Supplementary Material for "Adversarial Partial Domain Adaptation by Cycle Inconsistency"

Kun-Yu Lin[1♢], Jiaming Zhou[1♢], Yukun Qiu[1], and Wei-Shi Zheng[1,2,3,4‡]

[1] School of Computer Science and Engineering, Sun Yat-sen University, China
[2] Peng Cheng Laboratory, Shenzhen, China
[3] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
[4] Guangdong Province Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou
{linky5,zhoujm55,qiuyk}@mail2.sysu.edu.cn, wszheng@ieee.org

S1 Verification of Prototype-based Transformation

In Section 3.2, we introduced the cycle transformation consisting of two cross-domain feature transformations, which are implemented by searching the most similar samples across domains in the whole feature space. However, such a vanilla nearest-neighbor-based scheme is time-consuming and noisy (due to the large intra-class variation). Therefore, in Section 3.3, we propose prototype-based cross-domain feature transformations to reduce the computation cost and noise of the vanilla scheme. Since we calculate the feature similarity only $|\mathcal{C}_s|/K$ times (rather than n_s/n_t times) in one feature transformation, the computation cost is naturally reduced. Besides, as shown in Fig. S1, the prototype-based transformation widens the gap of ACCs (accuracy gap of cycle transformations between samples in shared class and samples in outlier classes) compared with the vanilla scheme, which verifies that the prototype-based scheme reduces the noise of the vanilla scheme for accurate weight assignment.

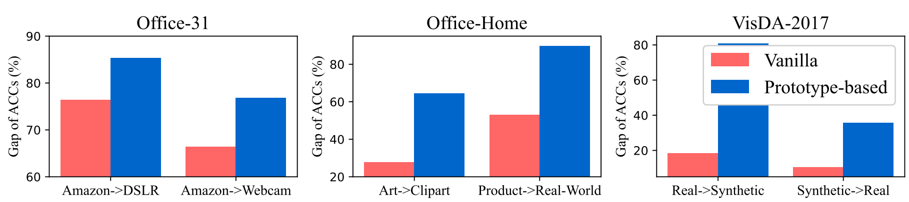
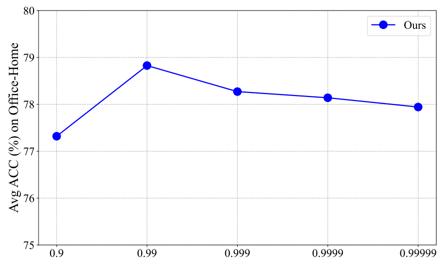
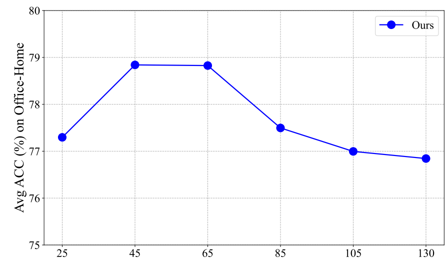


Fig. S1. Empirical verification of prototype-based cross-domain feature transformations using source-only models as feature extractors on three real-world datasets. We conduct cross-domain feature transformations by searching the most similar prototypes across domains and calculate the accuracy (ACC) of cycle transformations as that in Fig. 1b. In the figures, we compare the prototype-based scheme with the vanilla nearest-neighbor-based scheme in terms of the gap of ACCs, *i.e.*, Gap of ACCs = ACC of samples in shared classes − ACC of samples in outlier classes. Best viewed in color.

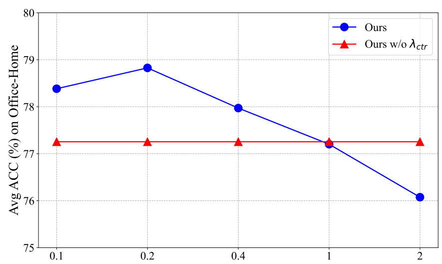
♢ indicates equal contributions and ‡ indicates the corresponding author.



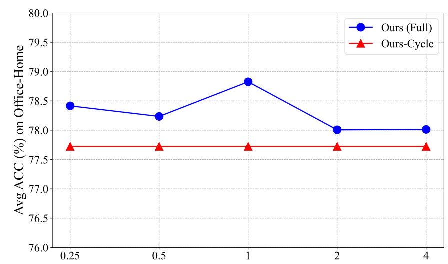
(a) Momentum λ_m



(b) The number of target prototypes K



(c) λ_{ctr} for center losses



(d) λ_w for mixed strategy

Fig. S2. Hyperparameter analysis of our method on (a) momentum λ_m, (b) the number of target prototypes K, (c) λ_{ctr} for center losses and (d) λ_w for mixed strategy. Best viewed in color.

S2 Hyperparameter Analysis

In this section, we analyze the sensitivity of our method to four hyperparameters, namely momentum λ_m for updating prototypes, the number of prototypes K in the target domain, λ_{ctr} for center losses and the weighting coefficient λ_w for the mixed strategy. We conduct quantitative analysis on the Office-Home datasest.
- Momentum λ_m. As shown in Fig. S2a, a value of λ_m close to 1 (*e.g.*, 0.99) is usually appropriate, since the update rate of prototypes should be low. This is because there are usually only one or two samples for each class within each batch and a large update rate harms the prototype update due to the intra-class variation.
- The number of target prototypes K. As shown in Fig. S2b, a moderate value of K (*e.g.*, 45, 65) is appropriate, which keeps a balance between abstracting the dataset and reducing noise caused by the large intra-class variation. Usually, setting $K = |\mathcal{C}_s|$ ($|\mathcal{C}_s| = 65$ in Office-Home) is appropriate in PDA scenarios (since the number of target classes in unknown).
- λ_{ctr} for center losses. As shown in Fig. S2c, our method with a small value of λ_{ctr} (*e.g.*, 0.1, 0.2) improves the performance of that without center losses. However, the value of λ_{ctr} cannot be too large (larger than 1) otherwise negative effects are triggered, since a very large λ_{ctr} disrupts the feature distributions within individual domains and thus the performance drops significantly.
- λ_w for mixed strategy. As shown in Fig. S2d, a mixed strategy with different values of λ_w consistently improves the performance of a cycle-inconsistency-

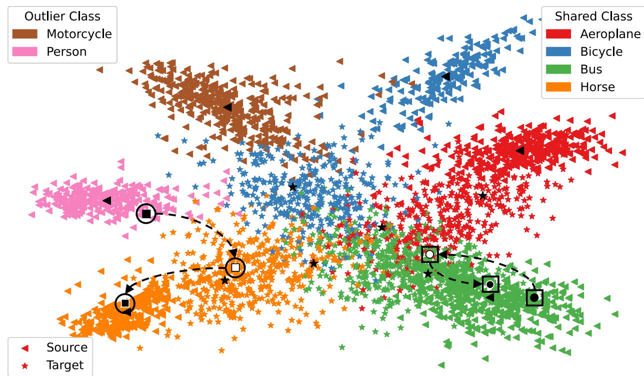


Fig. S3. Visualization of the cycle transformation process on VisDA-2017. Black triangle and star markers denote source and target prototypes, respectively. Besides, triangle and star markers in other colors denote the source and target samples, and different colors denote different classes. Best viewed in color.

based strategy, which shows that our method is not sensitive to the hyperparameter λ_w.

S3 Cycle Transformation Visualization

In Fig. S3, we show examples for visualizing the proposed prototype-based cycle transformation process. We directly show the feature distribution of a source-only model on the VisDA-2017 dataset by setting the features to be two-dimensional (rather than by t-SNE). For clearer visualization, we select six classes (four shared and two outlier classes). As shown in the figure, a source sample of the green class (the boxed **black** circle, of shared classes) falls close to the target prototype in the green region after the source-to-target transformation. By contrast, a source sample of the pink class (the circled **black** rectangle, of outlier classes) falls close to the target prototype in the orange region. After cycle transformations, the source sample of the green class falls close to the green source prototype, while the source sample of the pink class falls close to the orange source prototype. The examples show that, compared with source samples of shared classes, source samples of outlier classes more likely alter their categories after the proposed cycle transformations. According to the difference in cycle transformations, we distinguish outlier classes from shared classes for PDA. Also, in the figure, we find that the cross-domain feature transformation functions based on soft nearest neighbor improves the representation power of transformed features, *i.e.*, the transformed data points have many possible positions in the feature space (more than $|\mathcal{C}_s|/K$).

S4 Grad-CAM Visualization

In this section, we qualitatively show the effectiveness of our model by Grad-CAM [1]. As shown in Fig. S4, our model focuses on similar visual cues across the

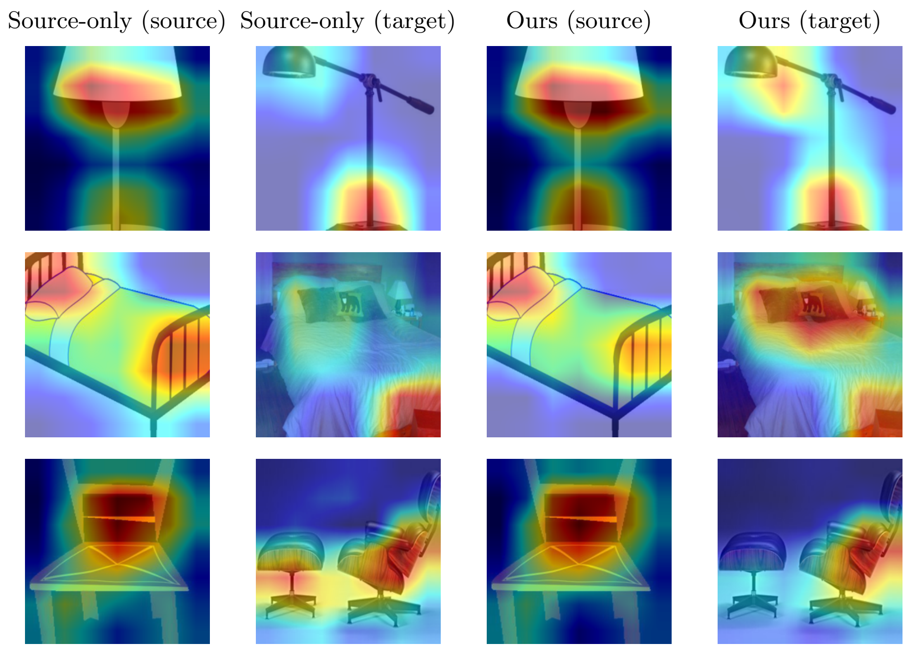


Fig. S4. Grad-CAM visualization of the source-only model and our model. In each row, we show a group of samples from the same category, where we show samples in both the source and target domains for both models. Best viewed in color.

source and target domains, compared with the source-only model. For example, as shown in the first row, the source-only model focuses on the base of the lamp in the target domain, which is different from that in the source domain. By contrast, our model focuses on the cap and base of the lamps in both the source and target domains. The results indicate that our model aligns the shared classes well across domains.

References

1. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. Int. J. Comput. Vis. **128**(2), 336–359 (2020)