Select, Label, and Mix: Learning Discriminative Invariant Feature Representations for Partial Domain Adaptation (Supplementary Material)

Aadarsh Sahoo¹ Rameswar Panda¹ Rogerio Feris¹ Kate Saenko^{1,2} Abir Das³ ¹ MIT-IBM Watson AI Lab, ² Boston University, ³ IIT Kharagpur

{aadarsh@, rpanda@, rsferis@us.}ibm.com, saenko@bu.edu, abir@cse.iitkgp.ac.in

Section	Content
А	Dataset Details
В	Implementation Details
С	Additional Experiments
D	Qualitative Results
Е	Broader Impact and Limitations

Table 1: Overview of Supplementary Material.

A. Dataset Details

We evaluate the performance of our approach on several benchmark datasets for partial domain adaptation, namely Office31 [12], Office-Home [14], ImageNet-Caltech and VisDA-2017 [10]. The following are the detailed descriptions of the above datasets:

Office31. This dataset contains 4,110 images distributed among 31 different classes and collected from three different domains: Amazon (A), Webcam (W) and DSLR (D), resulting in 6 transfer tasks. The dataset is imbalanced across domains with 2,817 images belonging to Amazon, 795 images to Webcam, and 498 images to DSLR, making Amazon a larger domain as compared to Webcam and DSLR. For all our experiments, we select the 10 classes shared by Office31 and Caltech256 [6] as the target categories and obtain the following label spaces:

 $\mathcal{L}_{source} = \{0, 1, 2, ..., 30\}.$

 $\mathcal{L}_{target} = \{0, 1, 5, 10, 11, 12, 15, 16, 17, 22\}.$

Number of Outlier Classes = 21.

Figure 1 shows few randomly sampled images from this dataset. The dataset is publicly available to download at: https://people.eecs.berkeley.edu/~jhoffman/domainadapt/#datasets_code.

Office-Home. This dataset contains 15,588 images distributed among 65 different classes and collected from four different domains: Art (Ar), Clipart (Cl), Product (Pr), and RealWorld (Rw), resulting in 12 transfer tasks. The dataset



Figure 1: **Sampled Images from Office31 Dataset**. Each row from top to bottom corresponds to the domains Amazon, Dslr and Webcam, respectively. The images in the same column belong to the same class. Best viewed in color.

is split across domains with 2427 images belonging to Art, 4365 images to Clipart, 4439 images to Product, and 4347 images to RealWorld. We select the first 25 categories (in alphabetic order) in each domain as the target classes and obtain the following label spaces:

 $\mathcal{L}_{source} = \{0, 1, 2, ..., 64\}.$ $\mathcal{L}_{target} = \{0, 1, 2, ..., 24\}.$

Number of Outlier Classes = 40.

Figure 2 displays a gallery of sample images for this dataset. The dataset is publicly available to download at: http://hemanthdv.org/OfficeHome-Dataset/.



Figure 2: Sampled Images from Office-Home Dataset. Each row from top to bottom corresponds to the domains Art, Clipart, Product and RealWorld, respectively. The images in the same column belong to the same class. Best viewed in color.



Figure 3: Sampled Images from ImageNet-Caltech Dataset. The top row corresponds to the ImageNet domain, while the bottom row to the Caltech domain. The images in the same column belong to the same class. Best viewed in color.



Figure 4: **Sampled Images from VisDA-2017 Dataset**. The top row corresponds to the Synthetic domain, while the bottom row to the Real domain. The images in the same column belong to the same class. Best viewed in color.

ImageNet-Caltech. This large-scale dataset consists of two datasets (ImageNet1K [11] (I) & Caltech256 [6] (C)) as two separate domains and consist of over 14 million images combined. 2 transfer tasks are formed for this dataset. While source domain contains 1,000 and 256 classes for ImageNet and Caltech respectively, each target domain contains only 84 classes that are common across both domains. As it is a general practice to use ImageNet pretrained weights for network initialization, we use the validation set images when using ImageNet as the target domain. Number of Outlier Classes = 172 for $C \rightarrow I$, 916 for $I \rightarrow C$. Figure 3 displays a gallery of sample images for this dataset. The datasets are publicly available to download at: http://www.image-net.org/

http://www.vision.caltech.edu/Image_ Datasets/Caltech256/.

VisDA-2017. This dataset contains 280,157 images distributed among 12 different classes and two domains. The dataset contains three sets of images: training, validation and testing. The training set contains 152,397 synthetic (S) images, the validation set contains 55,388 real-world (R) images, while the test set contains 72,372 real-world images. For the experiments, the training set is considered as the Synthetic (S) domain, while the validation set as the Real (R) domain, following [8]. This results in 2 transfer tasks. The first 6 categories (in alphabetical order) are selected in each of the domains as the target classes, and the following label spaces are obtained:

 $\mathcal{L}_{source} = \{0, 1, 2, ..., 11\}.$ $\mathcal{L}_{target} = \{0, 1, 2, ..., 5\}.$

Number of Outlier Classes = 6.

Figure 4 displays a gallery of sample images for this dataset. The dataset is publicly available to download at:

http://ai.bu.edu/visda-2017/#download.

B. Implementation Details

The training pipeline pseudo-code for **SLM** is shown in Algorithm 1. Following are the detailed description of the implementation we follow for various components of the framework:

Feature Extractor (\mathcal{G}). We use ResNet-50 [7] backbone for the feature extractor. The overall structure of ResNet-50 is Initial Layers, Layer-1, Layer-2, Layer-3, Layer-4, AvgPool, Fc. The model is initialized with ImageNet [11] pretrained weights. Additionally, we add a bottleneck layer of width 256 just after the AvgPool layer to obtain the features and replace all the BatchNorm layers with Domain-Specific Batch-Normalization [4] layers. All the layers till Layer-3 are frozen and only the rest of the layers are fine-tuned.

Selector Network (\mathcal{H}). We use a ResNet-18 [7] network with the Fc layer replaced with a binary-length fully connected layer as the selector network in our framework. The network is initialized with ImageNet pretrained weights and all the layers are trained while optimization.

Classifier (\mathcal{F}). The final Fc layer of ResNet-50 described above is replaced with a task-specific fully-connected layer to form the classifier network of our framework.

Domain Discriminator (D). A three-layer fully-connected network is used as the domain discriminator network. It takes the 256-length features obtained from the feature extractor as input. The adversarial training is incorporated using a gradient reversal layer (GRL).

Hyperparameters. All the networks are optimised using mini-batch stochastic gradient descent with a momentum of 0.9. A batch size of 64 is used for Office31 and VisDA-2017 while a batch size of 128 is used for Office-Home and ImageNet-Caltech. For feature extractor an initial learning rate of 5e-5 for the convolutional layers while an initial learning rate of 5e-4 for all the fully-connected layers is used. For the selector network and the domain discriminator an initial learning rate of 5e-3 and 5e-4 are used respectively. The learning rates are decayed following a cosineannealing strategy as the training progresses. The best models are captured by obtaining the performance on a validation set. We do NOT follow the ten-crop technique [2, 3], to improve the performance in the inference phase. We obtain the best hyperparameters using grid search. All the experiments were averaged over three runs, which used random seed values of 1, 2, and 3 respectively.

Hardware and Software Details. All the experiments were conducted using a single NVIDIA Tesla V100-DGXS GPU with 32 GigaBytes of memory, equipped with a Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz. We used PyTorch v1.4.0, Python v3.6.10 to implement the codes.

Algorithm 1 The training pipeline for SLM

Data: source data \mathcal{D}_{source} and target data \mathcal{D}_{target} .

Networks: Selector Network $\mathcal{H}(.)$, Feature Extractor $\mathcal{G}(.)$, Classifier $\mathcal{F}(.)$, and Domain Discriminator $\mathcal{D}(.)$.

- 1: Initialize networks $\mathcal{G}(.), \mathcal{F}(.), \mathcal{H}(.)$, and $\mathcal{D}(.)$ in **SLM**.
- 2: for $itrn = 1 \rightarrow num_itrn$ do
- 3: Obtain the mini-batches \mathcal{D}_{source}^{b} and \mathcal{D}_{target}^{b} . # "Select" Module
- 4: Obtain the binary decisions from \$\mathcal{H}(\mathcal{D}_{source}^b)\$ & use them to obtain \$\mathcal{D}_{sel}^b\$ & \$\mathcal{D}_{dis}^b\$.
 # "Label" Module
- 5: Obtain soft pseudo-labels \hat{y}^b from $\mathcal{F}(\mathcal{G}(\mathcal{D}_{target}))$ for \mathcal{D}_{target} .

"Mix" Module

- 6: Obtain $\mathcal{D}_{inter_mix}^{b}$, $\mathcal{D}_{intra_mix_s}^{b}$, and $\mathcal{D}_{intra_mix_t}^{b}$.
- 7: Compute \mathcal{L}_{sup} , \mathcal{L}_{adv} , \mathcal{L}_{select} , \mathcal{L}_{label} , and \mathcal{L}_{mix} .
- 8: Compute the gradients & backpropagate for optimization using gradient descent.

9: end for

C. Additional Experimental Results

Results on Office-Home Dataset. In Table 2, along with the performance accuracies, we have included the standard deviation for each adaptation task for the Office-Home dataset, as promised in Table-2 of the main paper.

Effectiveness on Different Backbone Networks. To show that the proposed framework is backbone-agnostic, i.e., it provides the best performance irrespective of the architecture of the feature extractor, we conduct experiments using a VGG-16 [13] backbone for the feature extractor. We report the results on the transfer tasks from the Office31 dataset in Table 3 and compare it with the current state-ofthe-art methods. Our method outperforms the previously best results by a margin of **3.0**% on average and achieves new state-of-the-art results. This confirms that our proposed framework for partial domain adaptation is robust with respect to the change of backbone network.

Effectiveness of Individual Modules. In Section 4.3 of the main paper, we discussed the importance of the proposed three unique modules on Office-Home dataset. Here, we extend the experiments to Office31 and VisDA-2017 and provide the performance on the transfer tasks in Table 4. Similar to the results on Office-Home dataset, our approach with all the three modules (Select, Label and Mix) working jointly, works the best on both datasets.

Effectiveness of Hausdorff Distance. We investigate the effect of Hausdorff distance (Eqn. 2 in the main paper) in selector network training and find that removing it lowers down performance from 76.0% to 73.7% on Office-Home

dataset, showing its importance in guiding the selector to discard the outlier source samples for effective reduction in negative transfer. We provide the individual performance of all the transfer tasks on Office-Home dataset in Table 5, which shows that our approach with Hausdorff distance loss works the best in all cases.

Effectiveness of Soft Pseudo-Labels. We also test the effectiveness of soft pseudo-labels by replacing them with hard pseudo-labels for the target samples and observe that the average performance decreases from 76.0% to 72.0% on Office-Home dataset. This confirms that soft pseudo-labels are critical in attenuating the unwanted deviations caused by the false and noisy hard pseudo-labels. We provide the performance on each of the transfer tasks from Office-Home in Table 6.

Effectiveness of Different MixUp. We examined the effect of mixup regularization on both domain discriminator and classifier separately in Section 4.3 of the main paper. We concluded that our Mix strategy not only helps to explore intrinsic structures across domains, but also helps to stabilize the domain discriminator. Here, we provide the corresponding performance on each of the transfer tasks of Office-Home in Table 7.

D. Qualitative Results

Feature Visualizations. We use t-SNE [9] to visualize the features learned using different components of our SLM framework. We choose an UDA setup (similar to DANN [5]) as a vanilla method and add different modules one-by-one to visualize their individual contribution in learning discriminative features for partial domain adaptation. As seen from Figure 5, the feature space for vanilla setup lacks dicriminability for both source and target features. The discriminability improves for both source as well as target features as we add "Select" and "Label" to the Vanilla setup. The best results are obtained when all three modules "Select", "Label" and "Mix" i.e., SLM are added and trained jointly in an end-to-end manner.

E. Broader Impact and Limitations

Our research can help reduce burden of collecting largescale supervised data in many real-world applications of visual classification by transferring knowledge from models trained on large broad datasets to specific datasets possessing a domain shift. This scenario is quite common as large datasets (e.g. ImageNet [11]) can be used for training which contain a broader range of categories while our goal can be to transfer the knowledge to smaller datasets with a smaller number of categories. The positive impact that our work could have on society is in making technology more accessible for institutions and individuals that do not have rich resources for annotating newly collected datasets. We

						Office-	Home						
Method	Ar→Cl	Ar→Pr	$Ar \rightarrow Rw$	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	$Pr \rightarrow Rw$	$Rw \rightarrow Ar$	R w→Cl	$Rw \rightarrow Pr$	Average
ResNet-50	$47.2_{\pm 0.2}$	66.8 ± 0.3	$76.9_{\pm 0.5}$	$57.6_{\pm 0.2}$	58.4 ± 0.1	$62.5_{\pm 0.3}$	$59.4_{\pm 0.3}$	$40.6_{\pm 0.2}$	$75.9_{\pm 0.3}$	$65.6_{\pm0.1}$	$49.1_{\pm 0.2}$	75.8 ± 0.4	61.3
DANN	43.2 ± 0.5	$61.9_{\pm 0.2}$	$72.1_{\pm 0.4}$	52.3 ± 0.4	$53.5_{\pm 0.2}$	$57.9_{\pm 0.1}$	47.2 ± 0.3	$35.4_{\pm 0.1}$	70.1 ± 0.3	$61.3_{\pm 0.2}$	$37.0_{\pm 0.2}$	$71.7_{\pm 0.3}$	55.3
CORAL	38.2 ± 0.1	55.6 ± 0.3	$65.9_{\pm 0.2}$	48.4 ± 0.4	$52.5_{\pm 0.1}$	$51.3_{\pm 0.2}$	48.9 ± 0.3	$32.6_{\pm 0.1}$	$67.1_{\pm 0.2}$	$63.8_{\pm0.4}$	35.9 ± 0.2	$69.8_{\pm 0.1}$	52.5
ADDA	45.2	68.8	79.2	64.6	60.0	68.3	57.6	38.9	77.5	70.3	45.2	78.3	62.8
RTN	49.4	64.3	76.2	47.6	51.7	57.7	50.4	41.5	75.5	70.2	51.8	74.8	59.3
CDAN+E	47.5	65.9	75.7	57.1	54.1	63.4	59.6	44.3	72.4	66.0	49.9	72.8	60.7
JDDA	45.8 ± 0.4	$63.9_{\pm 0.2}$	$74.1_{\pm 0.3}$	51.8 ± 0.2	55.2 ± 0.3	$60.3_{\pm 0.2}$	$53.7_{\pm 0.2}$	$38.3_{\pm0.1}$	$72.6_{\pm 0.2}$	$62.5_{\pm0.1}$	$43.3_{\pm0.3}$	$71.3_{\pm 0.1}$	57.7
SPL	46.4 ± 0.0	$70.5_{\pm 0.6}$	$77.2_{\pm 0.0}$	$61.0_{\pm0.0}$	$65.2_{\pm 0.0}$	73.2 ± 0.0	64.3 ± 0.0	$44.7_{\pm 0.0}$	$79.1_{\pm 0.0}$	$69.5_{\pm0.0}$	$58.0_{\pm0.0}$	$79.8_{\pm0.0}$	65.7
PADA	53.2 ± 0.2	$69.5_{\pm 0.1}$	$78.6_{\pm 0.1}$	$61.7_{\pm 0.2}$	$62.7_{\pm 0.3}$	$60.9_{\pm 0.1}$	$56.4_{\pm 0.5}$	44.6 ± 0.2	$79.3_{\pm 0.1}$	$74.2_{\pm 0.1}$	55.1 ± 0.3	$77.4_{\pm 0.2}$	64.5
SAN	44.4	68.7	74.6	67.5	65.0	77.8	59.8	44.7	80.1	72.2	50.2	78.7	65.3
IWAN	53.9	54.5	78.1	61.3	48.0	63.3	54.2	52.0	81.3	76.5	56.8	82.9	63.6
ETN	60.4 ± 0.3	76.5 ± 0.2	77.2 ± 0.3	$64.3_{\pm 0.1}$	67.5 ± 0.3	75.8 ± 0.2	$69.3_{\pm 0.1}$	54.2 ± 0.1	83.7 ± 0.2	75.6 ± 0.3	56.7 ± 0.2	84.5 ± 0.3	70.5
SAFN	58.9 ± 0.5	76.3 ± 0.3	81.4 ± 0.3	70.4 ± 0.5	73.0 ± 1.4	77.8 ± 0.5	72.4 ± 0.3	55.3 ± 0.5	80.4 ± 0.8	75.8 ± 0.4	60.4 ± 0.8	$79.9_{\pm 0.2}$	71.8
DRCN	54.0	76.4	83.0	62.1	64.5	71.0	70.8	49.8	80.5	77.5	59.1	79.9	69.0
RTNet	62.7 ± 0.1	$79.3_{\pm 0.2}$	$81.2_{\pm 0.1}$	$65.1_{\pm 0.1}$	68.4 ± 0.3	$76.5_{\pm 0.1}$	70.8 ± 0.2	$55.3_{\pm 0.1}$	85.2 ± 0.3	$76.9_{\pm 0.2}$	$59.1_{\pm 0.2}$	83.4 ± 0.3	72.0
RTNet _{adv}	$63.2_{\pm 0.1}$	$80.1_{\pm 0.2}$	$80.7_{\pm0.1}$	$66.7_{\pm 0.1}$	$69.3_{\pm0.2}$	$77.2_{\pm 0.2}$	71.6 ± 0.3	$53.9_{\pm 0.3}$	84.6 ± 0.1	$77.4_{\pm 0.2}$	57.9 ± 0.3	85.5 ± 0.1	72.3
BA ³ US	$60.6_{\pm 0.5}$	$\underline{83.2}{\scriptstyle \pm 0.1}$	$\underline{88.4}_{\pm 0.2}$	$\underline{71.8}_{\pm 0.2}$	$72.8_{\pm1.1}$	$83.4_{\pm0.6}$	$\textbf{75.5}_{\pm 0.2}$	$61.6_{\pm 0.4}$	$\underline{86.5}{\scriptstyle \pm 0.2}$	$\underline{79.3}_{\pm 0.7}$	$62.8_{\pm0.5}$	$86.1_{\pm0.3}$	76.0

Table 2: **Performance on Office-Home.** We highlight the **best** and <u>second best</u> method on each task. While the upper section shows results of unsupervised domain adaptation approaches, the lower section shows results of existing partial domain adaptation methods.**SLM** achieves the best average performance among all compared methods.

Office31												
Method	$\mathbf{A} \rightarrow \mathbf{W}$	$\mathbf{D} ightarrow \mathbf{W}$	$\mathbf{W} ightarrow \mathbf{D}$	$\mathbf{A} \rightarrow \mathbf{D}$	$\mathbf{D} \rightarrow \mathbf{A}$	$\mathbf{W} \rightarrow \mathbf{A}$	Average					
VGG-16 [13] (ICLR'15)	$60.3_{\pm 0.8}$	$98.0_{\pm 0.6}$	$99.4_{\pm 0.4}$	$76.4_{\pm 0.5}$	$73.0_{\pm 0.6}$	$79.1_{\pm 0.5}$	81.0					
PADA [2] (ECCV'18)	86.1 ± 0.4	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$81.7_{\pm 0.3}$	$93.0_{\pm 0.2}$	95.3 ± 0.3	92.5					
SAN [1] (CVPR'18)	83.4 ± 0.4	$99.3 _{\pm 0.5}$	$100.0_{\pm 0.0}$	90.7 ± 0.2	87.2 ± 0.2	91.9 ± 0.4	92.1					
IWAN [15] (CVPR'18)	82.9 ± 0.3	79.8 ± 0.3	88.5 ± 0.2	91.0 ± 0.3	89.6 ± 0.2	93.4 ± 0.2	87.5					
ETN [3] (CVPR'19)	85.7 ± 0.2	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$89.4_{\pm 0.2}$	95.9 ± 0.2	$92.3_{\pm 0.2}$	93.9					
SLM (Ours)	$92.0_{\pm 0.1}$	99.8 ± 0.2	99.6 ± 0.5	$98.1_{\pm 0.0}$	$96.1_{\pm 0.0}$	$96.0_{\pm 0.1}$	96.9					

Table 3: **Performance on Office31 with VGG-16 backbone.** Numbers show the accuracy (%) of different methods on partial domain adaptation setting. We highlight the **best** and <u>second best</u> method on each transfer task. Our proposed framework, **SLM** achieves the best performance on 4 out of 6 transfer tasks including the best average performance among all compared methods.

	Modules	5				VisDA-2017						
Select	Label	Mix	$ \mathbf{A} \rightarrow \mathbf{W} $	$\mathbf{D} \to \mathbf{W}$	$W \rightarrow D$	$\mathbf{A} \rightarrow \mathbf{D}$	$\mathbf{D} \to \mathbf{A}$	$W \rightarrow A$	Average	$\mathbf{R} \rightarrow \mathbf{S}$	$S S \rightarrow R$	Average
X	X	X	88.0	98.3	95.8	88.8	84.5	80.2	89.3	57.7	56.4	57.0
\checkmark	X	X	91.8	99.3	96.6	93.8	94.2	93.5	94.9	69.0	68.4	68.7
\checkmark		X	92.4	99.9	99.2	94.9	95.5	93.8	96.0	77.2	84.8	81.0
✓	 ✓ 	1	99.8	100.0	99.8	98.7	96.1	95.9	98.4	77.5	91.7	84.6

Table 4: Effectiveness of Different Modules on Office31 and VisDA-2017 Datasets. Our proposed approach achieves the best performance with all the modules working jointly for learning discriminative invariant features in partial domain adaptation.

	$Ar \rightarrow 0$	$Cl Ar \rightarrow Pr$	$\mathbf{r} \rightarrow \mathbf{R}\mathbf{w}$	V Cl \rightarrow Ar	$\mathbf{Cl} \rightarrow \mathbf{Pr}$	$\mathbf{Cl} \rightarrow \mathbf{Rv}$	$\mathbf{Pr} \to \mathbf{Ar}$	$\mathbf{Pr} \rightarrow \mathbf{Cl}$	$\mathbf{Pr} \rightarrow \mathbf{Rw}$	$\mathbf{R}\mathbf{W} \rightarrow \mathbf{A}\mathbf{W}$	$\mathbf{Pr} \rightarrow \mathbf{Cl}$	$\mathbf{Pr} \rightarrow \mathbf{Rw}$	Average
W/o Hausdorff Loss	56.2	83.1	90.3	72.6	71.5	80.8	71.4	51.6	84.8	82.5	57.5	81.7	73.7
Ours (SLM)	61.1	84.0	91.4	76.5	75.0	81.8	74.6	55.6	87.8	82.3	57.8	83.5	76.0

Table 5: Effectiveness of Hausdorff Triplet Loss on Office-Home Dataset. The table shows the performance of the framework without (top-row) and with (bottom-row) the inclusion of the Hausdorff distance triplet loss. The results highlight the importance of the Hausdorff distance loss in our proposed framework.

	$Ar \rightarrow Cl$	$\mathbf{Ar} \rightarrow \mathbf{Pr}$	$Ar \rightarrow Rw$	$Cl \rightarrow Al$	$r \operatorname{Cl} \to \operatorname{Pr}$	$Cl \rightarrow Rw$	$V \mathbf{Pr} \to \mathbf{Ar}$	$\mathbf{Pr} \rightarrow \mathbf{Cl}$	$\mathbf{Pr} \rightarrow \mathbf{Rw}$	$\mathbf{R}\mathbf{W} \to \mathbf{A}\mathbf{r}$	$\mathbf{Pr} \rightarrow \mathbf{C}$	$\mathbf{Pr} \to \mathbf{Rw}$	Average
W/ Hard Pseudo-labels	52.5	79.9	90.2	73.5	72.6	78.2	69.9	47.5	87.5	78.6	50.6	82.7	72.0
Ours (SLM)	61.1	84.0	91.4	76.5	75.0	81.8	74.6	55.6	87.8	82.3	57.8	83.5	76.0

Table 6: Effectiveness of Soft Pseudo-labels on Office-Home Dataset. Table shows the performance of the framework when we replace the soft pseudo-labels with hard pseudo-labels (top-row) for the target samples. The results justify that the soft pseudo-labels are critical for our framework and attenuate unwanted deviations caused by hard pseudo-labels.

	$\mathrm{Ar} ightarrow$	$\operatorname{Cl}\operatorname{Ar} \to \operatorname{I}$	$\mathbf{Pr} \mathbf{Ar} \to \mathbf{Rw}$	$\mathbf{Cl} \rightarrow \mathbf{A}$	$\mathbf{r} \operatorname{\mathbf{Cl}} ightarrow \mathbf{Pr}$	$Cl \rightarrow Rv$	$\mathbf{Pr} \to \mathbf{Ar}$	$\mathbf{Pr} \rightarrow \mathbf{C}$	$\mathbf{Pr} \rightarrow \mathbf{Rv}$	$\mathbf{w} \mathbf{R} \mathbf{w} \to \mathbf{A} \mathbf{w}$	$\mathbf{r} \ \mathbf{Pr} \to \mathbf{C}$	$Pr \rightarrow Rw$	Average
No Domain Discriminator MixUp	56.2	81.5	90.0	74.0	71.8	80.3	72.2	50.9	86.3	79.8	58.0	82.0	73.6
No Classifier MixUp	57.8	8 82.9	88.5	75.1	73.6	79.3	69.0	54.9	86.6	79.8	57.6	81.2	73.9
Ours (SLM)	61.1	. 84.0	91.4	76.5	75.0	81.8	74.6	55.6	87.8	82.3	57.8	83.5	76.0

Table 7: Effectiveness of Different MixUp on Office-Home Dataset. The table shows the performance of the framework with the exclusion of mixup regularization from the domain discriminator (top-row) and the classifier (middle-row). The final row shows the results of the proposed SLM framework, which provides the best performance confirming the importance of our Mix strategy.



Figure 5: Feature Visualizations using t-SNE. Plots show visualization of our approach with different modules on $A \rightarrow W$, $A \rightarrow D$, $W \rightarrow A$, and $D \rightarrow A$ tasks repectively (top to down) from Office31 dataset. Blue and red dots represent source and target data respectively. As can be seen, features for both target as well as source domain become progressively discriminative and improve from left to right by adoption of our proposed modules. Best viewed in color.

also believe our approach of selecting relevant source data would motivate the research community to extend it to various open-world problems and would help in training more generalizable models. Negative impacts of our research are difficult to predict, however, it shares many of the pitfalls associated with standard deep learning models such as susceptibility to adversarial attacks and lack of interpretability.

References

- Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. In *Proceedings of the IEEE Conference* on Computer Vision and Pattern Recognition, pages 2724– 2732, 2018.
- [2] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings* of the European Conference on Computer Vision (ECCV), pages 135–150, 2018.
- [3] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2985–2994, 2019.
- [4] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 2019.
- [5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.
- [6] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [8] Shuang Li, Chi Harold Liu, Qiuxia Lin, Qi Wen, Limin Su, Gao Huang, and Zhengming Ding. Deep residual correction network for partial domain adaptation. *IEEE Transactions* on Pattern Analysis and Machine Intelligence, 2020.
- [9] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [10] 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.
- [11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- [12] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.
- [13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.
- [14] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 5018–5027, 2017.

[15] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *Proceedings of the IEEE Conference* on Computer Vision and Pattern Recognition, pages 8156– 8164, 2018.