– APPENDIX –
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

1 Dataset Details

This section details the class distribution of each dataset under class distribution shift (CDS), and the splits used for training the source model and adapting it to the target.

1.1 VisDA-C RSUT

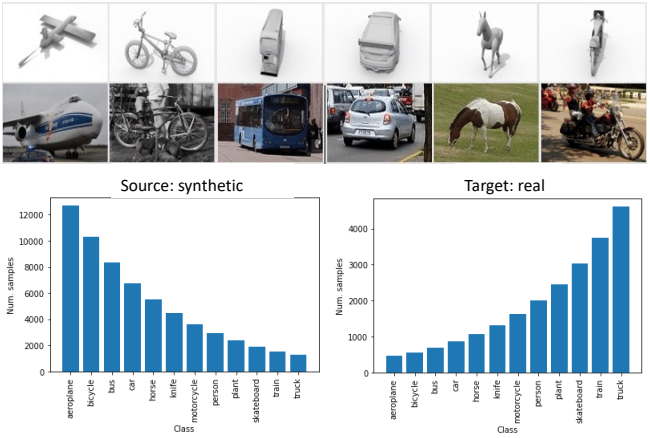

Fig. 1: Class distribution in the VisDA-C RSUT dataset.

As shown in the main text, Fig. 1 details the class distribution of the version of VisDA-C under class distribution shift (CDS), VisDA-C RSUT (Reversely-unbalanced Source and Unbalanced Target). It follows a Pareto distribution,

which represents the worst case scenario for a class distribution shift between the source and the target domains. The source domain consists of computer generated *synthetic* images, while the target domain are *real* world photos. Although there are two domains, only the `synthetic` → `real` setting is considered.

This dataset only provides an RS and a UT split for each domain. When training the source model, the RS split of the source domain is used, from which a 20% of the data is used for validation, as in [3]. Similarly, model adaptation is conducted on the UT split of the target domain, and accuracy is calculated on a 20% of the data. This 20% subsampling follows the same distribution as the original data.

1.2 Office-Home RSUT

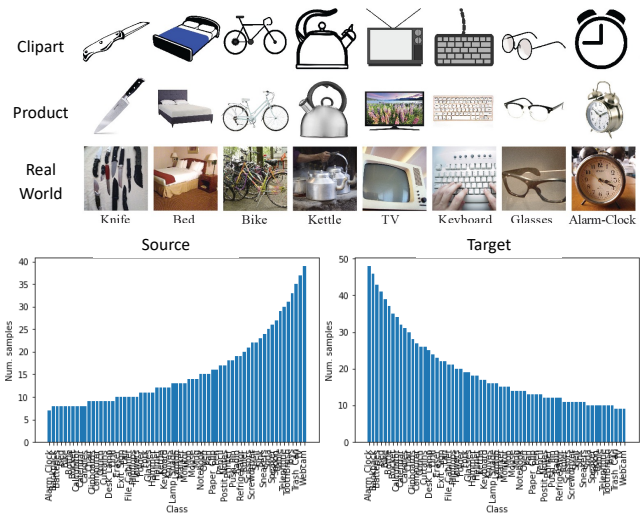

Fig. 2: Class distribution in the Office-Home RSUT dataset.

The CDS version of the Office-Home dataset also follows a Pareto distribution, although this time is inverted with respect to VisDA-C RSUT. Object classes are contained in three different domains (*Clipart*, *Product*, *RealWorld*), which results in six different adaptation combinations.

As in VisDA-C RSUT, only an RS and a UT split for each domain is provided. Thus, the source model is trained on the RS split of the source domain (20% of the data for validation [3]) and adapted on the UT split of the target domain. The adaptation accuracy is calculated on a 20% of the UT split of the target domain. Whatever domain is used for source or target, it will follow the corresponding distribution shown in the figure.

1.3 DomainNet Subset

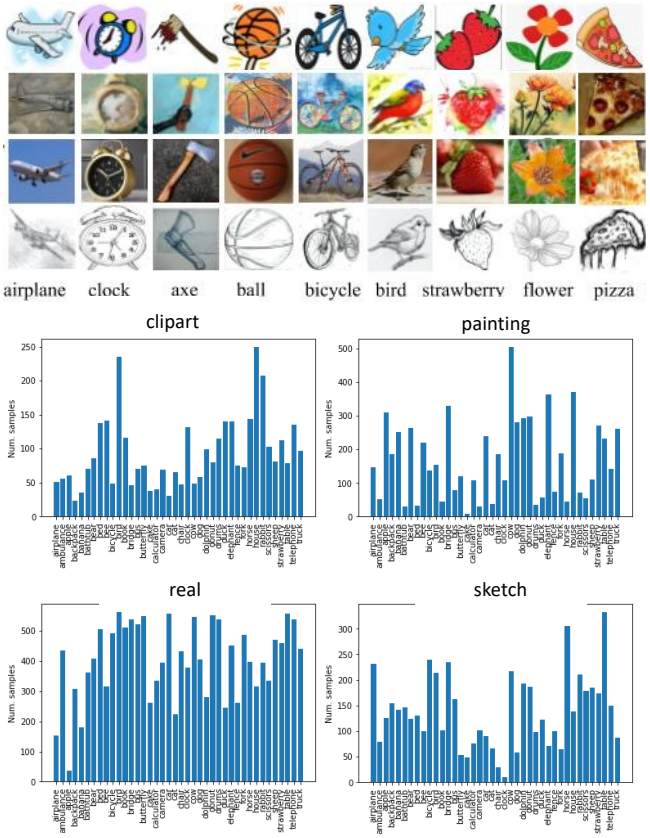

Fig. 3: Class distribution in the subset of DomainNet dataset.

The subset of DomainNet created by [4], also called DomainNet mini by [1], contains four domains (*clipart*, *painting*, *real*, *sketch*), which results in sixteen combinations for adaptation. However, since the class distributions of each domain are inherently imbalanced, no RSUT subsampling is performed.

This dataset provides a train, validation and test splits for each domain, in which train and test follow the same class distribution as the figure, and validation follows a uniform distribution. Thus, the source model is trained on the train split and validated on the test split of the source domain. Then, during adaptation, the train split of the target domain is used, and accuracy is calculated on the test split.

2 Additional Examples of Robust Nearest Neighbors

Figure 4 visualizes additional examples of the effectiveness our proposed method to help removing bias noise in the nearest neighbors (NN) algorithm under class distribution shift (CDS). Specifically, the results are obtained with a ResNet-101 source model trained in the VisDA-C RSUT dataset [4] and adapted via guided pseudolabels [3]. Similarly to Fig. 2 in the main text (Sec. 3), each row displays the neighbors of a target sample of a given class label (in red). Plot (a) shows that the $k = 10$ NN in the source model contain more samples from unrelated classes, esp. those with majority representation in the source domain. Plots (b) and (c) show the $K = 100$ NN in the source and generic feature spaces respectively, and (d) shows the $k = 10$ robust NN (*i.e.* common to both the source and generic models). As the robust NN contain less bias noise than the vanilla nearest neighbors (a), the subsequent pseudolabeling includes more samples of the correct class, which leads to higher classification accuracy.

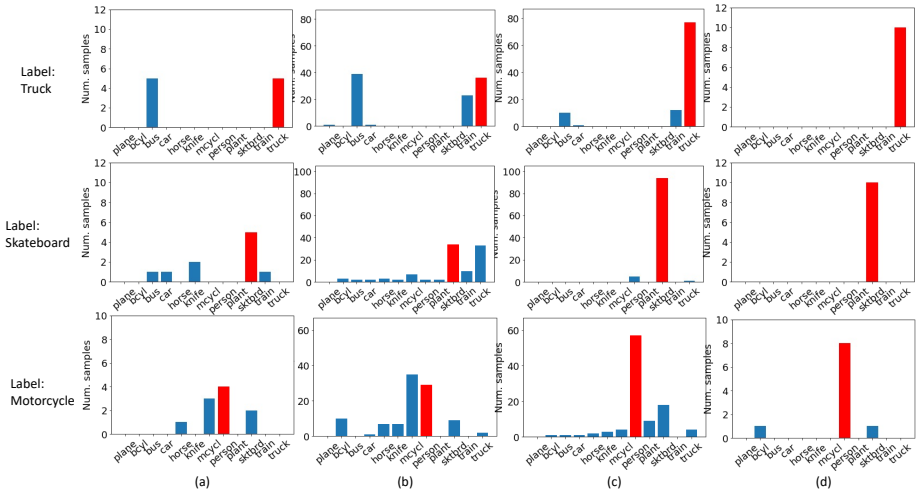

Fig. 4: Class histogram of the nearest neighbors (NN) in [3] given a target sample (class in red) on the VisDA-C RSUT dataset. (a) Source NN ($k = 10$), (b) Source NN ($K = 100$), (c) Generic NN ($K = 100$), (d) Robust NN ($k = 10$).

3 Generic Model Details

We employ three different backbones in our experiments.

ResNet-101 is used in VisDA-C, and ResNet-50 is used in Office-Home and DomainNet. We use the pretrained models available in `torchvision`, which contain the training weights of ImageNet-1K.

For Vision Transformer, we employ the image encoder provided by `huggingface transformers`. Specifically, we set a ViT-B/32 architecture pretrained on the model weights of `openai/clip-vit-base-patch32`.

Finally, Swin Transformer uses also the base model available in `huggingface transformers`, which is pretrained on ImageNet-21K at resolution 224×224 (`microsoft/swin-base-patch4-window7-224-in22k`).

4 Effect of Different Levels of CDS (Additional Results)

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. 1. The stronger the generic model is, the more resilience to CDS our method provides. The previous work in SFDA-CDS [1], although as resilient as our method, cannot provide high accuracy. 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.

Table 1: 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
SHOT [2]	61.59	67.44	65.03	-3.44
ISFDA [1]	76.69	75.25	73.85	<u>-2.84</u>
PL base	81.01	79.22	76.86	-4.15
+ Ours (ResNet101)	83.85	80.23	77.34	-6.51
+ Ours (ViT-B)	83.88	81.32	80.65	-3.23
+ Ours (Swin-B)	86.64	84.98	81.66	-4.98
PL guided [3]	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	83.95	**-2.77**
+ Ours (Swin-B)	88.84	87.57	85.52	-3.32

5 Selection of the generic model

As with every deep learning research, accuracy naturally varies with the data and architecture. As a general result, our method performs better on the *real-world* target domain, in which even the performance of the ResNet backbone is very close to that of the other stronger generic models. The reason is that the training data of the generic models is mostly based on photos and real images. Despite this variations, understanding which generic architecture will provide better adaptation results on the target domain is intuitive. Given a set

of architectures and few shot target labels for evaluation (*e.g.*, 10%), a simple probe on their features allows picking up the best for each target (Tab. 2).

Table 2: Probing accuracy (Avg.) on each CDS target data by adding the unadapted source classifier to each generic architecture.

Dataset	ResNet	ViT-B	Swin-B
VISDA-C RSUT	23.13	25.81	**27.58**
OfficeHome RSUT	8.28	12.05	**15.09**
DomainNet	10.97	**22.34**	21.61

Instead of using a generic model, one could leverage the label-less target data to self-supervisedly pretrain or finetune a model so its feature space reflects better the target distribution. However, in the case of small datasets, self-supervised methods did not prove helpful to improve the performance of the generic model (more so under the CDS constraints).

6 Differences with related SFDA methodologies that employ an auxiliary model

Our paper proves that the nearest neighbors algorithm used in source-free domain adaptation (SFDA) is weak to imbalance, and that it can be fixed via features free of the source data bias. These features are indeed extracted from an auxiliary (*i.e.*, generic) model, and, while [5] also employs an auxiliary model, this is the only similarity. While their aim is extracting target features z_* that are more discriminative than those of the source model, ours is extracting target features free of the source class distribution bias. Thus, their pipeline is not designed for class distribution shift (CDS) settings. First, [5] assumes that the class distribution on source and target is the same, and thus, they can afford freezing the source classifier and focus on improving data representation. However, intuitively, freezing a biased source classifier won't facilitate the adaptation to a target domain with a different class distribution bias, as the classifier is directly responsible for making the final class predictions. In [5], k_a and q_* denote the frozen source and auxiliary model's classifier, whose input target features are z_a and z_*, and output probabilities are p_a and p_* respectively. SFDA-CDS introduces majority-minority bias in both the source (*i.e.*, $b_s(z_a), b_s(p_a)$) and the auxiliary branch (*i.e.*, $b_t(z_*), b_t(p_*)$). Their auxiliary branch is used to optimize the source model via *MatchOrConf* pseudolabels $\tilde{y}$ (Eq. 5 in [5]), in which $b_s(p_a)$ would be used for weighting $b_t(p_*)$ (Eq. 2 in [5]), which would result in pseudolabels biased by both branches $b_s(b_t(\tilde{y}))$. Even in the improbable case in which those biased pseudolabels would allow adapting the source model removing b_s completely from features (*i.e.*, $b_s(z_a) \rightarrow b_t(z_a)$), a frozen k_a will keep providing biased probabilities $k_a(b_t(z_a)) = b_s(b_t(p_a))$. Thus, the training would consist of

a push-and-pull between b_s and b_t, and it is unclear if the model would converge to an optimal solution. While proving this is out of the scope of our work, this intuition lead our methodology to tackle CDS directly at the *feature level* and not at the *pseudolabeling (*i.e.*, logits) level*, and *adapt* (not *freeze*) the source classifier. Thus, the *MatchOrConf* pseudolabeling method in [5] is conceptually and algorithmically unrelated to our *robust nearest neighbors*.

References

1. 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)
2. 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)
3. 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)
4. 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)
5. 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)