RETHINKING EFFECTIVENESS OF UNSUPERVISED DOMAIN ADAPTATION METHODS

SUPPLEMENTARY MATERIAL

Anonymous authors
Paper under double-blind review

1 TRAINING DETAILS FOR ADAPTATION METHODS

We re-implement several adaptation methods in a new framework, which is designed with an aim to standardize evaluation across these methods. Specifically, we keep the adaptation independent hyper-parameters (such as architectures, batch sizes) same across the methods, and use the adaptation-specific hyper-parameters as recommended in the respective methods. We provide a basic version of our framework that incorporates DANN and CDAN implementations along with the supplementary material. The full version of the framework with implementations of several other adaptation methods will be released upon acceptance. We use the open-source repositories of prior UDA methods from the links given below.

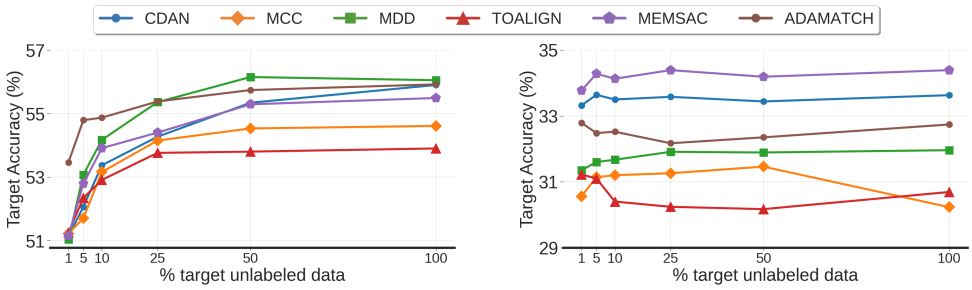
- **CDAN:** https://github.com/thuml/CDAN/tree/master
- **MCC:** https://github.com/thuml/Versatile-Domain-Adaptation
- **MDD:** https://github.com/thuml/MDD
- **ToAlign:** https://github.com/microsoft/UDA
- **MemSAC:** https://github.com/ViLab-UCSD/MemSAC_ECCV2022
- **AdaMatch:** https://github.com/google-research/adamatch
- **DALN:** https://github.com/xiaoachen98/DALN

Architecture-specific training details In our ablation on benchmarking UDA across architectures, we use all pre-trained checkpoints from the timm library, and all of them are pre-trained on ImageNet-1k. Across the architectures, we uniformly use a batch size of 32, SGD optimizer with an initial learning rate of 0.003 and cosine decay. It might be possible that ViT models benefit from other algorithms such as Adam Zhang et al. (2020), which we do not explore in this paper. For data augmentation, we first resize the images so that the shorter size is 256 and then choose a random 224×224 crop followed by random horizontal flip. However, we use a crop size of 256 instead of 224 for Swin transformer due to its input size. We train the networks for a total of 75k iterations on DomainNet and CUB200 with validation performed at every 5k steps, and for 30k iterations on the smaller OfficeHome dataset with validation at every 500 steps. We use early stopping on the test set to choose the best accuracy.

For the classifier, we use a 2-layer MLP with a hidden dimension of 256. The input dimension for the MLP, though, varies depending on the output dimension of the backbone architecture used. For Resnet-50, it is 2048, for Swin-t and ConvNext-t it is 768 and for Deit-s and ResMLP-s it is 384.

2 EFFECT OF UNLABELED DATA IN THE TARGET

Stratified Sampling Procedure In our experiments on studying the effect of the volume of the data used, we adopt a stratified sampling procedure across the categories. Specifically, to sample $x\%$ of data, we take samples from each category individually and perform the sampling, as opposed to sampling $x\%$ from the dataset as a whole. While the former performs per-class sampling, the latter performs global sampling which might change the tail properties of the resulting sub-sampled dataset. We also make sure that all categories which have non-zero images in the original dataset have at least 1 image in the sub-sampled dataset. Note that we only use the label information from the target only during sampling, but not during training.

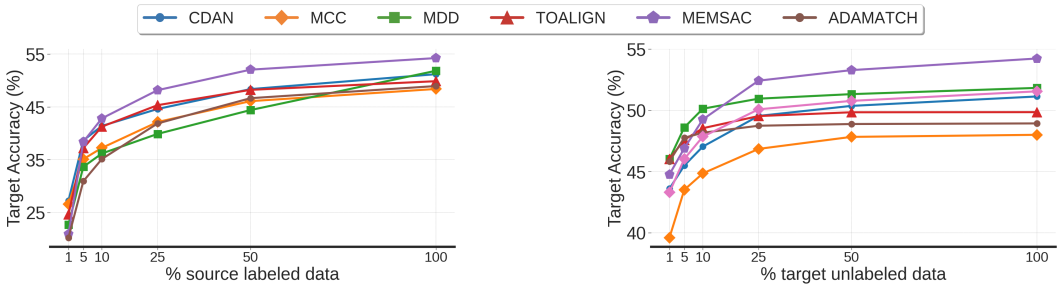


(a) Real→Clipart (DeiT Backbone) (b) GeoPlaces (USA→Asia)

Figure 1: **Effect of unlabeled data** In addition to the transfer settings shown in the main paper, we show the effect of target unlabeled data on the target accuracy on two more settings - Real→Clipart from DomainNet on a DeiT Backbone in (a) and on USA→Asia from GeoPlaces using ResNet-50 backbone in (b). The trends remain similar, where we observe that most UDA methods under-utilize unlabeled data.

As demonstrated in main paper, current UDA methods under-utilize unlabeled data, and the performance saturates even when more unlabeled data is accessible to the algorithms. This trend holds across most datasets, architectures and methods. In Fig. 1, we show results using two additional settings. For Real→Clipart from DomainNet using DeiT-III backbone in Fig. 1a, we observe similar trends as in paper, where the accuracy plateaus around 25% for most methods. Additionally, we also show the scaling trends for USA→Asia from GeoPlaces using ResNet-50 backbone in Fig. 1b, where we observe that unlabeled data rarely helps, even hurting the adaptation accuracy in some cases.

To further investigate the factors effecting the target accuracy, we conduct a similar experiment by using subsets of source labeled data, while using the full target unlabeled data each time. Specifically, we use $\{1, 5, 10, 25, 50, 100\}\%$ of source labels and train the UDA methods on each subset. We run three random seeds and plot the mean accuracy in Fig. 2. We observe that the scaling trends of target accuracy with respect to source labeled data are much more favorable towards improving performance. For example, doubling the number of source labels from 50% to 100% improves target accuracy by $\sim 9\%$ on average across UDA methods. In contrast, the improvement in doubling the target unlabeled data from 50% to 100% is less than 0.5% on average. This confirms the fact that labels have a more pronounced impact on target accuracy even when they arise from a different domain, compared to unlabeled data from the same domain.

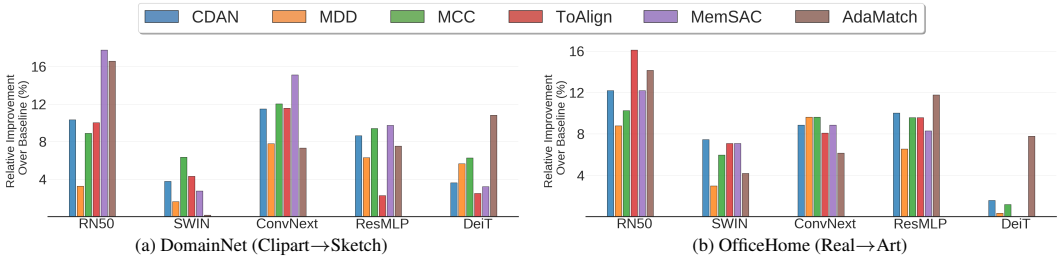


(a) Source Labeled Data (b) Target Unlabeled Data

Figure 2: **Source labels vs. Target unsupervised data** We show that collecting more labels from source dataset, even when it is from a different domain, has a more profound influence on the target accuracy (a) compared to collecting more unlabeled data from the target domain using current UDA methods (b).

3 Effect of Backbone Architecture

Additional Results We show results of effect of change in backbone for two additional settings in Fig. 3, namely Clipart→Sketch from DomainNet in Fig. 3a and Real→Art from OfficeHome in Fig. 3b. We observe same trends as discussed in main paper, with vision transformer architecture like



(a) DomainNet (Clipart→Sketch) (b) OfficeHome (Real→Art)

Figure 3: **Effect of backbone.** For each of the UDA methods, we show the gain in accuracy relative to a baseline trained only using source-data. Matching the trends shown in the main paper, we observe that the benefits offered by UDA approaches over the baseline diminish when using better backbones that have improved domain-robustness properties.

Swin and Deit diminishing the benefits of most UDA methods, that otherwise yield good gains with Resnet-50 as the backbone.

4 Effect of Pre-training Data

Pre-training details We use the official repositories for SwAV, MoCo-v3, MAE to pre-train the models on our datasets. Note that we subsample an image set of 1M images from ImageNet, Places205 and iNat2021 to normalize the effects of data volume, using the same per-class sampling strategy described in Sec. 2. We use the official repositories for Swav, MoCo-V3 and MAE, and use the code for supervised pre-training from PyTorch. We train Swav for 150 epochs, MoCo-v3 for 250 epochs, MAE for 400 epochs and supervised pre-training for 90 epochs. The training for all the methods is performed on 8 GPUs with a total batch size of 1024 in each case. For all other hyperparameters, we follow the ones recommended in the respective repositories.

5 Broader Impact Statement

The prospects of success in UDA has broad implications beyond accuracy, as it stands to directly benefit fairness, inclusivity and democratization of ML models to under-represented societies as not all demographies or domains can be adequately represented or labeled in training datasets (Prabhu et al., 2022). Therefore, it is imperative to hold a deeper understanding of UDA methods and the factors impacting their effectiveness in real-world scenarios, as studied in this work. Furthermore, the analysis and recommendations presented through our work has broader impact by serving a dual purpose: assisting researchers in identifying future research opportunities as well as guiding practitioners in maximizing the benefits derived from adaptation models.

References

Viraj Prabhu, Ramprasaath R Selvaraju, Judy Hoffman, and Nikhil Naik. Can domain adaptation make object recognition work for everyone? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3981–3988, 2022. 3

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020. 1