# Aggregating From Multiple Target-Shifted Sources Supplementary Material

**Changjian Shui** [1] **Zijian Li** [2] **Jiaqi Li** [3] **Christian Gagné** [1 4] **Charles X.Ling** [3] **Boyu Wang** [3]

## 1. Additional Related Work

**Additional Multi-source DA Theory** has been investigated in the previous literature. In the unsupervised DA, (Ben-David et al., 2010; Zhao et al., 2018; Peng et al., 2019) adopted $\mathcal{H}$-divergence of marginal distribution $\mathcal{D}(x)$ to estimate the domain relations.(Li et al., 2018a) also applied Wasserstein distance of $\mathcal{D}(x)$ to estimate pair-wise domain distance. (Mansour et al., 2009; Wen et al., 2020) used the Discrepancy distance to derive a tighter theoretical bound. The motivated practice from the aforementioned method used the feature information to learn the task relations, with the general following forms:

$$R_{\mathcal{T}}(h) \leq \sum_t \boldsymbol{\lambda}[t] R_{\mathcal{S}}(h) + \sum_t \boldsymbol{\lambda}[t] d(\mathcal{S}_t(x), \mathcal{T}(x)) + \beta$$

However, as we stated in the paper, $d(\mathcal{S}_t(x), \mathcal{T}(x))$ is not a proper to measure the task's relations. Besides, (Hoffman et al., 2018) used Rényi divergence that requires $\text{supp}(\mathcal{T}(x)) \subseteq \text{supp}(\mathcal{S}(x))$, which generally does not hold in the complicated real-world applications. (Konstantinov & Lampert, 2019; Mansour et al., 2020) adopted $\mathcal{Y}$-discrepancy (Mohri & Medina, 2012) to measure the joint distribution similarity. However, $\mathcal{Y}$ discrepancy is practically difficult to estimate from the data and we empirically show it is difficult to handle the target-shifted sources.

**Multi-source DA Practice** has been proposed from various prospective. The key idea is to estimate the importance of different sources and then select the most related ones, to mitigate the influence of negative transfer. In the multi-source unsupervised DA, (Sankaranarayanan et al., 2018; Balaji et al., 2019; Pei et al., 2018; Zhao et al., 2019; Zhu et al., 2019; Zhao et al., 2020; 2019; Stojanov et al., 2019; Li et al., 2019b; Wang et al., 2019; Lin et al., 2020) proposed different practical strategies in the classification, regression and semantic segmentation problems. In the presence of available labels on the target domain, (Hoffman et al., 2012; Tan et al., 2013; Wei et al., 2017; Yao & Doretto, 2010; Konstantinov & Lampert, 2019) used generalized linear model to learn the target. (Christodoulidis et al., 2016; Li et al., 2019a; Chen et al., 2019) focused on deep learning approaches and (Lee et al., 2019) proposed an ad-hoc strategy to combine to sources in the few-shot target domains. In contrast, these ideas are generally *data-driven approaches* and do not propose a principled practice to understand the source combination and understand task relations.

**Label-Partial Unsupervised DA** Label-Partial can be viewed as a special case of the target-shifted DA. [1] Most existing works focus on one-to-one partial DA (Zhang et al., 2018; Chen et al., 2020; Bucci et al., 2019; Cao et al., 2019) by adopting the re-weighting training approach without a principled understanding. In our paper, we first analyzed this common practice and adopt the label distribution ratio as its weights, which provides a principled approach to detect the non-overlapped classes in the representation learning.

### 1.1. Other scenarios related to Multi-Source DA

**Domain Generalization** The domain generalization (DG) resembles multi-source transfer but aims at different goals. A common setting in DG is to learn multiple source but directly predict on the unseen target domain. The conventional DG approaches generally learn a distribution invariant features (Balaji et al., 2018; Saenko et al., 2010; Motiian et al., 2017; Ilse et al., 2019) or conditional distribution invariant features (Li et al., 2018b; Akuzawa et al., 2019). However, our theoretical results reveal that in the presence of label shift (i.e $\alpha_t(y) \neq 1$) and outlier tasks then learning conditional or marginal

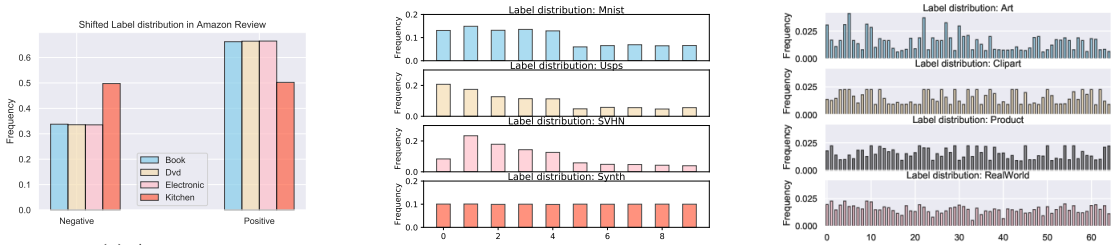
[1] Since $\text{supp}(\mathcal{T}(y)) \subseteq \text{supp}(\mathcal{S}_t(y))$ then we naturally have $\mathcal{T}(y) \neq \mathcal{S}_t(y)$.

invariant features can not guarantee a small target risk. Our theoretical result enables a formal understanding about the inherent difficulty in DG problems.

**Multi-Task Learning** The goal of multi-task learning (Zhang & Yang, 2017) aims to improve the prediction performance of **all** the tasks. In our paper, we aim at controlling the prediction risk of a specified target domain. We also notice some practical techniques are common such as the shared parameter (Zhang & Yeung, 2012), shared representation (Ruder, 2017), etc.

## 2. Additional Figures

We additionally visualize the label distributions in our experiments.




(a) Amazon (b) Digits (c) Office-Home

*Figure 1.* Label distribution visualization. (a) One example in Amazon Review dataset with sources: Book, Dvd, Electronic and target: Kitchen. We randomly drop $50\%$ of the negative reviews in all the sources while keeping target label distribution unchanged. (b) One example in Digits dataset with Sources: MNIST, USPS, SVHN and Target Synth. We randomly drop $50\%$ data on digits 5-9 in all sources while keeping target label distribution unchanged. (c) Office-Home dataset. The original label distribution is non-uniform. See Appendix 12 for details.

## 3. Notation Tables

*Table 1.* Table of Notations

| | |
|---|---|
| $R_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\ell(h(x,y))$ | Expected Risk on distribution $\mathcal{D}$ w.r.t. hypothesis $h$ |
| $\hat{R}_{\mathcal{D}}(h) = \frac{1}{N}\sum_{i=1}^{N}\ell(h(x_i,y_i))$ | Empirical Risk on observed data $\{(x_i,y_i)\}_{i=1}^N$ that are i.i.d. sampled from $\mathcal{D}$. |
| $\alpha$ and $\hat{\alpha}_t$ | True and empirical label distribution ratio $\alpha(y) = \mathcal{T}(y)/\mathcal{S}(y)$ |
| $\hat{R}^{\alpha}_{\mathcal{S}}(h) = \frac{1}{N}\sum_{i=1}^{N}\alpha(y_i)\ell(h(x_i,y_i))$ | Empirical Weighted Risk on observed data $\{(x_i,y_i)\}_{i=1}^N$. |
| $\mathcal{S}(z\|y) = \int_x g(z\|x)S(x\|Y=y)dx$ | Conditional distribution w.r.t. latent variable $Z$ that induced by feature learning function $g$. |
| $W_1(\mathcal{S}_t(z\|y)\|\mathcal{T}(z\|y))$ | Conditional Wasserstein distance on the latent space $Z$ |

## 4. Proof of Theorem 1

**Proof idea** Theorem 1 consists three steps in the proof:

**Lemma 1.** *If the prediction loss is assumed as $L$-Lipschitz and the hypothesis is $K$-Lipschitz w.r.t. the feature $x$ (given the same label), i.e. for $\forall Y = y$, $\|h(x_1,y) - h(x_2,y)\|_2 \le K\|x_1 - x_2\|_2$. Then the target risk can be upper bounded by:*

$$R_{\mathcal{T}}(h) \le \sum_t \boldsymbol{\lambda}[t]R^{\alpha_t}_{\mathcal{S}}(h) + LK\sum_t \boldsymbol{\lambda}[t]\mathbb{E}_{y\sim\mathcal{T}(y)}W_1(\mathcal{T}(x|Y=y)\|\mathcal{S}(x|Y=y)) \tag{1}$$

*Proof.* The target risk can be expressed as:

$$R_{\mathcal{T}}(h(x,y)) = \mathbb{E}_{(x,y)\sim\mathcal{T}}\ell(h(x,y)) = \mathbb{E}_{y\sim\mathcal{T}(y)}\mathbb{E}_{x\sim\mathcal{T}(x|y)}\ell(h(x,y))$$

By denoting $\alpha(y) = \frac{\mathcal{T}(y)}{\mathcal{S}(y)}$, then we have:

$$\mathbb{E}_{y\sim\mathcal{T}(y)}\mathbb{E}_{y\sim\mathcal{T}(x|y)}\ell(h(x,y)) = \mathbb{E}_{y\sim\mathcal{S}(y)}\alpha(y)\mathbb{E}_{x\sim\mathcal{T}(x|y)}\ell(h(x,y))$$

Then we aim to upper bound $\mathbb{E}_{x\sim\mathcal{T}(x|y)}\ell(h(x,y))$. For any fixed $y$,

$$\mathbb{E}_{x\sim\mathcal{T}(x|y)}\ell(h(x,y)) - \mathbb{E}_{x\sim\mathcal{S}(x|y)}\ell(h(x,y)) \leq |\int_{x\in\mathcal{X}} \ell(h(x,y))d(\mathcal{T}(x|y) - \mathcal{S}(x|y))|$$

Then according to the Kantorovich-Rubinstein duality, for **any** distribution coupling $\gamma \in \Pi(\mathcal{T}(x|y), \mathcal{S}(x|y))$, then we have:

$$\begin{aligned}
&= \inf_{\gamma} |\int_{\mathcal{X}\times\mathcal{X}} \ell(h(x_p,y)) - \ell(h(x_q,y))d\gamma(x_p,x_q)| \\
&\leq \inf_{\gamma} \int_{\mathcal{X}\times\mathcal{X}} |\ell(h(x_p,y)) - \ell(h(x_q,y))|d\gamma(x_p,x_q) \\
&\leq L\inf_{\gamma} \int_{\mathcal{X}\times\mathcal{X}} |h(x_p,y)) - h(x_q,y)|d\gamma(x_p,x_q) \\
&\leq LK\inf_{\gamma} \int_{\mathcal{X}\times\mathcal{X}} \|x_p - x_q\|_2 d\gamma(x_p,x_q) \\
&= LKW_1(\mathcal{T}(x|Y=y)\|\mathcal{S}(x|Y=y))
\end{aligned}$$

The first inequality is obvious; and the second inequality comes from the assumption that $\ell$ is $L$-Lipschitz; the third inequality comes from the hypothesis is $K$-Lipschitz w.r.t. the feature $x$ (given the same label), i.e. for $\forall Y = y$, $\|h(x_1,y) - h(x_2,y)\|_2 \leq K\|x_1 - x_2\|_2$.

Then we have:

$$\begin{aligned}
R_{\mathcal{T}}(h) &\leq \mathbb{E}_{y\sim\mathcal{S}(y)}\alpha(y)[\mathbb{E}_{x\sim\mathcal{S}(x|y)}\ell(h(x,y)) + LKW_1(\mathcal{T}(x|y)\|\mathcal{S}(x|y))] \\
&= \mathbb{E}_{(x,y)\sim\mathcal{S}}\alpha(y)\ell(h(x,y)) + LK\mathbb{E}_{y\sim\mathcal{T}(y)}W_1(\mathcal{T}(x|Y=y)\|\mathcal{S}(x|Y=y)) \\
&= R^{\alpha}_{\mathcal{S}}(h) + LK\mathbb{E}_{y\sim\mathcal{T}(y)}W_1(\mathcal{T}(x|Y=y)\|\mathcal{S}(x|Y=y))
\end{aligned}$$

Supposing each source $\mathcal{S}_t$ we assign the weight $\boldsymbol{\lambda}[t]$ and label distribution ratio $\alpha_t(y) = \frac{\mathcal{T}(y)}{\mathcal{S}_t(y)}$, then by combining this $T$ source target pair, we have:

$$R_{\mathcal{T}}(h) \leq \sum_t \boldsymbol{\lambda}[t]R^{\alpha_t}_{\mathcal{S}_t}(h) + LK\sum_t \boldsymbol{\lambda}[t]\mathbb{E}_{y\sim\mathcal{T}(y)}W_1(\mathcal{T}(x|Y=y)\|\mathcal{S}_t(x|Y=y))$$

□

Then we will prove Theorem 1 from this result, we will derive the non-asymptotic bound, estimated from the finite sample observations. Supposing the empirical label ratio value is $\hat{\alpha}_t$, then for any simplex $\boldsymbol{\lambda}$ we can prove the high-probability bound.

### 4.1. Bounding the empirical and expected prediction risk

*Proof.* We first bound the first term, which can be upper bounded as:

$$\sup_h |\sum_t \boldsymbol{\lambda}[t]R^{\alpha_t}_{\mathcal{S}_t}(h) - \sum_t \boldsymbol{\lambda}[t]\hat{R}^{\hat{\alpha}_t}_{\mathcal{S}_t}(h)| \leq \underbrace{\sup_h |\sum_t \boldsymbol{\lambda}[t]R^{\alpha_t}_{\mathcal{S}_t}(h) - \sum_t \boldsymbol{\lambda}[t]\hat{R}^{\alpha_t}_{\mathcal{S}_t}(h)|}_{\text{(I)}} + \underbrace{\sup_h |\sum_t \boldsymbol{\lambda}[t]\hat{R}^{\alpha_t}_{\mathcal{S}_t}(h) - \sum_t \boldsymbol{\lambda}[t]\hat{R}^{\hat{\alpha}_t}_{\mathcal{S}_t}(h)|}_{\text{(II)}}$$

**Bounding term** $(\mathbf{I})$ According to the McDiarmid inequality, each item changes at most $|\frac{2\boldsymbol{\lambda}[t]\alpha_t(y)\ell}{N_{\mathcal{S}_t}}|$. Then we have:

$$P\left((\mathrm{I}) - \mathbb{E}(\mathrm{I}) \geq t\right) \leq \exp\left(\frac{-2t^2}{\sum_{t=1}^{T} \frac{4}{\beta_t N}\boldsymbol{\lambda}^2[t]\alpha_t(y)^2\ell^2}\right) = \delta$$

By substituting $\delta$, at high probability $1-\delta$ we have:

$$(\mathrm{I}) \leq \mathbb{E}(\mathrm{I}) + L_{\max} d_{\infty}^{\sup} \sqrt{\sum_{t=1}^{T} \frac{\boldsymbol{\lambda}[t]^2}{\beta_t}} \sqrt{\frac{\log(1/\delta)}{2N}}$$

Where $L_{\max} = \sup_{h\in\mathcal{H}} \ell(h)$ and $N = \sum_{t=1}^{T} N_{\mathcal{S}_t}$ the total source observations and $\beta_t = \frac{N_{\mathcal{S}_t}}{N}$ the frequency ratio of each source. And $d_{\infty}^{\sup} = \max_{t=1,\dots,T} d_{\infty}(\mathcal{T}(y)\|\mathcal{S}(y)) = \max_{t=1,\dots,T} \max_{y\in[1,\mathcal{Y}]} \alpha_t(y)$, the maximum true label shift value (constant).

Bounding $\mathbb{E}\sup(\mathrm{I})$, the expectation term can be upper bounded as the form of Rademacher Complexity:

$$\begin{aligned}
\mathbb{E}(\mathrm{I}) &\leq 2\mathbb{E}_{\sigma}\mathbb{E}_{\hat{\mathcal{S}}_1^T} \sup_h \sum_{t=1}^{T} \boldsymbol{\lambda}[t] \sum_{(x_t,y_t)\in\hat{\mathcal{S}}_t} \frac{1}{TN} \left(\alpha_t(y)\ell(h(x_t,y_t))\right) \\
&\leq 2\sum_t \boldsymbol{\lambda}[t]\mathbb{E}_{\sigma}\mathbb{E}_{\hat{\mathcal{S}}_1^T} \sup_h \sum_{(x_t,y_t)\in\hat{\mathcal{S}}_t} \frac{1}{TN} \left(\alpha_t(y)\ell(h(x_t,y_t))\right) \\
&\leq 2\sup_t \mathbb{E}_{\sigma}\mathbb{E}_{\hat{\mathcal{S}}_t} \sup_h \sum_{(x_t,y_t)\in\hat{\mathcal{S}}_t} \frac{1}{TN} \left[\alpha_t(y)\ell(h(x_t,y_t))\right] \\
&= \sup_t 2\mathcal{R}_t(\ell,\mathcal{H}) = 2\bar{R}(\ell,\mathcal{H})
\end{aligned}$$

Where $\bar{R}(\ell,\mathcal{H}) = \sup_t \mathcal{R}_t(\ell,\mathcal{H}) = \sup_t \sup_{h\sim\mathcal{H}} \mathbb{E}_{\hat{\mathcal{S}}_t,\sigma} \sum_{(x_t,y_t)\in\hat{\mathcal{S}}_t} \frac{1}{TN} \left[\alpha_t(y)\ell(h(x_t,y_t))\right]$, represents the Rademacher complexity w.r.t. the prediction loss $\ell$, hypothesis $h$ and *true* label distribution ratio $\alpha_t$.

Therefore with high probability $1-\delta$, we have:

$$\sup_h |\sum_t \boldsymbol{\lambda}[t] R_{\mathcal{S}}^{\alpha_t}(h) - \sum_t \boldsymbol{\lambda}[t] \hat{R}_{\mathcal{S}}^{\alpha_t}(h)| \leq \bar{\mathcal{R}}(\ell,h) + L_{\max} d_{\infty}^{\sup} \sqrt{\sum_{t=1}^{T} \frac{\boldsymbol{\lambda}[t]^2}{\beta_t}} \sqrt{\frac{\log(1/\delta)}{2N}}$$

**Bounding Term** $(\mathbf{II})$ For all the hypothesis $h$, we have:

$$\begin{aligned}
|\sum_t \boldsymbol{\lambda}[t]\hat{R}_{\mathcal{S}_t}^{\alpha_t}(h) - \sum_t \boldsymbol{\lambda}[t]\hat{R}_{\mathcal{S}_t}^{\hat{\alpha}_t}(h)| &= |\sum_t \boldsymbol{\lambda}[t] \frac{1}{N_{\mathcal{S}_t}} \sum_{i}^{N_{\mathcal{S}_t}} (\alpha(y(i)) - \hat{\alpha}(y(i)))\ell(h)| \\
&= \sum_t \boldsymbol{\lambda}[t] \frac{1}{N_{\mathcal{S}_t}} |\sum_{y}^{|\mathcal{Y}|} (\alpha(Y=y) - \hat{\alpha}(Y=y))\bar{\ell}(Y=y)|
\end{aligned}$$

Where $\bar{\ell}(Y=y) = \sum_i^{N_{\mathcal{S}_t}} \ell(h(x_i, y_i=y))$, represents the cumulative error, conditioned on a given label $Y=y$. According to the Holder inequality, we have:

$$\begin{aligned}
\sum_t \boldsymbol{\lambda}[t] \frac{1}{N_{\mathcal{S}_t}} |\sum_{y}^{|\mathcal{Y}|} (\alpha_t(Y=y) - \hat{\alpha}_t(Y=y))\bar{\ell}(Y=y)| &\leq \sum_t \boldsymbol{\lambda}[t] \frac{1}{N_{\mathcal{S}_t}} \|\alpha_t - \hat{\alpha}_t\|_2 \|\bar{\ell}(Y=y)\|_2 \\
&\leq L_{\max} \sum_t \boldsymbol{\lambda}[t] \|\alpha_t - \hat{\alpha}_t\|_2 \\
&\leq L_{\max} \sup_t \|\alpha_t - \hat{\alpha}_t\|_2
\end{aligned}$$

Therefore, $\forall h \in \mathcal{H}$, with high probability $1-\delta$ we have:

$$\sum_t \boldsymbol{\lambda}[t] R_{\mathcal{S}}^{\alpha_t}(h) \leq \sum_t \boldsymbol{\lambda}[t] \hat{R}_{\mathcal{S}}^{\hat{\alpha}_t}(h) + 2\bar{\mathcal{R}}(\ell, h) + L_{\max} d_{\infty}^{\sup} \sqrt{\sum_{t=1}^{T} \frac{\boldsymbol{\lambda}[t]^2}{\beta_t}} \sqrt{\frac{\log(1/\delta)}{2N}} + L_{\max} \sup_t \|\alpha_t - \hat{\alpha}_t\|_2$$

### 4.2. Bounding empirical Wasserstein Distance

Then we need to derive the sample complexity of the empirical and true distributions, which can be decomposed as the following two parts. For any $t$, we have:

$$\begin{aligned}
&\mathbb{E}_{y\sim\mathcal{T}(y)} W_1(\mathcal{T}(x|Y=y)\|\mathcal{S}_t(x|Y=y)) - \mathbb{E}_{y\sim\hat{\mathcal{T}}(y)} W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y)) \\
&\underbrace{\leq \mathbb{E}_{y\sim\mathcal{T}(y)} W_1(\mathcal{T}(x|Y=y)\|\mathcal{S}_t(x|Y=y)) - \mathbb{E}_{y\sim\mathcal{T}(y)} W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y))}_{\text{(I)}} \\
&+ \underbrace{\mathbb{E}_{y\sim\mathcal{T}(y)} W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y)) - \mathbb{E}_{y\sim\hat{\mathcal{T}}(y)} W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y))}_{\text{(II)}}
\end{aligned}$$

**Bounding (I)** We have:

$$\begin{aligned}
&\mathbb{E}_{y\sim\mathcal{T}(y)} W_1(\mathcal{T}(x|Y=y)\|\mathcal{S}_t(x|Y=y)) - \mathbb{E}_{y\sim\mathcal{T}(y)} W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y)) \\
&= \sum_y \mathcal{T}(y) \left( W_1(\mathcal{T}(x|Y=y)\|\mathcal{S}_t(x|Y=y)) - W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y) \right) \\
&\leq |\sum_y \mathcal{T}(y)| \sup_y \left( W_1(\mathcal{T}(x|Y=y)\|\mathcal{S}_t(x|Y=y)) - W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y) \right) \\
&= \sup_y \left( W_1(\mathcal{T}(x|Y=y)\|\mathcal{S}_t(x|Y=y)) - W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y) \right) \\
&\leq \sup_y \; [W_1(\mathcal{S}_t(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y)) + W_1(\hat{\mathcal{S}}_t(x|Y=y)\|\hat{\mathcal{T}}(x|Y=y)) \\
&+ W_1(\hat{\mathcal{T}}(x|Y=y)\|\mathcal{T}(x|Y=y)) - W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y))] \\
&= \sup_y W_1(\mathcal{S}_t(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y)) + W_1(\hat{\mathcal{T}}(x|Y=y)\|\mathcal{T}(x|Y=y))
\end{aligned}$$

The first inequality holds because of the Holder inequality. As for the second inequality, we use the triangle inequality of Wasserstein distance. $W_1(P\|Q) \leq W_1(P\|P_1) + W_1(P_1\|P_2) + W_1(P_2\|Q)$.

According to the convergence behavior of Wasserstein distance (Weed et al., 2019), with high probability $\geq 1-2\delta$ we have:

$$W_1(\mathcal{S}_t(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y)) + W_1(\hat{\mathcal{T}}(x|Y=y)\|\mathcal{T}(x|Y=y)) \leq \kappa(\delta, N_{\mathcal{S}_t}^y, N_{\mathcal{T}}^y)$$

Where $k(\delta, N_{\mathcal{S}_t}^y, N_{\mathcal{T}}^y) = C_{t,y}(N_{\mathcal{S}_t}^y)^{-s_{t,y}} + C_y(N_{\mathcal{T}}^y)^{-s_y} + \sqrt{\frac{1}{2}\log(\frac{2}{\delta})}(\sqrt{\frac{1}{N_{\mathcal{S}_t}^y}} + \sqrt{\frac{1}{N_t^y}})$, where $N_{\mathcal{S}_t}^y$ is the number of $Y=y$ in source $t$ and $N_{\mathcal{T}}^y$ is the number of $Y=y$ in target distribution. $C_{t,y}$, $C_y$ $s_{t,y} > 2$, $s_y > 2$ are positive constant in the concentration inequality. This indicates the convergence behavior between empirical and true Wasserstein distance.

If we adopt the union bound (over all the labels) by setting $\delta \leftarrow \delta/|\mathcal{Y}|$, then with high probability $\geq 1-2\delta$, we have:

$$\sup_y W_1(\mathcal{S}(x|Y=y)\|\hat{\mathcal{S}}(x|Y=y)) + W_1(\hat{\mathcal{T}}(x|Y=y)\|\mathcal{T}(x|Y=y)) \leq \kappa(\delta, N_{\mathcal{S}_t}^y, N_{\mathcal{T}}^y)$$

where $\kappa(\delta, N_{\mathcal{S}_t}^y, N_{\mathcal{T}}^y) = C_{t,y}(N_{\mathcal{S}_t}^y)^{-s_{t,y}} + C_y(N_{\mathcal{T}}^y)^{-s_y} + \sqrt{\frac{1}{2}\log(\frac{2|\mathcal{Y}|}{\delta})}(\sqrt{\frac{1}{N_{\mathcal{S}_t}^y}} + \sqrt{\frac{1}{N_{\mathcal{T}}^y}})$

Again by adopting the union bound (over all the tasks) by setting $\delta \leftarrow \delta/T$, with high probability $\geq 1-2\delta$, we have:

$$\sum_t \boldsymbol{\lambda}[t] \mathbb{E}_{y\sim\mathcal{T}(y)} W_1(\mathcal{T}(x|Y=y)\|\mathcal{S}(x|Y=y)) - \sum_t \boldsymbol{\lambda}[t] \mathbb{E}_{y\sim\mathcal{T}(y)} W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}(x|Y=y)) \leq \sup_t \kappa(\delta, N_{\mathcal{S}_t}^y, N_{\mathcal{T}}^y)$$

Where $\kappa(\delta, N_{\mathcal{S}_t}^y, N_{\mathcal{T}}^y) = C_{t,y}(N_{\mathcal{S}_t}^y)^{-s_{t,y}} + C_y(N_{\mathcal{T}}^y)^{-s_y} + \sqrt{\frac{1}{2}\log(\frac{2T|\mathcal{Y}|}{\delta})}(\sqrt{\frac{1}{N_{\mathcal{S}_t}^y}} + \sqrt{\frac{1}{N_{\mathcal{T}}^y}})$.

**Bounding (II)** We can bound the second term:

$$\begin{aligned}
&\mathbb{E}_{y\sim\mathcal{T}(y)}W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y)) - \mathbb{E}_{y\sim\hat{\mathcal{T}}(y)}W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y)) \\
&\leq \sup_y W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y))|\sum_y \mathcal{T}(y) - \hat{\mathcal{T}}(y)| \\
&\leq C^t_{\max}|\sum_y \mathcal{T}(y) - \hat{\mathcal{T}}(y)|
\end{aligned}$$

Where $C^t_{\max} = \sup_y W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}(x|Y=y))$ is a positive and bounded constant. Then we need to bound $|\sum_y \mathcal{T}(y) - \hat{\mathcal{T}}(y)|$, by adopting MicDiarmid's inequality, we have at high probability $1-\delta$:

$$\begin{aligned}
|\sum_y \mathcal{T}(y) - \hat{\mathcal{T}}(y)| &\leq \mathbb{E}_{\hat{\mathcal{T}}}|\sum_y \mathcal{T}(y) - \hat{\mathcal{T}}(y)| + \sqrt{\frac{\log(1/\delta)}{2N_{\mathcal{T}}}} \\
&= 2\mathbb{E}_\sigma\mathbb{E}_{\hat{\mathcal{T}}}\sum_y \sigma\hat{\mathcal{T}}(y) + \sqrt{\frac{\log(1/\delta)}{2N_{\mathcal{T}}}}
\end{aligned}$$

Then we bound $\mathbb{E}_\sigma\mathbb{E}_{\hat{\mathcal{T}}}\sum_y \sigma\hat{\mathcal{T}}(y)$. We use the properties of Rademacher complexity [Lemma 26.11, (Shalev-Shwartz & Ben-David, 2014)] and notice that $\hat{\mathcal{T}}(y)$ is a probability simplex, then we have:

$$\mathbb{E}_\sigma\mathbb{E}_{\hat{\mathcal{T}}}\sum_y \sigma\hat{\mathcal{T}}(y) \leq \sqrt{\frac{2\log(2|\mathcal{Y}|)}{N_{\mathcal{T}}}}$$

Then we have $|\sum_y \mathcal{T}(y) - \hat{\mathcal{T}}(y)| \leq \sqrt{\frac{2\log(2|\mathcal{Y}|)}{N_{\mathcal{T}}}} + \sqrt{\frac{\log(1/\delta)}{2N_{\mathcal{T}}}}$

Then using the union bound and denoting $\delta \leftarrow \delta/T$, with high probability $\geq 1-\delta$ and for any simplex $\boldsymbol{\lambda}$, we have:

$$\begin{aligned}
\sum_t \boldsymbol{\lambda}[t]\mathbb{E}_{y\sim\mathcal{T}(y)}W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y)) &\leq \sum_t \boldsymbol{\lambda}[t]\mathbb{E}_{y\sim\hat{\mathcal{T}}(y)}W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}_t(x|Y=y)) \\
&C_{\max}(\sqrt{\frac{2\log(2|\mathcal{Y}|)}{N_{\mathcal{T}}}} + \sqrt{\frac{\log(T/\delta)}{2N_{\mathcal{T}}}})
\end{aligned}$$

where $C_{\max} = \sup_t C^t_{\max}$.

Combining together, we can derive the PAC-Learning bound, which is estimated from the finite samples (with high probability $1-4\delta$):

$$\begin{aligned}
R_{\mathcal{T}}(h) \leq &\sum_t \boldsymbol{\lambda}_t \hat{R}^{\hat{\alpha}_t}_{\mathcal{S}_t}(h) + LH\sum_t \boldsymbol{\lambda}_t\mathbb{E}_{y\sim\hat{\mathcal{T}}(y)}W_1(\hat{\mathcal{T}}(x|Y=y)\|\hat{\mathcal{S}}(x|Y=y)) + L_{\max}d^{\sup}_\infty\sqrt{\sum_{t=1}^T\frac{\boldsymbol{\lambda}_t^2}{\beta_t}}\sqrt{\frac{\log(1/\delta)}{2N}} \\
&+ 2\bar{\mathcal{R}}(\ell,h) + L_{\max}\sup_t\|\alpha_t - \hat{\alpha}_t\|_2 + \sup_t\kappa(\delta, N^y_{\mathcal{S}_t}, N^y_{\mathcal{T}}) + C_{\max}(\sqrt{\frac{2\log(2|\mathcal{Y}|)}{N_{\mathcal{T}}}} + \sqrt{\frac{\log(T/\delta)}{2N_{\mathcal{T}}}})
\end{aligned}$$

Then we denote $\text{Comp}(N_{\mathcal{S}_1},\ldots,N_{\mathcal{T}},\delta) = 2\bar{\mathcal{R}}(\ell,h) + \sup_t\kappa(\delta, N^y_{\mathcal{S}_t}, N^y_{\mathcal{T}}) + C_{\max}(\sqrt{\frac{2\log(2|\mathcal{Y}|)}{N_{\mathcal{T}}}} + \sqrt{\frac{\log(T/\delta)}{2N_{\mathcal{T}}}})$ as the convergence rate function that decreases with larger $N_{\mathcal{S}_1},\ldots,N_{\mathcal{T}}$. Bedsides, $\bar{\mathcal{R}}(\ell,h) = \sup_t\mathcal{R}_t(\ell,\mathcal{H})$ is the re-weighted Rademacher complexity. Given a fixed hypothesis with finite VC dimension [2], it can be proved $\bar{\mathcal{R}}(\ell,h) = \min_{N_{\mathcal{S}_1},\ldots,N_{\mathcal{S}_T}}\mathcal{O}(\sqrt{\frac{1}{N_{\mathcal{S}_t}}})$ i.e (Shalev-Shwartz & Ben-David, 2014). □

[2] If the hypothesis is the neural network, the Rademacher complexity can still be bounded analogously through recent theoretical results in deep neural-network

## 5. Proof of Theorem 2

We first recall the stochastic feature representation $g$ such that $g : \mathcal{X} \to \mathcal{Z}$ and *scoring hypothesis* h $h : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$ and the prediction loss $\ell$ with $\ell : \mathbb{R} \to \mathbb{R}$. [3]

*Proof.* The marginal distribution and conditional distribution w.r.t. latent variable $Z$ that are induced by $g$, which can be reformulated as:

$$\mathcal{S}(z) = \int_x g(z|x)\mathcal{S}(x)dx \qquad \mathcal{S}(z|y) = \int_x g(z|x)\mathcal{S}(x|Y=y)dx$$

In the multi-class classification problem, we additionally define the following distributions:

$$\mu^k(z) = \mathcal{S}(Y=k,z) = \mathcal{S}(Y=k)\mathcal{S}(z|Y=k)$$
$$\pi^k(z) = \mathcal{T}(Y=k,z) = \mathcal{T}(Y=k)\mathcal{T}(z|Y=k)$$

Based on (Nguyen et al., 2009) and $g(z|x)$ is a stochastic representation learning function, the loss conditioned a fixed point $(x, y)$ w.r.t. $h$ and $g$ is $\mathbb{E}_{z\sim g(z|x)}\ell(h(z, y))$. Then taking the expectation over the $\mathcal{S}(x, y)$ we have: [4]

$$\begin{aligned}
R_{\mathcal{S}}(h,g) &= \mathbb{E}_{(x,y)\sim\mathcal{S}(x,y)}\mathbb{E}_{z\sim g(z|x)}\ell(h(z,y)) \\
&= \sum_{k=1}^{|\mathcal{Y}|}\mathcal{S}(y=k)\int_x \mathcal{S}(x|Y=k)\int_z g(z|x)\ell(h(z,y=k))dzdx \\
&= \sum_{k=1}^{|\mathcal{Y}|}\mathcal{S}(y=k)\int_z[\int_x \mathcal{S}(x|Y=k)g(z|x)dx]\ell(h(z,y=k))dz \\
&= \sum_{k=1}^{|\mathcal{Y}|}\mathcal{S}(y=k)\int_z \mathcal{S}(z|Y=k)\ell(h(z,y=k))dz \\
&= \sum_{k=1}^{|\mathcal{Y}|}\int_z \mathcal{S}(z,Y=k)\ell(h(z,y=k))dz \\
&= \sum_{k=1}^{|\mathcal{Y}|}\int_z \mu^k(z)\ell(h(z,y=k))dz
\end{aligned}$$

Intuitively, the expected loss w.r.t. the joint distribution $\mathcal{S}$ can be decomposed as the expected loss on the label distribution $\mathcal{S}(y)$ (weighted by the labels) and conditional distribution $\mathcal{S}(\cdot|y)$ (real valued conditional loss).

Then the expected risk on the $\mathcal{S}$ and $\mathcal{T}$ can be expressed as:

$$R_{\mathcal{S}}(h,g) = \sum_{k=1}^{|\mathcal{Y}|}\int_z \ell(h(z,y=k))\mu^k(z)dz$$
$$R_{\mathcal{T}}(h,g) = \sum_{k=1}^{|\mathcal{Y}|}\int_z \ell(h(z,y=k))\pi^k(z)dz$$

[3] Note this definition is different from the conventional binary classification with binary output, and it is more suitable in the multi-classification scenario and cross entropy loss (Hoffman et al., 2018). For example, if we define $l = -\log(\cdot)$ and $h(z, y) \in (0, 1)$ as a scalar score output. Then $\ell(h(z, y))$ can be viewed as the cross-entropy loss for the neural-network.

[4] An alternative understanding is based on the Markov chain. In this case it is a DAG with $Y \xleftarrow{\mathcal{S}(y|x)} X \xrightarrow{g} Z$, $X \xrightarrow{\mathcal{S}(y|x)} Y \xrightarrow{h} S \xleftarrow{h} Z \xleftarrow{g} X$. (S is the output of the scoring function). Then the expected loss over the all random variable can be equivalently written as $\int \mathbb{P}(x, y, z, s)\ \ell(s)\ d(x, y, z, s) = \int \mathbb{P}(x)\mathbb{P}(y|x)\mathbb{P}(z|x)\mathbb{P}(s|z, y)\ell(s) = \int \mathbb{P}(x, y)\mathbb{P}(z|x)\mathbb{P}(s|z, y)\ell(s)d(x, y)d(z)d(s)$. Since the scoring $S$ is determined by $h(x, y)$, then $\mathbb{P}(s|y, z) = 1$. According to the definition we have $\mathbb{P}(z|x) = g(z|x)$, $\mathbb{P}(x, y) = \mathcal{S}(x, y)$, then the loss can be finally expressed as $\mathbb{E}_{\mathcal{S}(x,y)}\mathbb{E}_{g(z|x)}\ell(h(z, y))$

By denoting $\alpha(y) = \frac{\mathcal{T}(y)}{\mathcal{S}(y)}$, we have the $\alpha$-weighted loss:

$$R^{\alpha}_{\mathcal{S}}(h,g) = \mathcal{T}(Y=1)\int_z \ell(h(z,y=1))\mathcal{S}(z|Y=1) + \mathcal{T}(Y=2)\int_z \ell(h(z,y=2))\mathcal{S}(z|Y=2) + \cdots + \mathcal{T}(Y=k)\int_z \ell(h(z,y=k))\mathcal{S}(z|Y=k)dz$$

Then we have:

$$R_{\mathcal{T}}(h,g) - R^{\alpha}_{\mathcal{S}}(h,g) \leq \sum_k \mathcal{T}(Y=k)\int_z \ell(h(z,y=k))d|\mathcal{S}(z|Y=k) - \mathcal{T}(z|Y=k)|$$

Under the same assumption, we have the loss function $\ell(h(z,Y=k))$ is KL-Lipschitz w.r.t. the cost $\|\cdot\|_2$ (given a fixed $k$). Therefore by adopting the same proof strategy (Kantorovich-Rubinstein duality) in Lemma 2, we have

$$\begin{aligned}&\leq KL\mathcal{T}(Y=1)W_1(\mathcal{S}(z|Y=1)\|\mathcal{T}(z|Y=1)) + \cdots + KL\mathcal{T}(Y=k)W_1(\mathcal{S}(z|Y=k)\|\mathcal{T}(z|Y=k))\\ &= KL\mathbb{E}_{y\sim\mathcal{T}(y)}W_1(\mathcal{S}(z|Y=y)\|\mathcal{T}(z|Y=y))\end{aligned}$$

Therefore, we have:

$$R_{\mathcal{T}}(h,g) \leq R^{\alpha}_{\mathcal{S}}(h,g) + LK\mathbb{E}_{y\sim\mathcal{T}(y)}W_1(\mathcal{S}(z|Y=y)\|\mathcal{T}(z|Y=y))$$

Based on the aforementioned result, we have $\forall t = 1,\ldots,T$ and denote $\mathcal{S} = \mathcal{S}_t$ and $\alpha(y) = \alpha_t(y) = \mathcal{T}(y)/\mathcal{S}_t(y)$:

$$\boldsymbol{\lambda}[t]R_{\mathcal{T}}(h,g) \leq \boldsymbol{\lambda}[t]R^{\alpha_t}_{\mathcal{S}_t}(h,g) + LK\boldsymbol{\lambda}[t]\mathbb{E}_{y\sim\mathcal{T}(y)}W_1(\mathcal{S}_t(z|Y=y)\|\mathcal{T}(z|Y=y))$$

Summing over $t = 1,\ldots,T$, we have:

$$R_{\mathcal{T}}(h,g) \leq \sum_{t=1}^{T}\boldsymbol{\lambda}[t]R^{\alpha_t}_{\mathcal{S}_t}(h,g) + LK\sum_{t=1}^{T}\boldsymbol{\lambda}[t]\mathbb{E}_{y\sim\mathcal{T}(y)}W_1(\mathcal{S}_t(z|Y=y)\|\mathcal{T}(z|Y=y))$$

□

## 6. Approximation $W_1$ distance

According to Jensen inequality, we have

$$W_1(\hat{\mathcal{S}}_t(z|Y=y)\|\hat{\mathcal{T}}(z|Y=y)) \leq \sqrt{[W_2(\hat{\mathcal{S}}_t(z|Y=y)\|\hat{\mathcal{T}}(z|Y=y))]^2}$$

Supposing $\hat{\mathcal{S}}_t(z|Y=y) \approx \mathcal{N}(\mathbf{C}^y_t, \boldsymbol{\Sigma})$ and $\hat{\mathcal{T}}(z|Y=y) \approx \mathcal{N}(\mathbf{C}^y, \boldsymbol{\Sigma})$, then we have:

$$[W_2(\hat{\mathcal{S}}_t(z|Y=y)\|\hat{\mathcal{T}}(z|Y=y)]^2 = \|\mathbf{C}^y_t - \mathbf{C}^y\|^2_2 + \text{Trace}(2\boldsymbol{\Sigma} - 2(\boldsymbol{\Sigma}\boldsymbol{\Sigma})^{1/2}) = \|\mathbf{C}^y_t - \mathbf{C}^y\|^2_2$$

We would like to point out that assuming the identical covariance matrix is more computationally efficient during the matching. This is advantageous and reasonable in the deep learning regime: we adopted the mini-batch (ranging from 20-128) for the neural network parameter optimization, in each mini-batch the samples of each class are **small**, then we compute the empirical covariance/variance matrix will be surely **biased** to the ground truth variance and induce a much higher complexity to optimize. By the contrary, the empirical mean is **unbiased** and computationally efficient, we can simply use the moving the moving average to efficiently update the estimated mean value (with a unbiased estimator). The empirical results verify the effectiveness of this idea.

## 7. Proof of Lemma 1

For each source $\mathcal{S}_t$, by introducing the duality of Wasserstein-1 distance, for $y \in \mathcal{Y}$, we have:

$$\begin{aligned}W_1(\mathcal{S}_t(z|y)\|\mathcal{T}(z|y)) &= \sup_{\|d\|_L\leq 1}\mathbb{E}_{z\sim\mathcal{S}_t(z|y)}d(z) - \mathbb{E}_{z\sim\mathcal{T}(z|y)}d(z)\\ &= \sup_{\|d\|_L\leq 1}\sum_z \mathcal{S}_t(z|y)d(z) - \sum_z \mathcal{T}(z|y)d(z)\\ &= \frac{1}{\mathcal{T}(y)}\sup_{\|d\|_L\leq 1}\frac{\mathcal{T}(y)}{\mathcal{S}_t(y)}\sum_z \mathcal{S}_t(z,y)d(z) - \sum_z \mathcal{T}(z,y)d(z)\end{aligned}$$

Then by defining $\bar{\alpha}_t(z) = \mathbf{1}_{\{(z,y)\sim\mathcal{S}_t\}}\frac{\mathcal{T}(Y=y)}{\mathcal{S}_t(Y=y)} = \mathbf{1}_{\{(z,y)\sim\mathcal{S}_t\}}\alpha_t(Y=y)$, we can see for each pair observation $(z,y)$ sampled from the same distribution, then $\bar{\alpha}_t(Z=z) = \alpha_t(Y=y)$. Then we have:

$$\begin{aligned}\sum_y \mathcal{T}(y)W_1(\mathcal{S}_t(z|y)\|\mathcal{T}(z|y)) &= \sum_y \sup_{\|d\|_L\le 1}\{\sum_z \alpha_t(y)\mathcal{S}_t(z,y)d(z) - \sum_z \mathcal{T}(z,y)d(z)\} \\ &= \sup_{\|d\|_L\le 1}\sum_z \bar{\alpha}_t(z)\mathcal{S}_t(z)d(z) - \sum_z \mathcal{T}(z)d(z) \\ &= \sup_{\|d\|_L\le 1}\mathbb{E}_{z\sim\mathcal{S}_t(z)}\bar{\alpha}_t(z)d(z) - \mathbb{E}_{z\sim\mathcal{T}(z)}d(z)\end{aligned}$$

We propose a simple example to understand $\bar{\alpha}_t$: supposing three samples in $\mathcal{S}_t = \{(z_1, Y=1), (z_2, Y=1), (z_3, Y=0)\}$ then $\bar{\alpha}_t(z_1) = \bar{\alpha}_t(z_2) = \alpha_t(1)$ and $\bar{\alpha}_t(z_3) = \alpha_t(0)$. Therefore, the conditional term is equivalent to the label-weighted Wasserstein adversarial learning. We plug in each source domain as weight $\boldsymbol{\lambda}[t]$ and domain discriminator as $d_t$, we finally have Lemma 1.

## 8. Derive the label distribution ratio Loss

In GLS, we have $\mathcal{T}(z|y) \approx \mathcal{S}_t(z|y), \forall t$, then we suppose the predicted target distribution as $\bar{\mathcal{T}}(y)$. By simplifying the notation, we define $f(z) = \text{argmax}_y h(z,y)$ the most possible prediction label output, then we have:

$$\begin{aligned}\bar{\mathcal{T}}(y) &= \sum_{k=1}^{\mathcal{Y}}\mathcal{T}(f(z)=y|Y=k)\mathcal{T}(Y=k) = \sum_{k=1}^{\mathcal{Y}}\mathcal{S}_t(f(z)=y|Y=k)\mathcal{T}(Y=k) \\ &= \sum_{i=1}^{\mathcal{Y}}\mathcal{S}_t(f(z)=y, Y=k)\alpha_t(k) = \bar{\mathcal{T}}_{\alpha_t}(y)\end{aligned}$$

The first equality comes from the definition of target label prediction distribution, $\bar{\mathcal{T}}(y) = \mathbb{E}_{\mathcal{T}(z)}\mathbf{1}\{f(z)=y\} = \mathcal{T}(f(z)=y) = \sum_{k=1}^{\mathcal{Y}}\mathcal{T}(f(z)=y, Y=k) = \sum_{k=1}^{\mathcal{Y}}\mathcal{T}(f(z)=y|Y=k)\mathcal{T}(Y=k)$.

The second equality $\mathcal{T}(f(z)=y|Y=k) = \mathcal{S}_t(f(z)=y|Y=k)$ holds since $\forall t$, $\mathcal{T}(z|y) \approx \mathcal{S}_t(z|y)$, then for the shared hypothesis $f$, we have $\mathcal{T}(f(z)=y|Y=k) = \mathcal{S}_t(f(z)=y|Y=k)$.

The term $\mathcal{S}_t(f(z)=y, Y=k)$ is the (expected) source prediction confusion matrix, and we denote its empirical (observed) version as $\hat{\mathcal{S}}_t(f(z)=y, Y=k)$.

Based on this idea, in practice we want to find a $\hat{\alpha}_t$ to match the two predicted distribution $\bar{\mathcal{T}}$ and $\bar{\mathcal{T}}_{\hat{\alpha}_t}$. If we adopt the KL-divergence as the metric, we have:

$$\begin{aligned}\min_{\hat{\alpha}_t} D_{\text{KL}}(\bar{\mathcal{T}}\|\bar{\mathcal{T}}_{\hat{\alpha}_t}) &= \min_{\hat{\alpha}_t}\mathbb{E}_{y\sim\bar{\mathcal{T}}}\log(\frac{\bar{\mathcal{T}}(y)}{\bar{\mathcal{T}}_{\hat{\alpha}_t}(y)}) = \min_{\hat{\alpha}_t} -\mathbb{E}_{y\sim\bar{\mathcal{T}}}\log(\bar{\mathcal{T}}_{\hat{\alpha}_t}(y)) \\ &= \min_{\hat{\alpha}_t} -\sum_y \bar{\mathcal{T}}(y)\log(\sum_{k=1}^{\mathcal{Y}}\mathcal{S}_t(f(z)=y, Y=k)\hat{\alpha}_t(k))\end{aligned}$$

We should notice the nature constraints of label ratio: $\{\hat{\alpha}_t(y) \ge 0, \sum_y \hat{\alpha}_t(y)\hat{\mathcal{S}}_t(y) = 1\}$. Based on this principle, we proposed the optimization problem to estimate each label ratio. We adopt its empirical counterpart, the empirical confusion matrix $C_{\hat{\mathcal{S}}_t}[y,k] = \hat{\mathcal{S}}_t[f(z)=y, Y=k]$, then the optimization loss can be expressed as:

$$\begin{aligned}\min_{\hat{\alpha}_t} \quad & -\sum_{y=1}^{|\mathcal{Y}|}\bar{\mathcal{T}}(y)\log(\sum_{k=1}^{|\mathcal{Y}|}C_{\hat{\mathcal{S}}_t}[y,k]\hat{\alpha}_t(k)) \\ \text{s.t.} \quad & \forall y\in\mathcal{Y}, \hat{\alpha}_t(y) \ge 0, \quad \sum_y \hat{\alpha}_t(y)\hat{\mathcal{S}}_t(y) = 1\end{aligned}$$

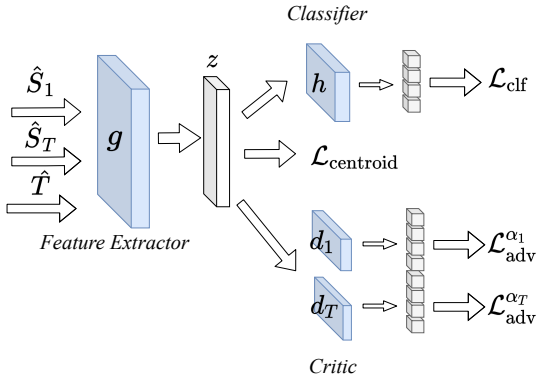



*Figure 2.* Network Structure of Proposed Approach. It consists three losses: the weighted Classification losses; the centroid matching for explicit conditional matching; the weighted adversarial loss for implicit conditional matching, showed in Eq. (3)

## 9. Label Partial Multi-source unsupervised DA

The key difference between multi-conventional and partial unsupervised DA is the estimation step of $\hat{\alpha}_t$. In fact, we only add a sparse constraint for estimating each $\hat{\alpha}_t$:

$$\begin{aligned} \min_{\hat{\alpha}_t} \quad & -\sum_{y=1}^{|\mathcal{Y}|} \bar{\mathcal{T}}(y) \log(\sum_{k=1}^{|\mathcal{Y}|} C_{\hat{\mathcal{S}}_t}[y,k]\hat{\alpha}_t(k)) + C_2\|\hat{\alpha}_t\|_1 \\ \text{s.t.} \quad & \forall y \in \mathcal{Y}, \hat{\alpha}_t(y) \geq 0, \quad \sum_y \hat{\alpha}_t(y)\hat{\mathcal{S}}_t(y) = 1 \end{aligned} \tag{2}$$

Where $C_2$ is the hyper-parameter to control the level of target label sparsity, to estimate the target label distribution. In the paper, we denote $C_2 = 0.1$.

## 10. Explicit and Implicit conditional learning

Inspired by Theorem 2, we need to learn the function $g : \mathcal{X} \to \mathcal{Z}$ and $h : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$ to minimize:

$$\min_{g,h} \sum_t \boldsymbol{\lambda}[t]\hat{R}_{\mathcal{S}_t}^{\hat{\alpha}_t}(h,g) + C_0 \sum_t \boldsymbol{\lambda}[t]\mathbb{E}_{y\sim\hat{\mathcal{T}}(y)} W_1(\hat{\mathcal{S}}_t(z|Y=y)\|\hat{\mathcal{T}}(z|Y=y))$$

This can be equivalently expressed as:

$$\begin{aligned} \min_{g,h} \sum_t \boldsymbol{\lambda}[t]\hat{R}_{\mathcal{S}_t}^{\alpha_t}(h,g) + \epsilon C_0 \sum_t \boldsymbol{\lambda}[t]\mathbb{E}_{y\sim\hat{\mathcal{T}}(y)} W_1(\hat{\mathcal{S}}_t(z|Y=y)\|\hat{\mathcal{T}}(z|Y=y)) \\ + (1-\epsilon)C_0 \sum_t \boldsymbol{\lambda}[t]\mathbb{E}_{y\sim\hat{\mathcal{T}}(y)} W_1(\hat{\mathcal{S}}_t(z|Y=y)\|\hat{\mathcal{T}}(z|Y=y)) \end{aligned}$$

Due to the explicit and implicit approximation of conditional distance, we then optimize an alternative form:

$$\begin{aligned} \min_{g,h} \max_{d_1,\ldots,d_T} & \underbrace{\sum_t \boldsymbol{\lambda}[t]\hat{R}_{\mathcal{S}_t}^{\hat{\alpha}_t}(h,g)}_{\text{Classification Loss}} + \epsilon C_0 \underbrace{\sum_t \boldsymbol{\lambda}[t]\mathbb{E}_{y\sim\hat{\mathcal{T}}(y)}\|\mathbf{C}_t^y - \mathbf{C}^y\|_2}_{\text{Explicit Conditional Loss}} \\ & + (1-\epsilon)C_0 \underbrace{\sum_t \boldsymbol{\lambda}[t][\mathbb{E}_{z\sim\hat{\mathcal{S}}_t(z)}\bar{\alpha}^t(z)d(z) - \mathbb{E}_{z\sim\hat{\mathcal{T}}(z)}d(z)]}_{\text{Implicit Conditional Loss}} \end{aligned} \tag{3}$$

Where

- $\mathbf{C}_t^y = \sum_{(z_t,y_t)\sim\hat{\mathcal{S}}_t} \mathbf{1}_{\{y_t=y\}} z_t$ the centroid of label $Y=y$ in source $\mathcal{S}_t$.
- $\mathbf{C}^y = \sum_{(z_t,y_p)\sim\hat{\mathcal{T}}} \mathbf{1}_{\{y_p=y\}} z_t$ the centroid of pseudo-label $Y=y_p$ in target $\mathcal{S}_t$. (If it is the unsupervised DA scenarios).
- $\bar{\alpha}_t(z) = \mathbf{1}_{\{(z,y)\sim\mathcal{S}_t\}}\hat{\alpha}_t(Y=y)$, namely if each pair observation $(z,y)$ from the distribution, then $\bar{\alpha}_t(Z=z) = \hat{\alpha}_t(Y=y)$.
- $d_1,\cdots,d_T$ are domain discriminator (or critic function) restricted within 1-Lipschitz function.
- $\epsilon\in[0,1]$ is the adjustment parameter in the trade-off of explicit and implicit learning. We fix $\epsilon=0.5$ in the experiments.
- $\hat{\mathcal{T}}(y)$ empirical target label distribution. (In the unsupervised DA scenarios, we approximate it by predicted target label distribution $\bar{\mathcal{T}}(y)$.)

**Gradient Penalty** In order to enforce the Lipschitz property of the statistic critic function, we adopt the gradient penalty term (Gulrajani et al., 2017). More concretely, given two samples $z_s\sim\mathcal{S}_t(z)$ and $z_t\sim\mathcal{T}(z)$ we generate an interpolated sample $z_{\text{int}} = \xi z_s + (1-\xi)z_t$ with $\xi\sim\text{Unif}[0,1]$. Then we add a gradient penalty $\|\nabla d(z_{\text{int}})\|_2^2$ as a regularization term to control the Lipschitz property w.r.t. the discriminator $d_1,\cdots,d_T$.

## 11. Algorithm Descriptions

We propose a detailed pipeline of the proposed algorithm in the following, shown in Algorithm 1 and 2. As for updating $\boldsymbol{\lambda}$ and $\alpha_t$, we iteratively solve the convex optimization problem after each training epoch and updating them by using the moving average technique.

For solving the $\boldsymbol{\lambda}$ and $\alpha_t$, we notice that frequently updating these two parameters in the mini-batch level will lead to an instability result during the training. [5] As a consequence, we compute the accumulated confusion matrix, weighted prediction risk, and conditional Wasserstein distance for the whole training epoch and then solve the optimization problem. We use CVXPY to optimize the two standard convex losses. [6]

**Comparison with different time and memory complexity.** We discuss the time and memory complexity of our approach.

Time complexity: In computing each batch we need to compute $T$ re-weighted loss, $T$ domain adversarial loss and $T$ explicit conditional loss. Then our computational complexity is still $(O)(T)$ during the mini-batch training, which is comparable with recent SOTA such as MDAN and DARN. In addition, after each training epoch we need to estimate $\alpha_t$ and $\boldsymbol{\lambda}$, which can have time complexity $\mathcal{O}(T|\mathcal{Y}|)$ with each epoch. (If we adopt SGD to solve these two convex problems). Therefore, the our proposed algorithm is time complexity $\mathcal{O}(T|\mathcal{Y}|)$. The extra $\mathcal{Y}$ term in time complexity is due to the approach of label shift in the designed algorithm.

Memory Complexity: Our proposed approach requires $\mathcal{O}(T)$ domain discriminator and $\mathcal{O}(T|\mathcal{Y}|)$ class-feature centroids. By the contrary, MDAN and DARN require $\mathcal{O}(T)$ domain discriminator and M3SDA and MDMN require $\mathcal{O}(T^2)$ domain discriminators. Since our class-feature centroids are defined in the latent space $(z)$, then the memory complexity of the class-feature centroids can be much smaller than domain discriminators.

[5] In the label distribution shift scenarios, the mini-batch datasets are highly labeled imbalanced. If we evaluate $\alpha_t$ over the mini-batch, it can be computationally expensive and unstable.

[6] The optimization problem w.r.t. $\alpha_t$ and $\boldsymbol{\lambda}$ is not large scale, then using the standard convex solver is fast and accurate.

**Algorithm 1** Wasserstein Aggregation Domain Network (unsupervised scenarios, one iteration)

**Require:** Labeled source samples $\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_T$, Target samples $\hat{\mathcal{T}}$
**Ensure:** Label distribution ratio $\hat{\alpha}_t$ and task relation simplex $\boldsymbol{\lambda}$. Feature Learner $g$, Classifier $h$, Statistic critic function $d_1, \ldots, d_T$, class centroid for source $\mathbf{C}_t^y$ and target $\mathbf{C}^y$ ($\forall t = [1, T], y \in \mathcal{Y}$).
$\triangleright\triangleright\triangleright$ DNN Parameter Training Stage (fixed $\alpha_t$ and $\boldsymbol{\lambda}$) $\triangleleft\triangleleft\triangleleft$
**for** mini-batch of samples $(\mathbf{x}_{\mathcal{S}_1}, \mathbf{y}_{\mathcal{S}_1}) \sim \hat{\mathcal{S}}_1, \ldots, (\mathbf{x}_{\mathcal{S}_T}, \mathbf{y}_{\mathcal{S}_T}) \sim \hat{\mathcal{S}}_T, (\mathbf{x}_{\mathcal{T}}) \sim \hat{\mathcal{T}}$ **do**
  Predict target pseudo-label $\bar{\mathbf{y}}_{\mathcal{T}} = \text{argmax}_y h(g(\mathbf{x}_{\mathcal{T}}), y)$
  Compute source confusion matrix for each batch (un-normalized)
    $C_{\hat{\mathcal{S}}_t} = \#[\text{argmax}_{y'} h(z, y') = y, Y = k]$ $(t = 1, \ldots, T)$
  Compute the *batched* class centroid for source $C_t^y$ and target $C^y$.
  Moving Average for update source/target class centroid: (We set $\epsilon_1 = 0.7$)
    Source class centroid update $\mathbf{C}_t^y = \epsilon_1 \times \mathbf{C}_t^y + (1 - \epsilon_1) \times C_t^y$
    Target class centroid update $\mathbf{C}^y = \epsilon_1 \times \mathbf{C}^y + (1 - \epsilon_1) \times C^y$
  Updating $g, h, d_1, \ldots, d_T$ (SGD and Gradient Reversal), based on Eq.(3)
**end for**
$\triangleright\triangleright\triangleright$ Estimation $\hat{\alpha}_t$ and $\boldsymbol{\lambda}$ $\triangleleft\triangleleft\triangleleft$
Compute the global(normalized) source confusion matrix
$C_{\hat{\mathcal{S}}_t} = \hat{\mathcal{S}}_t[\text{argmax}_{y'} h(z, y') = y, Y = k]$ $(t = 1, \ldots, T)$
Solve $\alpha_t$ (denoted as $\{\alpha_t'\}_{t=1}^T$) (Or Eq.(2)) in the partial scenario).
Update $\alpha_t$ by moving average: $\alpha_t = \epsilon_1 \times \alpha_t + (1 - \epsilon_1) \times \alpha_t'$
Compute the weighted loss and weighted centroid distance, then solve $\boldsymbol{\lambda}$ (denoted as $\boldsymbol{\lambda}'$) from Sec. 2.3.
Updating $\boldsymbol{\lambda}$ by moving average: $\boldsymbol{\lambda} = 0.8 \times \boldsymbol{\lambda} + 0.2 \times \boldsymbol{\lambda}'$

## 12. Dataset Description and Experimental Details

### 12.1. Amazon Review Dataset

We used the amazon review dataset (Blitzer et al., 2007). It contains four domains (Books, DVD, Electronics, and Kitchen) with positive (label "1") and negative product reviews (label "0"). The data size is 6465 (Books), 5586 (DVD), 7681 (Electronics), and 7945 (Kitchen). We follow the common data pre-processing strategies (Chen et al., 2012): use the bag-of-words (BOW) features then extract the top-5000 frequent unigram and bigrams of all the reviews.

We also noticed the original data-set are label balanced $\mathcal{D}(y = 0) = \mathcal{D}(y = 1)$. To enhance the benefits of the proposed approach, we create a new dataset with label distribution drift. Specifically, in the experimental settings, we randomly drop $50\%$ data with label "0" (negative reviews) for all the source data while keeping the target identical, showing in Fig (3).

We choose the MLP model with

- feature representation function $g$: $[5000, 1000]$ units
- Task prediction and domain discriminator function $[1000, 500, 100]$ units,

We choose the dropout rate as $0.7$ in the hidden and input layers. The hyper-parameters are chosen based on cross-validation. The neural network is trained for $50$ epochs and the mini-batch size is 20 per domain. The optimizer is Adadelta with a learning rate of 0.5.

**Experimental Setting** We use the amazon Review dataset for two transfer learning scenarios (limited target labels and unsupervised DA). We first randomly select 2K samples for each domain. Then we create a drifted distribution of each source, making each source $\approx 1500$ and target sample still 2K.

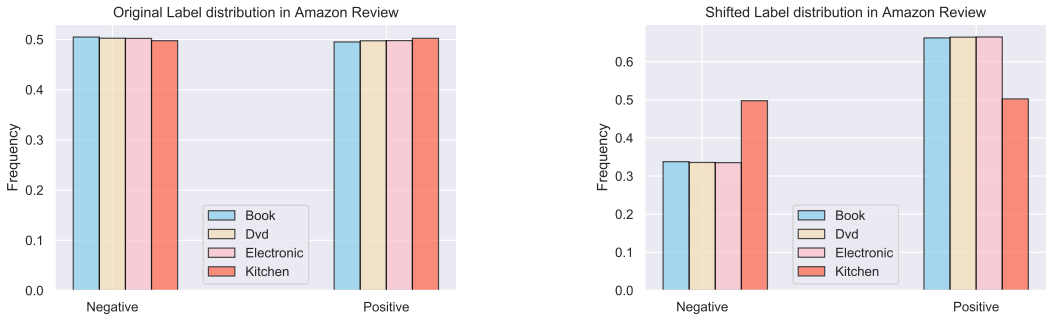
In the unsupervised DA, we use these labeled source tasks and *unlabelled* target task, which aims to predict the labels on the target domain.

In the conventional transfer learning, we random sample only $10\%$ dataset ($\approx 200$ samples) as the target training set and the rest $90\%$ samples as the target test set.

**Algorithm 2** Wasserstein Aggregation Domain Network (Limited Target Data, one iteration)

**Require:** Labeled source samples $\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_T$, Target samples $\hat{\mathcal{T}}$, Label shift ratio $\alpha_t$
**Ensure:** Task relation simplex $\boldsymbol{\lambda}$. Feature Learner $g$, Classifier $h$, Statistic critic function $d_1, \ldots, d_T$, class centroid for source $\mathbf{C}_t^y$ and target $\mathbf{C}^y$ ($\forall t = [1, T], y \in \mathcal{Y}$).
1: $\triangleright\triangleright\triangleright$ DNN Parameter Training Stage (fixed $\boldsymbol{\lambda}$) $\triangleleft\triangleleft\triangleleft$
2: **for** mini-batch of samples $(\mathbf{x}_{\mathcal{S}_1}, \mathbf{y}_{\mathcal{S}_1}) \sim \hat{\mathcal{S}}_1, \ldots, (\mathbf{x}_{\mathcal{S}_T}, \mathbf{y}_{\mathcal{S}_T}) \sim \hat{\mathcal{S}}_T, (\mathbf{x}_{\mathcal{T}}) \sim \hat{\mathcal{T}}$ **do**
3: Compute the *batched* class centroid for source $C_t^y$ and target $C^y$.
4: Moving Average for update source/target class centroid: (We set $\epsilon_1 = 0.7$)
5: Source class centroid update $\mathbf{C}_t^y = \epsilon_1 \times \mathbf{C}_t^y + (1 - \epsilon_1) \times C_t^y$
6: Target class centroid update $\mathbf{C}^y = \epsilon_1 \times \mathbf{C}^y + (1 - \epsilon_1) \times C^y$
7: Updating $g, h, d_1, \ldots, d_T$ (SGD and Gradient Reversal), based on Eq.(3).
8: **end for**
9: $\triangleright\triangleright\triangleright$ Estimation $\boldsymbol{\lambda}$ $\triangleleft\triangleleft\triangleleft$
10: Solve $\boldsymbol{\lambda}$ by Sec. 2.3. (denoted as $\boldsymbol{\lambda}'$)
11: Updating $\boldsymbol{\lambda}$ by moving average: $\boldsymbol{\lambda} = \epsilon_1 \times \boldsymbol{\lambda} + (1 - \epsilon_1) \times \boldsymbol{\lambda}'$

We select $C_0 = 0.01$ and $C_1 = 1$ for these two transfer scenarios. In both practical settings, we set the maximum training epoch as 50.




(a) (b)

*Figure 3.* Amazon Review dataset (a) Original Label Training Distribution; (b) Label-Shifted distribution with sources tasks: Book, Dvd, Electronic, and target task Kitchen. We randomly drop $50\%$ of the negative reviews for all the source distribution while keeping the target label distribution unchanged.

### 12.2. Digit Recognition

We follow the same settings of (Ganin et al., 2016) and we use four-digit recognition datasets in the experiments MNIST, USPS, SVHN, and Synth. MNIST and USPS are the standard digits recognition task. Street View House Number (SVHN) (Ganin et al., 2016) is the digit recognition dataset from house numbers in Google Street View Images. Synthetic Digits (Synth) (Ganin et al., 2016) is a synthetic dataset that by various transforming SVHN dataset.

We also visualize the label distribution in these four datasets. The original datasets show an almost uniform label distribution on the MNIST as well as Synth, (showing in Fig. 5 (a)). In our paper, we generate a label distribution drift on the source datasets for each multi-source transfer learning. Concretely, we drop $50\%$ of the data on digits 5-9 of all the sources while we keep the target label distribution unchanged. (Fig. 5 (b) illustrated one example with sources: Mnist, USPS, SVHN, and Target Synth. We drop the labels only on the sources.)

MNIST and USPS images are resized to $32 \times 32$ and represented as 3-channel color images to match the shape of the other three datasets. Each domain has its own given training and test sets when downloaded. Their respective training sample sizes are 60000, 7219, 73257, 479400, and the respective test sample sizes are 10000, 2017, 26032, 9553.

The model structure is shown in Fig. 4. There is no dropout and the hyperparameters are chosen based on cross-validation. It is trained for 60 epochs and the mini-batch size is 128 per domain. The optimizer is Adadelta with a learning rate of 1.0. We adopted $\gamma = 0.5$ for MDAN and $\gamma = 0.1$ for DARN in the baseline (Wen et al., 2020).

**Experimental Setting** We use the Digits dataset for two transfer learning scenarios (limited target labels and unsupervised DA). Notice the USPS data has only 7219 samples and the digits dataset is relatively simple. We first randomly select 7K samples for each domain. We create a drifted distribution of each source, making each source $\approx 5300$, and the target sample still 7K.

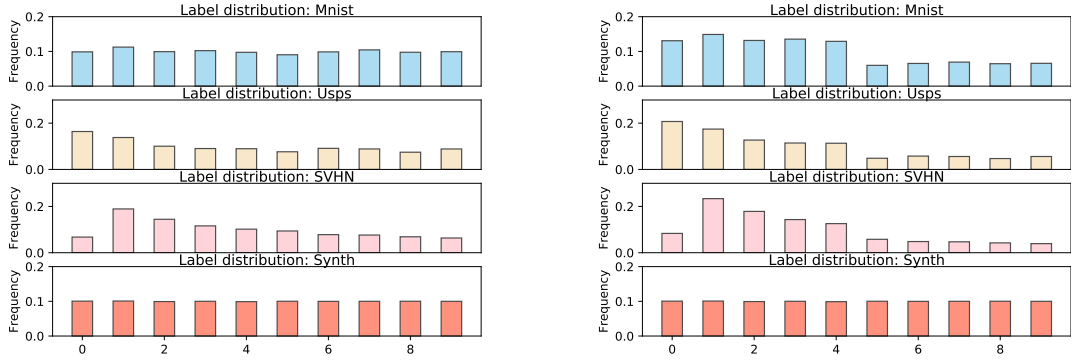
In the unsupervised DA, we use these labeled source tasks and *unlabelled* target task, which aims to predict the labels on the target domain.

In the transfer learning with limited data, we random sample only $10\%$ dataset ($\approx 700$ samples) as the target training set and the rest $90\%$ samples as the target test set.

We select $C_0 = 0.01$ and $C_1$ as the maximum prediction loss $C_1 = \max_t R^{\alpha_t}(h)$ as the hyper-parameters across these two scenarios. The maximum training epoch is $60$.

1. Feature extractor: with 3 convolution layers.
   'layer1': 'conv': [3, 3, 64], 'relu': [], 'maxpool': [2, 2, 0],
   'layer2': 'conv': [3, 3, 128], 'relu': [], 'maxpool': [2, 2, 0],
   'layer3': 'conv': [3, 3, 256], 'relu': [], 'maxpool': [2, 2, 0],
2. Task prediction: with 3 fully connected layers.
   'layer1': 'fc': [*, 512], 'act_fn': 'relu',
   'layer2': 'fc': [512, 100], 'act_fn': 'relu',
   'layer3': 'fc': [100, 10],
3. Domain Discriminator: with 2 fully connected layers.
   *reverse_gradient*()
   'layer1': 'fc': [*, 256], 'act_fn': 'relu',
   'layer2': 'fc': [256, 1],

*Figure 4.* Neural Network Structure in the digits recognition (Ganin et al., 2016)




(a) (b)

*Figure 5.* One example in Digits dataset with Sources: MNIST, USPS, SVHN and Target Synth. We randomly drop $50\%$ data on digits 5-9 in all sources while keeping target label distribution unchanged.

### 12.3. Office-Home dataset

To show the dataset in the complex scenarios, we use the challenging Office-Home dataset (Venkateswara et al., 2017). It contains images of 65 objects such as a spoon, sink, mug, and pen from four different domains: Art (paintings, sketches, and/or artistic depictions), Clipart (clipart images), Product (images without background), and Real-World (regular images

captured with a camera). One of the four datasets is chosen as an unlabelled target domain and the other three datasets are used as labeled source domains.

The dataset size is 2427 (Art), 4365 (Clipart), 4439 (Product), 4357 (Real-World). We follow the same training/test procedure as (Wen et al., 2020). We additionally visualize the label distribution $\mathcal{D}(y)$ in four domains in Fig.**??**, which illustrated the inherent different label distributions. We did not re-sample the source label distribution to uniform distribution in the data pre-processing step. All the baselines are evaluated under the same setting.

We use the ResNet50 (He et al., 2016) pretrained from the ImageNet in PyTorch as the base network for feature learning and put an MLP with the network structure shown in Fig. 7.

**Experimental Settings** We use the original Office-Home dataset for two transfer learning scenarios (unsupervised DA and label-partial unsupervised DA). We use SGD optimizer with learning rate 0.005, momentum 0.9 and weight_decay value 1e-3. It is trained for 100 epochs and the mini-batch size is 32 per domain. As for the baselines, MDAN use $\gamma$ = 1.0 while DARN use $\gamma$ = 0.5. We select $C_0 = 0.01$ and $C_1$ as the maximum prediction loss $C_1 = \max_t R^{\alpha_t}(h)$ as the hyper-parameters across these two scenarios.

In the multi-source unsupervised partial DA, we randomly select 35 classes from the target (by repeating 3 samplings), then at each sampling we run 5 times. The final result is based on these $3 \times 5 = 15$ repetitions.

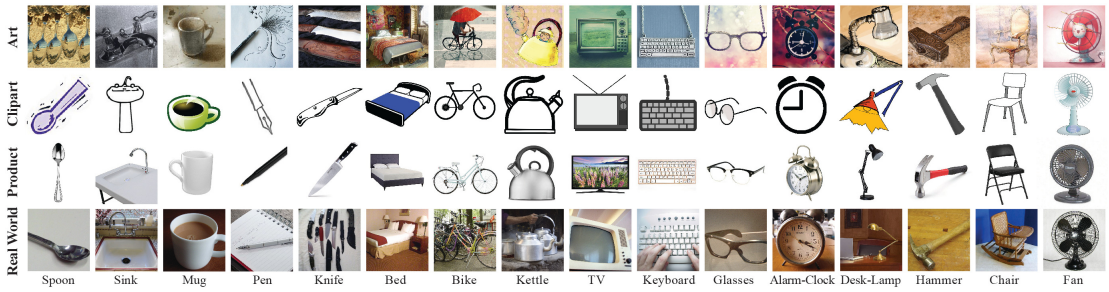



*Figure 6.* Samples Images From Office-Home dataset (Venkateswara et al., 2017), which consists four domains with non-uniform label distribution.

1. Feature extractor: ResNet50 (He et al., 2016),
2. Task prediction: with 3 fully connected layers.
   'layer1': 'fc': [*, 256], 'batch_normalization', 'act_fn': 'Leaky_relu',
   'layer2': 'fc': [256, 256], 'batch_normalization', 'act_fn': 'Leaky_relu',
   'layer3': 'fc': [256, 65],
3. Domain Discriminator: with 3 fully connected layers.
   *reverse_gradient*()
   'layer1': 'fc': [*, 256], 'batch_normalization', 'act_fn': 'Leaky_relu',
   'layer2': 'fc': [256, 256], 'batch_normalization', 'act_fn': 'Leaky_relu',
   'layer3': 'fc': [256, 1], 'Sigmoid',

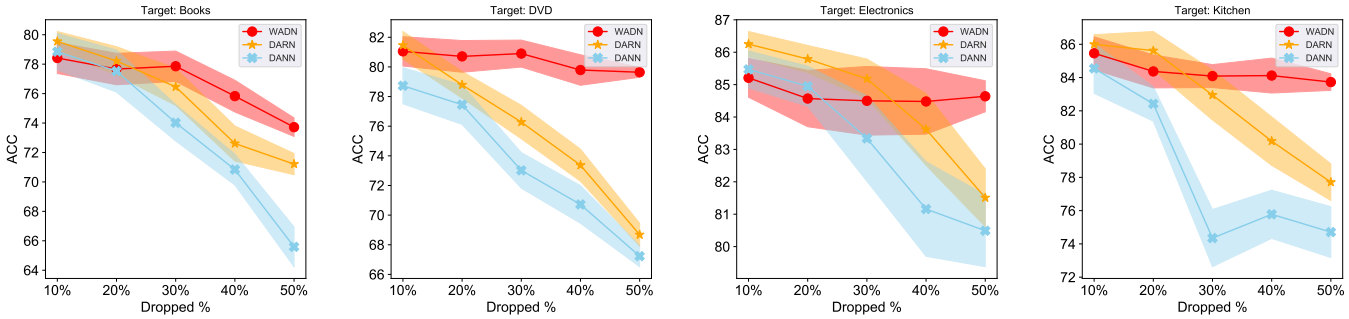
*Figure 7.* Neural Network Structure in the Office-Home

## 13. Analysis in Unsupervised DA

### 13.1. Ablation Study: Different Dropping Rate

To show the effectiveness of our proposed approach, we change the drop rate of the source domains, showing in Fig.(8). We observe that in task Book, DVD, Electronic, and Kitchen, the results are significantly better under a large label-shift. In the

initialization with almost no label shift, the state-of-the-art DARN illustrates a slightly better ($< 1\%$) result.




(a) Target: Book (b) Target: DVD (c) Target: Electronics (d) Target: Kitchen

*Figure 8.* Different label drift levels on Amazon Dataset. Larger dropping rate means higher label shift.

### 13.2. Additional Analysis on Amazon Dataset

We present two additional results to illustrate the working principles of WADN, showing in Fig. (9).

We visualize the evolution of $\boldsymbol{\lambda}$ between DARN and WADN, which both used theoretical principled approach to estimate $\boldsymbol{\lambda}$. We observe that in the source shifted data, DARN shows an inconsistent estimator of $\boldsymbol{\lambda}$. This is different from the observation of (Wen et al., 2020). We think it may in the conditional and label distribution shift problem, using $\hat{R}_{\mathcal{S}}(h(z)) + \text{Discrepancy}(\mathcal{S}(z), \mathcal{T}(z))$ to update $\boldsymbol{\lambda}$ is unstable. In contrast, WADN illustrates a relative consistent estimator of $\boldsymbol{\lambda}$ under the source shifted data.

In addition, WARN gradually and correctly estimates the unbalanced source data and assign higher wights $\alpha_t$ for label $y = 0$ (first row of Fig.(9)). These principles in WADN jointly promote significantly better results.

### 13.3. Additional Analysis on Digits Dataset

We show the evolution of $\hat{\alpha}_t$ on WADN, which verifies the correctness of our proposed principle. Since we drop digits 5-9 in the source domains, the results in Fig. (10) illustrate a higher $\hat{\alpha}_t$ on these digits.

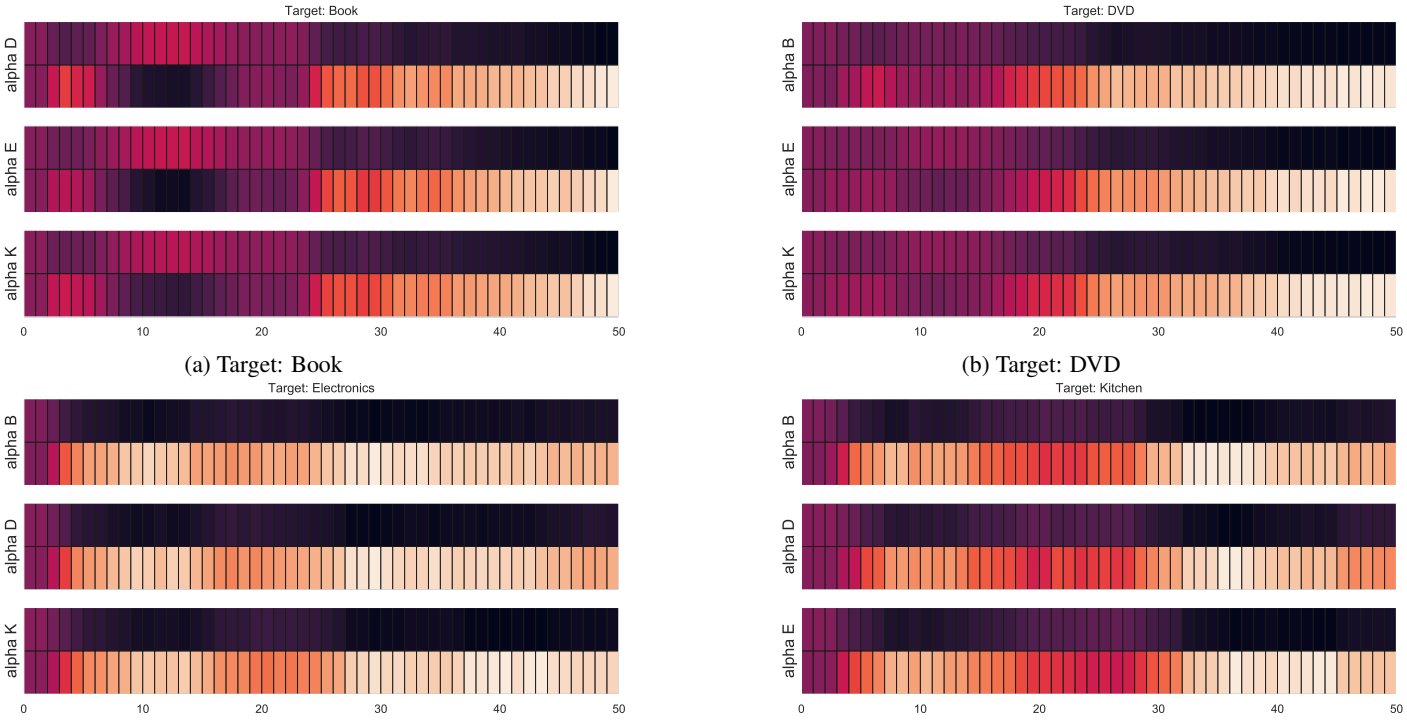
## 14. Partial multi-source Unsupervised DA

From Fig. (12), WADN is consistently better than other baselines, given different selected classes.

Besides, when fewer classes are selected, the accuracy in DANN, PADA, and DARN is not drastically dropping but maintaining a relatively stable result. We think the following possible reasons:

- The reported performances are based on the **average of different selected sub-classes rather than one sub-class selection.** From the statistical perspective, if we take a close look at the **variance**, the results in DANN are *much more unstable* (higher std) inducing by the different samplings. Therefore, the conventional domain adversarial training is improper for handling the partial transfer since it is not reliable and negative transfer still occurs.
- In multi-source DA, it is equally important to detect the non-overlapping classes and find the most similar sources. Comparing the baselines that only focus on one or two principles shows the importance of unified principles in multi-source partial DA.
- We also observe that in the Real-World dataset, the DANN improves the performance by a relatively large value. This is due to the inherent difficultly of the learning task itself. In fact, the Real-World domain illustrates a much higher performance compared with other domains. According to the Fano lower bound, *a task with smaller classes is generally easy to learn*. It is possible the vanilla approach showed improvement but still with a much higher variance.

Fig (13), (14) showed the estimated $\hat{\alpha}_t$ with different selected classes. The results validate the correctness of WADN in estimating the label distribution ratio.




(a) Target: Book

(b) Target: DVD

(c) Target: Electronics

(d) Target: Kitchen

*Figure 9.* Amazon Dataset. WADN approach: evolution of $\hat{\alpha}_t$ during the training. Darker indicates higher Value. Since we drop $y = 0$ in the sources, then the true $\alpha_t(0) > 1$ will be assigned with higher value.

## References


Akuzawa, K., Iwasawa, Y., and Matsuo, Y. Adversarial invariant feature learning with accuracy constraint for domain generalization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 315–331. Springer, 2019.

Balaji, Y., Sankaranarayanan, S., and Chellappa, R. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, pp. 998–1008, 2018.

Balaji, Y., Chellappa, R., and Feizi, S. Normalized wasserstein distance for mixture distributions with applications in adversarial learning and domain adaptation. *arXiv preprint arXiv:1902.00415*, 2019.

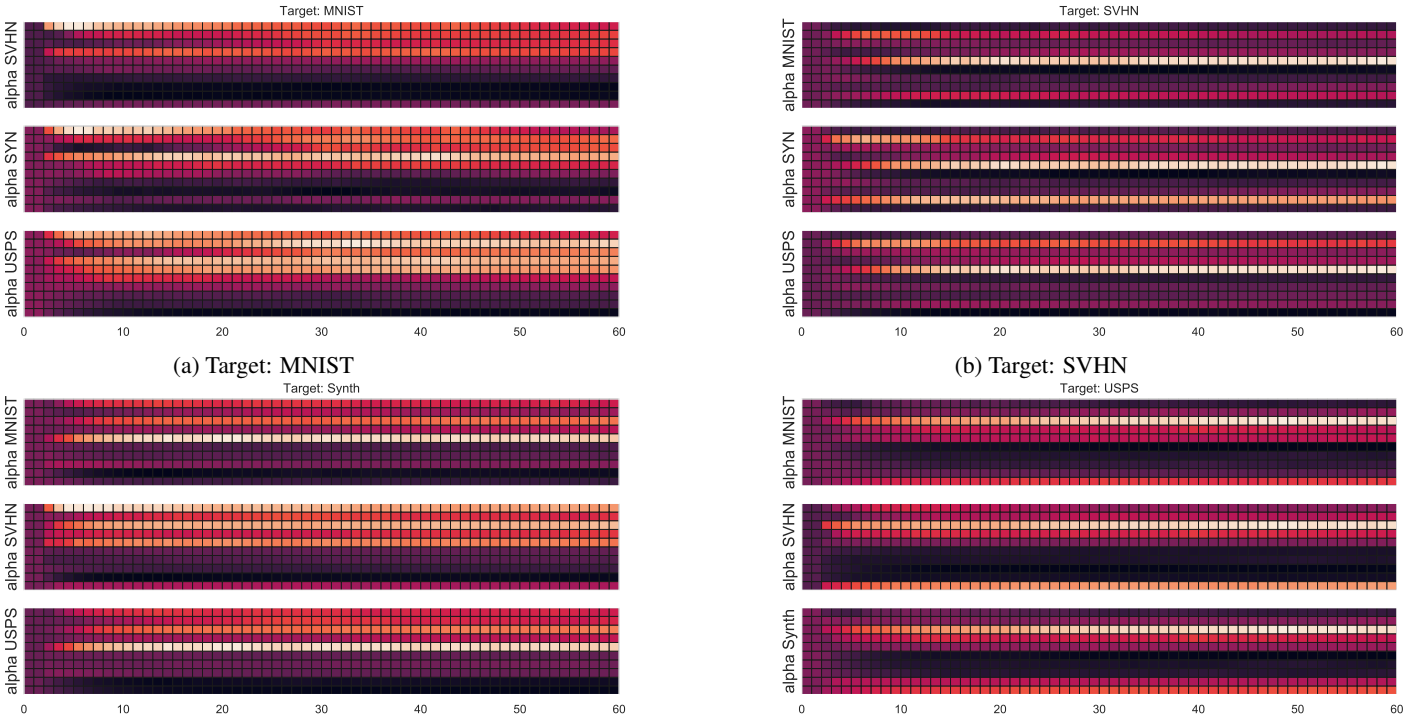
Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

Blitzer, J., Dredze, M., and Pereira, F. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 440–447, 2007.

Bucci, S., D'Innocente, A., and Tommasi, T. Tackling partial domain adaptation with self-supervision. In *International Conference on Image Analysis and Processing*, pp. 70–81. Springer, 2019.

Cao, Z., You, K., Long, M., Wang, J., and Yang, Q. Learning to transfer examples for partial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2985–2994, 2019.

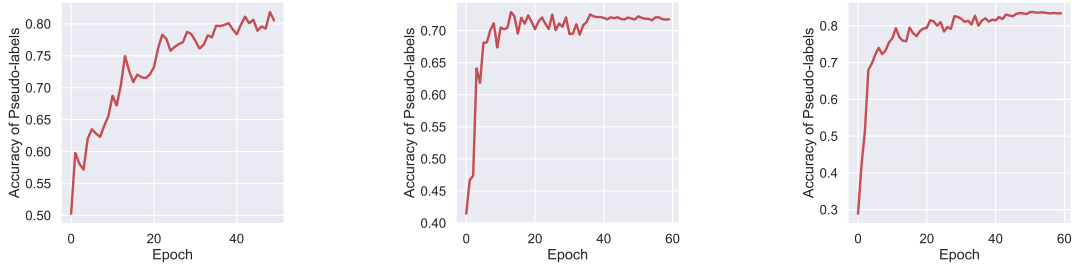
Chen, M., Xu, Z., Weinberger, K. Q., and Sha, F. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1627–1634, 2012.




(a) Target: MNIST (b) Target: SVHN

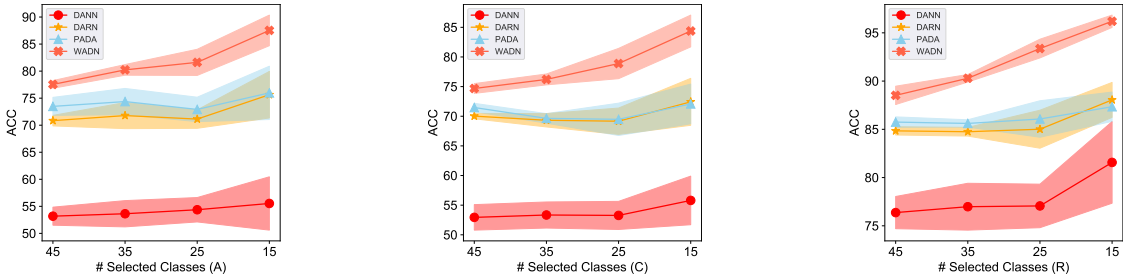
(c) Target: Synth (d) Target: USPS

*Figure 10.* Digits Dataset. WADN approach: evolution of $\hat{\alpha}_t$ during the training. Darker indicates higher value. Since we drop digits $5-9$ on source domain, therefore, $\alpha_t(y)$, $y \in [5, 9]$ will be assigned with a relative higher value.




(a) Amazon Review: target DVDs (b) Digits: target SVHN (c) Digits: target Synth

*Figure 11.* Evolution of accuracy w.r.t. the predicted target pseudo-labels in different tasks in unsupervised DA.




(a) Target: Art (b) Target: Clipart (c) Target: Real-World

*Figure 12.* Multi-source Label Partial DA: Performance with different target selected classes.

Chen, X., Awadallah, A. H., Hassan, H., Wang, W., and Cardie, C. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Chen, Z., Chen, C., Cheng, Z., Fang, K., and Jin, X. Selective transfer with reinforced transfer network for partial domain adaptation. In *AAAI Conference on Artificial Intelligence*, 2020.

Christodoulidis, S., Anthimopoulos, M., Ebner, L., Christe, A., and Mougiakakou, S. Multisource transfer learning with convolutional neural networks for lung pattern analysis. *IEEE journal of biomedical and health informatics*, 21(1):76–84, 2016.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hoffman, J., Kulis, B., Darrell, T., and Saenko, K. Discovering latent domains for multisource domain adaptation. In *European Conference on Computer Vision*, pp. 702–715. Springer, 2012.

Hoffman, J., Mohri, M., and Zhang, N. Algorithms and theory for multiple-source adaptation. In *Advances in Neural Information Processing Systems*, pp. 8246–8256, 2018.

Ilse, M., Tomczak, J. M., Louizos, C., and Welling, M. Diva: Domain invariant variational autoencoders. *arXiv preprint arXiv:1905.10427*, 2019.

Konstantinov, N. and Lampert, C. Robust learning from untrusted sources. In *International Conference on Machine Learning*, pp. 3488–3498, 2019.

Lee, J., Sattigeri, P., and Wornell, G. Learning new tricks from old dogs: Multi-source transfer learning from pre-trained networks. In *Advances in Neural Information Processing Systems*, pp. 4370–4380, 2019.

Li, J., Wu, W., Xue, D., and Gao, P. Multi-source deep transfer neural network algorithm. *Sensors*, 19(18):3992, 2019a.

Li, Y., Carlson, D. E., et al. Extracting relationships by multi-domain matching. In *Advances in Neural Information Processing Systems*, pp. 6798–6809, 2018a.

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., and Tao, D. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 624–639, 2018b.

Li, Y., Murias, M., Major, S., Dawson, G., and Carlson, D. On target shift in adversarial domain adaptation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 616–625, 2019b.

Lin, C., Zhao, S., Meng, L., and Chua, T.-S. Multi-source domain adaptation for visual sentiment classification. *arXiv preprint arXiv:2001.03886*, 2020.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Multiple source adaptation and the rényi divergence. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 367–374. AUAI Press, 2009.

Mansour, Y., Mohri, M., Suresh, A. T., and Wu, K. A theory of multiple-source adaptation with limited target labeled data. *arXiv preprint arXiv:2007.09762*, 2020.

Mohri, M. and Medina, A. M. New analysis and algorithm for learning with drifting distributions. In *International Conference on Algorithmic Learning Theory*, pp. 124–138. Springer, 2012.

Motiian, S., Piccirilli, M., Adjeroh, D. A., and Doretto, G. Unified deep supervised domain adaptation and generalization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Nguyen, X., Wainwright, M. J., Jordan, M. I., et al. On surrogate loss functions and f-divergences. *The Annals of Statistics*, 37(2):876–904, 2009.

Pei, Z., Cao, Z., Long, M., and Wang, J. Multi-adversarial domain adaptation. *arXiv preprint arXiv:1809.02176*, 2018.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. Adapting visual category models to new domains. In *European conference on computer vision*, pp. 213–226. Springer, 2010.

Sankaranarayanan, S., Balaji, Y., Castillo, C. D., and Chellappa, R. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8503–8512, 2018.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Stojanov, P., Gong, M., Carbonell, J. G., and Zhang, K. Data-driven approach to multiple-source domain adaptation. *Proceedings of machine learning research*, 89:3487, 2019.

Tan, B., Zhong, E., Xiang, E. W., and Yang, Q. Multi-transfer: Transfer learning with multiple views and multiple sources. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 243–251. SIAM, 2013.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027, 2017.

Wang, H., Yang, W., Lin, Z., and Yu, Y. Tmda: Task-specific multi-source domain adaptation via clustering embedded adversarial training. In *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1372–1377. IEEE, 2019.

Weed, J., Bach, F., et al. Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance. *Bernoulli*, 25(4A):2620–2648, 2019.

Wei, P., Sagarna, R., Ke, Y., Ong, Y.-S., and Goh, C.-K. Source-target similarity modelings for multi-source transfer gaussian process regression. In *International Conference on Machine Learning*, pp. 3722–3731, 2017.

Wen, J., Greiner, R., and Schuurmans, D. Domain aggregation networks for multi-source domain adaptation. *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Yao, Y. and Doretto, G. Boosting for transfer learning with multiple sources. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1855–1862. IEEE, 2010.

Zhang, J., Ding, Z., Li, W., and Ogunbona, P. Importance weighted adversarial nets for partial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8156–8164, 2018.

Zhang, Y. and Yang, Q. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

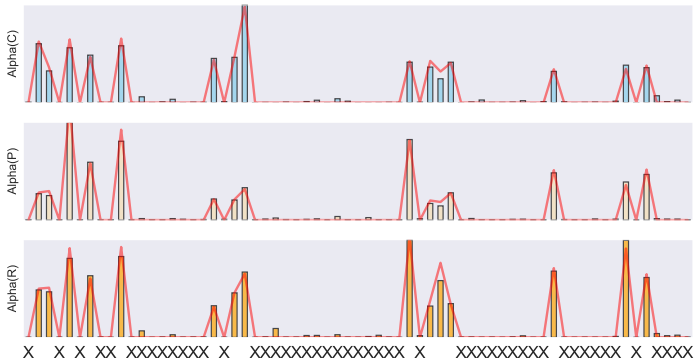
Zhang, Y. and Yeung, D.-Y. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2012.

Zhao, H., Zhang, S., Wu, G., Moura, J. M., Costeira, J. P., and Gordon, G. J. Adversarial multiple source domain adaptation. In *Advances in neural information processing systems*, pp. 8559–8570, 2018.

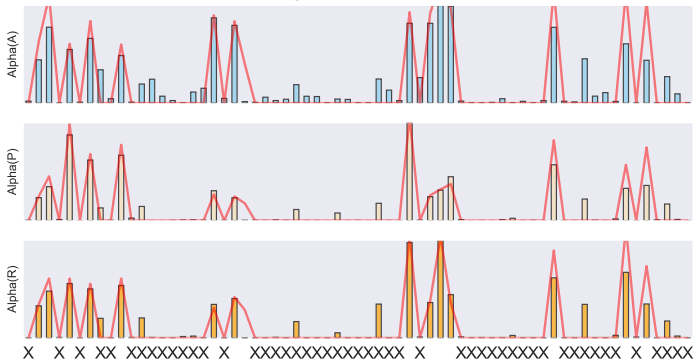
Zhao, S., Wang, G., Zhang, S., Gu, Y., Li, Y., Song, Z., Xu, P., Hu, R., Chai, H., and Keutzer, K. Multi-source distilling domain adaptation. *arXiv preprint arXiv:1911.11554*, 2019.

Zhao, S., Li, B., Xu, P., and Keutzer, K. Multi-source domain adaptation in the deep learning era: A systematic survey. *arXiv preprint arXiv:2002.12169*, 2020.

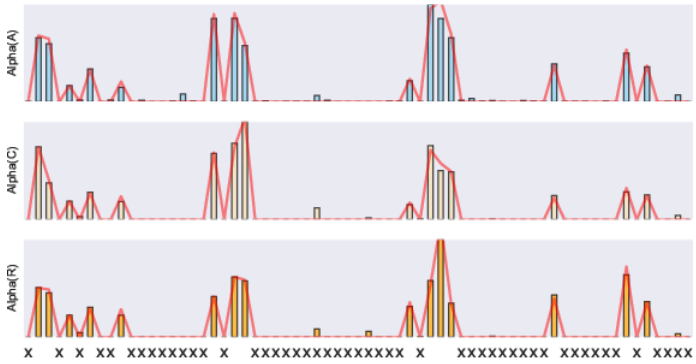
Zhu, Y., Zhuang, F., and Wang, D. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5989–5996, 2019.



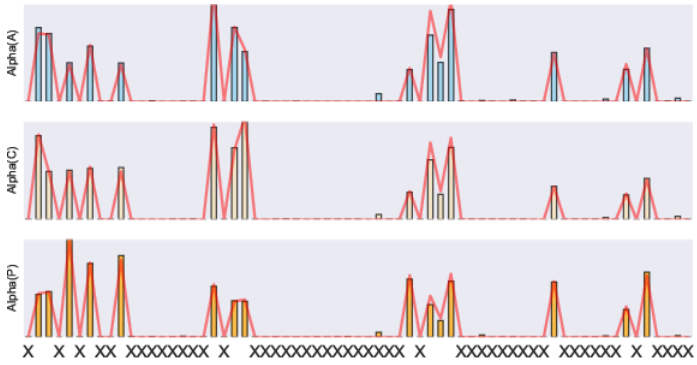

(a) Target: Art




(b) Target: Clipart

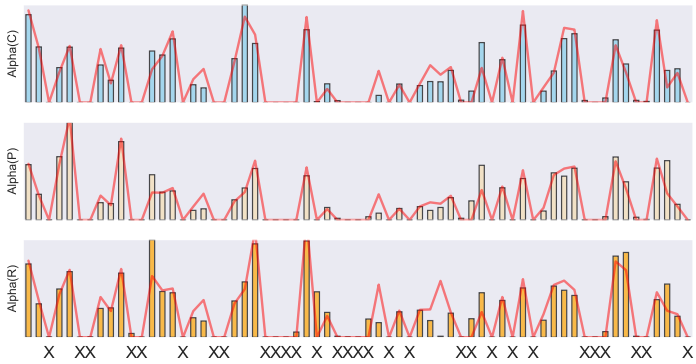



(c) Target: Product

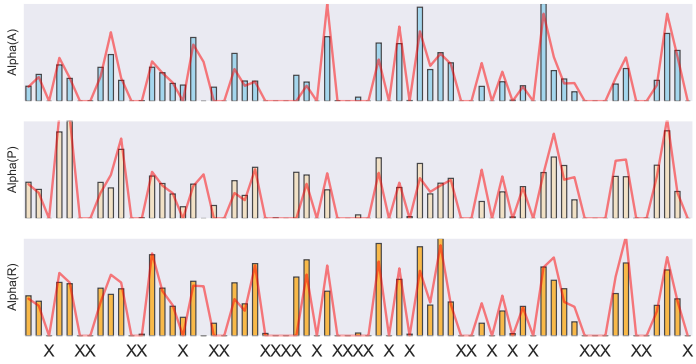



(d) Target: Real-World

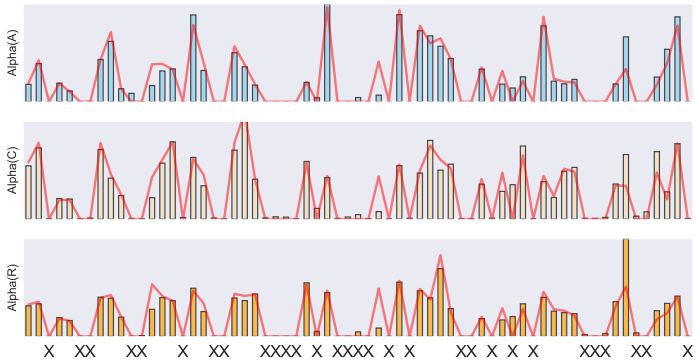
*Figure 13.* We select 15 classes and visualize estimated $\hat{\alpha}_t$ (the bar plot). The "X" along the x-axis represents the index of **dropped** 50 classes. The red curves are the ground-truth label distribution ratio.



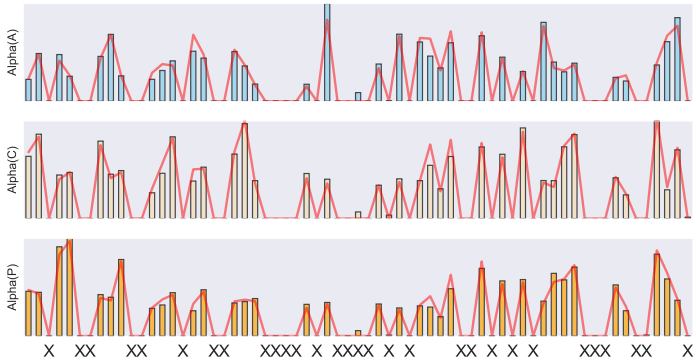

(a) Target: Art




(b) Target: Clipart




(c) Target: Product




(d) Target: Real-World

*Figure 14.* We select 35 classes and visualize estimated $\hat{\alpha}_t$ (the bar plot). The "X" along the x-axis represents the index of **dropped** 30 classes. The red curves are the ground-truth label distribution ratio.