CNG-SFDA: Clean-and-Noisy Region Guided Online-Offline Source-Free Domain Adaptation

Hyeonwoo Cho[1], Chanmin Park[2], Dong-Hee Kim[1], Jinyoung Kim[1], and Won Hwa Kim[3]

[1] VUNO Inc, [2] POSCO DX, [3] POSTECH
hyeonwoo.cho@gmail.com

1 Related Works

1.1 Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (UDA) [12, 25] has been widely proposed to address the domain shift problem. Adversarial learning methods, as in [5, 8, 12, 24, 39, 50], are a primary approach, focusing on aligning distribution discrepancies between the source and the target domains. Another approach proposed by [19, 35] introduces a discrepancy metric, which is used to minimize discrepancies in both the source and target distributions. Generative methods [3,28,29] have been utilized to tackle the domain shift by generating images that are indistinguishable between the source and target domains. In addition, [11,38] have explored the utilization of cluster structure information to understand class relationships between instances with the help of the labeled source data. More recently, self-training methods [6, 7, 17, 27, 42, 49, 52] produce pseudo-labels for the target data to transfer the target domain's knowledge to the source model. However, all these UDA approaches require access to both source and target data during the adaptation.

1.2 Learning with noisy labels.

Noisy labels typically negatively impact the performance of deep neural networks (DNNs) during training. Recent approaches [22, 23, 51] of learning with noise labels have been proposed to mitigate the negative effect of noise labels.

To address this issue, most methods, such as using noise-robust loss [13, 44, 46,51] functions to minimize effect of noisy labels, estimating the noise-transition matrix [14, 45], selecting clean samples from noisy data [16, 48], and label correction methods [1, 21, 32, 37, 47] that replace noisy labels with more reliable labels, are proposed. In particular, the label correction methods have performed better than other methods. Unsupervised Label Noise Modeling (ULNM) [1] partitions data into noisy and clean samples by applying a beta mixture model from the higher loss incurred by each sample. Similarly, DivideMix [21] employs two networks to select samples with clean labels and applies semi-supervised learning. MOIT [32] utilizes robust feature representation for noise detection, using k-nearest neighbors to infer label distributions and distinguish clean and noisy data by comparing them with the ground truth. While previous methods of training with noisy labels have benchmarks based on the ground truth for clean and noisy data, Source-Free Domain Adaptation (SFDA) is more challenge setting due to the absence of the ground truth for clean and noisy data.

Moreover, most existing methods [1, 21, 32] rely on identifying false labels within given neighbors structures. However, our proposed method partitions clusters into the clean (i.e., close to cluster prototypes) and noisy (i.e., far from cluster prototypes) regions by using cluster prototypes (i.e., centroids of clusters) and employs distinct training strategies for each regions within the clusters.

2 Notation

In this section, we provide notations including symbols and brief descriptions for our algorithm addressed in Sec. 3, as shown in Tab. 1. We subdivide notations into 5 parts: *Data, Augmentation, Architecture, Loss, General.*

	Symbol	Description
Data	D_t	Target domain data
	x_t	Batch sampled from D_t
	x_{cr}	Samples in clean regions from D_t
	$\tilde{x}_t$	Mixup samples from D_t
Augmentation	T_w	Weak augmentation
	T_s	Strong augmentation
	$T_{\bar{s}}$	Strong augmentation
Architecture	$f(\cdot)$	Source model
	$g(\cdot)$	Encoder of $f(\cdot)$
	$h(\cdot)$	Classifier of $f(\cdot)$
	$\bar{f}(\cdot)$	Momentum model
	$\bar{g}(\cdot)$	Encoder of $\bar{f}(\cdot)$
	$\bar{h}(\cdot)$	Classifier of $\bar{f}(\cdot)$
	θ	Parameters of $f(\cdot)$
	$\bar{\theta}$	Parameters of $\bar{f}(\cdot)$
Loss	L_{cr}	Loss for clean samples x_{cr}
	L_{ccp}	Cluster compactness loss
	L_{div}	Loss for regularization
	L_{inst}	Instance-aware contrast loss
	L_{prt}	Prototype-aware contrast loss
	$L_{overall}$	Total loss
General	Q_w	Memory queue of w_t
	Q_k	Memory queue of k_t
	P_w	Memory queue of $\bar{f}(T_w(x_t))$
	μ_k	Prototype for k-th cluster
	S	Similarity
	p_c	Clean probability
	α	Threshold of p_c
	$\tilde{w}_t$	Mixup weight
	P_{prt}	Positive pairs for L_{prt}
	N_{prt}	Negative pairs for L_{prt}
	P_{inst}	Positive pairs for L_{inst}
	N_{inst}	Negative pairs for L_{inst}

Table 1: Notation Table

Algorithm 1: CNG-SFDA algorithm

```
Input: Unlabeled target set $\mathcal{D}_t = \{x_t^i\}_{i=1}^{N_t}$, source model $f(\cdot) = g(h(\cdot))$, source
       encoder $g(\cdot)$, source classifier $h(\cdot)$ ;     // Let $\cdot||\cdot$ denote the append
       operation, C be the number of classes.
Output : updated $\theta$
Initialization: Momentum model $\bar{f}(\cdot) = \bar{h}(\bar{g}(\cdot))$, $\bar{f}_{\bar{\theta}} = f_\theta$
for $epoch = 1$ to $MaxEpoch$ do
    step1: Stacking Memory Queue
    $Q_w, Q_k, P_w \leftarrow [\,], [\,], [\,]$
    for $iter = 1$ to $MaxIter$ do
        $x_t \leftarrow batch\ sampled\ from\ \mathcal{D}_t$
        $w_t = g(T_w(x_t))$
        $q_t = g(T_s(x_t))$
        $k_t = \bar{g}(T_{\bar{s}}(x_t))$
        $\bar{p}_w = \sigma(\bar{f}(T_w(x_t)))$
        $Q_w, Q_k, P_w \leftarrow Q_w||w_t, Q_k||k_t, P_w||\bar{p}_w$
    end
    step2 : Introduce Clean Probability
    $\mu = ||Q_w \cdot P_w/P_w||_{\ell_2} \leftarrow$ Calculate Cluster Prototype
    $S = Q_w \cdot \mu/||Q_w|| \cdot ||\mu|| \leftarrow$ Similarity features and all cluster prototypes
    for $iter \leftarrow 1$ to $MaxIter$ do
        $x_t \leftarrow batch\ sampled\ from\ \mathcal{D}_t$
        $w_t = g(T_w(x_t))$
        $q_t = g(T_s(x_t))$
        $k_t = \bar{g}(T_{\bar{s}}(x_t))$
        $p_s = \sigma(f(T_s(x_t)))$
        $p_w = \sigma(f(T_w(x_t)))$
        $\hat{p}_w^{(i,c)} = \frac{1}{K}\sum_{j=1}^{K} p_w^{(j,c)}$
        $\hat{y}_t = argmax_c\ \hat{p}_w^{(i,c)}$ ;                  // Generates pseudo-labels
        $s_t = w_t \cdot \mu_k/||w_t|| \cdot ||\mu_k||$
         $p_c = exp(s_t)/exp(S) \leftarrow$ clean probabilities of batch
        step3 : Train Clean and Noise Regions
        if $p_c > \alpha$ then
            $x_{cr} \leftarrow$ Samples in Clean Regions
        Compute loss $L_{cr}$ for $x_{cr}, \hat{y}_t$
        $\tilde{x}_t = \lambda x_t^i + (1-\lambda)x_t^j \leftarrow$ mixed images
        $\tilde{y}_t = \lambda \hat{y}_t^i + (1-\lambda)\hat{y}_t^j \leftarrow$ mixed pseudo-labels
        $\tilde{w}_t = exp(\lambda p_c^i + (1-\lambda)p_c^j) \leftarrow$ Mixup weight
        Compute loss $L_{ccp}$ using $\tilde{w}_t$ for $\tilde{x}_t, \tilde{y}_t$
        Compute loss $L_{div}$ for $p_s$
        step4 : Contrastive Learning
        $\bar{c} \leftarrow$ class of $\hat{y}_t$
        $P_{prt} \leftarrow \{q_t, \mu_{\bar{c}}\}$
        $N_{prt} \leftarrow \{q_t, \{\mu_{k \neq \bar{c}}\}_{k=1}^{C}\}$ ;       // Filter $\mu_k$ with same class of $q_t$
        Compute loss $L_{prt}$ for $P_{prt}$, $N_{prt}$
        $P_{inst} \leftarrow \{q_t, k_t\}, N_{inst} \leftarrow \{q_t, Q_k\}$
        Compute loss $L_{inst}$ for $P_{inst}$, $N_{inst}$
        Update $\theta$ by minimizing $L_{overall}$ using SGD optimizer
        Update $\bar{\theta}$ by EMA of $\theta$
    end
end
```

3 Algorithm

Algorithm 1 outlines a training procedure of CNG-SFDA. As detailed in the methodology section of the main paper, CNG-SFDA algorithm comprises 4 distinct steps: step1 (stacking the memory queue), step2 (Distinguishing Clean and Noisy Regions in Clusters), step3 (Training Strategy for Clean and Noisy Regions), and step4 (Instance and Prototype-aware Contrastive Learning). In particular, for online SFDA, we change a max epoch to 1 and activate the soft voting for generating pseudo-labels once the memory queue reaches a certain length.

4 Experiments

4.1 Implementation details.

In this section, we describe the implementation details. We categorize this section into *Training Source* and *Training Target* (i.e., adaptation phase). For all implementations, we use Pytorch [33] and follow AdaContrast implementations * for a baseline method. For all training, we use 2 NVIDIA A6000 GPUs. Code is here: https://github.com/hyeonwoocho7/CNG-SFDA.

Training Source For image classification tasks, we initialize the ResNet backbone with ImageNet-1K [10] pre-trained weights from the Pytorch model zoo. We split the source dataset into training and validation sets in a ratio of 9:1. Following [4], we train the pretrained model for 10, 60, and 100 epochs for VisDA-C, DomainNet-126, and PACS, respectively, using a consistent learning rate of 2e-4. Additionally, we set a same batch size of 128 for all datasets.

Training Target We categorize this section into implementation details for offline and online SFDA. As shown Tab. 1 augmentation part, we leverage 3 different augmentations. For all augmentations, we use Pytorch default implementations.

Offline SFDA In the offline SFDA phase, we conduct training for 30 and 15 epochs with a batch size of 128 for VisDA-C and DomainNet-126. For PACS, we train the model with batch size 32 for 30 and 50 epochs in single-source and multi-source settings, respectively. We use SGD optimizer with a momentum of 0.9, a constant learning rate of 2e-4, and weight decay of 1e-4 for all datasets.

Online SFDA For online adaptation, we set the epoch to 1 on all datasets since the target data is seen to model during only inference. Other hyper-parameters are the same as offline settings.

4.2 SFDA on DomainNet

Tab. 2 shows the performance of the online CNG-SFDA and other online SFDA methods on DomainNet [34] in a multi-source setting. DomainNet is a large-scale dataset with 345 classes, unlike DomainNet-126 which has 126 classes. As shown in Tab. 2, CNG-SFDA shows competitive performance with the SOTA [43].

*https://github.com/DianCh/AdaContrast

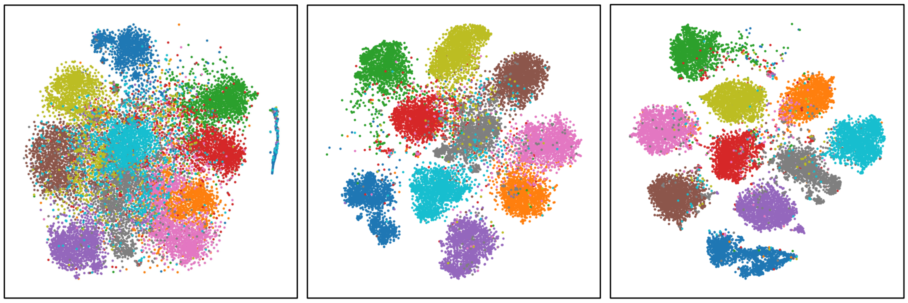
Method	clipart	infograph	painting	quickdraw	real	sketch	Avg.
ERM [40]	64.8	22.1	51.8	13.8	64.7	54.0	45.2
PL (ICMLW'13) [20]	65.0	19.0	50.3	4.21	54.4	54.2	41.2
BN (NeurIPS'20) [36]	64.5	15.6	50.6	11.8	63.9	53.9	43.4
Tent (ICLR'21) [41]	65.8	18.2	53.0	10.8	64.9	55.7	44.7
SHOT-IM (ICML'20) [43]	65.6	18.7	52.4	19.0	66.5	55.5	46.3
T3A (NeurIPS'21) [18]	64.8	22.1	50.9	**19.4**	65.9	54.0	46.2
ETA (ICML'22) [31]	65.1	19.4	52.7	18.2	65.9	55.5	46.1
LAME (CVPR'22) [2]	64.2	15.6	50.5	11.8	63.5	53.7	43.2
TSD (CVPR'23) [43]	**66.1**	**24.1**	52.8	18.2	68.5	**56.7**	**47.7**
CNG-SFDA (online)	64.6	21.4	**57.1**	14.9	**69.9**	54.0	47.0

Table 2: Compare classification accuracy (%) on DomainNet with other SFDA methods (ResNet-50 Backbone).

5 Analysis

5.1 Feature visualization

Fig. 1 shows the feature distributions from the source model (referred to as 'Source Only'), AdaContrast [4] method, and CNG-SFDA through t-SNE [26]. We observe that the feature distribution of CNG-SFDA has better cluster compactness than that of other methods.



(a) Source Only (b) Adacontrast [4] (c) CNG-SFDA

Fig. 1: t-SNE of the feature distributions from 'Source Only', Adacontrast [4], and CNG-SFDA on VisDA-C.

5.2 Model calibration

We consider that model calibration in SFDA enhances the reliability, robustness, and interpretability of the adapted target model, making it a crucial aspect of

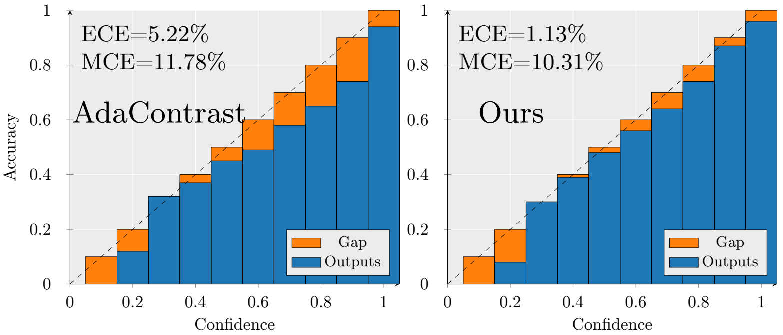


Fig. 2: Model calibration analysis of AdaContrast [4] and CNG-SFDA on **VisDA-C**.

deploying machine learning solutions in dynamic environments. In Fig. 2, we compare the model calibration [9, 15, 30] for AdaContrast and CNG-SFDA on VisDA-C. We divide the probability of the model into 10 bins and compute the average accuracy in relation to the average confidence for each bin. The more close the model's outputs (blue bar) to $y = x$ line, the better calibration it has. As shown in Fig. 2, CNG-SFDA has better calibration than AdaContrast. In addition, we use two intuitive statistics that measure calibration [9, 30]: expected calibration error (ECE) and maximum calibration error (MCE). The more close ECE and MCE to zero, the better calibration the model has. CNG-SFDA achieves 4.09(%) ECE and 1.47(%) MCE lower than AdaContrast's.

5.3 Similarity distribution of Clean vs mix-up vs Noisy.

Fig. 3 presents the density distribution of similarity between clean, noisy, and mix-up samples based on cluster prototypes on VisDA-C. We calculated this similarity using the cosine similarity between each sample's embedding and its closest cluster prototype embedding. We observe that clean samples exhibit features closer to the prototype, while noisy samples exhibit more distant features. Notably, features of mix-up tend to be distributed between the clean and noisy distributions. These findings support our hypothesis that mix-up features serve as a bridge connecting clean and noisy features.

5.4 Comparative Analysis: Detecting Clean and Noisy Data

We conduct a comparative analysis on detecting clean and noisy data. To compare the effectiveness of detecting clean and noisy data with other method, we leverage accuracy, recall, and precision metrics. We compare CNG-SFDA with MOIT [32] method which detects clean and noisy data using neighbors features. Specifically, we measure the detection performance for noisy data from the source model. As illustrated in Fig. 4, our method improves the detection performance

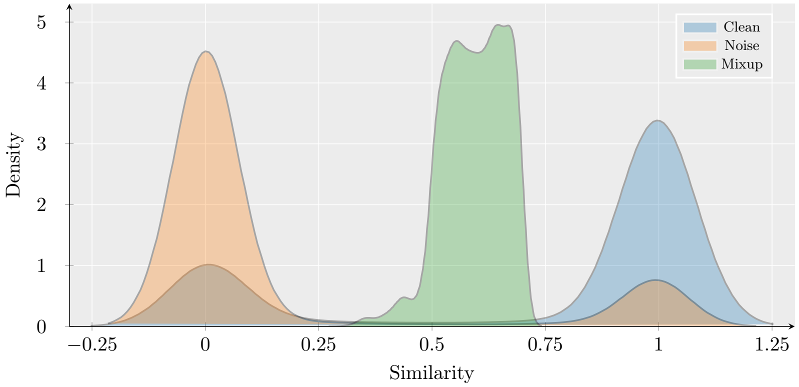


Fig. 3: Similarity distribution of cluster prototype with Clean vs Noisy vs Mixup.

on VisDA-C, DomainNet-126, and PACS. Also, MOIT exhibits good performance in the precision score, as it tends to capture less noise data. However, we observe that our approach selects as many clean samples as possible, leading to outstanding recall on all datasets.

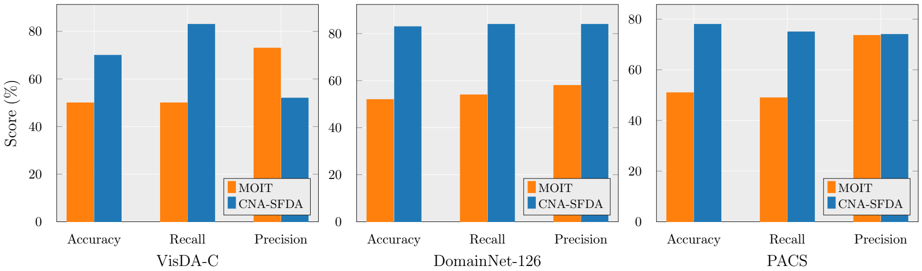


Fig. 4: Compare the performance of MOIT [32] and CNG-SFDA for detecting clean and noisy samples on **all datasets**.

5.5 Component ablation of online SFDA

Tab. 3 shows the ablation results of each component for all datasets in the online SFDA setting. 'PL' is a baseline method that uses the online pseudo labeling proposed in [4]. As shown in Tab. 3, we observe improved performance in most cases as we apply each component. This demonstrates the effectiveness of each element of CNG-SFDA in the online SFDA setting.

PL	L_{cr}	L_{ccp}	L_{div}	L_{inst}	L_{prt}	VisDA-C	DomainNet	PACS
✓	✗	✗	✗	✗	✗	83.3	62.6	70.2
✓	✓	✗	✗	✗	✗	82.7	64.5	70.8
✓	✓	✓	✗	✗	✗	86.5	63.8	71.6
✓	✓	✓	✓	✗	✗	86.6	64.2	73.5
✓	✓	✓	✓	✓	✗	86.1	66.3	74.0
✓	✓	✓	✓	✓	✓	**86.8**	**66.9**	**74.4**

Table 3: The effectiveness of each component **all datasets** is validated by classification accuracy (%).

5.6 Insensitivity of hyper-parameter

Fig. 5 presents the sensitivity analysis of different hyper-parameters for online and offline SFDA, respectively. These results demonstrate the insensitivity of CNG-SFDA to hyper-parameter tuning.

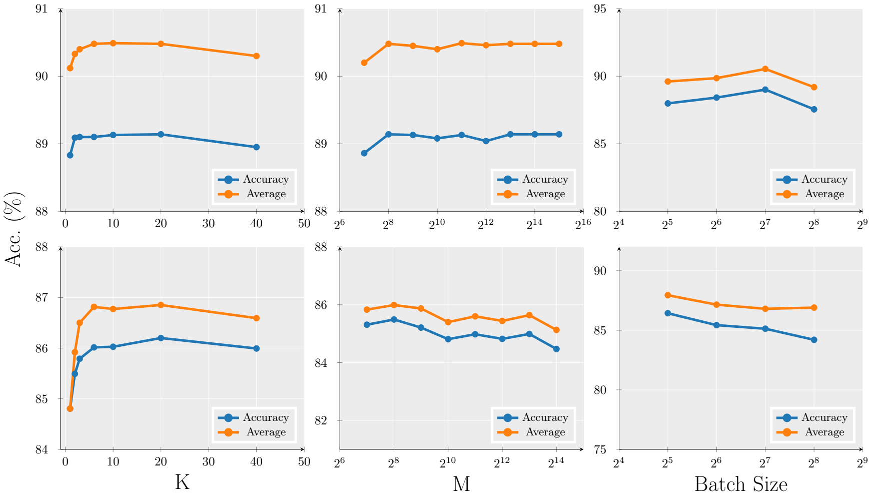


Fig. 5: Sensitivity analysis of offline SFDA (Top) and online SFDA (Bottom) with different hyper-parameters. Classification Accuracy (%) with K nearest neighborhoods for pseudo labeling, the length M of memory queue, and the batch size.

References

1. Arazo, E., Ortego, D., Albert, P., O'Connor, N., McGuinness, K.: Unsupervised label noise modeling and loss correction. In: International conference on machine learning. pp. 312–321. PMLR (2019)
2. Boudiaf, M., Mueller, R., Ben Ayed, I., Bertinetto, L.: Parameter-free online test-time adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8344–8353 (2022)
3. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3722–3731 (2017)
4. Chen, D., Wang, D., Darrell, T., Ebrahimi, S.: Contrastive test-time adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 295–305 (2022)
5. Chen, L., Chen, H., Wei, Z., Jin, X., Tan, X., Jin, Y., Chen, E.: Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7181–7190 (2022)
6. Cho, H., Nishimura, K., Watanabe, K., Bise, R.: Effective pseudo-labeling based on heatmap for unsupervised domain adaptation in cell detection. Medical Image Analysis **79**, 102436 (2022)
7. Choi, J., Jeong, M., Kim, T., Kim, C.: Pseudo-labeling curriculum for unsupervised domain adaptation. arXiv preprint arXiv:1908.00262 (2019)
8. Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., Tian, Q.: Gradually vanishing bridge for adversarial domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12455–12464 (2020)
9. DeGroot, M.H., Fienberg, S.E.: The comparison and evaluation of forecasters. Journal of the Royal Statistical Society: Series D (The Statistician) **32**(1-2), 12–22 (1983)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
11. Deng, Z., Luo, Y., Zhu, J.: Cluster alignment with a teacher for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9944–9953 (2019)
12. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. pp. 1180–1189. PMLR (2015)
13. Ghosh, A., Kumar, H., Sastry, P.S.: Robust loss functions under label noise for deep neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)
14. Goldberger, J., Ben-Reuven, E.: Training deep neural-networks using a noise adaptation layer. In: International conference on learning representations (2016)
15. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International conference on machine learning. pp. 1321–1330. PMLR (2017)
16. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. Advances in neural information processing systems **31** (2018)

17. He, T., Shen, L., Guo, Y., Ding, G., Guo, Z.: Secret: Self-consistent pseudo label refinement for unsupervised domain adaptive person re-identification. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 879–887 (2022)
18. Iwasawa, Y., Matsuo, Y.: Test-time classifier adjustment module for model-agnostic domain generalization. Advances in Neural Information Processing Systems **34**, 2427–2440 (2021)
19. Lee, C.Y., Batra, T., Baig, M.H., Ulbricht, D.: Sliced wasserstein discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10285–10295 (2019)
20. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML. vol. 3, p. 896. Atlanta (2013)
21. Li, J., Socher, R., Hoi, S.C.: Dividemix: Learning with noisy labels as semi-supervised learning. arXiv preprint arXiv:2002.07394 (2020)
22. Li, J., Xiong, C., Hoi, S.C.: Learning from noisy data with robust representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9485–9494 (2021)
23. Liu, S., Niles-Weed, J., Razavian, N., Fernandez-Granda, C.: Early-learning regularization prevents memorization of noisy labels. Advances in neural information processing systems **33**, 20331–20342 (2020)
24. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. Advances in neural information processing systems **31** (2018)
25. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. Advances in neural information processing systems **29** (2016)
26. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
27. Mei, K., Zhu, C., Zou, J., Zhang, S.: Instance adaptive self-training for unsupervised domain adaptation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16. pp. 415–430. Springer (2020)
28. Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R., Kim, K.: Image to image translation for domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4500–4509 (2018)
29. Nam, H., Lee, H., Park, J., Yoon, W., Yoo, D.: Reducing domain gap by reducing style bias. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8690–8699 (2021)
30. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: Proceedings of the 22nd international conference on Machine learning. pp. 625–632 (2005)
31. Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., Tan, M.: Efficient test-time model adaptation without forgetting. In: International conference on machine learning. pp. 16888–16905. PMLR (2022)
32. Ortego, D., Arazo, E., Albert, P., O'Connor, N.E., McGuinness, K.: Multi-objective interpolation training for robustness to label noise. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6606–6615 (2021)
33. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)

34. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1406–1415 (2019)
35. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3723–3732 (2018)
36. Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., Bethge, M.: Improving robustness against common corruptions by covariate shift adaptation. Advances in neural information processing systems **33**, 11539–11551 (2020)
37. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5552–5560 (2018)
38. Tang, H., Chen, K., Jia, K.: Unsupervised domain adaptation via structurally regularized deep clustering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8725–8735 (2020)
39. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7167–7176 (2017)
40. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience (1998)
41. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. arXiv preprint arXiv:2006.10726 (2020)
42. Wang, Q., Breckon, T.: Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 6243–6250 (2020)
43. Wang, S., Zhang, D., Yan, Z., Zhang, J., Li, R.: Feature alignment and uniformity for test time adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20050–20060 (2023)
44. Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., Bailey, J.: Symmetric cross entropy for robust learning with noisy labels. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 322–330 (2019)
45. Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., Sugiyama, M.: Are anchor points really indispensable in label-noise learning? Advances in neural information processing systems **32** (2019)
46. Xu, Y., Cao, P., Kong, Y., Wang, Y.: L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. Advances in neural information processing systems **32** (2019)
47. Yi, K., Wu, J.: Probabilistic end-to-end noise correction for learning with noisy labels. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7017–7025 (2019)
48. Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., Sugiyama, M.: How does disagreement help generalization against label corruption? In: International Conference on Machine Learning. pp. 7164–7173. PMLR (2019)
49. Yu, Y.C., Lin, H.T.: Semi-supervised domain adaptation with source label adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24100–24109 (2023)
50. Zhang, W., Ouyang, W., Li, W., Xu, D.: Collaborative and adversarial network for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3801–3809 (2018)
51. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems **31** (2018)

52. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European conference on computer vision (ECCV). pp. 289–305 (2018)