Supplementary: *CoNMix* for Source-free Single and Multi-target Domain Adaptation

1. Algorithmic Details

In the algorithm 1, we have provided the implementation details of the proposed framework *CoNMix* (Refer Page- 2).

2. Theoretical Insights

2.1. Vision Transformer Preliminary

Recently, Vision Transformer (VT) [6, 29] is getting a lot of traction for solving computer vision tasks and surpassing the existing SOTA results. Self-attention module, which provides weight for each input feature vector in VT is very crucial. Let, image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$, where H, W and C are image height, width and channels respectively. If patches are of size $P \times P$ then total number of patches per channel will be $N = \frac{H \times W}{P \times P}$. In VT, these images patches act as sequence of inputs and projected into three different vectors namely Query $Q \in \mathbb{R}^{N \times d_q}$, Key $K \in \mathbb{R}^{N \times d_q}$ and values $V \in \mathbb{R}^{N \times d_v}$. Here d_q and d_v are projected dimension. Finally, we compute the output using the weighted combination of values (V) where the weight is calculated as, $W_{attn} = \text{softmax}(\frac{QK^T}{\sqrt{d_q}})$. Therefore, the weighted output due to self-attention is formulated as $V_{out} = W_{attn}V$.

2.2. Backbone Selection

Notation: We will use h as hypothesis or classifier. $\xi_S(h)$ and $\xi_T(h)$ are expected risk/error of hypothesis h for source domain and target domain respectively. $\xi_T^{VT}(h)$ and $\xi_T^{RN}(h)$ are expected risk/error of hypothesis h for target domain using Vision Transformer (VT) and ResNet (RN) based backbone respectively. $\mathcal{D_S}$ and $\mathcal{D_T}$ are source and target domain distribution respectively. $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D_S}, \mathcal{D_T})$ is divergence between source and target domain distribution and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{X_S}, \mathcal{X_T})$ is it's empirical measure.

Let $\mathcal{H}$ be a hypothesis space of VC dimension d, $\mathcal{X_S}, \ \mathcal{X_T}$ are m unlabeled source and target domain samples, each drawn from $\mathcal{D_S}$ and $\mathcal{D_T}$ respectively, then theorem 2 of [1] states that for any $\delta \in (0, 1)$, with probability at least $1-\delta$ (over the choice of the samples) and, for every $h \in \mathcal{H}$:

$$\xi_T(h) \leq \xi_S(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{X_S}, \ \mathcal{X_T}) + 4\sqrt{\frac{2d\log 2m + \log\frac{2}{\delta}}{m}} + \lambda \tag{1}$$

where λ is the sum of source error and target error w.r.t best available hypothesis $h^* \in \mathcal{H}$. Eq. 1 represents goodness of hypothesis h in target domain, where smaller expected risk $\xi_T(h)$ is desirable for a chosen hypothesis h.
We analyse feature space projection of same set of image samples $\mathcal{X_S}, \ \mathcal{X_T}$ using Vision transformer (VT) as well as ResNet based backbone. For the given $\mathcal{X_S}, \mathcal{Y_S}$ and h, $\xi_S(h)$ will be close to zero because we do supervised training with labeled source data. The term λ will be very small because it is defined w.r.t best possible hypothesis. The 3rd term on the right-hand side of Eq. 1 is constant. Therefore, we can safely assume that the effect on upper bound of ξ_T is largely due to divergence $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{X_S}, \ \mathcal{X_T})$. If we denote the remaining terms with a constant term c then Eq. 1 can be re-formulated as

$$\xi_T(h) \leq \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{X_S}, \ \mathcal{X_T}) + c \tag{2}$$

$\mathcal{A}$-distance is a measure of distance between two distributions. It can be approximated using Eq. 3, if we have large number of samples [13]. Here h' is a binary domain classifier and $d_{\mathcal{A}}$ is denoted as $\mathcal{A}$ - distance between the distribution $\mathcal{D_S}$ and $\mathcal{D_T}$. $\xi(h')$ is the expected risk associated with classifier h' which we measure on latent representation of samples from $\mathcal{D_S}$ and $\mathcal{D_T}$.

$$d_{\mathcal{A}} = 2[1 - 2\min_{h' \in H'} \xi(h')] \tag{3}$$

Ben *et al.* in [2] analysed the bound on expected risk associated with target domain using $\mathcal{A}$ - distance which can be seen as an approximation of divergence $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$. If we replace $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{X_S}, \ \mathcal{X_T})$ by $\mathcal{A}$ - distance ($d_{\mathcal{A}}$) in Eq. 2 then the new expression will be dependent on $\mathcal{A}$ - distance which is shown in Eq. 4

$$\xi_T(h) \leq \frac{1}{2}d_{\mathcal{A}}(\mathcal{X_S}, \ \mathcal{X_T}) + c \tag{4}$$

We can calculate $d_{\mathcal{A}}(\mathcal{X_S}, \ \mathcal{X_T})$ using domain classifier h' for source and target domain samples for VT as well as

Algorithm 1: *CoNMix* Implementation Details

```
Stage 1: Source Training
Input: Source Feature Extractor $g_s(x;\theta_s)$; Source Classifier $h_s(x;\phi_s)$; Source Data $\mathcal{D}_s$;
Maximum Stage-i Epoch $(E_i)_{max}$; Convergence threshold $e_{th}$
while epoch $< (E_1)_{max}$ or $\mathcal{L}_{CE} > e_{th}$ do
    $\{x_i^s, y_i^s\}_{i=1}^B \sim \mathcal{D}_s$ ▷ Randomly sample a batch of source image
    Update $\theta_s, \phi_s$ by minimizing $\mathcal{L}_{CE}$
end
Stage 2: Single Target Domain Adaptation
Input: Set of target domains $\{T_j\}_{j=1}^{N_d}$; $j^{th}$ Target domain data $\mathcal{D}_{t_j}$;Maximum Epoch $(E_2)_{max}$;
Hyper-parameter $\lambda_1, \lambda_2$ and $\lambda_3$; ▷ $N_d$ is Number of target domains
Initialization: $g_{t_j}(x;\theta_{t_j}) \leftarrow g_s(x;\theta_s)$; $h_t(x;\phi_{t_j}) \leftarrow h_s(x;\phi_s)$;
while epoch $< (E_2)_{max}$ do
    $\{x_i^{t_j}\}_{i=1}^B \sim \mathcal{D}_{t_j}$ ▷ Randomly sample a batch of target image
    $\{x_{iw}^{t_j}\}_{i=1}^B = \{A_w(x_i^{t_j})\}_{i=1}^B$ ▷ Weak Augmentation of $i^{th}$sample
    $\{x_{is}^{t_j}\}_{i=1}^B = \{A_s(x_i^{t_j})\}_{i=1}^B$ ▷ Strong Augmentation of $i^{th}$sample
    Obtain Pseudo-Labels from Eq. (9)
    Update $\theta_{t_j}, \phi_{t_j}$ by minimizing loss from Eq. (12)
end
Repeat Stage-2 $\forall j \in \{1, 2, 3, ..., N_d\}$
Stage 3: Multi Target Domain Adaptation
Input: Student Feature Extractor $g_l(x;\theta_l)$; Student Classifier $h_l(x;\phi_l)$; Target Data $\{\mathcal{D}_{t_j}\}_{j=1}^{N_d}$; $j^{th}$ Target Feature
  Extractor $g_{t_j}(x:\theta_{t_j})$; $j^{th}$ Target Classifier $h_{t_j}(x;\phi_{t_j})$;
Initialization: $g_l(x;\theta_l) \leftarrow$ ImageNet pre-trained weights; $h_l(x;\phi_l) \leftarrow$ Random
while epoch $< (E_3)_{max}$ do
    while $j \in \{1, 2, 3, ..., N_d\}$ do
        $\{x_{i_1}^{t_j}, \hat{y}_{i_1}^{t_j}\}_{i_1=1}^B \sim \mathcal{D}_{t_j}$ and $\{x_{i_2}^{t_k}, \hat{y}_{i_2}^{t_k}\}_{i_2=1}^B \sim \mathcal{D}_{t_k}$ ▷ Sample two random batch of target image (any
          domain) with pseudo label
        Get the MixUp sample and label, $\widetilde{x}_{ij} = \lambda x_{i_1}^{t_j} + (1-\lambda) x_{i_2}^{t_k}$ and $\widetilde{y}_{ij} = \lambda \hat{y}_{i_1}^{t_j} + (1-\lambda)\hat{y}_{i_2}^{t_k}$
        Update $\theta_l$ and $\phi_l$ by minimizing $\mathcal{L}_{MKD} = \mathcal{L}_{CE}^{P_l}(\widetilde{x}_{ij}, \widetilde{y}_{ij})$
    end
end
```

MCD	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	15.4	25.5	3.3	44.6	31.2	24.0
inf	24.1	-	24.0	1.6	35.2	19.7	20.9
pnt	31.1	14.8	-	1.7	48.1	22.8	23.7
qdr	8.5	2.1	4.6	-	7.9	7.1	6.0
rel	39.4	17.8	41.2	1.5	-	25.2	25.0
skt	37.3	12.6	27.2	4.1	34.5	-	23.1
Avg.	28.1	12.5	24.5	2.4	34.1	21.2	20.5

CDAN	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	13.5	28.3	9.3	43.8	30.2	25.0
inf	18.9	-	21.4	1.9	36.3	21.3	20.0
pnt	29.6	14.4	-	4.1	45.2	27.4	24.2
qdr	11.8	1.2	4.0	-	9.4	9.5	7.2
rel	36.4	18.3	40.9	3.4	-	24.6	24.7
skt	38.2	14.7	33.9	7.0	36.6	-	26.1
Avg.	27.0	12.4	25.7	5.1	34.3	22.6	21.2

CGDM	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	16.9	35.3	10.8	53.5	36.9	30.7
inf	27.8	-	28.2	4.4	48.2	22.5	26.2
pnt	37.7	14.5	-	4.6	59.4	33.5	30.0
qdr	14.9	1.5	6.2	-	10.9	10.2	8.7
rel	49.4	20.8	47.2	4.8	-	38.2	32.0
skt	50.1	16.5	43.7	11.1	55.6	-	35.4
Avg.	36.0	14.0	32.1	7.1	45.5	28.3	27.2

DeiT-B	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	24.2	49.8	16.8	65.3	53.3	41.9
inf	47.0	-	44.5	5.1	61.2	37.5	39.1
pnt	52.0	21.9	-	7.2	64.4	41.8	37.5
qdr	2.9	0.3	0.3	-	0.5	4.7	1.7
rel	48.4	18.9	47.2	6.4	-	31.5	30.5
skt	59.3	18.8	44.8	17.6	57.5	-	39.6
Avg.	41.9	16.8	37.3	10.6	49.8	33.8	31.7

CDTrans	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	27.9	57.6	27.9	73.0	58.8	49.0
inf	58.6	-	53.4	9.6	71.1	47.6	48.1
pnt	60.7	24.0	-	13.0	69.8	49.6	43.4
qdr	2.9	0.4	0.3	-	0.7	4.7	1.8
rel	49.3	18.7	47.8	9.4	-	33.5	31.7
skt	66.8	23.7	54.6	27.5	68.0	-	48.1
Avg.	47.7	18.9	42.7	17.5	56.5	38.8	37.0

CoNMix	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	24.5	54.3	14.5	71.9	56.3	44.3
inf	42.2	-	39.9	4.9	52.1	34.7	34.8
pnt	59.7	24.0	-	10.6	71.8	53.3	43.9
qdr	27.8	3.0	10.6	-	15.0	22.2	15.7
rel	63.5	27.7	59.5	12.9	-	52.9	43.3
skt	60.7	20.1	49.1	18.8	59.7	-	41.7
Avg.	50.8	19.9	42.7	12.3	54.1	43.9	**37.3**

Table 1: SOTA results comparison on DomainNet for STDA settings. For a given matrix, row and column represents source and target domains respectively for different algorithms. CoNMix currently shows the best performance on DomainNet.

			VisDA-2017				
Domains	*Training*	*Validation*					*Total*
No. of Sample	152,397	55,388					207,785
			Office-31				
Domains	*Amazon*	*Dslr*	*Webcam*				*Total*
No. of Sample	2817	795	498				4,110
			Office-Caltech				
Domains	*Amazon*	*Caltech*	*Dslr*	*Webcam*			*Total*
No. of Sample	958	1123	157	295			2533
			Office-Home				
Domains	*Art (Ar)*	*Clipart (Cl)*	*Product (Pr)*	*RealWorld (Rw)*			*Total*
No. of Sample	2427	4365	4439	4357			15,588
			DomainNet				
Domains	*Clipart (C)*	*Infograph (I)*	*Painting (P)*	*Quickdraw (Q)*	**Real (R)**	*Sketch (S)*	*Total*
No. of Sample	48,837	53,201	75,759	172,500	175,327	70,386	596,010

Table 2: Detailed information for the datasets used.

Method	params	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
EfficientNetV2b3	14.0 M	52.9	73.6	78.7	63.8	76.1	76.5	64.6	53.2	80.7	69.4	57.6	82.5	69.1
EfficientNetV2S	21.5 M	57.5	80.9	85.6	66.7	78.8	83.6	69.6	55.7	85.4	76.2	59.3	87.7	73.9
DeiT-S	22.0 M	63.8	83.7	84.4	73.7	83.3	82.2	73.4	59.9	84.4	75.7	62.3	86.3	76.1

Table 3: Effect of popular backbone selection on *CoNMix*. Even when the backbone parameters are same, DeiT-S backbones perform better than EfficientNet backbones highlighting the benefit of VT backbone in solving adaptation tasks.

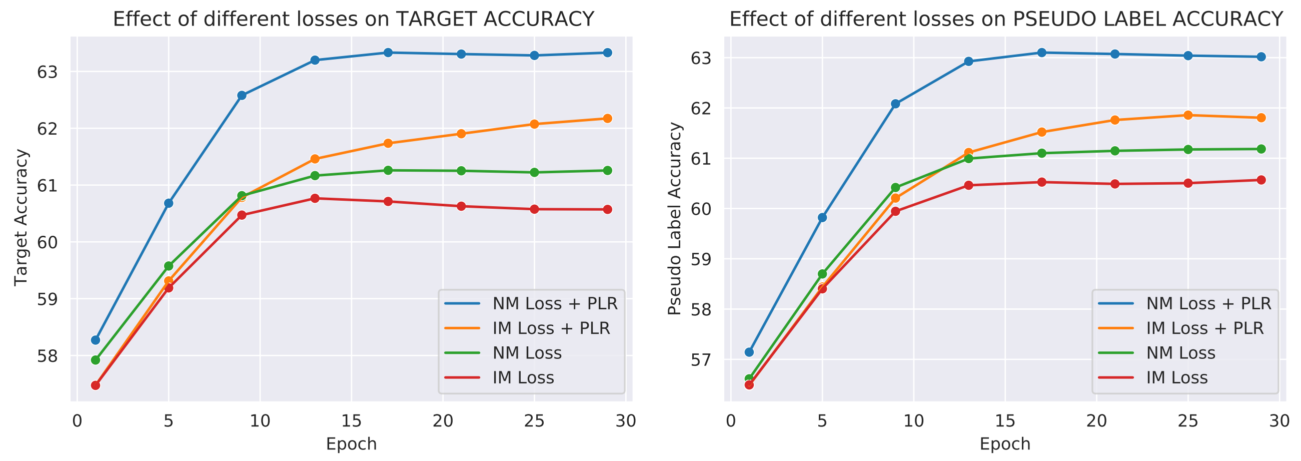

Figure 1: Effect of PLR on NM as well as IM loss for Art → Clipart split of Office-Home dataset. **Target accuracy as well as the pseudo label accuracy improves when add PLR.** Combination of NM with PLR achieves the best performance.

ResNet based backbone. Clearly, the smaller $d_{\mathcal{A}}(\mathcal{X}_{\mathcal{S}},\ \mathcal{X}_{\mathcal{T}})$ will lead to tighter upper bound for $\xi_T(h)$ which will give improved performance. In Fig. 4 of the main paper we see that $d_{\mathcal{A}}(\mathcal{X}_{\mathcal{S}},\ \mathcal{X}_{\mathcal{T}})$ for VT (DeiT-S) backbone is smaller compared to ResNet50. We also observe the performance gain on almost all the datasets when we use VT backbone compared to its ResNet counterpart. Hence, we should prefer VT during backbone selection for solving domain adaptation tasks.

2.3. Theorems

Theorem 2.1. *Let $\|A\|_F$ is the Frobenius norm of classification-response matrix $A \in \mathbb{R}^{B\times K}$ and $\|A\|_*$ is the nuclear norm of A. Then $\|A\|_*$ is the upper bound for $\|A\|_F$. Here, B is batch-size and K is total number of classes.*

$$\|A\|_F \leq \|A\|_* \tag{5}$$

Proof.

$$\|A\|_F = \sqrt{\langle A, A\rangle} = \sqrt{Tr(A^T A)}$$

Method	SF	aero	bicycle	bus	car	horse	knife	motor	person	plant	skate	train	truck	Avg
DANN [8]	✗	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN [18]	✗	68.1	15.4	76.5	87	71.1	48.9	82.3	51.5	88.7	33.2	88.9	42.2	61.1
CaCo [10]	✗	90.4	80.7	78.8	57.0	88.9	87.0	81.3	79.4	88.7	88.1	86.8	63.9	80.9
CAN [12]	✗	97.0	87.2	82.5	74.3	97.8	96.2	90.8	80.7	96.6	96.3	87.5	59.9	87.2
FixBi [20]	✗	96.1	87.8	90.5	90.3	96.8	95.3	92.8	88.7	97.2	94.2	90.9	25.7	87.2
ResNet-101 [9]	✓	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
SHOT-IM [15]	✓	93.7	86.4	78.7	50.7	91.0	93.5	79.0	78.3	89.2	85.4	**87.9**	51.1	80.4
MA [14]	✓	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
SHOT [15]	✓	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
SHOT++ [16]	✓	97.7	88.4	90.2	86.3	97.9	98.6	92.9	84.1	97.1	92.2	93.6	28.8	87.3
CPGA [24]	✓	95.6	89.0	75.4	64.9	91.7	97.5	89.7	83.8	93.9	93.4	87.7	69.0	86.0
CoNMix(ours)	✓	96.2	89.2	83.0	67.8	95.1	95.4	85.0	75.0	90.7	93.2	86.6	55.1	84.4

Table 4: Accuracy (%) on VisDA-2017 [23] for Single Target Domain Adaptation. Only methods within Shaded regions are Source-Free. Magenta represents the best performance for given split across all Source-Free methods.

Performing the Eigen decomposition of A^TA We have,

$$A^TA = VD^2V^T \tag{6}$$

Where, V is the eigen vector of A^TA. After Substituting the value of A^TA in Eq. 6, we have

$$\|A\|_F = \sqrt{Tr(VD^2V^T)} = \sqrt{Tr(D^2V^TV)} = \sqrt{Tr(D^2)}$$

$$\therefore \|A\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i^2} \tag{7}$$

Where r is the rank of A and maximum value of r will be

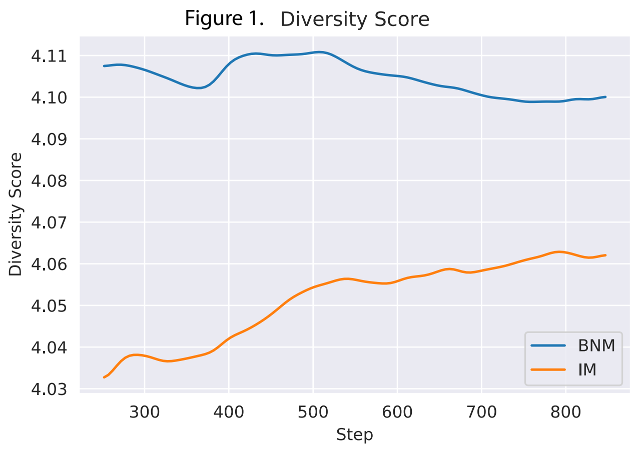

Figure 2: Diversity comparison for NM and IM. Diversity is higher for NM when compared to IM throughout the training.

$\min(B, K)$. Squaring both the sides of Eq.7 we have,

$$(\|A\|_F)^2 = \sum_{i=1}^{r} \sigma_i^2 = (\sum_{i=0}^{r} \sigma_i)^2 - \sum_{i \neq j}^{r} \sigma_i\sigma_j$$

$$= (\|A\|_*)^2 - \sum_{i \neq j}^{r} \sigma_i\sigma_j \leq (\|A\|_*)^2 \quad (\because \sigma \geq 0)$$

$$\therefore \|A\|_F \leq \|A\|_* \tag{8}$$

Please Note that, $\|A\|_* = \sum_{i=1}^{r} \sigma_i$

$$\|A\|_* = \sqrt{(\sum_{i=1}^{r} \sigma_i)^2} \tag{9}$$

$$\leq \sqrt{\sum_{i=1}^{r}(1) \sum_{i=1}^{r}(\sigma_i)^2} \quad \text{(cauchy-schwarz inequality)}$$

$$= \sqrt{r\sum_{i=1}^{r}(\sigma_i)^2}$$

$$= \sqrt{r}\|A\|_F \tag{10}$$

Using Eq. 8 and Eq. 10, we can show that

$$\|A\|_F \leq \|A\|_* \leq \sqrt{r}\|A\|_F \tag{11}$$

□

2.4. MixUP Knowledge Distillation Loss ($\mathcal{L}_{MKD}$)

We represent mixup input and output as $\widetilde{x}_{ij} = \lambda x_i + (1 - \lambda)x_j$ and $\widetilde{y}_{ij} = \lambda y_i + (1 - \lambda)y_j$. Here (x_i, y_i) represents image and pseudo label pairs sampled from i^{th} domain. Let student model output for mixup input $(\widetilde{x}_{ij})$ is $\widetilde{P}_{ij}$ which can be shown as $\widetilde{P}_{ij} = \delta_k(h_l(g_l(\widetilde{x}_{ij})))$. We show the expansion

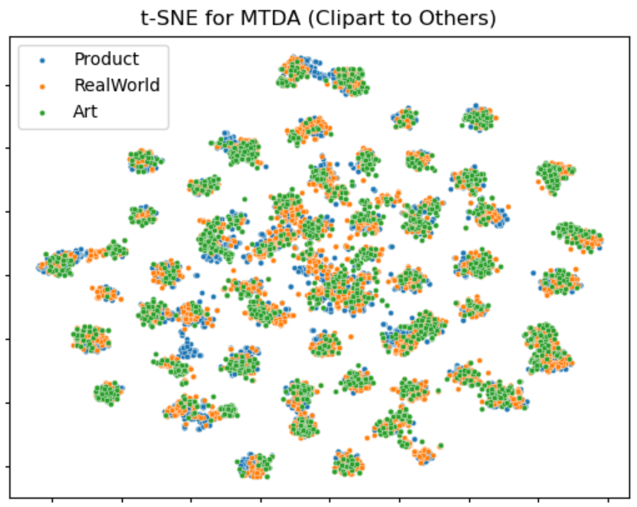

Figure 3: t-SNE plot obtained using sample features from multiple domains are passed through the CoNMix's MTDA model (student network) feature extractor. We can observe that the samples from different domains but belonging to same classes are grouped together.

of $\mathcal{L}_{MKD}$ in Eq. 12.

$$\begin{aligned}
\mathcal{L}_{MKD} &= \mathcal{L}_{CE}^{P_t}(\widetilde{x}_{ij}, \widetilde{y}_{ij}) \\
&= -\sum \widetilde{y}_{ij} \log \tilde{P}_{ij} \\
&= -\sum (\lambda y_i + (1-\lambda) y_j) \log \tilde{P}_{ij} \\
&= -\lambda \sum y_i \log(\tilde{P}_{ij}) \\
&\quad - (1-\lambda) \sum y_j \log(\tilde{P}_{ij}) \\
\mathcal{L}_{MKD} &= \lambda \times \mathcal{L}_{CE}^{P_t}(\widetilde{x}_{ij}, y_i) + (1-\lambda) \times \mathcal{L}_{CE}^{P_t}(\widetilde{x}_{ij}, y_j)
\end{aligned} \tag{12}$$

3. Experiments

3.1. Visualization of Multi Domain Features

We visualise the feature representation of multi-domain samples using t-SNE plots 3. We can observe that the samples from same classes are grouped together irrespective of the domains they come from. It further shows the effectiveness of representation learned using student network in our proposed framework.

3.2. Comparison with Additional Backbones

To further investigate the role backbone in our proposed framework, we perform adaptation using EfficientNetV2-B3 (14M param) as the *CoNMix* backbone. We obtained an accuracy of 69.1% for Office-Home, which is 3% lesser w.r.t RN50 and 7% lesser w.r.t DeiT-S backbone. For further analysis, we change the *CoNMix* backbone to EfficientNetV2-S (22M param, with better ImageNet accuracy compared to DeiT-S) and obtained average accuracy of 73.9%, which is 1.8% higher w.r.t RN50 but 2.2% lesser w.r.t DeiT-S. These observations reaffirm our analysis that the transformer based backbone provides more meaningful representation compared to their CNN counterparts for solving adaptation tasks.

3.3. Source Free Results for DomainNet

SF-STDA results: We can observe from Table. 1 that *CoNMix* is able to achieve state-of-the-art results for source free single target domain adaptation task. One important result to highlight is that *CoNMix* shows an improvement of **0.3%** over CDTrans [32], whereas CDTrans requires source dataset during domain adaptation. Given the large-scale nature of DomainNet dataset, this gain for source-free adaptation task is significant. For VisDA dataset, there is only 2 domains hence MTDA is not possible. We achieved 84.4% on VidDA as it can be seen in Table. 4.

SF-MTDA results: For source-free multi target domain adaptation, we use Vision Transformer (VT) [6] backbone as Teacher network and ResNet101 backbone as Student network. We perform additional experiment by changing the student network as Vision Transformer. Please refer to Table. 6 for the detailed results. Interestingly, our proposed approach CoNMix (VT) outperforms SOTA results on DomainNet by a margin of **3.3%** even though we do not have access to labeled source dataset. Finally, through our extensive experiments, we observe that we achieve the best performance when both the student and teacher network's backbones are vision transformers (VT). We have also evaluated on VisDA dataset in Table. 4.

3.4. Open Domain Test

The aim of open domain test is to check how the model performs when subjected to an unseen domain i.e, the domain whose samples are available only during test time and not available for target adaptation. For this study we divide all the domains into three parts. We assign one domain as source $\mathcal{S}$, other domain T_i for testing (will not be used during target adaptation) and the remaining domains we use for target adaptation $(T_1, T_2, \ldots, T_{i-1}, T_{i+1}, \ldots, T_n)$ where n is the total number of target domains. We show the obtained results in Table. 5. Here, Ar$\rightarrow$ Cl implies that we perform the adaptation experiment by keeping Art domain from Office-Home dataset as source and use Clipart as unseen domain for test. The target domains used for adaptation in this settings are all other domains (Rw,Pr) except Art and Clipart domain. *Please note that the model has no access to unseen test domain during adaptation and it is used only for inference*. Therefore this analysis is important for analyzing the generalization capability of our proposed approach. Table. 5 also shows the effectiveness of using

Method	SF	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
Source train (DeiT-S)	✓	51.7	74.2	79.3	62.6	72.5	74.7	64.0	47.5	**79.6**	**69.9**	49.8	**80.9**	67.2
CoNMix (Hard Distil $\lambda = 1$)	✓	52.5	75.6	78.3	68.6	74.3	78.1	69.1	51.8	77.0	65.8	51.3	74.2	68.1
CoNMix (MixUp Distil $\lambda \neq 1$)	✓	**53.9**	**77.0**	**79.7**	**69.7**	**75.3**	**78.4**	**70.5**	**54.0**	77.7	68.7	**54.0**	74.1	**69.4**

Table 5: Open Domain test Accuracy (%) on Office-Home dataset. *CoNMix* achieves highest Average accuracy. This experiment shows that the proposed approach can generalise over unseen target domains.We can clearly observe that *CoNMix* which uses MixUp KD (MKD i.e, $\lambda \neq 1$) is able to generalise better for unseen target domains. All experiments are performed using DeiT-S backbone.

Method	SF	Cli.	Inf.	Pai.	Qui.	Rea.	Ske.	Avg
SE [7]	✗	21.3	8.5	14.5	13.8	16.0	19.7	15.6
MCD [27]	✗	25.1	19.1	27.0	10.4	20.2	22.5	20.7
DADA [22]	✗	26.1	20.0	26.5	12.9	20.7	22.8	21.5
CDAN [17]	✗	31.6	27.1	31.8	12.5	33.2	35.8	28.7
MCC [11]	✗	33.6	30.0	32.4	13.5	28.0	35.3	28.8
CDAN+DCL [25]	✗	35.1	31.4	37.0	20.5	35.4	41.0	33.4
CGCT [25]	✗	36.1	33.3	35.0	10.0	39.6	39.7	32.3
D-CGCT [25]	✗	37.0	32.2	37.3	19.3	39.8	40.8	34.4
Source train (RN101)	✓	25.6	16.8	25.8	9.2	20.6	22.3	20.1
CoNMix (RN101)	✓	41.8	29.2	39.9	17.5	32.7	41.2	33.7
CoNMix (VT)	✓	**46.7**	**37.6**	**44.6**	**18.3**	**35.2**	**43.7**	**37.7**

Table 6: Accuracy (%) for SF-MTDA on DomainNet dataset. Only methods in shared region are source-free. Only methods within Shaded regions are Source-Free. Magenta represents the best performance for given split across all Source-Free methods. Interestingly, our Source-free approach outperforms the methods which access the labeled source data during target adaptation.

$\mathcal{L}_{NM}$	$\mathcal{L}_{Cons}$	$\mathcal{L}_{CE}^{P_l}$	DeiT	RN50
		✓	6.3	19.3
	✓		3.8	13.8
✓			59.5	50.2
✓	✓		60.3	55.6
✓	✓	✓	63.8	57.6

Table 7: Analysis for adaptation performance when loss components are introduced sequentially for $Ar \rightarrow Cl$ adaptation task of Office-Home dataset. We compare the effect of competitive backbones on proposed *CoNMix* framework and observe that the Vision Transformer is more suitable for the proposed loss.

MixUp Knowledge Distillation (MKD) for distilling knowledge from multiple SF-STDA teacher models. Since the average accuracy of *CoNMix* which uses MKD ($\lambda \neq 1$) is greater than source only training hence we can say that student model isn't memorizing the teacher's prediction. We also checked the variant of MKD which uses simple hard distillation methods ($\lambda = 1$). We can observe that *CoNMix* which uses MKD ($\lambda \neq 1$) is able to generalise better over unseen domains because MixUp creates an intermediate domains and acts as a implicit regularizer. MixUp has already been shown to be effective when used with Vision Transformers [4]. Experiments of Table. 5 shows that *CoNMix+MKD* has better domain generalization capability and therefore, it would be interesting to explore further in this direction.

3.5. Loss ablation study

To gauge the effect of our proposed loss functions on two competitive backbone, we sequentially add all the loss components used in *CoNMix* and show their result in the Table. 7. We can observe that the proposed loss $\mathcal{L}_{NM}$ for source-free adaptation task, works better with DeiT backbone compared to its ResNet50 counterpart. As expected, the performance of $\mathcal{L}_{CL}^{P_l}$ and $\mathcal{L}_{Cons}$ individually is poor due to their inability to handle noise present in the pseudo labels. Further, We perform an experiment using DeiT backbone by omitting $\mathcal{L}_{NM}$ and using only $\mathcal{L}_{CE}^{P_l} + \mathcal{L}_{Cons}$. We can observe from the Fig. 5 that the model performance on adaptation task drops as the training progresses.

3.6. Performance analysis for NM and IM

We define diversity as entropy measures over expected per-class prediction probability (P_c) and denote it as $\mathbb{H} = -\sum_{c=1}^{K} P_c \log P_c$, where $P_c = \frac{1}{N}\sum_{i=1}^{N} P_c^i$, N is the total samples, K is total classes). Higher diversity implies expected per-class prediction distribution should be closer to uniform distribution therefore, higher entropy score is desirable. We analyse diversity for information maximization (IM) as well as nuclear-norm maximization (NM) in Fig. 2 and observe that the NM provides better diversity compared to IM. We also provide comparative analysis using different metrics such target accuracy, pseudo label accuracy for NM and IM loss in Fig. 1 and showcased the benefits of NM over IM. NM provides better class-discriminability as well better diversity in its formulation, which is crucial for its improved performance against IM.

3.7. Analysis using Grad-CAM visualization

To have better class-discriminability and domain-transferability, it is important that the network gives more importance to salient object of interest and discard highly domain-dependent components such as background information. Background information can be the reason for its spurious correlation with the model predictions and it can impact the overall model performance [31]. We plot the Grad-Cam Visualization [28] using target adapted weights for ResNet as well as Vision Transformer based backbone in Fig.4, we observe that Vision Transformer cap-

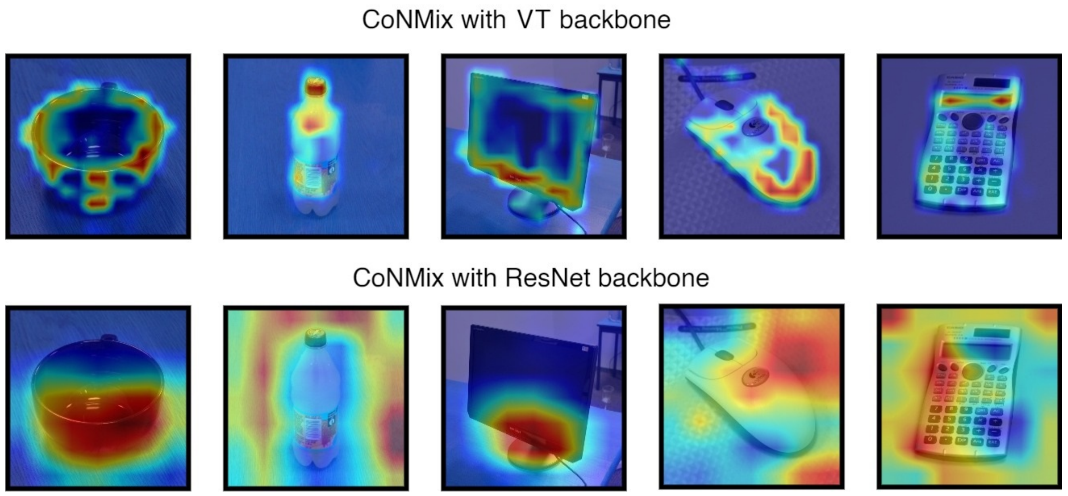

Figure 4: **Grad-CAM visualization**. The above visualization is obtained using two backbone for SF-STDA adapted model on Office31 dataset with Amazon $\rightarrow$ Dslr source-target pair. (*Top*) Grad-CAM visualization using Vision Transformer (Deit-S) backbone shows that the model is able to focus mainly on the salient object of interest i.e, foreground and mostly discards the background. (*Bottom*) Grad-CAM visualization using ResNet-50 backbone is not able to discard background information effectively. Therefore, the learned representation using Vision Transformer will have better domain-transferability and class-discriminability.

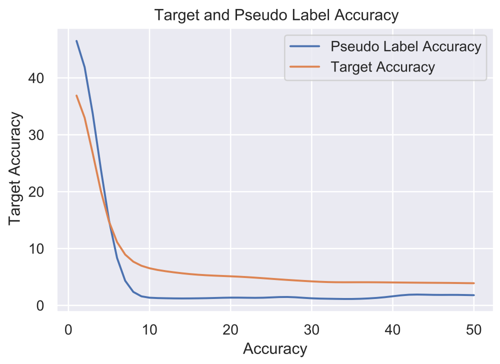

Figure 5: Plot for Pseudo label accuracy and Target Accuracy when we only use $\mathcal{L}_{CE}^{P_l}$ and $\mathcal{L}_{Cons}$. We can observe that the final test accuracy decreases as training proceeds. This is mainly because unlike $\mathcal{L}_{NM}$, $\mathcal{L}_{CE}^{P_l}$ and $\mathcal{L}_{Cons}$ can not handle noise present in the pseudo label.

tures precise details of the important salient object of interest and discard the domain-dependent background information. ResNet based backbone gives importance to the background along with the foreground object of interest. Therefore, it may lead to poor class-discriminability and domain-transferability. This analysis further strengthens the choice of Vision Transformer based backbone for solving domain adaptation tasks.

3.8. Effect of Label Smoothing

Muller *et al*. highlights the effect of label smoothing on generalization and learning speed deep neural network

Src	Target	Test	Accuracy
Cl	Ar	Ar, Pr, Rw	73.5
Cl	Ar, Rw	Ar, Pr, Rw	79.1
Cl	Ar,Pr	Ar, Pr, Rw	79.4
Cl	Ar, Pr, Rw	Ar, Pr, Rw	81.4

Table 8: Effect of the STDA model when we introduce more number of domains. The model is better able to generalise better when more domains are taken as target dataset.

[19] due to implicit regularisation. It is also adopted during source training for solving many unsupervised domain adaptation as well as source-free domain adaptation tasks. To verify the effect of label smoothing on source training, we replaced the soft labels with their one hot class representation. We take this source trained model and perform source-free adaptation. We obtained average performance of 74.2 % Office-Home dataset. Here, we observe a drop of $\approx$ 1.8% for source-free STDA task when compared the situation when we use label smoothing during source-training. This indicates the significance of label smoothing for solving source-free tasks.

3.9. Effect of Seed on Model Performance

We analyse the effect of different seeds on SF-STDA task on Office-31 and Office-Home dataset. We observes small standard deviation of 0.1 and 0.2 on average model performance across different seeds for Office-31 and Office-Home dataset respectively. We performed this analysis for all the task of Office-31 dataset and Real-World to others task for Office-Home dataset. We have added the experiment results in Table. 9 & 10.

4. Training Details

In this section, we discuss image augmentation and hyperparameter details used in our work.

4.1. Augmentation Details

Data Augmentation is an artificial technique used to create transformed versions of images from existing training examples and leads the model to improve performance and generalize well on unseen examples. To enhance the generalizability of our model, we use two types of augmentations i.e. weak augmentation and strong augmentation. In weak augmentation, we transform the image by random cropping or flipping or its combination. To produce strong augmentations, we transform the image by employing RandAugment [5].

4.2. Hyperparameter Details

In this section, we discuss the evaluation protocol, and hyperparameter search. We perform random search for hyperparameter using wandb [3] sweep.

Evaluation Protocol: For evaluating the model performance, we use classification accuracy as our evaluation metric. We compute classification accuracy on unlabelled target dataset for STDA. For multi-target domain adaptation, we report average classification accuracy on the union of target datasets. For e.g, Amazon $\rightarrow$ rest (i.e, Webcam $\cup$ Dslr) we report average accuracy on (Webcam $\cup$ Dslr).

Hyperparameter search: We have three parameters λ_1, λ_2 and λ_3 for each loss component $\mathcal{L}_{NM}, \mathcal{L}_{CE}^{Pl}$ and $\mathcal{L}_{cons}$ respectively and an initial learning rate *lr*. Based on performance evaluation using random search we choose the optimal values for λ_1, λ_2 and λ_3 and *lr*. The learning rate (*lr*) is set to 5e-3 for Office-Home, 3e-3 for Office-31 and 1e-3 for Office-Caltech, VisDA and DomainNet. λ_1 is set to 4.0 for Office-Home, Office-31 and Office-Caltech, 0.5 for VisDA and 7.0 for DomainNet. λ_2 is 0.2 for Office-Home, Office-31 and VisDA, 1.0 for Office-Caltech and 0.6 for DomainNet. λ_3 is equal to 0.2 for all datasets. α in PLR is set to 0.9 for all the datasets.

5. Dataset Details

We provide the detailed instruction of the datasets used in our experiments in Table 2. Additionally, we provide the number of samples in each domain used for training the network on datasets like Office-31 [26], Office-Caltech, Office-Home [30] and large-scale dataset like DomainNet [21] and VisDA-2017 [23]. Please note that, for the DomainNet there is a separate training and test split for each domain.

6. Broader Impact of Domain Adaptation

Domain Adaptation generally works without strong supervision and is targeted toward making a model more generalized and not tied to any particular dataset. Therefore, any work along this direction is bound to have some positive impact, thereby increasing the deployed technology's credibility. Domain Adaptation work like ours acts as a bridge between machine learning driven research and real-world applications by accounting for existing domain shifts. Source-free paradigm of domain adaptation allows to extend it for the applications where data privacy is of utmost importance.

Seed	A→D	A→W	D→A	D→W	W→A	W→D	Avg
2060	92.8	91.7	77.5	98.2	75.4	100	89.3
2080	91.0	92.7	77.6	98.2	76.6	100.0	89.4
2100	92.8	91.6	78.1	98.0	76.7	100.0	89.5
	92.2±1.1	92±0.6	77.7±0.3	98.1±0.1	76.2±0.7	100.0 ±0.0	89.4±0.1

Table 9: SF-STDA results on Office-31 dataset for different seed values. We observes standard deviation of 0.1 on average model performance across different seeds, which indicates the stability of model performance.

Seed	$Rw \rightarrow Ar$	$Rw \rightarrow Cl$	$Rw \rightarrow Pr$	Avg
2060	75.3	62.0	86.0	74.4
2080	75.6	61.0	86.0	74.4
2100	74.8	62.1	85.5	74.1
	75.2 ±0.4	61.7±0.6	85.8±0.3	74.2±0.2

Table 10: SF-STDA results on Office-Home dataset for different seed values. We observes standard deviation of 0.2 on average model performance across different seeds, which indicates the stability of model performance.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

[3] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[4] Jie-Neng Chen, Shuyang Sun, Ju He, Philip HS Torr, Alan Yuille, and Song Bai. Transmix: Attend to mix for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12135–12144, 2022.

[5] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical data augmentation with no separate search. *CoRR*, abs/1909.13719, 2019.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *Proc. ICLR*, 2021.

[7] Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *Proc. ICLR*, 2018.

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

[9] 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.

[10] Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu, and Ling Shao. Category contrast for unsupervised domain adaptation in visual tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1203–1214, June 2022.

[11] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *Proc. ECCV*, 2020.

[12] 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.

[13] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *VLDB*, volume 4, pages 180–191. Toronto, Canada, 2004.

[14] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] 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.

[16] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[17] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 1647–1657, 2018.

[18] Mingsheng Long and Jianmin Wang. Learning transferable features with deep adaptation networks. In *Proc. ICML*, 2015.

[19] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv*, 2019.

[20] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1094–1103, 2021.

[21] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019.

[22] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. *arXiv preprint arXiv:1904.12347*, 2019.

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

[24] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. *CoRR*, abs/2106.15326, 2021.

[25] Subhankar Roy, Evgeny Krivosheev, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Curriculum graph co-teaching for multi-target domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5351–5360, 2021.

[26] 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.

[27] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

[28] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[29] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[30] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proc. CVPR*, 2017.

[31] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

[32] Tongkun Xu, Weihua Chen, Pichao Wang, Fan Wang, Hao Li, and Rong Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. *arXiv preprint arXiv:2109.06165*, 2021.