Consistency Regularization for Generalizable Source-free Domain Adaptation Supplementary Materials

1. Results on DomainNet testing set

We further evaluate the generalization ability of existing SFDA methods on the DomainNet dataset, as shown in Table 1. The train-test split is the same as [3], and testing set results of previous methods are reproduced with their official codes. Our consistency regularization based paradigm shows high accuracy both in training and testing data, avoiding a giant performance degradation on testing data like what appeared in previous works.

Table 1. Accuracy (%) and its drop between target training and testing data on DomainNet dataset. Best results are highlighted.

Method	SF	Target	Test	Drop
SHOT [1]	✓	65.1	62.7	2.4(3.7%↓)
G-SFDA [5]	✓	63.3	58.1	5.2(8.2%↓)
NRC [4]	✓	64.7	60.6	4.1(6.3%↓)
DaC [6]	✓	68.3	66.6	1.7(2.5%↓)
Ours	✓	**69.2**	**67.9**	**1.3(1.9%↓)**

2. Full results on testing set

Table 2 and Table 3 shows accuracy results on VisDA-2017 and DomainNet testing set. Our consistency regularization based SFDA method gains improvement on most classes, especially for those who suffer from severe performance degradation with previous SFDA works. It proves that our method can avoid overfitting issues and develop a more generalizable model under the SFDA setting, which are essential for real-world scenarios.

Table 2. Per-class and mean accuracy (%) on VisDA-2017 testing set. SF means source-free. We highlight the best results and underline the second-best ones in SFDA methods. † means these results from *VisDA2017 Classification Challenge* [2] leaderboard, which are under **easier** vanilla UDA setting and may use **stronger** backbone causing **unfair** comparisons.

Method	SF	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
BUPT†	×	95.7	67.0	93.4	97.2	90.6	86.9	92.0	74.2	96.3	66.9	95.2	69.2	85.4
IISC_SML†	×	92.4	81.0	86.4	92.2	87.9	89.4	69.4	84.2	96.4	85.3	83.5	89.3	86.4
NLE†	×	94.3	86.5	86.9	95.1	91.1	90.0	82.1	77.9	96.4	77.2	86.6	88.0	87.7
SHOT [1]	✓	81.4	73.4	77.2	81.9	80.0	79.7	54.0	63.3	91.8	45.7	78.7	78.9	73.2
G-SFDA [5]	✓	89.2	76.2	**88.0**	93.4	87.7	83.0	71.4	62.4	92.8	74.6	82.9	79.7	81.8
NRC [4]	✓	90.3	**84.9**	85.7	86.0	88.9	83.2	69.2	66.1	95.4	69.1	85.4	79.2	82.0
DaC [6]	✓	93.3	73.5	85.2	93.6	**90.7**	86.0	**80.0**	68.5	96.5	76.6	81.8	84.7	<u>84.2</u>
Ours	✓	**93.7**	80.6	87.7	**93.9**	88.2	**97.9**	77.8	**81.3**	**96.9**	**82.3**	**87.2**	**87.3**	**87.8**

3. Adaptation procedure

Algorithm 1 shows the overall adaptation process of our proposed SFDA method. Instead of introducing multiple loss terms with some balance hyper-parameters by previous works, our method only has one loss term, which can effectively

Table 3. Accuracy (%) on DomainNet testing set. SF means source-free. We highlight the best results and underline the second-best ones.

Method	SF	Rw→Cl	Rw→Pt	Pt→Cl	Cl→Sk	Sk→Pt	Rw→Sk	Pt→Rw	Avg.
SHOT [1]	✓	64.1	59.0	64.9	59.0	59.6	58.8	73.3	62.7
G-SFDA [5]	✓	57.4	58.5	57.4	51.9	54.0	54.8	73.0	58.1
NRC [4]	✓	61.2	60.7	61.5	55.1	58.9	52.2	74.5	60.6
DaC [6]	✓	68.0	**68.5**	65.9	60.0	63.5	60.3	**79.6**	<u>66.6</u>
Ours	✓	**71.4**	68.3	**71.2**	**63.2**	**65.1**	**62.2**	74.1	**67.9**

erase the burden of the hyper-parameters optimization process.

Algorithm 1 Process of our proposed SFDA method.

Input: target unlabelled dataset $D_t = \{x_i^t\}_{i=1}^{n_t}$, source model $h_s = f_s \circ g_s$.
Parameters: maximum iteration I, sharpening temperature T, selection threshold τ, mapping functions $\mathcal{M}$ and $\mathcal{T}$.

```
Initialize prediction bank P and feature bank F
iter ← 0
while iter < I do
    Fetch mini-batch samples x_i from D_t
    Apply strong image augmentation x'_i = A(x_i)
    Compute model prediction p_i, p'_i = h(x_i, x'_i)

    Derive soft pseudo-labels p̂_k(x_i) and sampling probabilities ξ_i        ▷ Eq. (1,3)
    Compute consistency regularization loss L̂_cr                             ▷ Eq. (4)

    Compute class-wise weight w_div with prediction bank P                    ▷ Eq. (5,6)
    Update prediction bank P with current mini-batch predictions

    Update prototype and compute assignments of samples with feature bank F   ▷ Eq. (7)
    Compute prototype calibration weight w_proto                              ▷ Eq. (8)
    Update feature bank F with current mini-batch features

    Compute final loss L = w_div · w_proto · L̂_cr and update model
end while
```

References

[1] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020. 1, 2

[2] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 1

[3] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8050–8058, 2019. 1

[4] Shiqi Yang, Joost van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Advances in Neural Information Processing Systems*, 34:29393–29405, 2021. 1, 2

[5] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021. 1, 2

[6] Ziyi Zhang, Weikai Chen, Hui Cheng, Zhen Li, Siyuan Li, Liang Lin, and Guanbin Li. Divide and contrast: Source-free domain adaptation via adaptive contrastive learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2