KD3A: Unsupervised Multi-Source Decentralized Domain Adaptation via Knowledge Distillation (Appendix)

Hao-zhe Feng [1] **Zhaoyang You** [1] **Minghao Chen** [1] **Tianye Zhang** [1] **Minfeng Zhu** [1]
Fei Wu [1] **Chao Wu** [1] **Wei Chen** [1]

1. Appendix A

Claim *For the extended source domain* $\mathbb{D}_S^{K+1} = \{(\mathbf{X}_i^T, \mathbf{p}_i)\}_{i=1}^{N_T}$*, training the related source model* h_S^{K+1} *with the knowledge distillation loss* $L^{kd}(\mathbf{X}_i^T, q_S^{K+1}) = D_{\mathrm{KL}}(\mathbf{p}_i \| q_S^{K+1}(\mathbf{X}_i^T))$ *equals to optimizing the task risk* $\epsilon_{\mathbb{D}_S^{K+1}}(h) = \Pr_{(\mathbf{X},\mathbf{p})\sim\mathbb{D}_S^{K+1}}[h(\mathbf{X}) \neq \arg_c \max \mathbf{p}_c]$.

Proof:

First, we prove that $\forall c = 1, \ldots, C$,

$$|q_S^{K+1}(\mathbf{X}_i^T))_c - \mathbf{p}_{i,c}| \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(\mathbf{p}_i \| q_S^{K+1}(\mathbf{X}_i^T))} \quad (1)$$

The widely used **Pinsker's inequality** states that, if P and Q are two probability distributions on a measurable space $(\mathbf{X}, \Sigma)$, then

$$\delta(P, Q) \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(P \| Q)}$$

where

$$\delta(P, Q) = \sup\{|P(\mathbf{A}) - Q(\mathbf{A})| | \mathbf{A} \in \Sigma, \\ \Sigma \text{ is a measurable event.}\}$$

In our situation, we choose the event $\mathbf{A}$ as the probability of classifying the input $\mathbf{X}_i^T$ into class c, and the related probability under P, Q is $\mathbf{p}_{i,c}$ and $q_S^{K+1}(\mathbf{X}_i^T))_c$. With **Pinsker's inequality**, it is easy to prove (1). Since the inequality (1) holds for all class c, minimizing the knowledge distillation loss will make $q_S^{K+1}(\mathbf{X}_i^T)) \to \mathbf{p}_i$, that is, $\epsilon_{\mathbb{D}_S^{K+1}}(h) \to 0$.

2. Appendix B

Proposition 1 (*The generalization bound for knowledge distillation). Let* $\mathcal{H}$ *be the model space and* $\epsilon_{\mathbb{D}_S^{K+1}}(h)$ *be the task risk of the new source domain* $\mathbb{D}_S^{K+1}$ *based on knowledge distillation. Then for all* $h_T \in \mathcal{H}$*, we have:*

$$\epsilon_{\mathbb{D}_T}(h_T) \leq \epsilon_{\mathbb{D}_S^{K+1}}(h_T) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T) \\ + \min\{\lambda_1, \sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_T}(h)|\} \quad (2)$$

where λ_1 is a constant for the task risk of the optimal model.

Proof:

Following the Theorem 2 in Ben-David et al. (2010), for the source domain $\mathbb{D}_S^{K+1}$ and the target domain $\mathbb{D}_T$, for all $h_T \in \mathcal{H}$, we have

$$\epsilon_{\mathbb{D}_T}(h_T) \leq \epsilon_{\mathbb{D}_S^{K+1}}(h_T) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T) + \lambda_1 \quad (3)$$

where λ_1 is constant of the optimal model on the source domain and the target domain as $\lambda_1 = \min_{h\in\mathcal{H}} \epsilon_{\mathbb{D}_S^{K+1}}(h) + \epsilon_{\mathbb{D}_T}(h)$.

In addition, the following inequality also holds for all $h_T \in \mathcal{H}$:

$$\epsilon_{\mathbb{D}_T}(h_T) - \epsilon_{\mathbb{D}_S^{K+1}}(h_T) \leq \sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_T}(h) - \epsilon_{\mathbb{D}_S^{K+1}}(h)| \quad (4)$$

where $\sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_T}(h) - \epsilon_{\mathbb{D}_S^{K+1}}(h)|$ is the upper bound of the task risk gap between the target domain $\mathbb{D}_T$ and the extended domain $\mathbb{D}_S^{K+1}$. Notice $\mathbb{D}_S^{K+1}$ shares the same input space with $\mathbb{D}_T$ since they all use $\{X_i^T\}_{i=1}^{N_T}$ as inputs. Therefore, we have

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T) = 0 \quad (5)$$

Substituting (5) into (4), we have

$$\epsilon_{\mathbb{D}_T}(h_T) \leq \epsilon_{\mathbb{D}_S^{K+1}}(h_T) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T) + \\ \sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_T}(h) - \epsilon_{\mathbb{D}_S^{K+1}}(h)| \quad (6)$$

Combining (3) and (6), we get the **Proposition 1**.

The learning bound with empirical risk error. Proposition 1 shows how to relate the extended source domain $\mathbb{D}_S^{K+1}$ and the target domain $\mathbb{D}_T$. Since we use the finite samples to empirically estimate the $\hat{\epsilon}_{\mathbb{D}_S^{K+1}}(h)$ and

[1]College of Computer Science and Technology, Zhejiang University, Zhejiang, China. Correspondence to: Wei Chen <chenvis@zju.edu.cn>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

$\hat{d}_{\mathcal{H}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T)$ at the training time, We now proceed to give a learning bound for empirical risk minimization using N_T sampled training data.

Following the learning bound **Lemma 1,5** in Ben-David et al. (2010), for all $0 < \delta < 1$, with probability at least $1-\delta$, we have:

$$\begin{aligned}\epsilon_{\mathbb{D}_S^{K+1}}(h) &\leq \hat{\epsilon}_{\mathbb{D}_S^{K+1}}(h) + \sqrt{\frac{4}{N_T}(d\log\frac{2eN_T}{d} + \log\frac{4}{\delta})} \\ d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T) &\leq \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T) \\ &\quad + 4\sqrt{\frac{d\log(2N_T + \log(\frac{2}{\delta})}{N_T}}\end{aligned} \tag{7}$$

where d is the VC-dimension of model space $\mathcal{H}$.

Combining (2) and (7), we get the generalization bound for knowledge distillation with the empirical learning error as follows:

$$\epsilon_{\mathbb{D}_T}(h_T) \leq \hat{\epsilon}_{\mathbb{D}_S^{K+1}}(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T) + C_1 \tag{8}$$

where C_1 is a constant as

$$\begin{aligned}&C_1 = \min\{ \\ &\lambda_1 + \sqrt{\frac{4}{N_T}(d\log\frac{2eN_T}{d} + \log\frac{4}{\delta})} + 4\sqrt{\frac{d\log(2N_T + \log(\frac{2}{\delta})}{N_T}} \\ &\sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_T}(h) - \hat{\epsilon}_{\mathbb{D}_S^{K+1}}(h)| + \sqrt{\frac{4}{N_T}(d\log\frac{2eN_T}{d} + \log\frac{4}{\delta})}. \\ &\}\end{aligned} \tag{9}$$

3. Appendix C

Proposition 2 *The KD3A bound is a tighter bound than the original bound, if the task risk gap between the knowledge distillation domain $\mathbb{D}_S^{K+1}$ and the target domain $\mathbb{D}_T$ is smaller than the following upper-bound for all source domain $k \in \{1, \cdots, K\}$, that is, $\epsilon_{\mathbb{D}_S^{K+1}}(h)$ should satisfy:*

$$\begin{aligned}\sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_T}(h)| &\leq \inf_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_S^k}(h)| \\ &\quad + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T) + \lambda_S^k\end{aligned} \tag{10}$$

Proof:

Following the Theorem 2 in Ben-David et al. (2010), for each source domain $\mathbb{D}_S^k$ and for all $h_T \in \mathcal{H}$, we have

$$\epsilon_{\mathbb{D}_T}(h_T) \leq \epsilon_{\mathbb{D}_S^k}(h_T) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T) + \lambda_S^k \tag{11}$$

where $\lambda_S^k = \min_{h\in\mathcal{H}} \epsilon_{\mathbb{D}_S^k}(h) + \epsilon_{\mathbb{D}_T}(h)$ is the optimal task risk of $\mathbb{D}_S^k$ and $\mathbb{D}_T$.

The original bound states that for all $h_T \in \mathcal{H}$, we have

$$\epsilon_{\mathbb{D}_T}(h) \leq \sum_{k=1}^K \boldsymbol{\alpha}_k \left(\epsilon_{\mathbb{D}_S^k}(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T)\right) + \lambda_0 \tag{12}$$

where $\lambda_0 = \min_{h\in\mathcal{H}} \sum_{k=1}^K \alpha_k \epsilon_{\mathbb{D}_S^k}(h) + \epsilon_{\mathbb{D}_T}(h)$ and we have the following relations between λ_0 and λ_S^k:

$$\begin{aligned}\lambda_0 &= \min_{h\in\mathcal{H}} \sum_{k=1}^K \alpha_k \epsilon_{\mathbb{D}_S^k}(h) + \epsilon_{\mathbb{D}_T}(h) \\ &\geq \sum_{k=1}^K \alpha_k (\min_{h\in\mathcal{H}} \epsilon_{\mathbb{D}_S^k}(h) + \epsilon_{\mathbb{D}_T}(h)) \\ &= \sum_{k=1}^K \alpha_k \lambda_S^k\end{aligned} \tag{13}$$

With $(11-13)$, the original bound (12) can be considered as the weighted combination of the source domains. In addition, the KD3A bound is also the combination of the original bound (12) and the knowledge distillation bound (2). Then we get that the KD3A bound is a tighter bound than the original bound if the knowledge distillation bound (2) is tighter than the single source bound (11) for each source domain $\mathbb{D}_S^k$, that is, for all source domain $k \in \{1, \cdots, K\}$ and all $h_T \in \mathcal{H}$, the knowledge distillation bound should satisfy:

$$\begin{aligned}&\epsilon_{\mathbb{D}_S^{K+1}}(h_T) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T) \\ &+ \min\{\lambda_1, \sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_T}(h)|\} \\ &\leq \epsilon_{\mathbb{D}_S^k}(h_T) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T) + \lambda_S^k\end{aligned} \tag{14}$$

Since $d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T) = 0$ and λ_1 is a constant, the task risk gap $\sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_T}(h)|$ should satisfy the following condition for all $h_T \in \mathcal{H}$, that is:

$$\begin{aligned}\sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_T}(h)| &\leq \epsilon_{\mathbb{D}_S^k}(h_T) - \epsilon_{\mathbb{D}_S^{K+1}}(h_T) \\ &\quad + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T) + \lambda_S^k\end{aligned} \tag{15}$$

Since condition (15) holds for all $h_T \in \mathcal{H}$, we have the tighter bound condition as

$$\begin{aligned}\sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_T}(h)| &\leq \inf_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_S^k}(h)| \\ &\quad + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{D}_S^k, \mathbb{D}_T) + \lambda_S^k\end{aligned} \tag{16}$$

Layer	Configuration
1	2D Convolution with kernel size 5*5 and output feature channels 64
2	BatchNorm, ReLU, MaxPool
3	2D Convolution with kernel size 5*5 and output feature channels 64
4	BatchNorm, ReLU, MaxPool
5	2D Convolution with kernel size 5*5 and output feature channels 128
6	BatchNorm, ReLU
7	Fully connection layer with output channels 10
8	Softmax

Table 1. The 3-layers CNN backbone for **Digit-5**.

Parameters	Benchmark Datasets			
	Amazon Review	**Digit-5**	**Office-Caltech10**	**DomainNet**
Data Augmentation	None	Mixup ($\alpha = 0.2$)		
Backbone	3-layers MLP	3-layers CNN	Resnet101 (pretrained = True)	
Optimizer	SGD with momentum = 0.9			
Learning rate schedule	From 0.05 to 0.001 with cosine decay		From 0.005 to 0.0001 with cosine decay	
Batchsize	50	100	32	50
Total epochs	40			
Communication rounds	r=1			
Confidence gate	From 0.9 to 0.95			From 0.8 to 0.95

Table 2. Implementation details of our KD3A on four benchmark datasets: Amazon Revoew, Digit-5, Office-Caltech10 and DomainNet.

4. Appendix D: Representation Invariant Bounds For KD3A.

One reviewer argues that the generalization bound in proposition 1 is not rigorous since the optimization process may change the value of λ. The optimal joint risk λ between source and target domain is defined as $\lambda := \min_{h\in\mathcal{H}} \epsilon_S(h) + \epsilon_T(h)$. λ is based on the hypothesis space $\mathcal{H}$ and is usually intractable to compute. Considering the fixed model backbones are used in in practice (where the hypothesis space $\mathcal{H}$ is implicitly determined), we follow previous works (i.e. Theorem 1 in Long et al. (2015) and Theorem 2 in Zhao et al. (2018)) and consider λ as a constant. However, we agree with the fact proposed in Zhao et al. (2019) (Section 4.1) that optimizing the $\mathcal{H}-$divergence can learn domain invariant representations, but can also change the representation space. This may change the value of λ. As such, we take the suggestions of the reviewer and replace the original bound with the new bound in Zhao et al. (2019), which utilizes the $\tilde{\mathcal{H}}-$divergence and the constant term C. With this upper bound, we propose a new version for our Proposition 1, Theorem 2 and Proposition 2 as follows:

Proposition 1. Denoting $C_1 := \min\{\mathbb{E}_{\mathbb{D}_S^{K+1}}[|f_S^{K+1} - f_T|], \mathbb{E}_{\mathbb{D}_T}[|f_S^{K+1} - f_T|]\}$, we have

$$\epsilon_{\mathbb{D}_T}(h_T) \leq \epsilon_{\mathbb{D}_S^{K+1}}(h_T) + d_{\tilde{\mathcal{H}}}(\mathbb{D}_S^{K+1}, \mathbb{D}_T) + \min\{C_1, \sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_T}(h)|\}$$

	Clipart	Infograph	Painting	Avg
KD3A†	$69.7_{\pm 0.67}$	$21.2_{\pm 0.35}$	$58.8_{\pm 0.66}$	48.8
KD3A	$\mathbf{72.5_{\pm 0.62}}$	$\mathbf{23.4_{\pm 0.43}}$	$\mathbf{60.9_{\pm 0.71}}$	**51.1**
	Quickdraw	Real	Sketch	
KD3A†	$15.1_{\pm 0.21}$	$70.4_{\pm 0.54}$	$57.9_{\pm 0.41}$	48.8
KD3A	$\mathbf{16.4_{\pm 0.28}}$	$\mathbf{72.7_{\pm 0.55}}$	$\mathbf{60.6_{\pm 0.32}}$	**51.1**

Table 3. The ablation study for data-augmentation strategies on DomainNet.†: Methods trained without data-augmentation.

Theorem 2. Denoting $C_2 := \sum_{k=1}^{K+1} \alpha_k^{CF} \min\{\mathbb{E}_{\mathbb{D}_S^k}[|f_S^k - f_T|], \mathbb{E}_{\mathbb{D}_T}[|f_S^k - f_T|]\}$, we have

$$\epsilon_{\mathbb{D}_T}(h_T) \leq \sum_{k=1}^{K+1} \boldsymbol{\alpha}_k^{\text{CF}} \left(\epsilon_{\mathbb{D}_S^k}(h_T) + d_{\tilde{\mathcal{H}}}(\mathbb{D}_S^k, \mathbb{D}_T)\right) + C_2$$

Proposition 2. Denoting $C_S^k := \min\{\mathbb{E}_{\mathbb{D}_S^k}[|f_S^k - f_T|], \mathbb{E}_{\mathbb{D}_T}[|f_S^k - f_T|]\}$, $\forall k$, the tighter condition should satisfy

$$\sup_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_T}(h)| \leq \inf_{h\in\mathcal{H}} |\epsilon_{\mathbb{D}_S^{K+1}}(h) - \epsilon_{\mathbb{D}_S^k}(h)| + d_{\tilde{\mathcal{H}}}(\mathbb{D}_S^k, \mathbb{D}_T) + C_S^k$$

The proof in Appendix A-C can directly apply to the new bounds. Moreover, KD3A also works on the above new bounds since the $\tilde{\mathcal{H}}-$divergence can be optimized by minimizing the Batchnorm-MMD distance.

Methods	mt	mm	sv	syn	usps	Avg
Oracle	$99.5_{\pm 0.08}$	$95.4_{\pm 0.15}$	$92.3_{\pm 0.14}$	$98.7_{\pm 0.04}$	$99.2_{\pm 0.09}$	97.0
Source-only	$92.3_{\pm 0.91}$	$63.7_{\pm 0.83}$	$71.5_{\pm 0.75}$	$83.4_{\pm 0.79}$	$90.71_{\pm 0.54}$	80.3
MDAN	$97.2_{\pm 0.98}$	$75.7_{\pm 0.83}$	$82.2_{\pm 0.82}$	$85.2_{\pm 0.58}$	$93.3_{\pm 0.48}$	86.7
M^{3}SDA	$98.4_{\pm 0.68}$	$72.8_{\pm 1.13}$	$81.3_{\pm 0.86}$	$89.6_{\pm 0.56}$	$96.2_{\pm 0.81}$	87.7
CMSS	$99.0_{\pm 0.08}$	$75.3_{\pm 0.57}$	$88.4_{\pm 0.54}$	$93.7_{\pm 0.21}$	$97.7_{\pm 0.13}$	90.8
DSBN*	97.2	71.6	77.9	88.7	96.1	86.3
FADA	$91.4_{\pm 0.7}$	$62.5_{\pm 0.7}$	$50.5_{\pm 0.3}$	$71.8_{\pm 0.5}$	$91.7_{\pm 1}$	73.6
FADA*	92.5	64.5	72.1	82.8	91.7	80.8
SHOT	$98.2_{\pm 0.37}$	$80.2_{\pm 0.41}$	$84.5_{\pm 0.32}$	$\mathbf{91.1_{\pm 0.23}}$	$97.1_{\pm 0.28}$	90.2
KD3A†	$99.1_{\pm 0.15}$	$86.9_{\pm 0.11}$	$82.2_{\pm 0.26}$	$89.2_{\pm 0.19}$	$98.4_{\pm 0.11}$	91.2
KD3A	$\mathbf{99.2_{\pm 0.12}}$	$\mathbf{87.3_{\pm 0.23}}$	$\mathbf{85.6_{\pm 0.17}}$	$89.4_{\pm 0.28}$	$\mathbf{98.5_{\pm 0.25}}$	**92.0**

Table 4. UMDA accuracy (%) on the **Digit-5**. *: The best results recorded in our re-implementation. †: Methods trained without data-augmentation. Our model KD3A achieves 92.0% accuracy and outperforms all other baselines.

Methods	*Books*	*DVDs*	*Elec.*	*Kitchen*	Avg.
Source-only	74.4	79.2	73.5	71.4	74.6
MDAN	78.6	**80.7**	85.4	86.3	82.8
FADA	78.1	82.7	77.4	77.5	78.9
KD3A	**79.0**	80.6	**85.6**	**86.9**	**83.1**

Table 5. The UMDA performance on Amazon Review dataset.

5. Appendix E: The Implementation of BatchNorm MMD

We have introduced the **BatchNorm MMD** with the following loss:

$$\sum_{l=1}^{L}\sum_{k=1}^{K+1}\boldsymbol{\alpha}_k\big(\|\mu(\boldsymbol{\pi}_l^T)-\mathbb{E}(\boldsymbol{\pi}_l^k)\|_2^2+\|\mu[\boldsymbol{\pi}_l^T]^2-\mathbb{E}[\boldsymbol{\pi}_l^k]^2\|_2^2\big) \tag{17}$$

However, directly optimizing the loss (17) requires to traverse all Batchnorm layers, which is time-consuming. Inspired by the suggestions of reviewers, we propose a computation-efficient method containing two steps. First, we directly derive the global optimal solution of $\mu(\boldsymbol{\pi}_l^T)$ for loss (17), that is, $\forall l, 1\le l\le L$, the optimal model h_{op}^T on target domain $\mathbb{D}_T$ should satisfy

$$\begin{aligned}\mu_{\text{op}}(\boldsymbol{\pi}_l^T)&=\sum_{k=1}^{K+1}\boldsymbol{\alpha}_k\mathbb{E}(\boldsymbol{\pi}_l^k)\\ \mu_{\text{op}}[\boldsymbol{\pi}_l^T]^2&=\sum_{k=1}^{K+1}\boldsymbol{\alpha}_k\mathbb{E}[\boldsymbol{\pi}_l^k]^2\end{aligned} \tag{18}$$

Then we calculate the optimal solution from (18) as $\{(\mu_{\text{op}}(\boldsymbol{\pi}_l^T),\mu_{\text{op}}[\boldsymbol{\pi}_l^T]^2)\}_{l=1}^L$, directly substitute this solution into every Batchnorm layer of h^T and use it as global model. Although this computation-efficient implementation may seem heuristic, we find it practically work and can achieve the same performance as the original maximization step.

6. Appendix F

6.1. Implementation Details.

We perform UMDA on those datasets with multiple domains. During experiments, we choose one domain as the target domain, and use the remained domains as source domains. Finally, we report the average UMDA results among all domains. The code, with which the most important results can be reproduced, is available at Github[1]. In this section, we discuss the implementation details. Following previous settings (Peng et al., 2019), we use a 3-layer MLP as backbone for Amazon Review, a 3-layer CNN for Digit-5 and the ResNet101 pre-trained on ImageNet for Office-Caltech10 and DomainNet. The details of hyper-parameters are provided in Table 2 and the backbones and training epochs are set to same in all method comparison experiments. In training process, We use the SGD as optimizer and take the cosine schedule to decay learning rate from high (0.05 for Amazon Review and Digit5, and 0.005 for Office-Caltech10 and DomainNet) to zero.

Data augmentations. Data augmentations are important in deep network training process. Since different datasets require different augmentation strategies (e.g. rotate, scale, and crop), which introduces extra hyper-parameters, we use mixup (Zhang et al., 2017) as a unified augmentation

[1] github.com/FengHZ/KD3A

Methods	A	C	D	W	Avg
Oracle	99.7	98.4	99.8	99.7	99.4
Source-only	86.1	87.8	98.3	99.0	92.8
MDAN	98.9	98.6	91.8	95.4	96.1
M^{3}SDA	94.5	92.2	**99.2**	99.5	96.4
CMSS	96.0	93.7	99.3	99.6	97.2
DSBN*	93.2	91.6	98.9	99.3	95.8
FADA	$84.2_{\pm 0.5}$	$88.7_{\pm 0.5}$	$87.1_{\pm 0.6}$	$88.1_{\pm 0.4}$	87.1
SHOT	96.4	96.2	98.5	**99.7**	97.7
KD3A†	$96.0_{\pm 0.07}$	$95.2_{\pm 0.08}$	$97.9_{\pm 0.11}$	$99.6_{\pm 0.03}$	97.2
KD3A	$\mathbf{97.4_{\pm 0.08}}$	$\mathbf{96.4_{\pm 0.11}}$	$98.4_{\pm 0.08}$	$\mathbf{99.7_{\pm 0.02}}$	**97.9**

Table 6. UMDA accuracy (%) on the Office-Caltech10. *: The best results recorded in our re-implementation. †: Methods trained without data-augmentation.

strategy and simply set the mix-parameter $\alpha = 0.2$ in all experiments. For fair comparison, we report the results on both conditions, i.e. with/without data-augmentations. The results are shown in Table 3,4 and 6. The ablation study in data augmentations indicates that mixup strategy can unify different augmentation strategies on different doman adaptation datasets with only one hyper-parameter. Moreover, KD3A can achieve good results even without data-augmentation.

6.2. Results on Amazon Review, Digit-5 And Office-caltech10.

In this section, we report the experiment results on **Amazon Review**, **Digit-5** and **Office-Caltech10**. Amazon Review is a sentimental analysis dataset including four domains: Books, DVDs, Electronics and Kitchen Appliances. Digit-5 is a digit classification dataset including MNIST (mt), MNISTM(mm), SVHN (sv), Synthetic (syn), and USPS (up). Office-Caltech10 contains 10 object categories from four domains, i.e. Amazon (A), Caltech (C), DSLR (D). and Webcam (W). **Note that results are directly cited from published papers if we follow the same setting.** The results on Table 5, 4 and 6 show that our KD3A outperforms other UMDA methods and advanced decentralized UMDA methods. Moreover, our KD3A provides better consensus knowledge on the hard domains such as the *MNISTM* domain on the **Digit-5**, which outperforms other methods by a large margin.

References

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., et al. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, 2010.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. Learning transferable features with deep adaptation networks. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 97–105. JMLR.org, 2015.

Peng, X., Bai, Q., Xia, X., et al. Moment matching for multi-source domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 1406–1415. IEEE, 2019.

Zhang, H., Cissé, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017. URL http://arxiv.org/abs/1710.09412.

Zhao, H., Zhang, S., Wu, G., Moura, J. M. F., Costeira, J. P., and Gordon, G. J. Adversarial multiple source domain adaptation. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pp. 8568–8579, 2018.

Zhao, H., des Combes, R. T., Zhang, K., and Gordon, G. J. On learning invariant representations for domain adaptation. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7523–7532. PMLR, 2019.