Supplementary Material for SENTRY: Selective Entropy Optimization via Committee Consistency for Unsupervised Domain Adaptation

1. Appendix

Contents

1.1. Ablating class-balancing

In Tab. 1, we include results for ablating class-balancing on both the source and target domains. As seen, both contribute a small gain: +0.5%, +0.3% on C→S (Row 1 v/s 2) for source class-balancing and +2%, +0.1% on Rw→Cl (Row 1 v/s 3) for target pseudo class-balancing. Using both together works best (Row 4). However, even *without any class balancing* (Row 1), SENTRY is still 3.8% and 6.2% better than the next best method on each shift (Tab. 1-2 in paper). This confirms that the gains are due to SENTRY's predictive consistency-based selective optimization.

#	CB (source)	pseudo CB (target)	DomainNet C→S	OH (RS-UT) Rw→Cl
1			76.6	56.2
2	✓		$77.1_{+0.5}$	$56.5_{+0.3}$
3		✓	$78.6_{+2.0}$	$56.3_{+0.1}$
4	✓	✓	$79.5_{+2.9}$	$56.8_{+0.6}$

Table 1: Ablating class balancing. Gray row=SENTRY. CB=class balancing. subscript=improvement v/s row 1.

1.2. Role of FS-architecture

Recall that in our experiments, we matched the "few-shot" style CNN architecture used in Tan *et al.* [12] (Sec 4.2 in main paper). We now quantify the effect of this choice. As before, we measure average accuracy on DomainNet Clipart→Sketch (C→S) and OfficeHome RS-UT Real World→Clipart (Rw→Cl). We rerun our method without the few-shot modification, and observe a 1.6% drop on C→S and a 0.03% increase in average accuracy on Rw→Cl. Overall, this modification seems to lead to a slight gain.

1.3. Additional analysis

Per-class accuracy change. In Fig. 2, we report the per-class accuracy (sorted by class cardinality) after adaptation using our method on DomainNet Clipart→Sketch, and contrast it against the next best-performing method, InstaPBM [6]. As seen, SENTRY outperforms InstaPBM on 37/40 categories, and is competitive on the others.

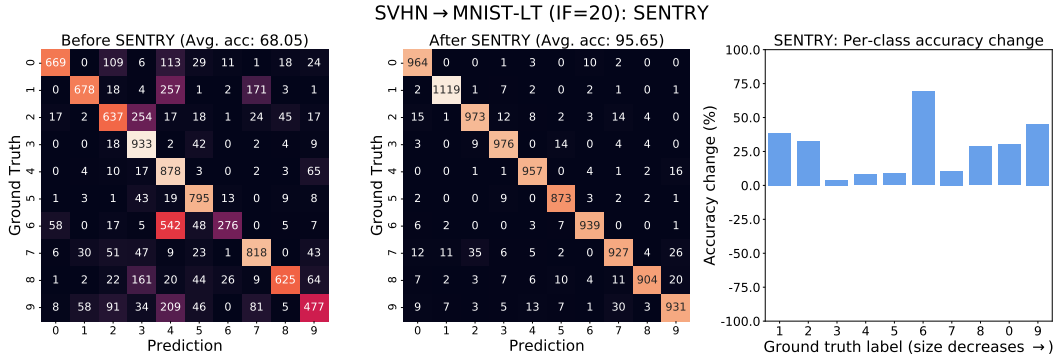


Figure 1: SVHN→MNIST-LT (IF=20): Performance on target test set after SENTRY.

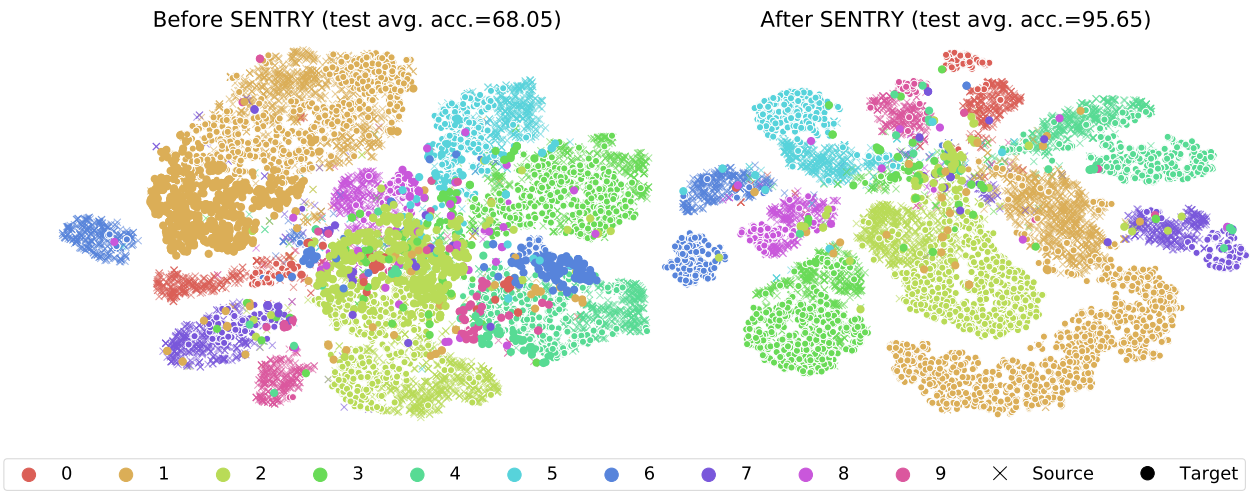


Figure 3: SVHN→MNIST: We use t-SNE [10] to visualize features for incorrect (large, opaque circles) and correct (partly transparent circles) model predictions on the imbalanced target train set and source train set before (left) and after (right) adaptation via SENTRY. Colors denote ground truth class, and × and ◯ denote source and target instances. SENTRY is able to overcome significant misalignments for both head classes with many examples (*e.g.* 1's and 2's) as well as tail classes with very few examples (*e.g.* 0's and 9's).

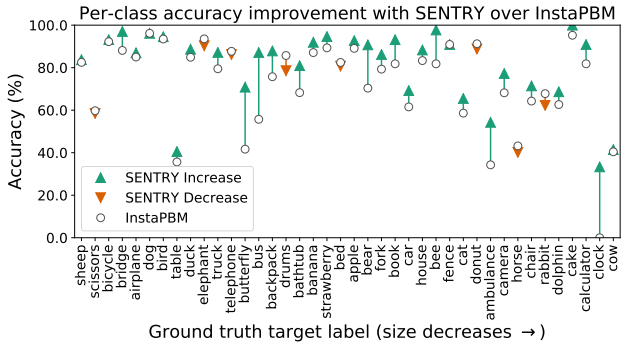
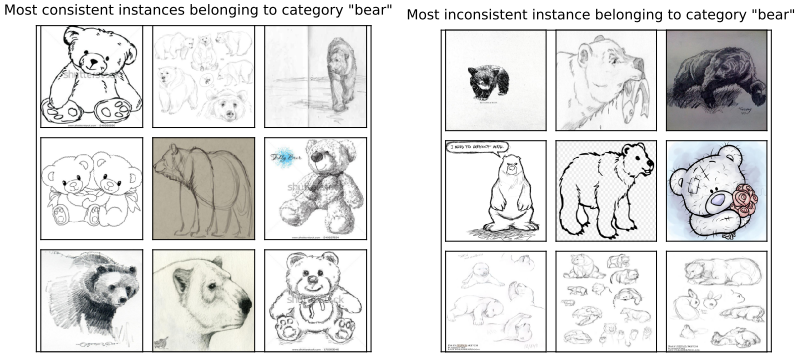


Figure 2: DomainNet C→S: Per-class accuracy gain with SENTRY over InstaPBM. Classes are sorted by size (largest → smallest).

We further analyze the performance of SENTRY on the SVHN→MNIST-LT (IF=20) shift, wherein the target train set has been manually long-tailed to create an imbalance factor of 20 (Sec 4.4 of main paper). In Fig 1, we show a confusion matrix of model predictions on the target test set after source training (left) and after target adaptation via SENTRY (middle). As seen, strong misalignments exist initially. However, after adaptation via our method, alignment improves dramatically across all classes. In Fig 1 (right), we show the *change* in per-class accuracy after adaptation, while sorting classes in decreasing order of size. As seen, SENTRY improves performance for both head and tail classes, often very significantly so.

t-SNE with SENTRY. Next, we use t-SNE [10] to visualize features (logits) extracted by the model for the source and target *train* sets. In Fig. 3, we visualize the feature landscape before and after adaptation via SENTRY. As seen, signifi-



(a) (b)

Figure 4: DomainNet Clipart→Sketch: Visualizing most consistent and inconsistent target instances.

Method	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr	AVG
Source	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [7]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [4]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [9]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [8]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
BSP [2]	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
MDD [16]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
MCS	55.9	73.8	79.0	57.5	69.9	71.3	58.4	50.3	78.2	65.9	53.2	82.2	66.3
InstaPBM [6]	54.4	75.3	79.3	65.4	**74.2**	**75.0**	63.3	49.7	80.2	72.8	57.8	83.6	69.7
MDD+I.A [5]	56.2	**77.9**	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
Ours	**61.8**	77.4	**80.1**	**66.3**	71.6	74.7	**66.8**	**63.0**	**80.9**	**74.0**	**66.3**	**84.1**	**72.2**

Table 2: Accuracies on standard OfficeHome. Bold and underscore denote the best and second-best performing methods respectively.

cant label imbalance exists: *e.g*. a lot more 1's and 2's are present as compared to 0's and 9's. Further, we denote target instances that are *incorrectly* classified by the model as large, opaque circles. Before adaptation (left), significant misalignments exist, particularly for head classes such as 1's and 2's. However, after adaptation via SENTRY, cross-domain alignment for most classes improves significantly, as does the average accuracy on the target test set (68.1% vs 95.7%).

Qualitative results. In Fig 4, we provide some qualitative examples to build intuition about our consistency-based selection. For the Clipart→Sketch shift, we visualize target (*i.e*. sketch) instances belonging to the ground truth category "bear". On the left, we visualize a random subset of target instances for which model predictions are *most consistent* under augmentations over the course of adaptation. On the right, we visualize a random subset of target instances for which model predictions are *most inconsistent* over the course of adaptation via SENTRY. Unsurprisingly, we find highly consistent instances to be "easier" to recognize, with more canonical poses and appearances. Similarly, inconsistent instances often tend to be challenging, and may even correspond to label noise, but our method appropriately avoids increasing model confidence on such instances.

1.4. OfficeHome Results

In Table 3a of the main paper, we presented accuracies averaged over all 12 shifts in the standard version of OfficeHome [14] for our proposed method against prior work. In Table 2, we include the complete table with performances on every shift. As seen, SENTRY achieves state-of-the-art performance on 9/12 shifts, and improves upon the next best method (InstaPBM [6]) by 2.5% overall.

1.5. Additional Implementation Details

In Sec 4.2 of the main paper, we presented implementation details for our method. We describe a few additional details to aid in reproducibility.

Training and Optimization. We match optimization details to Tan *et al*. [12]. On all benchmarks other than DIGITS, we use SGD with momentum of 0.9, a learning rate of 10^{-2} for the last layer and 10^{-3} for all other layers, and weight decay of 5×10^{-4}. We use the learning rate decay strategy proposed in Ganin *et al*. [4]. On DIGITS, we use Adam with a learning rate of 2×10^{-4} and no weight decay. We use a batch size of 16 on DomainNet, OfficeHome, and VisDA, and 128 on DIGITS. For data augmentation when training the source models on DomainNet, OfficeHome, and VisDA, we first resize to 256 pixels, extract a random crop of size (224x224), and randomly flip images with a 50% probability. For SENTRY, we use RandAugment [3] for generating augmented images, as described in Sec. 3.3 of the main paper and do not use any additional augmentations. For $L_{\texttt{SENTRY}}$, we average loss for consistent and inconsistent instances separately and weigh each loss by the proportion of instances assigned to each group. We select λ_{IE}=0.1 and $\lambda_{\texttt{SENTRY}}$=1.0 so as to approximately scale each loss term to the same order of magnitude.

Baseline implementations. For all baselines except InstaPBM [6], we directly report results from prior work. We base our InstaPBM implementation on code provided by authors and implement target information entropy, conditional entropy, contrastive, and mixup losses with loss weights 0.1, 1.0, 0.01, 0.1 respectively.

1.6. Analysis of DM-based methods under LDS

Prior work has already demonstrated the shortcomings of distribution-matching based UDA methods under additional label distribution shift [6, 15]. Wu *et al*. [15] show that DM-based domain adversarial methods optimize two out of a sum of three terms that bound target error as shown in Ben-David *et al*. [1]. Under matching task label distributions across domains, the contribution of the third term is small, which is the assumption under which these methods operate; absent this, the third term is unbounded and DM-based methods are not expected to succeed in domain alignment.

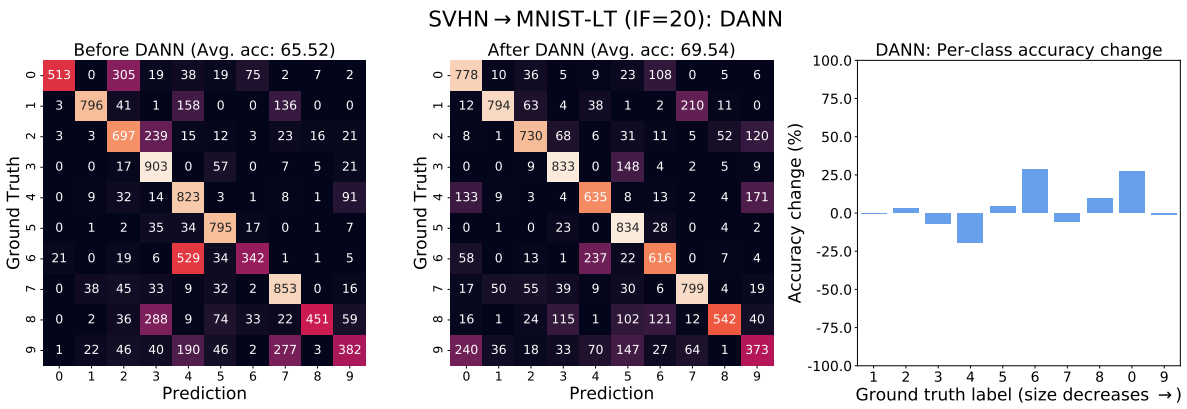


Figure 5: SVHN→MNIST-LT (IF=20): Performance on target test set after DANN [4].

We refer readers to Sec 2 of their paper for a formal proof.

Under LDS, such DM-based methods are expected to primarily mis-align majority (head) classes in the target domain with other classes in the source domain. We empirically test this hypothesis on the SVHN→MNIST-LT (IF=20) domain shift for digit recognition. In Fig 5, we repeat our per-class accuracy analysis for adaptation via DANN [4], a popular distribution matching UDA algorithm that uses domain adversarial feature matching. DANN has been shown to lead to successful domain alignment in the *absence* of label distribution shift (LDS); we now test its effectiveness in the presence of LDS. As seen, significant misalignments exist before adaptation (left). To match the source training strategy and architecture to the original paper, we do not use the few-shot architecture we use for our method, which leads to the slightly lower starting performance observed as compared to Fig. 1. However, due to label imbalance, DANN is unable to appropriately align instances and only slightly improves performance (69.54% in Fig 5, middle). In Fig 5, we show the *change* in accuracy for each class after adaptation, while sorting classes in decreasing order of size. As predicted by the theory, DANN does particularly poorly on head (majority) classes (1, 2, 3, 4), while slightly improving performance for classes with fewer examples. This is in contrast to our method SENTRY, which is able to improve performance for both head and tail classes (Fig 1, right).

1.7. Dataset Details

In Sec 4.1, we described our datasets in detail. For completeness, we also include label histograms and qualitative examples from each domain in the DomainNet and OfficeHome RS-UT benchmarks proposed in Tan *et al*. [12] in Figs. 6, 7.

Idiosyncrasy of the "clipart" domain. Some prior works in UDA (*e.g*. Tan *et al*. [12]) use center cropping at test time on DomainNet and OfficeHome, wherein they first resize a given image to 256 pixels and then extract a 224x224 crop from the center of the image. This practice has presumably carried over from ImageNet evaluation, where images are known to have a center bias [13]. Figs. 6a, 7c show qualitative examples from the clipart domain in DomainNet and OfficeHome. As seen, most clipart categories span the entire extent of the image and do *not* have a center bias. As a result, using center cropping at evaluation time can adversely affect performance when adapting to Clipart as a target domain. We show empirical evidence of this in Tables 3, 4 – when clipart is the target domain, performance drops consistently when using centercrop at test time. For SENTRY, we therefore do not use center crop at evaluation time. For comparison, we also include the performance of our strongest baseline in each setting in Tabs. 3, 4: InstaPBM [6] and MDD+I.A. [5], respectively. However, we note that in both settings, with and without centercrop, SENTRY still clearly outperforms our strongest baselines on both benchmarks.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010. 3

[2] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1081–1090, 2019. 3

[3] 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. 3

[4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–

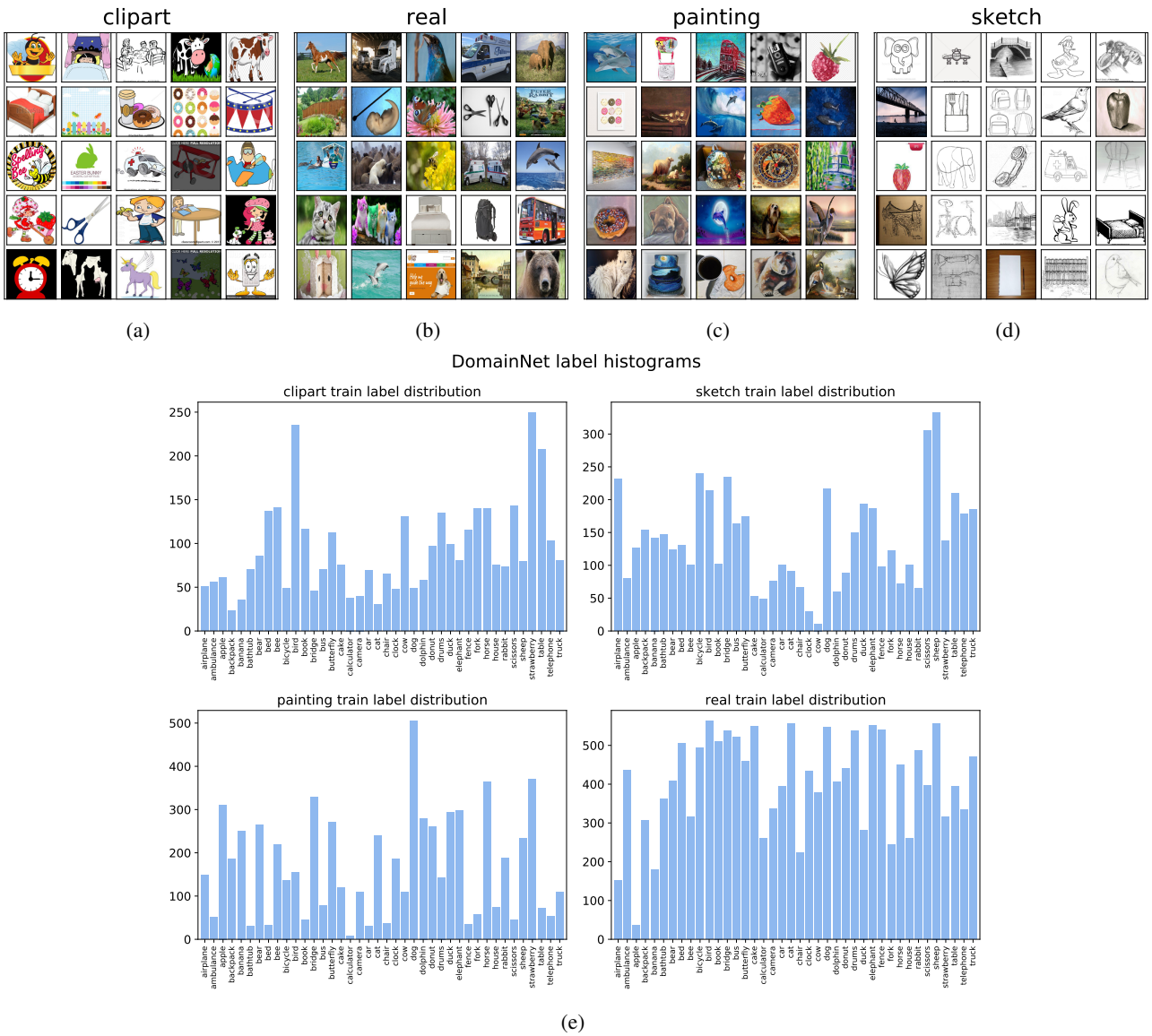


Figure 6: DomainNet [11] statistics: (a)-(d): Qualitative examples from each domain. (e): Label histograms for the splits proposed in Tan *et al*. [12].

Method	R → C	R → P	R → S	C → R	C → P	C → S	P → R	P → C	P → S	S → R	S → C	S → P	AVG
source	65.75	68.84	59.15	77.71	60.60	57.87	84.45	62.35	65.07	77.10	63.00	59.72	66.80
+CC-eval	62.42	69.31	59.28	79.74	59.49	58.46	84.55	60.42	66.26	78.61	58.31	61.31	66.51
InstaPBM [6]	80.10	75.87	70.84	89.67	70.21	72.76	89.60	74.41	72.19	87.00	79.66	71.75	77.84
SENTRY (Ours)	83.89	76.72	74.43	90.61	76.02	79.47	90.27	82.91	75.60	90.41	82.40	73.98	81.39
+CC-eval	78.81	78.15	71.62	89.84	75.98	77.69	89.50	77.34	73.82	89.96	80.66	75.02	79.87

Table 3: Idiosyncrasy of the "clipart" domain: Per-class average accuracies on DomainNet without (white rows) and with (gray rows) centercrop at test time. We highlight in red performance drops due to centercrop eval when adaptating to Clipart as a target domain. For comparison, we also include the performance of InstaPBM [6], the 2nd best method from Table 1 in the main paper.

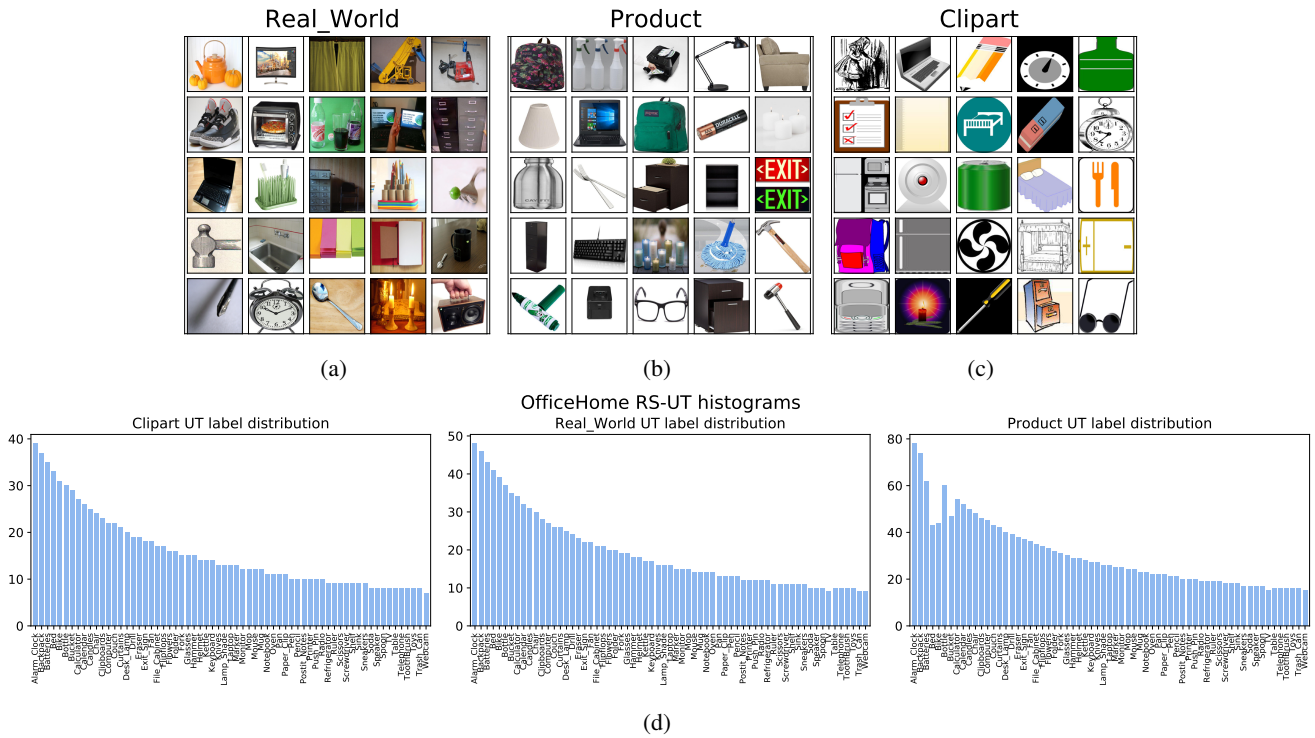


Figure 7: OfficeHome [14] statistics: (a)-(d): Qualitative examples from each domain. (e): Label histograms for the UT target splits proposed in Tan *et al*. [12].

Method	Rw → Pr	Rw → Cl	Pr → Rw	Pr → Cl	Cl → Rw	Cl → Pr	AVG
source	70.74	44.24	67.33	38.68	53.51	51.85	54.39
+CC-eval	70.25	38.20	67.74	35.61	55.08	52.90	53.30
MDD+I.A [5]	76.08	50.04	74.21	45.38	61.15	63.15	61.67
SENTRY (Ours)	76.12	56.80	73.60	54.75	65.94	64.29	65.25
+CC-eval	76.35	52.25	73.08	50.60	66.69	64.19	63.86

Table 4: Idiosyncrasy of the "clipart" domain: Per-class average accuracies on OfficeHome RS-UT without (white rows) and with (gray rows) centercrop at test time. We highlight in red performance drops due to centercrop eval when adaptating to Clipart as target. For comparison, we also include the performance of MDD+I.A. [5], the 2nd best method from Table 2 in the main paper.

1189, 2015. 3, 4

[5] Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. Implicit class-conditioned domain alignment for unsupervised domain adaptation. In *ICML*, 2020. 3, 4, 6

[6] Bo Li, Yezhen Wang, Tong Che, Shanghang Zhang, Sicheng Zhao, Pengfei Xu, Wei Zhou, Yoshua Bengio, and Kurt Keutzer. Rethinking distributional matching based domain adaptation. *arXiv preprint arXiv:2006.13352*, 2020. 1, 3, 4, 5

[7] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 3

[8] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018. 3

[9] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017. 3

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 2

[11] 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 International Conference on Computer Vision*, pages 1406–1415, 2019. 5

[12] Shuhan Tan, Xingchao Peng, and Kate Saenko. Class-imbalanced domain adaptation: An empirical odyssey. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2020. 1, 3, 4, 5, 6

[13] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011. 4

[14] 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. 3, 6

[15] Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International Conference on Machine Learning*, pages 6872–6881, 2019. 3

[16] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413, 2019. 3