Appendix: Reducing the Covariate Shift by Mirror Sample in Cross Domain Alignment

Yin Zhao[*][†]
Alibaba Group
yinzhao.zy@alibaba-inc.com

Minquan Wang[*]
Alibaba Group
minquan.wmq@alibaba-inc.com

Longjun Cai
Alibaba Group
longjun.clj@alibaba-inc.com

A Mirror Sample In Optimal Transport Theory

The mirror sample can be defined in terms of optimal transport (OT) theory[1]. It does not affect the understanding of the main paper without this session. We hope this formal definition of mirror samples in OT theory can inspire more theoretic understandings and further researches.

The proposed concept mirror is expected to reflect equivalent sample cross domains. Formally, define the mirror pair as the two realizations of the random variables from the supports of the source and target distributions that play "similar roles" w.r.t. their own distributions. In terms of the optimal transportation theory, let $\mathbb{T}^s$ and $\mathbb{T}^t$ be the two transforms (push-forwards operators) on p^s and p^t such that the resulting distributions are same, i.e. $\mathbb{T}^s_{\#}p^s = \mathbb{T}^t_{\#}p^t$. $x^s \in D^{\mathcal{S}}$ and $x^t \in D^{\mathcal{T}}$ are the mirror for each other if $\mathbb{T}^s_{\#}p^s(x^s) = \mathbb{T}^t_{\#}p^t(x^t)$. In general, infinite number of mirror pairs can be found since $\forall x^s \in D^{\mathcal{S}}$, we could always find $x^t \in D^{\mathcal{T}}$ such that $\mathbb{T}^s_{\#}p^s(x^s) = \mathbb{T}^t_{\#}p^t(x^t)$[1]. An ideal distribution alignment can be achieved by aligning every mirror pairs. However, it is impractical to directly find the mirrors in real application since we only have datasets $X^{\mathcal{S}}$ and $X^{\mathcal{T}}$, which are actually random samplings from the underlying distributions p^s and p^t. The real mirror for $x^s_i \in X^{\mathcal{S}}$ may not exist at all in $X^{\mathcal{T}}$ (but in the support $D^{\mathcal{T}}$). Fig.1 gives an illustrative example of this case. The middle ellipse refers to the aligned domain by the ideal transports $\mathbb{T}^s$ and $\mathbb{T}^t$ from source and target domains. The sample sets, i.e. $\{a, d, e, b, f\}$ and $\{\tilde{a}, \tilde{b}, \tilde{c}, \tilde{h}, \tilde{i}, \tilde{j}\}$ are the training data sampled from their own underlying distributions. So d, e, f and $\tilde{c}, \tilde{h}, \tilde{i}, \tilde{j}$ do not have their mirrors in the opposite domain dataset. Without the real mirrors, it is infeasible to investigate the optimal $\mathbb{T}^s$ and $\mathbb{T}^t$. Admittedly, it is beneficial to find those mirrors rather than imposing certain form of sample-to-sample mapping [10, 6, 7] since the mirrors reflect the natural correspondence of the underlying distributions. That means if the consistency of the mirrors is assured, the model trained in source domain can generalize well in target domain.

B Algorithm Details

The detailed algorithm is presented in Algorithm 1.

C Proofs of Propositions

In order to present the propositions and proofs clearly, we also present the lemma here.

---

[*] Equal Contribution

[†] Corresponding Author

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

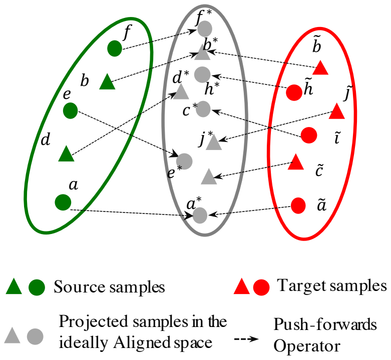


Figure 1: The mirror sample illustration in terms of optimal transport theory

Algorithm 1 Mirror Alignment Algorithm for UDA

```
Require: Source domain dataset $\{x_i^s, y_i^s\}_{i=1}^{n_s}$, target dataset $\{x_j^t\}_{j=1}^{n_t}$, epoch number $N$, batch size $B$, iteration number $T$ per epoch, where $T = \lfloor N/B \rfloor$, hyperparameter $\gamma$.
Ensure: the parameters of backbone $\theta_f$, newly-added FC layer $\theta_g$ and the classifier $\theta_c$
for $epoch = 1$ to $N$ do
    Calculate the source features $f_i^s$, $g_i^s$ for each sample and the class centers $\mu_{f,c}^s$, $\mu_{g,c}^s$ by $\mu_{f,c}^s = \frac{1}{n_c^s}\sum_{y_i^s=c} f_i^s$ and $\mu_{g,c}^s = \frac{1}{n_c^s}\sum_{y_i^s=c} g_i^s$.
    Calculate the target features $f_i^t$, $g_i^t$ using current $\theta_g$, $\theta_f$ and $\theta_c$, generate pseudo labels $y_j^t$ for each target sample by $k$-means [9] initialized by $\mu_{f,c}^s$ and $\mu_{g,c}^s$, then calculate $\mu_f^t = \frac{1}{n_c^t}\sum_{y_f^t=c} f_j^t$ and $\mu_{g,c}^t = \frac{1}{n_c^t}\sum_{y_j^t=c} g_j^t$
    for step =1 to $T$ do
        Choose a batch data $\{x_i^s\}_{b=1}^B$ and $\{x_j^t\}_{b=1}^B$ from $X^{\mathcal{S}}$, $X^{\mathcal{T}}$, with features $\{f_i^s\}_{b=1}^B$, $\{f_i^t\}_{b=1}^B$, $\{g_i^s\}_{b=1}^B$, $\{g_i^t\}_{b=1}^B$.
        Find the mirrors using anchors $\mu_{f,c}^s$, $\mu_{f,c}^t$ for $f$ and $\mu_{g,c}^s$, $\mu_{g,c}^t$ for $g$.
        Calculate total loss by Mirror Loss $\mathcal{L}_{mr,f}$, $\mathcal{L}_{mr,g}$ as well as $\mathcal{L}^s$ and $\mathcal{L}^t$.
        Update parameters $\theta_g$, $\theta_f$ and $\theta_c$ by SGD.
    end for
end for
```

Lemma 1. *[2] Given the hypothesis class $\mathcal{H}$, we have*

$$\forall h \in \mathcal{H}, \mathcal{R}_{\mathcal{T}}(h) \leq \mathcal{R}_{\mathcal{S}}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda \tag{1}$$

where

$$\lambda = \min_{h \in \mathcal{H}} \{\mathcal{R}_{\mathcal{S}}(h, h_{\mathcal{S}}) + \mathcal{R}_{\mathcal{T}}(h, h_{\mathcal{T}})\} \tag{2}$$

and

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) = 2 \sup_{h,h' \in \mathcal{H}} | \Pr_{x \sim D^{\mathcal{S}}}[h(x) \neq h'(x)] - \Pr_{x \sim D^{\mathcal{T}}}[h(x) \neq h'(x)]| \tag{3}$$

$h_{\mathcal{S}}$ and $h_{\mathcal{T}}$ are the labeling function in each domain.

The proof of Lemma 1 is in [2].

Proposition 1. *Denote $\Phi_{\mathcal{S}}(x), \Phi_{\mathcal{T}}(x)$ as the density function for domain $\mathcal{S}$ and $\mathcal{T}$, with supports as $D^{\mathcal{T}}$ and $D^{\mathcal{S}}$ respectively. $\mathcal{H}$ as the hypothesis class from features to label space. If $\Phi_{\mathcal{S}}(x) \overset{a.s.}{=} \Phi_{\mathcal{T}}(x)$, then $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) \to 0$.*

Proof. Denote $\Phi_{\mathcal{S}}(x)$ and $\Phi_{\mathcal{T}}(x)$ as the density function for domain $\mathcal{S}$ and $\mathcal{T}$ in the learned feature space $\mathcal{D}$, with supports as $D^{\mathcal{S}}$ and $D^{\mathcal{T}}$ respectively. $\mathcal{H}$ is the hypothesis class of functions mapping from $\mathcal{D}$ to $\mathcal{Y}$. Following the definition of $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S},\mathcal{T})$ in Lemma 1, we have

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S},\mathcal{T}) = \sup_{h,h'\in\mathcal{H}} | \int_{x\sim D^{\mathcal{S}}} &\Phi_{\mathcal{S}}(x)\mathbb{I}(h(x)\neq h'(x))dx \\ - \int_{x\sim D^{\mathcal{T}}} &\Phi_{\mathcal{T}}(x)\mathbb{I}(h(x)\neq h'(x))dx| \end{aligned} \tag{4}$$

considering the fact that $Pr[x]=E[\mathbb{I}(x)]$, where $\mathbb{I}$ is the indicator function. If $\mathcal{S}$ and $\mathcal{T}$ are aligned in space $\mathcal{D}$, then both the density functions Φ_S, Φ_T and their supports $D^{\mathcal{S}}$, $D^{\mathcal{T}}$ are same. We have

$$\begin{aligned} &| \int_{x\sim D^{\mathcal{S}}} \Phi_{\mathcal{S}}(x)\mathbb{I}(h(x)\neq h'(x))dx \\ &- \int_{x\sim D^{\mathcal{T}}} \Phi_{\mathcal{T}}(x)\mathbb{I}(h(x)\neq h'(x))dx| \\ &\leq \int_{x\sim\mathcal{D}} |\Phi_{\mathcal{S}}(x)-\Phi_{\mathcal{T}}(x)||\mathbb{I}(h(x)\neq h'(x))|dx \to 0 \end{aligned} \tag{5}$$

□

Proposition 2. *Define* $\lambda=\min_{h\in\mathcal{H}}\{\mathcal{R}_{\mathcal{S}}(h,h_{\mathcal{S}})+\mathcal{R}_{\mathcal{T}}(h,h_{\mathcal{T}})\}$ *same to [2], where* $h_{\mathcal{S}}$ *and* $h_{\mathcal{T}}$ *are the labeling functions in each domain. Denote* $\lambda_m+\frac{1}{2}d^m_{\mathcal{H}\Delta\mathcal{H}}$ *as the term of* $\lambda+\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}$ *when* $\mathcal{L}_{mr,x}$ *is minimized. If minimizing* $\mathcal{L}_{mr,x}$ *aligns the distribution in the learned space, we have*

$$\lambda_m+\frac{1}{2}d^m_{\mathcal{H}\Delta\mathcal{H}} \leq \lambda+\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}} \tag{6}$$

Proof. Based on [2], we have

$$\lambda=\min_{h\in\mathcal{H}}\{\mathcal{R}_{\mathcal{S}}(h,h_{\mathcal{S}})+\mathcal{R}_{\mathcal{T}}(h,h_{\mathcal{T}})\} \tag{7}$$

Based on the Proposition 1, $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S},\mathcal{T})$ will approach 0 empirically if $\mathcal{L}_{mr,x}$ is minimized indepent with $\mathcal{H}$. Thus $\forall h\in\mathcal{H}$, we have

$$\begin{aligned} \lambda_m+\frac{1}{2}d^m_{\mathcal{H}\Delta\mathcal{H}} &= \lambda_m \\ &= \min_{h\in\mathcal{H}}\{\mathcal{R}_{\mathcal{S}}(h,h_{\mathcal{S}})+\mathcal{R}_{\mathcal{T}}(h,h_{\mathcal{T}})\} \end{aligned} \tag{8}$$

When the model is trained without $\mathcal{L}_{mr,x}$, we can define a set $\mathcal{H}'\subset\mathcal{H}$ that satisfies $d_{\mathcal{H}'\Delta\mathcal{H}'}=0$.

If $\mathcal{H}'=\emptyset$, the Eq.(6) holds naturally.

If $\mathcal{H}'\neq\emptyset$, then $\forall h\in\mathcal{H}$, we have

$$\begin{aligned} \lambda+\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}} &= \min\{\min_{h\in\mathcal{H}'}\{\mathcal{R}_{\mathcal{S}}(h,h_{\mathcal{S}})+\mathcal{R}_{\mathcal{T}}(h,h_{\mathcal{T}})\}, \\ &\quad \min_{h\in\mathcal{H}-\mathcal{H}'}\{\mathcal{R}_{\mathcal{S}}(h,h_{\mathcal{S}})+\mathcal{R}_{\mathcal{T}}(h,h_{\mathcal{T}})\}+\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}\} \\ &\geq \min_{h\in\mathcal{H}}\{\mathcal{R}_{\mathcal{S}}(h,h_{\mathcal{S}})+\mathcal{R}_{\mathcal{T}}(h,h_{\mathcal{T}})\} \\ &= \lambda_m+\frac{1}{2}d^m_{\mathcal{H}\Delta\mathcal{H}} \end{aligned} \tag{9}$$

since for any subset $\mathcal{H}'\subseteq\mathcal{H}$, $\min_{h\in\mathcal{H}'}\{h\}\geq\min_{h\in\mathcal{H}}\{h\}$. $h_{\mathcal{S}}$ and $h_{\mathcal{T}}$ are the labeling functions for source and target domains [2].

□

D Details of Datasets

We use **Office-31** [25], **Office-Home**[28], **ImageCLEF** and **VisDA2017**[24] to validate our proposed method. Office-31 has three domains: Amazon(A), Webcam(W) and Dslr(D) with 4,110 images belonging to 31 classes. Office-Home is a more challenging benchmark dataset for unsupervised domain adaption. It contains 15,500 images of 65 classes with four domains: Art(Ar), Clipart(Cl), Product(Pr) and RealWorld(Rw). ImageCLEF contains 600 images of 12 classes, where the images are divided into three domains: Caltech-256(C), ILSVRC 2012(I), Pascal VOC 2012(P). For the above three datasets, we use all the adaption tasks. VisDA2017 is a large-scale dataset which contains ∼280K images belonging to 12 classes. These images are divided into three parts: train, validation and test. We use "train" as source domain and "validation" as target domain. The source domain is composed of 152,397 images, which are generated from synthetic renderings of 3D models. The target domain is composed of 55,388 images cropped from Microsoft COCO dataset [20].

E Detailed Results

E.1 Experiment Results with SOTA methods

In this section, we describe the task-level detailed results of our experiment comparing with SOTA methods. Table 1 shows the result of 6 tasks on Office-31 dataset. Compared with the SOTA result of SRDC [26], the average accuracy of our model increases by 0.3%. Specially, we achieve a 2.4% improvement on task A to W, 1.2% on task A to D. Table 2 shows the result on Office-Home. Table 3 shows the result of 6 tasks on ImageCLEF dataset. For all tasks, our method gains a new SOTA and the average accuracy is 91.6% which has a 0.7% improvement. For large-scale dataset VisDA2017, we migrated the proposed mirror loss to the existing method CAN [16]. We can observe from Table 4 that the average accuracy increases by 0.7% over the SOTA.

Table 1: Test accuracy(%) on Office-31 dataset for unsupervised domain adaptation based on ResNet50.

Method	A-W	D-W	W-D	A-D	D-A	W-A	Avg
Source Model [13]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DAN [22]	81.3±0.3	97.2±0.0	99.8±0.0	83.1±0.2	66.3±0.0	66.3±0.1	82.3
DANN [11]	81.7±0.2	98.0±0.2	99.8±0.0	83.9±0.7	66.4±0.2	66.0±0.3	82.6
ADDA [27]	86.2±0.3	78.8±0.4	96.8±0.2	99.1±0.2	69.5±0.1	68.5±0.1	83.2
JDDA [4]	82.6±0.4	95.2±0.2	99.7±0.0	79.8±0.1	57.4±0.0	66.7±0.2	80.2
MCSD [33]	94.9±0.3	99.1±0.1	100.0±0.0	95.6±0.3	77.6±0.4	77.0±0.3	90.7
DSR [3]	93.1	98.7	99.8	92.4	73.5	73.9	88.6
DM-ADA [29]	83.9±0.4	99.8±0.1	99.9±0.1	77.5±0.2	64.6±0.4	64.0±0.5	81.6
rRevGrad+CAT [8]	94.4±0.1	98.0±0.2	100.0±0.0	90.8±1.8	72.2±0.6	70.2±0.1	87.6
SAFN [30]	90.1±0.8	98.6±0.2	99.8±0.0	90.7±0.5	73.0±0.2	70.2±0.3	87.1
MDD [30]	94.5±0.3	98.4±0.1	100.0±0.0	93.5±0.2	74.6±0.3	72.2±0.1	88.9
CAN [16]	94.5±0.3	99.1±0.2	99.8±0.2	95.0±0.3	78.0±0.3	77.0±0.3	90.6
RSDA-MSTN [12]	96.1±0.2	99.3±0.2	100.0±0.0	95.8±0.3	77.4±0.8	78.9±0.3	91.1
SHOT [19]	90.9	98.8	99.9	93.1	74.5	74.8	88.7
SRDC [26]	95.7±0.2	99.2±0.1	100.0±0.0	95.8±0.2	76.7±0.3	77.1±0.1	90.8
BSP-TSA [5]	93.3±0.2	98.2±0.2	100.0±0.0	93.0±0.2	73.6±0.3	72.6±0.3	88.5
FixBi [23]	96.1±0.2	99.3±0.2	100.0±0.0	95.0±0.4	**78.7±0.5**	**79.4±0.3**	91.4
Ours	**98.5±0.3**	**99.3±0.1**	**100.0±0.0**	**96.2±0.1**	77.0±0.1	78.9±0.1	**91.7**

Table 2: Test accuracy(%) on Office-Home dataset for unsupervised domain adaptation based on ResNet50.

Method	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 Model [13]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [22]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [11]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
DSR [3]	53.4	71.6	77.4	57.1	66.8	69.3	56.7	49.2	75.7	68.0	54.0	79.5	64.9
MDD [34]	54.9	73.7	77.8	60.2	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
JDDA [4]	46.4	60.1	70.3	48.3	59.3	61.3	47.3	44.5	68.9	64.1	53.7	77.8	58.5
MCSD [33]	51.6	76.9	80.3	68.6	71.8	78.3	65.8	50.5	81.2	73.1	54.2	82.4	69.6
SAFN [30]	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
CAN [16]	53.4	76.8	77.6	63.0	75.0	73.4	63.3	53.8	77.5	72.9	58.2	81.7	68.9
RSDA-MSTN [12]	53.2	**77.7**	81.3	66.4	74.0	76.5	67.9	53.0	82.0	75.8	57.8	85.4	70.9
SHOT [19]	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
SRDC [26]	52.3	76.3	81.0	69.5	76.2	78.0	68.7	53.8	81.7	76.3	57.1	85.0	71.3
BSP-TSA [5]	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
FixBi [23]	**58.1**	77.3	80.4	67.7	**79.5**	78.1	65.8	**57.9**	81.7	76.4	**62.9**	**86.7**	72.7
Ours	57.6	77.6	**81.6**	**71.9**	77.8	**78.7**	**72.0**	56.3	**82.5**	**77.9**	61.3	85.3	**73.4**

Table 3: Test accuracy(%) on ImageCLEF dataset for unsupervised domain adaptation based on ResNet50.

Method	I-P	P-I	I-C	C-I	C-P	P-C	Avg
Source Model [13]	74.8±0.3	83.9±0.1	91.5±0.3	78.0±0.2	65.5±0.3	91.2±0.3	80.7
DAN [22]	74.5±0.3	82.2±0.2	92.8±0.2	86.3±0.4	69.2±0.4	89.8±0.4	82.5
DANN [11]	75.0±0.6	86.0±0.3	96.2±0.4	87.0±0.5	74.3±0.5	91.5±0.6	85.0
rRevGrad+CAT [8]	77.2±0.2	91.0±0.3	95.5±0.3	91.3±0.3	75.3±0.6	93.6±0.5	87.3
MDD [34]	77.8±0.3	92.2±0.1	97.2±0.2	92.2±0.1	76.7±0.3	95.0±0.2	88.5
JDDA [4]	77.5±0.2	86.7±0.2	86.3±0.1	86.3±0.3	72.5±0.2	90.6±0.1	83.3
SAFN [30]	79.3±0.1	93.3±0.4	96.3±0.4	91.7±0.0	77.6±0.1	95.3±0.1	88.9
CAN [16]	78.2±0.3	93.8±0.1	97.5±0.2	93.2±0.1	77.0±0.2	97.8±0.2	89.6
RSDA-MSTN [12]	79.8±0.2	94.5±0.5	98.0±0.4	94.2±0.4	79.2±0.3	97.3±0.3	90.5
SHOT [19]	78.3±0.2	90.2±0.1	94.3±0.3	88.8±0.2	76.3±0.2	95.2±0.5	87.2
SymNets [32]	80.2±0.3	93.6±0.2	97.0±0.3	93.4±0.3	78.7±0.3	96.4±0.1	89.9
MCSD [33]	79.2±0.2	**96.2±0.3**	96.8±0.1	93.8±0.2	77.8±0.4	96.2±0.0	90.0
SRDC [26]	80.8±0.3	94.7±0.2	97.8±0.2	94.1±0.2	80.0±0.3	97.7±0.1	90.9
BSP-TSA [5]	78.5	90.8	96.2	93.2	79.3	95.7	89.9
FixBi [23]	75.7±0.4	90.7±0.2	94.0±0.1	89.5±0.4	71.7±0.2	94.2±0.4	86.0
Ours	**82.4±0.1**	95.3±0.1	**97.9±0.2**	**95.2±0.2**	**81.0±0.1**	**98.0±0.1**	**91.6**

Table 4: Test accuracy(%) on VisDA dataset for unsupervised domain adaptation based on ResNet101.

Method	airplane	bicycle	bus	car	horse	knife	motorcycle	persion	plant	skateboard	train	truck	Avg
Source Model [13]	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
DAN [22]	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
JDDA [4]	88.2	65.4	77.5	44.9	90.6	44.3	86.3	54.2	52.3	37.5	85.9	23.0	62.5
SAFN [30]	93.6	61.3	84.1	70.6	94.1	79.0	**91.8**	79.6	89.9	55.6	**89.0**	24.4	76.1
SHOT [19]	92.6	81.1	80.1	58.5	89.7	86.1	81.5	77.8	89.5	84.9	84.3	49.3	79.6
BSP-TSA [5]	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
FixBi [23]	96.1	87.8	90.5	90.3	96.8	95.3	92.8	88.7	87.2	94.2	90.9	25.7	87.2
CAN [16]	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
CAN+Mirror(Ours)	**97.2**	**88.2**	**84.9**	**76.0**	97.2	95.8	89.2	**86.4**	96.1	**96.6**	85.9	**61.2**	**87.9**

E.2 Detailed Results for Ablation Study

Mirror Loss Ablation Study. The task-level results for Ablation Study for Office-Home and Office-31 are in Table 5 and Table 6 respectively.

Table 5: Ablation Results for Office-Home

Model Setups	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.
Baseline	46.3	70.0	76.4	66.0	71.4	73.1	67.0	49.8	76.3	70.6	52.0	79.4	66.5
Bk Mirror	51.4	76.2	82.2	70.3	76.8	78.5	70.6	55.0	82.4	76.7	56.8	85.2	71.8
FC Mirror	52.1	74.9	**81.7**	70.3	76.6	77.9	70.6	55.3	82.2	76.4	58.1	84.8	71.7
FC + Bk Mirror	**57.6**	**77.6**	81.6	**71.9**	**77.8**	**78.7**	**72.0**	**56.3**	**82.5**	**77.9**	**61.3**	**85.3**	**73.4**

Table 6: Ablation Results for Office-31

Model Setups	A-W	D–W	W–D	A–D	D–A	W–A	Avg.
Baseline	92.8	94.1	93.7	91.7	69.5	71.4	85.5
Bk Mirror	96.6	98.7	99.0	95.4	73.6	76.4	90.0
FC Mirror	95.5	98.9	99.0	94.8	74.3	76.0	89.7
FC + Bk Mirror	**98.5**	**99.3**	**100.0**	**96.2**	**77.0**	**78.9**	**91.7**

Sensitivity of k. The task-level results for parameters k w.r.t. Office-Home and Office-31 are given in Table 7 and Table 8. We can see that the best choice of k for both Office-Home and Office-31 is 3. But for Office-Home, when $k = 5$, there are two tasks (i.e. Ar to Rw and Pr to Rw) achieving the best accuracy; while for Office-31, when $k = 1$ there are also two tasks (i.e. A to W and W to D) having the best results. When k is large, such as 7 or 9, the overall accuracy drops significantly, meaning that large k leads to inaccurate mirror estimations.

Robustness of Mirror Construction The mirror sample is estimated as $\tilde{x}^s(x_j^t) = \sum_{x\in\tilde{X}^S(x_j^t)} \omega(x, x_j^t)x$, where $\omega(x, x_j^t)$ is the weight of the element x in the mirror set $\tilde{X}^{\mathcal{S}}(x_j^t)$. Besides the k, we tried different methods of calculating $\omega(x, x_j^t)$ to investigate the impacts. One is $\omega(x, x_j^t) = e^{-d(x,x_j^t)} / \sum_{x\in\tilde{X}^s(x_j^t)} e^{-d(x,x_j^t)}$, which is denoted as "weighted mirror sample" the other

Table 7: The influence of k in Mirror Selector for Office-Home

k	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.
1	52.3	75.3	81.4	71.0	76.9	77.3	70.0	54.6	82.3	76.6	57.2	85.2	71.7
3	**57.6**	**77.6**	**81.6**	**71.9**	**77.8**	**78.7**	**72.0**	**56.3**	**82.5**	**77.9**	**61.3**	**85.3**	**73.4**
5	51.5	75.4	81.5	70.7	76.2	76.9	70.9	54.9	82.4	77.0	58.7	84.9	71.8
7	51.2	74.7	80.9	70.3	76.4	76.1	70.3	54.6	81.6	76.6	58.2	84.9	71.3
9	51.1	74.7	80.5	70.9	76.3	76.4	70.9	55.0	81.6	77.2	57.9	84.7	71.4

Table 8: The influence of k in Mirror Selector for Office-31

k	A-W	D–W	W–D	A–D	D–A	W–A	Avg.
1	98.4	98.6	**100.0**	95.6	74.8	77.4	90.2
3	**98.5**	**99.3**	**100.0**	**96.2**	**77.0**	**78.9**	**91.7**
5	95.6	99.2	99.8	96.2	74.5	78.1	90.3
7	96.3	98.3	98.6	95.8	72.9	76.6	89.8
9	95.5	98.9	99.0	94.5	74.2	76.7	89.8

is $\omega(x, x_i^t) = 1/k$. For the distance d, besides the Eculidean distance, we also use Gaussian kernel based distance with standard deviation as 1. The results are in Table 9. We can see that the mentioned variation does not impact the average results too much. Although using the Euclidean distance with constant weight $1/k$ have the best results, the other combinations can still have competitive results. This means the proposed method is robust w.r.t. the distance and weight definition in the construction method.

Table 9: Robusness analysis w.r.t. the mirror construction($k = 3$) for Office-Home. The we use Gaussian kernel with standard devariation as 1.

d	weight	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
Euclidean	$\propto$ 1/d	52.0	76.7	81.6	71.2	76.5	78.9	70.5	55.5	82.2	78.2	59.8	85.3	72.4
Gaussian	$\propto$ 1/d	53.1	76.8	81	71.8	77.4	79.0	69.7	55.3	82.1	76.4	59.7	84.8	72.3
Gaussian	$1/k$	52.0	75.6	81.9	71.9	77	78.6	70.5	54.8	82.1	77.3	58.5	85.1	72.1
Euclidean	$1/k$	**57.6**	**77.6**	**81.6**	**71.9**	**77.8**	**78.7**	**72.0**	**56.3**	**82.5**	**77.9**	**61.3**	**85.3**	**73.4**

Sensitivity of γ. Then sensitivity analysis of γ is given in Table 10 and 12 on Office-Home and Office-31 Dataset. The best choice in those 2 datasets are 1.0. For Office-Home, γ can be between 1.0 and 2.0 with little performance decrease. For Office-31, the empirical γ should be between 0.0 and 1.0.

Comparison with Generative Methods Generating virtual sample of target domain to train the domain-agnostic classifier is one series of method. The typical works include CoGAN([21]), CyCADA([15]),CrDoCo([18]), DRANet[17] etc.. However as CrDoCo stated, those works are working to capture the pixel-wise domain adaptation. Although the generated samples can be seen as the mirror sample, they only apply for the domain gap like "style" difference. That's why CyCADA and its following work were justified mainly in image segmentation benchmark. Our proposed method, from another viewpoint, uses the samples of the target domain to generate the mirror sample. By definition, our proposed virtual sample generation does not presume any form of the domain shift. We also carry out experimental comparison on Digit Dataset with those typical methods in Table 11. We can see that our method outperforms them by large margins, justifying the claims above.

Table 10: The sensitivity of γs for the Mirror Loss for Office-Home

γ	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.
0.0	51.2	74.8	80.7	70.7	76.1	77.5	70.1	54.6	81.8	76.2	56.8	84.3	71.2
1.0	**57.6**	**77.6**	**81.6**	**71.9**	**77.8**	**78.7**	**72.0**	**56.3**	**82.5**	**77.9**	**61.3**	**85.3**	**73.4**
2.0	51.8	75.4	80.8	71.0	76.0	77.4	70.5	55.0	81.7	77.7	59.1	84.3	71.7
3.0	48.4	74.2	79.6	70.9	77.0	76.8	69.9	53.9	80.7	**77.9**	58.4	84.1	71.0

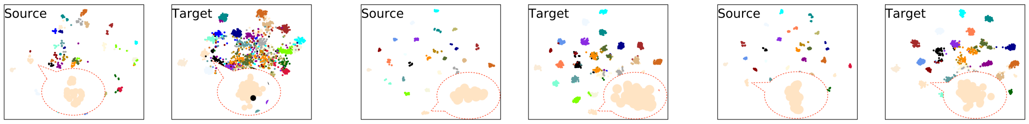
Visualizations by t-SNE on Office-31. In this section, we also visualize feature distribution by t-SNE [14] on Office-31 to visually show the alignment procedure for the underlying distribution. We can see that samples belonging to different classes gradually approach the center of their classes as the training progresses. Specially, in Fig.2(e) and Fig.2(f), the "shape" is more similar between source and target domains by using mirror loss.This reflects the proposed mirror and mirror loss have achieved higher consistency between the underlying distributions.

Table 11: Comparison with Generative Methods on Digits Dataset

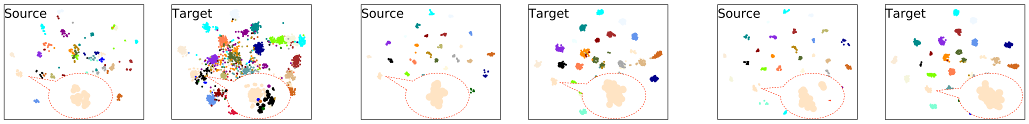
Methods	USPS → MNIST	SVHN → MNIST	MNIST → USPS
CoGAN[21]	89.1	–	91.2
LC+CycleGAN[31]	98.3	97.5	97.1
CyCADA[15]	96.5	90.4	95.6
DRANet[17]	97.8	–	97.8
Ours	**99.2**	**99.1**	**99.3**

Table 12: The sensitivity of γs for the Mirror Loss for Office-31

γ	A-W	D–W	W–D	A–D	D–A	W–A	Avg.
0.0	95.4	97.2	97.3	94.2	74.8	77.4	90.2
1.0	**98.5**	**99.3**	**100.0**	**96.2**	**77.0**	**78.9**	**91.7**
2.0	95.2	98.4	99.0	92.8	71.7	76.1	88.9
3.0	93.8	99.0	98.8	92.2	70.1	75.0	88.1



(a) W/O Mirror, W-A, epoch 1 (b) W/O Mirror, W-A, epoch 100 (c) W/O Mirror, W-A, epoch 200



(d) Mirror, W-A, epoch 1 (e) Mirror, W-A, epoch 100 (f) Mirror,W-A, epoch 200

Figure 2: Visualization of cluster evolvement for task W-A in Office-31. Different classes are distinguished by color.

References

[1] Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

[2] 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(1-2):151–175, 2010.

[3] Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. Learning disentangled semantic representation for domain adaptation. In *IJCAI: proceedings of the conference*, volume 2019, page 2060. NIH Public Access, 2019.

[4] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3296–3303, 2019.

[5] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International conference on machine learning*, pages 1081–1090. PMLR, 2019.

[6] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 3730–3739, 2017.

[7] Bharath Bhushan Damodaran, Benjamin Kellenberger, Remi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[8] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9944–9953, 2019.

[9] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556, 2004.

[10] R Flamary, N Courty, D Tuia, and A Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell*, 2016.

[11] 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.

[12] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9101–9110, 2020.

[13] 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.

[14] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 857–864, 2003.

[15] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[16] 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.

[17] Seunghun Lee, Sunghyun Cho, and Sunghoon Im. Dranet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15252–15261, 2021.

[18] Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Zi Huang. Cycle-consistent conditional adversarial transfer networks. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 747–755, 2019.

[19] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[21] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. *Advances in neural information processing systems*, 29:469–477, 2016.

[22] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105, 2015.

[23] 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.

[24] 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.

[25] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. permission. transferring visual category models to new domains. 2010.

[26] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8725–8735, 2020.

[27] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[28] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. 2017.

[29] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. *arXiv preprint arXiv:1912.01805*, 2019.

[30] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1426–1435, 2019.

[31] Shaokai Ye, Kailu Wu, Mu Zhou, Yunfei Yang, Sia Huat Tan, Kaidi Xu, Jiebo Song, Chenglong Bao, and Kaisheng Ma. Light-weight calibrator: a separable component for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13736–13745, 2020.

[32] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5031–5040, 2019.

[33] Yabin Zhang, Bin Deng, Hui Tang, Lei Zhang, and Kui Jia. Unsupervised multi-class domain adaptation: Theory, algorithms, and practice. *arXiv preprint arXiv:2002.08681*, 2020.

[34] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I Jordan. Bridging theory and algorithm for domain adaptation. *arXiv preprint arXiv:1904.05801*, 2019.