Robust Nearest Neighbors for Source-Free Domain Adaptation under Class Distribution Shift

Antonio Tejero-de-Pablos[1], Riku Togashi[1], Mayu Otani[1], and Shin'ichi Satoh[1,2]

[1] AI Lab, CyberAgent, Shibuya, Tokyo, Japan
{antonio_tejero,togashi_riku,otani_mayu}@cyberagent.co.jp
https://research.cyberagent.ai
[2] National Institute of Informatics, Chiyoda, Tokyo, Japan
satoh@nii.ac.jp

Abstract. The goal of source-free domain adaptation (SFDA) is retraining a model fit on data from a source domain (*e.g.* drawings) to classify data from a target domain (*e.g.* photos) employing only the target samples. In addition to the domain shift, in a realistic scenario, the number of samples per class on source and target would also differ (*i.e.* class distribution shift, or CDS). Dealing label-less with CDS via target data only is challenging, and thus previous methods assume no class imbalance in the source data. We study the SFDA pipeline and, for the first time, propose a SFDA method that can deal with class imbalance in both source and target data. While pseudolabeling is the core technique in SFDA to estimate the distribution of the target data, it relies on nearest neighbors, which makes it sensitive to class distribution shifts (CDS). We are able to calculate robust nearest neighbors by leveraging additional generic features free of the source model's CDS bias. This provides a "second-opinion" regarding which nearest neighbors are more suitable for adaptation. We evaluate our method using various types of features, datasets and tasks, outperforming previous methods in SFDA under CDS. Our code is available at `https://github.com/CyberAgentAILab/Robust_Nearest_Neighbors_SFDA-CDS`.

Keywords: Source-free domain adaptation · Class distribution shift · Robust nearest neighbors

1 Introduction

After a deep neural network model is deployed, it normally finds data whose distribution is slightly shifted from that of the training data. This "domain shift" (also referred as *covariate shift*) worsens the performance of the model and limits its practical use. The research field of domain adaptation approaches this problem in order to make deep models more robust against label-less unseen data. Several domain adaptation scenarios have been proposed over the years, being

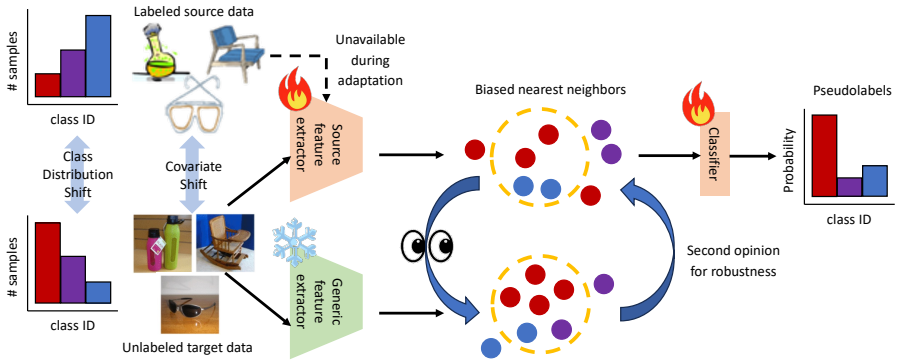

Fig. 1: We improve the pseudolabeling in source free domain adaptation by regulating the nearest neighbors calculation via the inclusion of a "generic" feature extractor. These features are not affected by the source model bias, and can be used as a reference to gain robustness against CDS. The *fire* emoji indicates model weight modification and the *snowflake* emoji indicates the model is frozen.

unsupervised domain adaptation (UDA) the most basic [3, 5, 30]. UDA methods aim to improve compatibility between the labeled data on which a model is trained (*i.e.* source domain) and the label-less data such model will be fed with after deployment (*i.e.* target domain). However, UDA methods require working with both source and target data simultaneously. This is impractical for applications in which, for security and privacy reasons, the source data is not available and only the already trained source model is provided, *e.g.* transfer learning among patient data in hospitals, or client data in companies. For this reason, source-free domain adaptation (SFDA) methods were proposed [7,22,40]. SFDA divides the adaptation process into two steps, the supervised training of the source model and the unsupervised adaptation of that model to the target data. Some real applications impose even further restrictions, such as the impossibility of modifying the model weights during adaptation [27]. This setting is referred as test-time adaptation (TTA).

In the strictest definition of source-free, there is no control over how the source model is trained, and the adaptation method needs to be solely based on the source model and the label-less target data. This leads to scenarios in which, unknowingly, a class distribution shift (CDS) exists between both domains. That is, the ratio of samples for each class is significantly different in the source and target domains. If not dealt with properly, the model becomes sensitive to such imbalance, and its predictions become biased to the majority class in the source domain. This is a very challenging scenario, as both *covariate* and *class distribution* shifts need to be tackled without source data nor labels.

The principle in SFDA methods is feeding the source model with the target data and assigning the most plausible labels (*i.e. pseudolabels*). This can be done by observing the outputs at either the feature level (*i.e.* observing the

nearest neighbors of a given sample) or at the logits level (*i.e.* observing the predicted probabilities). Previous work in SFDA under CDS approached the class distribution shift as a problem of noise at the logits level [20]. In order to avoid pseudolabels to be biased towards majority classes in the source data, they opt to modify the class distribution of the source data to be uniform. Although controlling the training of the source model is against the strict SFDA scenario, they had to relax the constraints since the CDS between source and target data cannot be estimated by the source model itself. We hypothesize that, since the CDS is unknown during adaptation, a model unrelated to the source data can provide a "second opinion" on the target data without the bias. Furthermore, instead of addressing pseudolabeling directly, we believe that a more fundamental problem lies in the previous step, the nearest neighbors calculation itself.

Figure 1 depicts the main idea of our proposal for SFDA under CDS. Unlike the previous work, we approach the majority/minority class bias at the nearest neighbors level by introducing an additional model to the SFDA pipeline. We opt for leveraging well-known pretrained models (*i.e.* ResNet, VisionTransformer, SwinTransformer) as they are widely available and have proved the generalizability of their features [4]. We regulate the nearest neighbors extracted in the source model's feature space by comparing them to the nearest neighbors in the generic feature space. We consider the neighbors are "robust" if they appear in both the source and generic models. In the teacher-student knowledge distillation fashion [13], the generic model remains frozen as the source model is adapted by taking generic features as a reference. As learning additional modules is not required, our method can also be applied to the TTA setting without modifications. In summary, our contributions are:

- We study the problem of SFDA under CDS, and identify a weakness in the nearest neighbors (NN) calculation, which was unnoticed by previous works.
- We propose a method for reducing bias in the NN by introducing generic features of an auxiliary model. Unlike previous works, we strictly adhere to the SFDA-CDS settings and do not manipulate the training of the source model.
- Our straightforward approach outperforms previous methods for both SFDA and TTA settings in several benchmarks under CDS, and for the first time in SFDA, without manipulating the training of the source model.

2 Related Work

Unsupervised Domain Adaptation (UDA). In order to tackle the covariate shift between the two domains, previous works proposed learning domain invariant features via adversarial learning [3, 5, 30], by minimizing inter-domain distance [11,14], the generation of intermediate domains between source and target [6,19], or making the source data similar to the target [26]. Such a variety of approaches is only possible since both source and target data are simultaneously available.

Source-Free Domain Adaptation (SFDA). In the strict definition of SFDA, only the model trained on the source data is provided for adaptation, and source data is never seen. However, a significant portion of the related work relaxes the restrictions of SFDA by employing specific policies when training the source model [16, 17, 39], which requires access to the actual source data. In our work, we consider a more realistic setting, in which the source model is provided but its training cannot be manipulated. In this situation, methods can just observe the outputs of the source model when fed with the target data; the labels assigned by the source model are called *pseudolabels* [18]. In order to understand the validity of these pseudolabels, SFDA methods study the uncertainty of the predicted class probabilities [37, 42], or directly the distribution of the target features extracted by the source model, via *e.g.* a nearest neighbors algorithm [7, 22, 40].

The current state of the art in SFDA [22] refines pseudolabels by combining uncertainty estimation of the nearest neighbors and contrastive learning. Contrastive learning brings samples with the same pseudolabel closer in feature space, and separates those with a different pseudolabel. This facilitates finding better neighbors, *i.e.* those belonging to the same class, but it assumes that there is no bias in the class distribution during adaptation (*e.g.* majority/minority classes). Thus, even the state of the art is vulnerable to CDS.

In order to overcome the lack of source data, a recent trend in SFDA is distilling generic knowledge from powerful feature extractors to obtain state-of-the-art accuracy in their respective tasks [41, 42]. Such feature extractors are publicly available and can be used off-the-shelf without any cost. However, these methods require learning an auxiliary classifier on the generic features along with the source model end-to-end. This not only increases the number of trainable parameters, but most importantly, since the auxiliary classifier is exposed to both source and target CDS biases during training, such a logits-based SFDA pseudolabeling is not guaranteed to succeed in a CDS scenario. Contrarily, we study the applicability of generic feature extractors to the problem of SFDA with class distribution shift, which, despite being a highly realistic scenario, has not been researched yet. Our methodology tackles CDS directly at the *feature level* and not at the *logits level*, and *adapts* (not *freezes*) the source classifier. The suppl. material further details these differences.

UDA under CDS. In machine learning, a difference in the class distribution between the training and test data causes a drop in performance [2]. Furthermore, when the training and test data belong to different domains, in addition to such class distribution shift (CDS), methods also need to deal with the covariate shift of the data samples. Previous approaches to this challenging scenario include weighting samples according to their class imbalance [38] and aligning features potentially from the same class [15, 32]. As with standard UDA, these methods are only possible because of the simultaneous availability of source and target data.

SFDA under CDS. In spite of being a realistic setting (*e.g.*, the SFDA dataset DomainNet class distribution between source and target is naturally imbal-

anced), most SFDA methods do not consider CDS. Moreover, many approaches use a "diversity loss" [20–22], assuring that all classes are equally represented to avoid converging to a posterior collapse (*i.e.* when all samples are assigned the same label). This technique is inherently sensitive to severe CDS, when the number of samples for the majority and minority classes is significantly different.

SFDA under CDS has been approached solely by treating CDS as noise in the logits output by the source model [20]. Specifically, in addition to the most probable class, the pseudolabeling losses are also calculated for the second most probable class in an attempt to mitigate bias. However, this work also relaxes the setting constraints by resampling the source data so its distribution is uniform among all classes. This gives an unreasonable advantage when dealing with CDS, since, as the source's class bias is gone, the pseudolabels are only influenced by the target's bias.

Test-Time Adaptation (TTA) under CDS. In test-time adaptation (TTA), the adaptation to the label-less target data needs to be done online during inference, *i.e.* without re-training the source model. For this, previous approaches range from interpolating target data statistics [25, 35] to optimizing the parameters of a batch normalization layer using entropy minimization [34]. Only recently, TTA methods robust to class imbalance in the target data were proposed [1, 10], although they assume no imbalance (*i.e.* majority/minority bias) on the source data. The full TTA under CDS setting is approached in [27] by including in the source model a label-shift adapter module optimized on the estimated target data class imbalance.

3 Methodology

3.1 SFDA Fundamentals

Our setting consists of a source domain $\{x_s \in X_s, y_s \in Y_s\}$ and a target domain $\{x_t \in X_t, y_t \in Y_t\}$. Both share the same label space $Y_s = Y_t$ with $|Y| = L$ classes, but a covariate shift exists (Fig. 1). Before applying SFDA, a source model consisting of a pretrained (*e.g.* ImageNet) feature extractor f_s and classifier h_s is trained on the source dataset in an unknown manner, and only the source model is provided, along with the unlabeled target samples x_t.

SFDA methods estimate the probability density $p(y_t, x_t)$ for target samples by adapting the source model without accessing the labels, as $\hat{p}(y_t|x_t)_{y_t \in Y_t} = \text{softmax}(h_s(f_s(x_t)))$. Our aim is optimizing the estimated probabilities of the source model $\hat{p}$ via a classification loss (*e.g.* cross-entropy loss) using pseudolabels. Pseudolabels p' are calculated alternatively to $\hat{p}$, and represent a closer estimate to the real probability distribution p of the target labels. The basic technique for calculating the pseudolabel of a certain sample x_t^i is applying a nearest neighbors algorithm in the feature space $f_s(x_t^i) = z_s^i \in Z_s$ and recalculate the probabilities of each class c accordingly. *E.g.*, via soft-voting:

$$p_c'^i = \frac{1}{k} \sum_{n \in N} \hat{p}_c^n \tag{1}$$

where N is the set of k neighbors around z_s^i, including itself. The set of neighbors is calculated as follows. Given a distance metric d, a feature sample z_s^i, and a bank of features F of size m, let $M = \{z_s^1, ..., z_s^m\}$ be a reordering of F such that $d(z_s^1, z_s^i) \leq ... \leq d(z_s^m, z_s^i)$. Then, N is the subset of the first k samples in M, where $N = \{z_s^1, ..., z_s^k\}$ and $k < m$. During the adaptation of the source model, $\hat{p}$ and p' are calculated for the samples within a batch, and f_s and h_s are updated via backpropagation through the cross-entropy loss as:

$$\ell_{\text{ce}} = -p' \cdot \log(\hat{\text{p}}) \tag{2}$$

After adaptation is completed, the predicted classes for the target samples are $y'_t = \text{argmax}_{c \in Y}(p')$.

3.2 Effect of CDS on the Nearest Neighbors

While using nearest neighbors (NN) as a reference for pseudolabeling has proved effective to combat covariate shift [7, 22, 40], the effect of simultaneously dealing with class distribution shift has not been studied yet.

Previous works [31, 36] proved that the NN algorithm is significantly impacted by imbalanced class distributions, due to its sensitivity to the local data structure. In a binary classification problem, let W and w be the number of neighbors from the majority and minority class respectively. Mathematically, W and w are a function of (k, Q, q), where $k = (W + w)$ is the number of neighbors considered, and Q and q the distributions of the majority and minority classes in the latent space. The probability of a sample being nearest to a majority class sample is $\frac{W}{k}$, and $\frac{w}{k}$ of being nearest to a minority class sample. Under severe imbalance, $W \gg w$, so inevitably $\frac{W}{k} \gg \frac{w}{k}$, which leads to a bias towards the majority class [36]. In the labeled single-domain scenario, since (Q, q) are known, this bias can be mitigated via oversampling/undersampling techniques, or a weighted classification algorithm [31]. However, in the SFDA-CDS scenario (Q, q) are not known nor estimable. This is because, when a certain sample is misclassified as a majority class, it can be because of the original bias of the source model or because the bias within the target data. Only the hyperparameter k is controllable, where a too-small k ignores the influence of minority samples, and a too-big k considers too many majority samples [31].

Providing a full theoretical proof of the impact of CDS on SFDA is complex and depends on specific assumptions about the data distribution. Therefore, we provide an empirical proof to illustrate this phenomenon. We trained a baseline classifier (*i.e.* ImageNet-pretrained ResNet101 [12]) in the source domain of the SFDA dataset VisDA-C [29] and applied the basic pseudolabeling pipeline to adapt the source model to the target domain. This pseudolabeling is used in state-of-the-art SFDA methods (*e.g.* guided pseudolabels [22]) which, despite proposing sophisticated modules for contrastive learning and pseudolabel reweighting, still employ a vanilla NN algorithm with cosine distance d and soft-voting. Although we accessed the ground truth labels $y_t \in \{plane, ..., truck\}$ for observation, they are unavailable during the actual adaptation. Figure 2 (a)

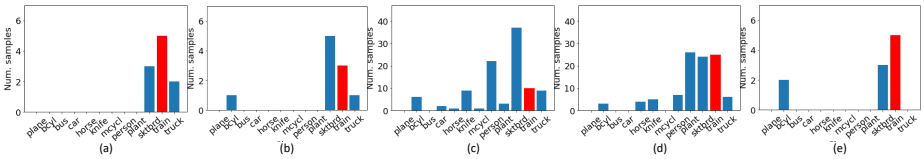

Fig. 2: Class histogram of the nearest neighbors (NN) in [22] given a target sample of the class *train* (in red) on the VisDA-C dataset. (a) Source NN on VisDA-C ($k = 10$), (b) Source NN on VisDA-C RSUT ($k = 10$), (c) Source NN on VisDA-C RSUT ($K = 100$), (d) Generic NN on VisDA-C RSUT ($K = 100$), (e) Robust NN on VisDA-C RSUT ($k = 10$).

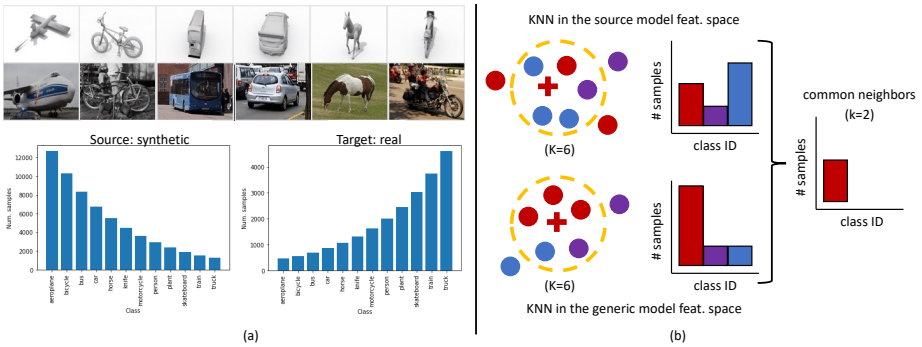

Fig. 3: (a) Class distribution in the VisDA-C RSUT dataset. (b) "Second opinion" strategy: The bias in the source nearest neighbors (NN) hinders pseudolabeling quality, so we take the intersection with the NN of a generic model to mitigate the bias.

displays a histogram of the classes of the $k = 10$ neighbors N used to calculate the pseudolabel p' of an input target sample with label y_t^i =*train*. The majority of the neighbors belong to the correct class, and consequently $y_t'^i$ = *train* is predicted. The accuracy of the adapted model is 90.0%.

Next, we use the CDS version of the dataset, VisDA-C RSUT [32]. RSUT stands for Reversely-unbalanced Source and Unbalanced Target, where source and target domains are subject to two reverse Pareto distributions (Fig. 3 (a)). For the same target sample, Fig. 2 (b) shows that the NN in the source model contain more samples from unrelated classes, including those with majority representation in the source domain (*i.e. bicycle*). As a result, the accuracy of the adapted model decreases to 83.59%. As mentioned above, this bias noise is inherent to the CDS setting, but unfeasible to estimate without either source or target labels, as the source model predictions are influenced by both. As a reasonable way to consider an unbiased estimation of the class distribution of the target domain, we propose relying on an external reference model.

3.3 Generic Features for Robust Nearest Neighbors

We propose leveraging an additional feature extractor g free of the majority/minority bias of the source data (Fig. 3 (b)). As an initial experiment, we use the same backbone as the source model before seeing the source data, *i.e.* ResNet101 pretrained on ImageNet-1K, which provides a set of features $g(x_t^i) = z_g^i \in Z_g$. As a multipurpose public dataset, ImageNet is less biased and more *generic* at least than the source data. Thus, while the feature space Z_g lacks bias in favor of generalizability, the feature space Z_s of the source model suffers from source bias but possesses domain knowledge (*e.g.* the label space). We hypothesize that combining both can lead to more "robust" nearest neighbors, and thus, to more accurate pseudolabeling. Then, the adaptation process can be regarded as a way of knowledge distillation from the generic model to the source model.

We propose replacing the set of source neighbors N with robust neighbors R that are present in both Z_s and Z_g. This requires, looking further than the original $k = 10$ samples, so we introduce an additional hyperparameter $K = 100$. We let $k \leq K \ll m$, since including the entire feature banks (*i.e.* $K = m$) is not desirable, as the generic model is not exempt of noise. Therefore, $R = \{z_s^1, ..., z_s^K\} \cap \{z_g^1, ..., z_g^K\} = \{z_r^1, ..., z_r^o\}$. To validate our hypothesis, we consider a conservative setting prioritizing the source neighbors, so that:

$$\text{If } o == k, N \leftarrow R \tag{3}$$

$$\text{If } o > k, N \leftarrow \{z_r^1, ..., z_r^k\} \tag{4}$$

$$\text{If } o < k, N \leftarrow \{z_s^1, ..., z_s^{k-o}\} \cup \{z_r^1, ..., z_r^o\} \tag{5}$$

Whereas the feature bank F needs to be updated on each training iteration [22], the bank of the generic features G only needs to be created once at the beginning.

Figure 2 (c) and (d) show the K NN in the source and generic feature spaces respectively, and (e) shows the set R of robust NN, which contains less bias noise. The final accuracy after adaptation is 86.6%, which means that pseudolabeling by employing NN of the correct class leads to higher classification accuracy. The suppl. material contains more visual examples of robust NN in this setting.

Although the state-of-the-art methods in both SFDA [20] and TTA [27] under CDS also aim to add robustness to the pseudolabeling scheme, instead of approaching the biased prediction logits, we approach the bias that stems from the nearest neighbors in the source model's feature space. This way, it was possible to understand why the performance dropped and how to improve it. Moreover, higher accuracy gains can be expected from stronger feature extractors, as shown on Sec. 4.

3.4 Validation of the Proposed Method

Hyperparameter K. We compared the performance of pseudolabeling (PL) in [22] on VisDA-C RSUT using different ranges of K for calculating robust

Table 1: Accuracy (%) on VisDA-C RSUT for different values of K.

Method	$K = 10$	$K = 50$	$K = 100$	$K = 500$
PL + ResNet101	83.96	86.34	**86.6**	83.56

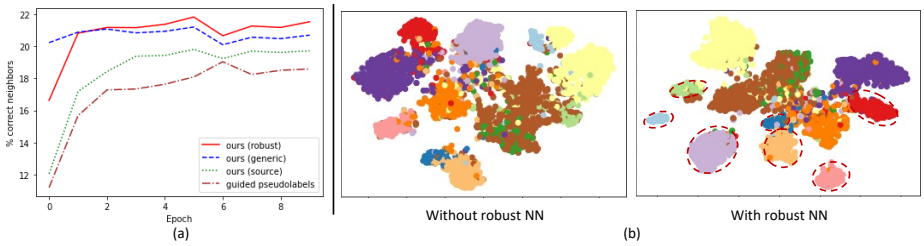

Fig. 4: (a) Percentage of correct NN used for pseudolabeling during the adaptation of the source model on VisDA RSUT. (b) Source model's feature space w/wo our method in VISDA-C RSUT. Minority classes (circled) are better clustered.

nearest neighbors. Table 1 shows that 100 is the best performing value. $K = 10$ is too small to find common neighbors, so the accuracy is almost that of the original method. Similarly, the range of $K = 500$ is too large and unrelated common neighbors are included, resulting in a performance drop.

Source, Generic and Robust Neighbors. Figure 4 (a) compares the percentage of correct neighbors, *i.e.* those belonging to the same class as y_t^i, as the source model is adapted. While the generic neighbors do not improve during adaptation, the source neighbors improve getting closer to the performance of the generic neighbors. This results in robust neighbors that surpass the performance of both the generic model and the original guided pseudolabels in [22].

Feature Space Visualization. Figure 4 (b) is a visualization via tSNE [24] of the feature space of our SFDA pipeline in which the NN are calculated. In particular, the minority classes appear better clustered when applying our method, which results in less biased pseudolabels.

4 Experimental Results

4.1 Experimental Settings

Metrics. For a fair comparison with the related work, we use the setting in [20]. We calculate the class-wise mean accuracy, as it reflects better the classification performance when all classes (majorities and minorities) should be properly classified. Otherwise, the model may display high accuracy just by classifying properly a subset of the majority classes only [8].

Datasets. We evaluate our method in three standard SFDA datasets. **VisDA-C** [29] contains twelve object classes in two domains: *real* and *synthetic*. **Office-Home** [33] contains sixty-five object classes in four domains, from which three are used as a benchmark in the relevant works [20, 27]: *clipart*, *product* and *real*. Finally, **DomainNet** [28] contains forty object classes in four domains: *clipart*, *painting*, *real* and *sketch*. Following the related work, we use the RSUT version of VisDA-C and Office-Home (see Fig. 3 (a)), created by [32] in order to apply CDS. On the other hand, the domains in DomainNet are naturally class distribution shifted, so no class resampling is applied, but a subset of the dataset is used instead [32]. As a result, each dataset represents a different type of CDS. The suppl. material details their respective class distributions, and the splits used for training the source model and adapting it to the target.

Source/Adapted Model. In order to evaluate our proposed robust nearest neighbors, we introduce them into the SFDA state-of-the-art, guided pseudolabels (**PL guided**) [22]. This method calculates pseudolabels for the test data via nearest neighbors, weights them according to the uncertainty of their predictions, and refines the feature space via contrastive learning. Additionally, as a reference, we provide the results on the pseudolabeling baseline (**PL base**), *i.e.*, no weighting nor contrastive learning. Following this benchmark's standards, the architecture used for the source models is: ResNet101 for VisDA-C and ResNet50 for Office-Home and DomainNet, pretrained in ImageNet-1K. The source model training and adaptation regimes are those of the original setting [22]; stochastic gradient descent is run during 100 epochs and batch size 64 in VisDA-C (RSUT), and 200 epoch and batch size 128 in Office-Home (RSUT) and DomainNet (subset). Likewise, the source data is learned with the standard cross-entropy loss and label-smoothing.

Generic Models. We chose a range of standard convolutional neural networks (CNNs) and transformers in order to study the performance variations when using different backbones. We use the same pretrained **ResNet** [12] backbones as the source/adapted model for each dataset. In addition, we employ the **Vision Transformer** (ViT-B/32) [9] and the **Swin Transformer** (Swin-B/4) [23] pretrained in ImageNet-21k. This list is not exhaustive and it demonstrates the ability of our method to successfully leverage different types of feature spaces. The suppl. material contains more details about the architectures and their weights. Robust nearest neighbors are calculated with hyperparameters $k = 10$ and $K = 100$.

4.2 Source-Free Domain Adaptation under CDS

Tables 2, 3 and 4 show the classification accuracy after adaptation on the target domain. The results are divided in seven blocks from top to bottom as follows: (1) the source model without adaptation, (2) UDA methods, (3) UDA under CDS methods, (4) SFDA methods, (5) SFDA under CDS method, (6) SFDA via basic

Table 2: Class-wise average accuracy of SFDA on VisDA-C (RSUT). The source domain are *synthetic* images and the target domain are *real* images.

Method	plane	bcyl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
Source model	64.49	16.67	46.86	78.81	63.28	7.49	75.86	17.55	65.02	12.88	70.48	3.18	43.55
PADA [3]	95.65	52.86	**87.74**	66.67	84.96	1.34	64.50	18.06	31.05	1.80	0.10	0.01	42.06
MCD [30]	63.04	41.43	83.96	67.28	86.59	93.85	85.59	76.27	84.09	11.26	5.04	2.95	58.45
BSP [5]	**100.0**	57.14	68.87	56.79	83.74	26.74	78.73	16.20	63.70	1.85	0.10	0.10	46.15
COAL [32]	86.96	40.0	71.70	79.63	89.43	22.46	86.47	46.18	82.95	34.99	72.76	7.04	60.05
MDD (Implicit) [15]	82.61	81.43	83.96	62.96	86.59	88.50	73.29	76.04	85.76	50.35	69.50	23.40	72.03
SHOT [21]	56.52	14.29	78.30	50.0	96.07	62.92	89.81	64.35	84.43	50.97	66.97	24.51	61.59
ISFDA [20]	86.95	64.29	82.71	60.7	95.53	96.17	84.94	78.97	90.01	71.35	80.97	27.63	76.69
PL base	84.95	90.0	86.11	80.68	97.54	88.69	89.1	81.31	92.73	84.81	63.07	33.15	81.01
+ Ours (ResNet101)	91.4	83.0	83.33	75.0	98.03	82.19	92.83	85.35	95.35	90.63	87.33	41.76	83.85
+ Ours (ViT-B)	90.32	**93.0**	81.94	81.25	97.04	94.16	83.48	81.06	94.34	87.24	82.61	40.13	83.88
+ Ours (Swin-B)	93.55	**93.0**	85.42	82.95	97.04	93.8	93.15	85.1	91.72	91.6	89.49	42.86	86.64
PL guided [22]	93.55	82.0	81.25	75.0	97.54	93.8	95.64	84.09	95.76	80.29	82.34	53.87	83.59
+ Ours (ResNet101)	94.62	84.0	72.92	77.84	96.06	91.24	95.33	**87.63**	95.96	**94.99**	93.66	**54.96**	86.6
+ Ours (ViT-B)	89.25	87.0	80.56	**84.09**	94.58	96.35	95.02	82.83	**96.36**	93.05	91.24	50.27	86.72
+ Ours (Swin-B)	93.55	**93.0**	79.86	82.39	**98.52**	**96.71**	**96.88**	86.36	**96.36**	94.51	**94.88**	53.11	**88.84**

Table 3: Class-wise average accuracy of SFDA on Office-Home (RSUT). The domains are *clipart* (C), *product* (P) and *real* (R).

Method	C→P	C→R	P→C	P→R	R→C	R→P	Avg.
Source model	52.27	53.31	35.84	67.31	38.35	69.77	52.81
PADA [3]	38.34	40.71	26.76	57.09	32.28	60.77	42.66
MCD [30]	39.01	44.47	29.99	62.95	33.17	66.03	45.94
BSP [5]	30.36	32.59	20.05	66.19	23.82	72.80	40.97
COAL [32]	57.33	59.22	40.61	73.26	42.58	73.65	58.40
MDD (I) [15]	63.15	61.15	45.38	74.21	50.04	76.08	61.67
SHOT [21]	65.07	63.55	46.1	74.81	50.16	77.37	62.84
ISFDA [20]	66.84	67.28	**50.33**	76.78	**53.69**	77.25	65.36
PL base	63.70	61.67	33.69	71.94	35.66	74.27	56.82
+ ResNet50	62.54	61.90	29.86	70.27	33.73	74.53	55.47
+ ViT-B	66.97	63.03	35.74	71.08	38.12	77.31	58.71
+ Swin-B	70.51	70.40	41.95	79.94	43.86	81.18	64.64
PL guided [22]	66.81	68.11	37.99	76.62	40.22	76.53	61.05
+ ResNet50	66.99	66.72	34.05	75.32	37.65	77.27	59.67
+ ViT-B	70.49	67.86	39.54	75.14	40.90	79.93	62.31
+ Swin-B	**75.50**	**76.59**	46.84	**82.85**	48.21	**84.26**	**69.04**

pseudolabeling with our proposal, and (7) SFDA via guided pseudolabeling with our proposal. The best performance is **bolded** and the second-best is underlined.

Similar patterns can be observed for all three datasets. Our method provides the best results when leveraging a strong generic feature extractor, outperforming the previous work in SFDA under CDS [20] without needing to impose a uniform distribution on the source data. The reason is that, although approaching bias reduction at the logits level can correct the source model's predictions partially, the performance improvement is limited compared to reducing bias at the nearest neighbors level. As a result, by empirically exploring the essence of the problem, we successfully provided a solution to the most challenging SFDA-CDS setting for the first time, via a simple method. In particular, Swin-B provides the best results in two of the three datasets. Unlike ViT-B, Swin-B model extracts features at different local and global levels, which makes it more robust against covariate shifts [42]. In VisDA-C, the difference in accuracy between the basic pseudolabeling and our method indicates that the robust nearest neighbors pro-

Table 4: Class-wise average accuracy of SFDA on DomainNet (subset). The domains are *clipart* (C), *painting* (P), *real* (R) and *sketch* (S).

Method	C→P	C→R	C→S	P→C	P→R	P→S	R→C	R→P	R→S	S→C	S→P	S→R	Avg.
Source model	53.55	76.70	53.06	55.55	84.39	60.19	58.84	67.89	53.08	54.60	57.78	74.62	62.52
PADA [3]	53.09	74.69	52.86	59.33	79.84	57.87	65.91	67.13	58.43	66.97	61.08	76.52	64.48
MCD [30]	56.61	79.78	53.66	58.31	83.38	60.98	61.97	69.33	56.26	56.27	66.78	81.74	65.42
BSP [5]	67.52	86.50	70.90	70.33	86.83	68.75	67.29	73.47	69.31	72.40	71.47	84.34	74.09
COAL [32]	69.98	89.63	71.29	68.01	89.81	70.49	73.58	75.37	70.50	73.21	70.53	87.97	75.89
MDD (Implicit) [15]	70.59	88.50	70.44	75.71	88.37	71.65	78.54	75.09	69.43	77.97	72.41	89.35	77.33
SHOT [21]	72.5	87.90	73.80	75.29	89.90	74.79	77.17	75.82	71.43	77.81	73.08	88.23	78.14
ISFDA [20]	75.11	90.09	74.78	76.70	89.57	76.07	81.52	77.29	73.55	79.70	73.13	87.55	79.58
PL base	76.32	92.13	75.53	76.92	91.42	74.90	73.44	78.55	72.93	76.14	73.66	91.86	79.48
+ Ours (ResNet50)	67.93	91.51	60.22	55.49	90.46	64.82	60.46	70.90	60.10	63.71	68.13	89.65	70.28
+ Ours (ViT-B)	75.96	**94.88**	79.19	**83.87**	**94.58**	**79.08**	80.83	78.16	78.25	79.11	73.70	92.47	82.51
+ Ours (Swin-B)	77.03	94.52	69.08	75.64	93.58	71.13	75.48	78.24	69.27	76.61	74.48	92.39	78.95
PL guided [22]	77.56	91.27	75.79	76.44	90.24	76.11	75.75	79.50	74.63	76.22	77.05	90.96	80.12
+ Ours (ResNet50)	71.25	90.80	65.69	62.18	89.42	67.35	61.19	72.50	64.30	65.62	72.09	90.46	72.74
+ Ours (ViT-B)	79.71	93.91	**80.26**	83.56	92.83	79.06	**82.68**	80.77	**79.27**	**82.80**	**78.44**	**93.52**	**83.9**
+ Ours (Swin-B)	**79.97**	93.69	74.40	77.49	92.51	75.09	79.43	**81.01**	72.48	79.42	77.86	93.43	81.4

Table 5: Class-wise average accuracy of TTA on VisDA-C (RSUT). The source domain are *synthetic* images and the target domain are *real* images.

Method	Avg.
Source model	51.45
ONDA [25]	50.68
LAME [1]	50.72
CoTTA [35]	49.88
NOTE [10]	49.37
TENT [34]	48.68
+ Label shift adapter [27]	**72.97**
Pseudolabel	47.12
+ Ours (ResNet101)	50.07
+ Ours (ViT-B)	49.60
+ Ours (Swin-B)	52.49

vide the highest boost on the minority "tail" classes of the source dataset (esp. *skateboard*, *train*, *truck*). This is coherent with our hypothesis that pseudolabeling is affected significantly by the majority/minority bias.

Regarding ResNet, while it is outperformed by the other generic models, it can provide comparable performance when the target domain are real images. In particular, given the similarity between the target domain in VisDA-C and ImageNet, ResNet's robust neighbors are as effective as the stronger generic models.

Moreover, our method is capable of improving basic pseudolabeling under the RSUT setting, without training additional modules and objective functions. When using strong feature extractors, the simpler base pseudolabeling can surpass the performance of the more complex guided pseudolabels method [22] (*e.g.* `PL base + Swin-B` vs. `PL guided` in Tab. 2 and 3).

4.3 Test-Time Adaptation under CDS

The nature of our method also allows it to improve the adaptation accuracy *on-the-fly* without retraining the source model, which suits the time-test adaptation (TTA) setting. For this, we run our method in inference mode, relying only on

Table 6: Class-wise average accuracy of TTA on Office-Home (RSUT). The domains are *clipart* (C), *product* (P) and *real* (R).

Method	C→P	C→R	P→C	P→R	R→C	R→P	Avg.
Source model	45.39	44.53	32.94	64.33	40.22	68.92	49.39
ONDA [25]	44.84	47.57	35.20	62.09	40.61	63.83	49.02
LAME [1]	41.68	42.27	32.40	63.57	37.92	66.94	47.46
CoTTA [35]	44.46	48.19	35.63	62.34	40.73	62.20	48.92
NOTE [10]	43.02	42.38	38.64	61.69	**41.40**	64.33	48.58
TENT [34]	49.60	49.51	**38.96**	63.08	41.25	64.52	51.15
+Adapter [27]	49.60	53.13	37.81	66.45	41.35	68.35	52.78
Pseudolabel	55.48	57.23	26.72	69.92	31.12	73.58	52.34
+Ours (ResNet50)	58.96	61.6	24.09	69.78	29.57	72.97	52.83
+Ours (ViT-B)	60.26	59.97	26.35	69.25	31.03	76.83	53.95
+Ours (Swin-B)	**65.23**	**67.62**	29.32	**79.87**	37.74	**81.18**	**60.16**

Table 7: Class-wise average accuracy of TTA on DomainNet (subset). The domains are *clipart* (C), *painting* (P), *real* (R) and *sketch* (S).

Method	C→P	C→R	C→S	P→C	P→R	P→S	R→C	R→P	R→S	S→C	S→P	S→R	Avg.
Source model	52.73	74.87	52.15	58.42	81.22	61.82	66.03	69.58	55.31	63.92	59.68	75.43	64.26
ONDA [25]	56.82	78.32	54.81	63.99	81.79	61.86	67.14	70.09	58.11	71.60	69.34	80.77	67.89
LAME [1]	49.20	72.45	48.69	57.81	80.09	60.85	65.25	68.19	53.97	61.0	55.66	73.25	62.20
CoTTA [35]	56.88	77.33	54.18	63.69	81.31	60.26	67.44	70.07	57.14	71.69	68.85	80.56	67.45
NOTE [10]	55.38	74.15	57.98	65.59	81.66	64.65	71.29	73.32	63.28	72.28	68.31	80.25	69.01
TENT [34]	63.26	77.10	59.76	66.69	80.02	64.32	**71.88**	74.34	62.25	**73.13**	**72.64**	78.73	70.34
+ Label shift adapter [27]	63.26	81.11	60.39	**67.38**	82.99	67.23	**71.88**	**74.83**	**64.40**	71.88	71.56	82.67	71.63
Pseudolabel	59.45	87.48	57.27	56.31	88.68	65.57	62.28	69.66	55.23	59.42	60.24	83.15	67.06
+ Ours (ResNet50)	60.39	89.94	50.13	48.41	88.70	56.68	51.28	66.48	47.60	52.19	59.38	84.89	63.01
+ Ours (ViT-B)	**66.15**	**92.6**	**62.81**	65.9	90.97	**71.36**	71.0	73.28	63.61	70.47	63.01	87.59	**73.23**
+ Ours (Swin-B)	65.78	92.41	58.47	61.06	**91.10**	66.34	66.1	72.03	56.84	65.6	63.57	**87.77**	70.59

the predicted pseudolabels of the robust nearest neighbors. Note that, since no learning is involved, the accuracy results are the same for both the base and guided pseudolabeling settings of our method.

Tables 5, 6 and 7 show the classification accuracy on the target domain. The results are divided in four blocks from top to bottom as follows: (1) the source model without adaptation, (2) TTA methods with partial support to CDS, (3) TTA method with full support to CDS, and (4) TTA via pseudolabeling with our proposal. Our method outperforms all TTA methods with the single exception of the state of the art [27] on VisDA-C. However, unlike [27], our method does not require optimizing an adapter module for CDS.

5 Discussion and Conclusions

Cost of the Generic Model and Robust NN. Our results are obtained employing pretrained models publicly available with fixed parameters, so applying our method incurs no extra training cost. Regarding the computational complexity of calculating our robust nearest neighbors (NN), given n samples, our method calculates the K NN in the d dimensional generic feature space, which has a complexity of $O(K \cdot n \cdot d)$. Then, it finds the common samples with the source k NN, which has a complexity of $O(n \cdot log(n))$.

Table 8: Mean class-accuracy (%) on VisDA-C RSUT for different degrees of CDS. The Δ indicates the performance drop between the strongest CDS vs. the weakest.

Method	CDS	CDS↑	CDS↑↑	Δ
PL guided [22]	83.59	80.24	77.79	-5.8
+ Ours (ResNet101)	86.60	82.60	80.46	-6.14
+ Ours (ViT-B)	86.72	84.84	83.88	-2.84
+ Ours (Swin-B)	88.84	87.57	85.52	-3.32

Effect of Different Levels of CDS. All our experiments considered the CDS configuration of the public datasets used in the comparison works. In addition, we included a study on different levels of CDS in Tab. 8. The stronger the generic model is, the more resilience to CDS our method provides. The suppl. material includes the same study for a subset of the baselines and comparison works.

Selection of the Generic Model. While our method is not tied to any specific architecture, our results suggest that as long as the generic features are stronger than the source features (*i.e.* more sophisticated architectures, more parameters), the adaptation will improve. Using the same source architecture (*i.e.*, ResNet) still produces reasonable results for target domains similar to ImageNet, but does not guarantee adaptation improvement. This effect is not abnormal; [42] also shows in Tab. 3 and 5 how ResNet decreases accuracy, while the stronger architecture always improves accuracy. More details in the suppl. material.

Extension to Multiple Generic Models. Our method allows for a straightforward way for adding new "robust opinions" by simply finding common nearest neighbors of the target samples in additional models' feature space. However, the more models added, the harder finding common neighbors would be, and may require adjusting hyperparameter K. Such study is left for future work.

5.1 Conclusions

This paper studied the effect of class distribution shift (CDS) in the task of source free domain adaptation (SFDA). Instead of proposing additional modules and objective functions to improve the SFDA's pseudolabeling process, we study the weakness of the nearest neighbors algorithm used in many previous works. We proved that, by adding robustness to the nearest neighbors via an external feature extractor, the accuracy of the subsequent adaptation improves, outperforming previous methods in both SFDA and test-time adaptation (TTA) tasks under CDS. This study stays within the scope of standard-size convolutional networks and transformers, but we believe that distillation from powerful generic models to custom architectures will gain relevancy with the widespread of large foundation models.

References

1. Boudiaf, M., Mueller, R., Ben Ayed, I., Bertinetto, L.: Parameter-free online test-time adaptation. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 8344–8353 (2022)
2. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural networks **106**, 249–259 (2018)
3. Cao, Z., Ma, L., Long, M., Wang, J.: Partial adversarial domain adaptation. In: Proc. European conference on computer vision. pp. 135–150 (2018)
4. Chen, W., Yu, Z., De Mello, S., Liu, S., Alvarez, J.M., Wang, Z., Anandkumar, A.: Contrastive syn-to-real generalization. In: Proc. International Conference on Learning Representations. pp. 1–12 (2021)
5. Chen, X., Wang, S., Long, M., Wang, J.: Transferability vs. Discriminability: Batch spectral penalization for adversarial domain adaptation. In: Proc. International Conference on Machine Learning. pp. 1081–1090 (2019)
6. Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., Tian, Q.: Gradually vanishing bridge for adversarial domain adaptation. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 12455–12464 (2020)
7. Dong, J., Fang, Z., Liu, A., Sun, G., Liu, T.: Confident anchor-induced multi-source free domain adaptation. Advances in Neural Information Processing Systems **34**, 2848–2860 (2021)
8. Dong, Q., Gong, S., Zhu, X.: Imbalanced deep learning by minority class incremental rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(6), 1367–1381 (2018)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
10. Gong, T., Jeong, J., Kim, T., Kim, Y., Shin, J., Lee, S.J.: NOTE: Robust continual test-time adaptation against temporal correlation. Proc. Advances in Neural Information Processing Systems **35**, 27253–27266 (2022)
11. Gu, X., Sun, J., Xu, Z.: Spherical space domain adaptation with robust pseudo-label loss. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 9101–9110 (2020)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
13. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
14. Hu, L., Kan, M., Shan, S., Chen, X.: Unsupervised domain adaptation with hierarchical gradient synchronization. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 4043–4052 (2020)
15. Jiang, X., Lao, Q., Matwin, S., Havaei, M.: Implicit class-conditioned domain alignment for unsupervised domain adaptation. In: Proc. International Conference on Machine Learning. pp. 4816–4827 (2020)
16. Kundu, J.N., Kulkarni, A.R., Bhambri, S., Mehta, D., Kulkarni, S.A., Jampani, V., Radhakrishnan, V.B.: Balancing discriminability and transferability for source-free domain adaptation. In: Proc. International Conference on Machine Learning. pp. 11710–11728 (2022)

17. Kundu, J.N., Venkat, N., Babu, R.V., et al.: Universal source-free domain adaptation. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 4544–4553 (2020)
18. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Proc. International Conference on Machine Learning Workshops. vol. 3-2, p. 896 (2013)
19. Li, S., Xie, M., Gong, K., Liu, C.H., Wang, Y., Li, W.: Transferable semantic augmentation for domain adaptation. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 11516–11525 (2021)
20. Li, X., Li, J., Zhu, L., Wang, G., Huang, Z.: Imbalanced source-free domain adaptation. In: Proc. ACM International Conference on Multimedia. pp. 3330–3339 (2021)
21. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In: Proc. International Conference on Machine Learning. pp. 6028–6039 (2020)
22. Litrico, M., Del Bue, A., Morerio, P.: Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 7640–7650 (2023)
23. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical vision transformer using shifted windows. In: Proc. International Conference on Computer Vision. pp. 10012–10022 (2021)
24. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of machine learning research **9**(11) (2008)
25. Mancini, M., Karaoguz, H., Ricci, E., Jensfelt, P., Caputo, B.: Kitting in the wild through online domain adaptation. In: Proc. International Conference on Intelligent Robots and Systems. pp. 1103–1109 (2018)
26. Na, J., Jung, H., Chang, H.J., Hwang, W.: FixBi: Bridging domain spaces for unsupervised domain adaptation. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 1094–1103 (2021)
27. Park, S., Yang, S., Choo, J., Yun, S.: Label shift adapter for test-time adaptation under covariate and label shifts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16421–16431 (2023)
28. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proc. International Conference on Computer Vision. pp. 1406–1415 (2019)
29. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: VisDA: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924 (2017)
30. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 3723–3732 (2018)
31. Shi, Z.: Improving k-nearest neighbors algorithm for imbalanced data classification. IOP Conference Series: Materials Science and Engineering **719**(1), 012072 (2020)
32. Tan, S., Peng, X., Saenko, K.: Class-imbalanced domain adaptation: An empirical odyssey. In: Proc. European Conference on Computer Vision Workshops. pp. 585–602 (2020)
33. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 5018–5027 (2017)
34. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: TENT: Fully test-time adaptation by entropy minimization. In: Proc. International Conference on Learning Representations. pp. 1–15 (2021)

35. Wang, Q., Fink, O., Van Gool, L., Dai, D.: Continual test-time domain adaptation. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 7201–7211 (2022)
36. Weiss, G.M., Provost, F.: The effect of class distribution on classifier learning: an empirical study. Tech. rep., Rutgers University (2001)
37. Xia, H., Zhao, H., Ding, Z.: Adaptive adversarial network for source-free domain adaptation. In: Proc. International Conference on Computer Vision. pp. 9010–9019 (2021)
38. Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., Zuo, W.: Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In: Proc. Conference on Computer Vision and Pattern Recognition. pp. 2272–2281 (2017)
39. Yang, S., Wang, Y., Van De Weijer, J., Herranz, L., Jui, S.: Generalized source-free domain adaptation. In: Proc. International Conference on Computer Vision. pp. 8978–8987 (2021)
40. Yang, S., van de Weijer, J., Herranz, L., Jui, S., et al.: Exploiting the intrinsic neighborhood structure for source-free domain adaptation. Advances in Neural Information Processing Systems **34**, 29393–29405 (2021)
41. Zara, G., Conti, A., Roy, S., Lathuilière, S., Rota, P., Ricci, E.: The unreasonable effectiveness of large language-vision models for source-free video domain adaptation. In: Proc. International Conference on Computer Vision. pp. 10307–10317 (2023)
42. Zhang, W., Shen, L., Foo, C.S.: Rethinking the role of pre-trained networks in source-free domain adaptation. In: Proc. International Conference on Computer Vision. pp. 18841–18851 (2023)