Instance Level Affinity-Based Transfer for Unsupervised Domain Adaptation
Supplementary Material

S1. Complete Training Details

All models are implemented in Pytorch. We use Nvidia 2080Ti GPUs for training. For training and testing on Office-31 and Birds-31 datasets, we take random crops of size 224x224 for all images, and use random horizontal flipping as a data augmentation strategy. For Digits, we resize the images to size 28x28. For creating one training batch, we use b source and b target samples, where b is set to 124 for Office-31 and Birds-31 and 120 for Digits.

We adjust the learning rate as $\eta_p = \eta_0(1 + \alpha p)^{-\beta}$, where p changes from 0 to 1 as training progresses, $\alpha = 10, \beta = 0.75$ and η_0 is the initial learning rate, following [1]. More details including the number of pre-training iterations and learning rate are presented in Table S1. We use the same hyperparameters across all the tasks within a dataset. All source code and trained models will be publicly released.

	Office-31	Birds-31	Digits
Initial Learning Rate	0.001	0.03	0.01
no. of Training Iterations	15,000	12,500	25,000
no. of GPUS	3	3	1
no. of pretraining iterations	2800	2800	5000
Co-eff MSC Loss (λ_2)	2.0	2.0	2.0
Co-eff Adv. Loss (λ_1)	1.0	1.0	1.0
Batch Size	124	124	120

Table S1: **Training details** for Office-31, Birds-31 and Digits.

S2. Evaluation on VisDA Dataset

We report results of our method on another challenging dataset, VisDA-2017 [5], which contains source images from a synthetic domain and target images from real domain. The dataset contains roughly 280K images across 12 categories in the training, validation and test domains. In Table S2, we observe that our method consistently improves performance on real target data for both DANN and CDAN backbones, showing the effectiveness of our method over global alignment strategies for a challenging adaptation task. For fair comparison with the reported results in previous works [3, 4], we use ResNet-101 and ResNet-50 for ILA-DA (with DANN) and ILA-DA (with CDAN), respectively.

Method	Synthetic → Real
DANN(Resnet-101) [1]	57.4
ILA-DA (with DANN, Resnet-101)	64.63
CDAN(Resnet-50) [4]	66.8
ILA-DA (with CDAN, Resnet-50)	68.85

Table S2: **VisDA dataset** Results for ILA-DA on VisDA adaptation setting using ResNet-101 for comparison with DANN[1] and ResNet-50 for comparison with CDAN[4].

S3. Effect of k

We provide results with $k = 1, 3, 5$ for Office-31 and Birds-31 in Table S3a and Table S3b, respectively. For Office-31, we observe that $k = 5$ gives optimal results for most of the individual tasks and also outperforms over $k = 1$ and $k = 3$ on the average. For Birds-31, we observe that $k = 3$ gives the best average on all the six tasks followed by $k = 5$ and $k = 1$. In general, we observe that $k > 1$ is useful for reliable pseudo-labeling in our approach. This is because using a larger value of k allows deciding the psuedo-label based on a larger number of neighboring source samples, which makes it robust to outliers near the decision boundaries. This hypothesis is verified by results on challenging tasks in Office-31, such as W→A in Table S3a.

S4. Effect of number of classes

We demonstrate the effectiveness of ILA-DA in improving class aware adaptation by running experiments with varying number of classes in the domains. For the purpose of this ablation, we choose the N→C task from Birds-31 and subsample the dataset to only contain the first 10,20 and 31 classes which correspond to using one-third, two-thirds and the complete label sets, respectively. In Table S4, we report the performance of our proposed method across these settings and observe that ILA-DA+DANN outperforms DANN across the board. More importantly, adaptation using ILA-DA+DANN provides increasing benefits with an increase in the number of classes. This can be attributed to the class aware nature of the adaptation using ILA-DA, which would be more useful in case of larger number of categories. These results indicate that our method contributes successfully towards class level alignment.

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg.
CDAN [4]	93.1	98.2	**100.0**	89.8	70.1	68.0	86.6
ILA-DA (k=1)	92.20	98.36	**100.0**	91.96	72.31	69.93	87.46
ILA-DA (k=3)	94.34	99.24	**100.0**	91.16	**73.34**	75.15	88.87
ILA-DA (k=5)	**95.72**	**99.25**	**100.0**	**93.37**	72.10	**75.40**	**89.30**

(a) Office-31

Method	C → I	I → C	I → N	N → I	C → N	N → C	Avg.
CDAN [4]	68.67	89.74	86.17	73.80	83.18	91.56	82.18
ILA-DA (k=1)	70.77	**93.94**	**90.46**	**78.47**	84.94	**94.64**	85.54
ILA-DA (k=3)	**72.77**	93.83	90.36	78.09	**86.58**	94.53	**86.03**
ILA-DA (k=5)	72.63	93.72	90.19	78.37	85.98	94.26	85.86

(b) Birds-31

Table S3: **Effect of k**. Results shown for domain adaptation on Office-31 (S3a) and Birds-31 (S3b) using Resnet-50 for all 6 transfer tasks in each setting. Results are shown for three values of $k = \{1, 3, 5\}$ with ILA-DA using CDAN as the backbone.

# Classes	Source Only	DANN	ILA-DA (with DANN)
$c = 10$	94.47	94.81	95.81 $(+1.00\%)$
$c = 20$	93.88	93.46	95.64 $(+2.18\%)$
$c = 31$	89.96	89.53	93.89 $(+4.36\%)$

Table S4: **Effect of number of classes**. Comparison of accuracies with different number of classes for N → C task from Birds-31. We show results using $c = \{10, 20, 31\}$, which correspond to using 1/3rd, 2/3rd and complete classes respectively. Owing to the class awareness of our adaptation, we find that ILA-DA provides increasing benefits with larger c, where traditional methods like DANN fail.

S5. Coefficient for MSC Loss

The total loss used for training ILA-DA model is given by Equation S1, with the notations as described in the main paper. The final loss used for training the network is given by the sum of the supervised, adversarial and our multi-sample contrastive(MSC) loss with a training objective.

$$\begin{aligned} &\min_{\mathcal{G},\mathcal{C}} \ \mathcal{L}_{sup} + \lambda_1 \mathcal{L}_{adv} + \lambda_2 \mathcal{L}_{MSC} \\ &\min_{\mathcal{D}} \qquad\quad \mathcal{L}_D \end{aligned} \tag{S1}$$

where λ_1 and λ_2 are the weights of the adversarial loss and the multi-sample contrastive loss respectively, as detailed in Section 3 of the main paper. In Table S5, we present results by varying λ_2 values to study the effect of MSC loss coefficient on target accuracy. We run our experiments on all six tasks for Birds-31 dataset for three values of λ_2, namely $\lambda_2 = \{0.1, 1.0, 2.0\}$. We observe that although the performance of ILA-DA is more or less robust to MSC loss coefficient, $\lambda_2 = 1.0$ performs optimally for Birds-31 with a slight improvement of 0.09% over $\lambda_2 = 2.0$. Note that we report results using $\lambda_2 = 2.0$ in our main paper for Birds-31, which we directly adopt from Office-31 experiments. Results in Table S5 indicate that there is further room for improvement by tuning loss coefficients specific to datasets.

Method	C → I	I → C	I → N	N → I	C → N	N → C	Avg.
ILA-DA , λ_2=0.1	72.31	93.56	90.39	77.14	86.31	94.37	85.68
ILA-DA , λ_2=1.0	72.98	93.56	90.09	78.79	86.91	94.43	86.12
ILA-DA , λ_2=2.0	72.77	93.83	90.36	78.09	86.58	94.53	86.03

Table S5: **Effect of MSC loss coefficient**(λ_2). Results for 6 transfer tasks on Birds-31 for three different values of $\lambda_2 = \{0.1, 1.0, 2.0\}$. Results shown for $k = 3$.

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg.
CDAN [4]	93.1	98.2	100.0	89.8	70.1	68.0	86.6
CAN [2]	94.5	99.1	99.8	95.0	78.0	77.0	90.6
CAN*	94.50	99.01	100.0	93.69	74.92	75.36	89.58
ILA-DA (with CDAN)	95.72	99.25	100.0	93.37	72.10	75.40	89.30

Table S6: **ILA-DA vs. CAN** Results for domain adaptation on Office-31 adaptation setting using Resnet-50. CAN* indicates our implementation of CAN using similar hyperparameter settings as ours.

S6. Further Comparisons on Office-31

A related yet different work to ours is Contrastive Adaptation Network (CAN) [2], which is based on MMD whereas our proposed technique is designed to work in an adversarial setup. We provide a comparison with CAN on Office-31 dataset in Table S6 using ILA-DA + CDAN. Since CAN uses task specific hyperparameters and domain specific batch norm while we do not, we re-implement their method using the same hyperparameter settings as ours (indicated by CAN* in Table S6). From Table S6, we observe that ILA-DA outperforms CAN on three tasks of the Office-31 dataset, namely A→W, D→W and W→D and performs comparably on other tasks. Overall, we find that ILA-DA + CDAN performs equally against CAN on the average.

References

[1] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[2] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[3] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019.

[4] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adap-

tation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018.

[5] 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.