation task that allows a small amount of labeled data per class (*i.e.*, k-shot per class) in the target domain. While SSDA and ADA are similar in that both utilize a small amount of labeled target data for domain adaptation, the focus of their methods is slightly different. While ADA methods focus on sampling the most performance-profitable data, SSDA methods focus on effective training strategies to utilize the few labeled target data.

In Table a6-a7, we compare LAMDA with state-of-the-art SSDA methods [6,7] using a 10%-budget for the source-target domain pair of OfficeHome and OfficeHome-RSUT. LAMDA surpasses previous arts of both ADA and SSDA in every setting of the datasets. The performance of state-of-the-art SSDA methods is often as competitive as or even outperforms the previous ADA methods, Thus, combining the training strategy of SSDA with ADA methods would boost the performance of ADA.

B.2 Previous methods equipped with cosine classifier

In Fig. a4, we combine the cosine classifier with the existing approaches and compare their performance with that of LAMDA varying budget for each of OfficeHome and OfficeHome-RSUT datasets. hods with 10%-budget. LAMDA

Table a6. Comparison with state-of-the-art Semi-Supervised Domain Adaptation (SSDA) methods [6,7] measured by accuracy (%) when using 10%-budget for each source-target domain pair of four domains of OfficeHome: **A**rt, **C**lipart, **P**roduct, and **R**eal. (†): Since ECACL requires the same number of labels for each class, we randomly sample the same number of instances within each class.

Task	Method	OfficeHome												
		A → C	A → P	A → R	C → A	C → P	C → R	P → A	P → C	P → R	R → A	R → C	R → P	Avg
SSDA	ECACL† [7]	72.2	86.7	82.8	70.5	85.0	82.6	70.9	71.5	82.9	76.0	74.0	88.9	78.7
	CDAC [6]	69.5	83.2	80.2	66.9	82.4	78.7	66.1	70.6	80.9	72.3	70.5	87.2	75.7
ADA	TQS [1]	64.3	84.8	83.5	66.1	81.0	76.7	66.5	61.4	82.0	73.7	65.9	88.5	74.5
	CLUE [9]	62.1	80.6	73.9	55.2	76.4	75.4	53.9	62.1	80.7	67.5	63.0	88.1	69.9
	S^3VAADA [10]	67.8	83.9	82.9	67.0	81.4	79.5	65.8	65.9	82.4	74.8	68.6	87.9	75.7
	Ours	**74.8**	**88.5**	**86.9**	**73.8**	**88.2**	**83.3**	**74.6**	**75.5**	**86.9**	**80.8**	**77.8**	**91.7**	**81.9**

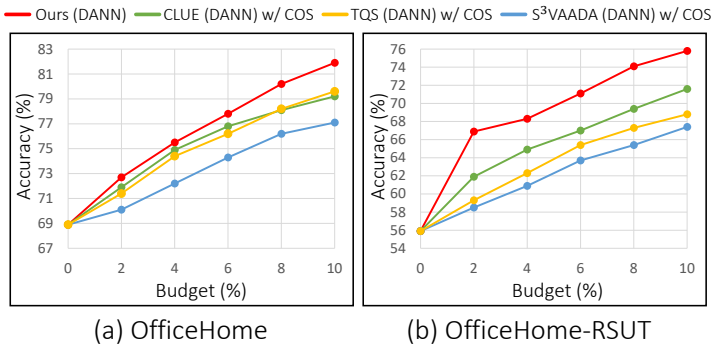

Fig. a4. Accuracy versus the percent of labeled target instances as budget, when each method is equipped with DANN [2] and cosine classifier. The accuracies are averaged on *all* scenarios of the OfficeHome and OfficeHome-RSUT. W/ COS: with cosine classifier.

constantly outperforms the previous arts in every setting on both datasets, where it with 6%-budget is often as competitive as or even surpasses the previous methods with 10%-budget. The performance gap between LAMDA and other methods increases as the budget increases (Fig. a4a), indicating that LAMDA effectively utilizes the budget in label distribution matching and supervised learning.

B.3 Evaluation on DomainNet

In Fig. a5, we compare the performance of LAMDA and the existing methods varying budget for two source-target domain pairs: Art to Product and Product to Clipart of DomainNet. LAMDA clearly outperforms the previous arts for the two domain adaptation settings, which demonstrates the scalability of LAMDA in the large-scale datasets. In particular, LAMDA with only 2%-budget is often

Table a7. Comparison with state-of-the-art Semi-Supervised Domain Adaptation (SSDA) methods [6,7] measured by accuracy (%) when using 10%-budget for each source-target domain pair of three domains of OfficeHome-RSUT: **C**lipart, **P**roduct, and **R**eal. (†): Since ECACL requires the same number of labels for each class, we randomly sample the same number of instances within each class.

Task	Method	OfficeHome-RSUT						
		C → P	C → R	P → C	P → R	R → C	R → P	Avg
SSDA	ECACL† [7]	78.6	68.6	59.5	77.1	61.9	82.0	71.3
	CDAC [6]	73.0	58.7	55.8	73.3	50.3	77.3	64.7
ADA	TQS [1]	69.4	65.7	53.0	76.3	53.1	81.1	66.4
	CLUE [9]	69.7	65.9	57.1	73.4	59.5	82.7	68.1
	S³VAADA [10]	73.0	63.0	50.7	69.6	52.6	78.3	64.5
	Ours	**81.2**	**75.7**	**64.1**	**81.6**	**65.1**	**87.2**	**75.8**

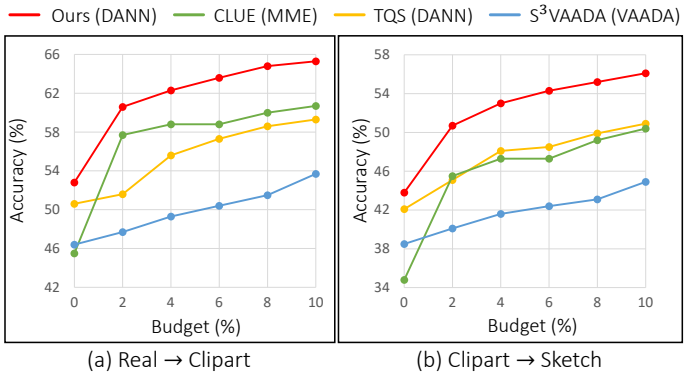

Fig. a5. Accuracy versus the percent of labeled target instances as budget for two source-target domain pair of DomainNet: (a) Real to Clipart, (b) Clipart to Sketch.

as competitive as or even outperforms the previous methods with a 10%-budget.

C Experiment details

C.1 Configuration of DANN

We adopt the implementation of DANN [2] from [5] and follow its default configurations. The discriminator consists of three fully connected layers of 1024, where each hidden layer is followed by batch normalization layer [3] and ReLU. We utilize Gradient Reverse Layer (GRL) for adversarial learning, where the coefficient λ of GRL is scheduled with $\lambda(s) = \frac{2}{1+\exp(-s)}$, where s denotes the training progress that scales from 0 to 1. We use the same DANN configuration and training schedule when combining previous ADA methods with DANN.

C.2 Implementation details of previous ADA methods

We evaluate the previous ADA methods based on the official implementations of TQS[1] [1], CLUE[2] [9], and S^{3}VAADA[3] [10]. We fit their implementations into our evaluation protocol with minimal modifications. We change the architecture of CLUE from ResNet34 into ResNet50, and we add missing parts of TQS implementation to make the code follow the original paper. When training each method, we use hyper-parameter values stated in each original paper, and if not stated, we tune them using the validation set.

[1] https://github.com/thuml/Transferable-Query-Selection
[2] https://github.com/virajprabhu/CLUE
[3] https://github.com/val-iisc/s3vaada

References

1. Fu, B., Cao, Z., Wang, J., Long, M.: Transferable query selection for active domain adaptation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 2, 3, 4, 6, 7, 8
2. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research (JMLR) **17**(1), 2096–2030 (2016) 1, 2, 5, 7, 8
3. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proc. International Conference on Machine Learning (ICML) (2015) 8
4. Jin, Y., Wang, X., Long, M., Wang, J.: Minimum class confusion for versatile domain adaptation. In: European Conference on Computer Vision. pp. 464–480. Springer (2020) 3
5. Junguang Jiang, Baixu Chen, B.F.M.L.: Transfer-learning-library. https://github.com/thuml/Transfer-Learning-Library (2020) 8
6. Li, J., Li, G., Shi, Y., Yu, Y.: Cross-domain adaptive clustering for semi-supervised domain adaptation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 6, 7
7. Li, K., Liu, C., Zhao, H., Zhang, Y., Fu, Y.: Ecacl: A holistic framework for semi-supervised domain adaptation. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 6, 7
8. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research (JMLR) **9**(11) (2008) 4
9. Prabhu, V., Chandrasekaran, A., Saenko, K., Hoffman, J.: Active domain adaptation via clustering uncertainty-weighted embeddings. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 1, 2, 3, 4, 6, 7, 8
10. Rangwani, H., Jain, A., Aithal, S.K., Babu, R.V.: S3vaada: Submodular subset selection for virtual adversarial active domain adaptation. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 1, 2, 3, 4, 6, 7, 8
11. Saito, K., Kim, D., Sclaroff, S., Darrell, T., Saenko, K.: Semi-supervised domain adaptation via minimax entropy. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 3