A Definitions and Proofs

Definition A.1. For some $K \geq 0$, the set of K-Lipschitz functions denotes the set of functions f that verify:

$$\|f(x) - f(x')\| \leq K\|x - x'\|, \ \forall x, x' \in \mathcal{X}$$

In the coming proofs, we assume that the hypothesis class $\mathbb{H}$ is a subset of λ_H-Lipschitz functions, where λ_H is a positive constant, and we assume that the true labeling functions are λ-Lipschitz for some positive real number λ.

Definition A.2. For a distribution $\mathcal{D}$, we define the discrepancy between two functions f and g as:

$$\gamma_{\mathcal{D}}(f, g) = \mathbb{E}_{x\sim\mathcal{D}}\left[|f(x) - g(x)|\right]$$

We use g_T and g_S to represent the true labeling functions of the target and source domains, respectively. We use $\gamma_S(f) \doteq \gamma_{P_S}(f, g_S)$ and $\gamma_T(f) \doteq \gamma_{P_T}(f, g_T)$ to respectively denote the discrepancies of a hypothesis f to the true labeling function for the source and target domains.

Definition A.3. For a distribution $\mathcal{D}$ that can be represented as mixture of K sub-distribution, we define the discrepancy between two functions f and g as:

$$\gamma^k_{\mathcal{D}}(f, g) = \mathbb{E}_{x\sim\mathcal{D}^k}\left[|f(x) - g(x)|\right]$$

where we use $\mathcal{D}^k$ to represent the distribution of the k-th sub-distribution.

We use P^k_S/P^k_T to represent the distribution of the k-th subdomain of the source domain/target domain respectively. Thus we can use $\gamma^k_S(f) \doteq \gamma_{P^k_S}(f, g_S)$ and $\gamma^k_T(f) \doteq \gamma_{P^k_T}(f, g_T)$ to respectively denote the discrepancies of a hypothesis f to the true labeling function of the k-th subdomain of the source domain and that of the target domain.

Theorem A.4 (Full Domain Generalization Bound)**.** *For a hypothesis $f : \mathcal{X} \to [0, 1]$,*

$$\gamma_T(f) \leq \gamma_S(f) + (\lambda + \lambda_H)W_1(P_S, P_T) + \gamma^\star \tag{20}$$

where $\gamma^\star = \min_{f\in\mathbb{H}} \gamma_S(f) + \gamma_T(f)$, $\mathbb{H}$ is a hypothesis class included in the set of λ_H-Lipschitz functions, and the true functions g_T and g_S are both λ-Lipschitz functions (as defined in Definition A.1).

Proof. For a hypothesis $f : \mathcal{X} \to [0, 1]$ with $f \in \mathbb{H}$, we have that

$$\gamma_{P_T}(f, g_T) = \gamma_{P_T}(f, g_T) + \gamma_{P_S}(f, g_S) - \gamma_{P_S}(f, g_S) + \gamma_{P_S}(f, g_T) - \gamma_{P_S}(f, g_T) \tag{21}$$

And then bound the RHS by taking the absolute value of differences:

$$\begin{aligned} \gamma_{P_T}(f, g_T) &\leq \gamma_{P_S}(f, g_S) + |\gamma_{P_S}(f, g_T) - \gamma_{P_S}(f, g_S)| + |\gamma_{P_T}(f, g_T) - \gamma_{P_S}(f, g_T)| \\ &\leq \gamma_{P_S}(f, g_S) + \mathbb{E}_{x\sim\mathcal{P}_S}[|g_S(x) - g_T(x)|] + |\gamma_{P_T}(f, g_T) - \gamma_{P_S}(f, g_T)| \end{aligned} \tag{22}$$

As stated in Li et al. (2018), the first two terms proceed exactly as in Ben-David et al. (2010); further derivations are not provided here. We next provide an upper bound for the last term. Let P_S and P_T be the densities of X_S and X_T, respectively.

$$|\gamma_{P_T}(f, g_T) - \gamma_{P_S}(f, g_T)| \leq \left|\int (P_T(x) - P_S(x))|f(x) - g_T(x)|dx\right| \tag{23}$$

Since our hypothesis class $\mathbb{H}$ is assumed to be λ_H-Lipschitz and the true labeling functions are λ-Lipschitz, we have that for every function $f \in \mathbb{H}$, $h : x \mapsto |f(x) - g_T(x)|$ is $\lambda + \lambda_H$-Lipschitz and it takes its values in $[0, 1]$. Therefore,

$$\begin{aligned} |\gamma_{P_T}(f, g_T) - \gamma_{P_S}(f, g_T)| &\leq \sup_{h:\mathcal{X}\to[0,1],||h||\leq\lambda+\lambda_H} \left|\int (P_T(x) - P_S(x))h(x)dx\right| \\ &= \sup_{h:\mathcal{X}\to[0,1],||h||\leq\lambda+\lambda_H} |\mathbb{E}_{x\sim P_T}[h(x)] - \mathbb{E}_{x\sim P_S}[h(x)]| \end{aligned} \tag{24}$$

Note that due to the symmetric nature of the function space, i.e., if h is K-Lipschitz then $-h$ is K-Lipschitz, we can just pick either side to lead with and drop the absolute value, yielding

$$|\gamma_{P_T}(f, g_T) - \gamma_{P_S}(f, g_T)| \leq (\lambda + \lambda_H) W_1(P_S, P_T) \tag{25}$$

Following the Theorem 2 of Ben-David et al. (2010), we can also easily bound the target error $\gamma_{P_T}(f, g_T)$ by:

$$\gamma_{P_T}(f, g_T) \leq \gamma_{P_S}(f, g_S) + (\lambda + \lambda_H) W_1(P_S, P_T) + \gamma^\star \tag{26}$$

where $\gamma^\star = \min_{f \in \mathbb{H}} \gamma_{P_S}(f, g_S) + \gamma_{P_T}(f, g_T)$.

For succinctness and clarity of the following analysis in this work, as defined in Definition A.2, we express equation 20 as:

$$\gamma_T(f) \leq \gamma_S(f) + (\lambda + \lambda_H) W_1(P_S, P_T) + \gamma^\star \tag{27}$$

where $\gamma^\star = \min\limits_{f \in \mathbb{H}} \gamma_S(f) + \gamma_T(f)$ □

Lemma A.5 (Decomposition of the Classification Error)**.** *For any hypothesis $f \in \mathbb{H}$,*

$$\begin{aligned} \gamma_S(f) &= \sum_{k=1}^{K} w_S^k \gamma_S^k(f) \\ \gamma_T(f) &= \sum_{k=1}^{K} w_T^k \gamma_T^k(f) \end{aligned} \tag{28}$$

Proof. As stated in Section 3, we assume that both X_S and X_T are mixtures of K sub-domains. In other words, we have $P_S = \sum_{k=1}^{K} w_S^k P_S^k$ and $P_T = \sum_{k=1}^{K} w_T^k P_T^k$ where we use P_S^k and P_T^k to represent the distribution of the k-th subdomain of the source domain and that of the target domain respectively and w_S^k and w_T^k correspond to the weights of each sub-domain in the respective domains.

We can write out $\gamma_S(f) = \gamma_{P_S}(f, g_S)$ as sub-domain specific component.

$$\begin{aligned} \gamma_S(f) &= \gamma_{P_S}(f, g_S) \\ &= \mathbb{E}_{x \sim \mathcal{P}_S} \left[|f(x) - g_S(x)| \right] \\ &= \int P_S(x) |f(x) - g_S(x)| dx \\ &= \int \sum_{k=1}^{K} w_S^k P_S^k |f(x) - g_S(x)| dx \\ &= \sum_{k=1}^{K} w_S^k \int P_S^k |f(x) - g_S(x)| dx \\ &= \sum_{k=1}^{K} w_S^k \mathbb{E}_{x \sim P_S^k} \left[|f(x) - g_S(x)| \right] \\ &= \sum_{k=1}^{K} w_S^k \gamma_{P_S^k}(f, g_S) \\ &\overset{\text{Def. A.3}}{=} \sum_{k=1}^{K} w_S^k \gamma_S^k(f) \end{aligned} \tag{29}$$

□

With similar proof, we have:

$$\gamma_T(f) = \gamma_{P_T}(f, g_T) = \sum_{k=1}^{K} w_T^k \gamma_{P_T^k}(f, g_T) = \sum_{k=1}^{K} w_T^k \gamma_T^k(f) \tag{30}$$

Theorem A.6 (Sub-domain-based Generalization Bound)**.**

$$\gamma_T(f) \leq \sum_{k=1}^{K} w_T^k \gamma_S^k(f) + \sum_{k=1}^{K} w_T^k W_1(P_S^k, P_T^k) + \sum_{k=1}^{K} w_T^k (\gamma^k)^\star \tag{31}$$

Proof.

$$\begin{aligned} \gamma_T(f) &\overset{\text{Lemma A.5}}{=} \sum_{k=1}^{K} w_T^k \gamma_T^k(f) \\ &\overset{\text{Proposition 4.3}}{\leq} \sum_{k=1}^{K} w_T^k \{\gamma_S^k(f, g_S) + W_1(P_S^k, P_T^k) + (\gamma^k)^\star\} \end{aligned} \tag{32}$$

□

We next show that, under reasonable assumptions, the weighted sum of distances between corresponding sub-domains of the source and target domains is at most as large as the distance between the marginal distribution of the source domain and that of the target domain.

First we define a Wasserstein-like distance between Gaussian Mixture Models in Definition A.7, which uses Wasserstein-1 distance that extends the Proposition 4 of Delon & Desolneux (2020).

Definition A.7. (Wasserstein-like distance between Gaussian Mixture Models) Assume that both X_S and X_T are mixtures of K sub-domains. In other words, we have $P_S = \sum_{k=1}^{K} w_S^k P_S^k$ and $P_T = \sum_{k=1}^{K} w_T^k P_T^k$ where we use P_S^k and P_T^k to represent the distribution of the k-th subdomain of the source domain and that of the target domain respectively and w_S^k and w_T^k correspond to the weights of each sub-domain in the respective domains. We define:

$$MW_1(P_S, P_T) = \min_{w \in \Pi(\mathbf{w_S}, \mathbf{w_T})} \sum_{k=1}^{K} \sum_{k'=1}^{K} w_{k,k'} W_1(P_S^k, P_T^{k'}) \tag{33}$$

where $\mathbf{w_S} \doteq [w_S^1, \ldots, w_S^K]$ and $\mathbf{w_T} \doteq [w_T^1, \ldots, w_T^K]$ belong to Δ^K (the $K-1$ probability simplex). $\Pi(w_S, w_T)$ represents the simplex $\Delta^{K \times K}$ with marginals $\mathbf{w_S}$ and $\mathbf{w_T}$.

To demonstrate that MW_1 is less or equal to the sum of W_1 and a term that is dependent on the trace of the covariance matrices of two Gaussian mixtures (extend the proof of Delon & Desolneux (2020)), we start with a lemma. This lemma makes more explicit the distance MW_1 between a Gaussian mixture and a mixture of Dirac distributions.

Lemma A.8 (Extension to Lemma 4.1 of Delon & Desolneux (2020))**.** *Let $\mu_0 = \sum_{k=1}^{K_0} \pi_0^k \mu_0^k$ with $\mu_0^k = \mathcal{N}(m_0^k, \Sigma_0^k)$ and $\mu_1 = \sum_{k=1}^{K_1} \pi_1^k \delta_{m_1^k}$. Let $\tilde{\mu_0} = \sum_{k=1}^{K_0} \pi_0^k \delta_{m_0^k}$ ($\tilde{\mu_0}$ only retains the means of μ_0). Then,*

$$MW_1(\mu_0, \mu_1) \leq W_1(\tilde{\mu_0}, \mu_1) + \sum_{k=1}^{K_0} \pi_0^k \sqrt{tr\ (\Sigma_0^k)}$$

where $\pi_\mathbf{0} \doteq [\pi_0^1, \ldots, \pi_0^k]$ and $\pi_\mathbf{1} \doteq [\pi_1^1, \ldots, \pi_1^k]$ belong to Δ^K (the $K-1$ probability simplex)

Proof.

$$\begin{aligned} MW_1(\mu_0,\mu_1) &= \inf_{w\in\Pi(\pi_0,\pi_1)} \sum_{k,l} w_{k,l} W_1(\mu_0^k, \delta_{m_1^l}) \\ &\le \inf_{w\in\Pi(\pi_0,\pi_1)} \sum_{k,l} w_{k,l} W_2(\mu_0^k, \delta_{m_1^l}) \\ &= \inf_{w\in\Pi(\pi_0,\pi_1)} \sum_{k,l} w_{k,l} \left[\sqrt{||m_1^l - m_0^k||^2 + \text{tr }(\Sigma_0^k)}\right] \\ &\le \inf_{w\in\Pi(\pi_0,\pi_1)} \sum_{k,l} w_{k,l} ||m_1^l - m_0^k|| + \sum_k \pi_0^k \sqrt{\text{tr }(\Sigma_0^k)} \\ &\le W_1(\tilde{\mu_0},\mu_1) + \sum_{k=1}^{K_0} \pi_0^k \sqrt{\text{tr }(\Sigma_0^k)} \end{aligned} \tag{34}$$

□

Remark A.9. We use μ_0, μ_1, and $\tilde{\mu_0}$ to represent a general scenario for measuring the distance between a Gaussian mixture and a mixture of Diract distributions. In the following proofs, we will utilize the defined notation. For instance, μ_0 can be denoted as P_S, while $\tilde{\mu_0}$ corresponds to $\tilde{P_S}$.

Theorem A.10 (Extension to Proposition 6 in (Delon & Desolneux, 2020))**.** *Let P_S and P_T be two Gaussian mixtures with $P_S = \sum_{k=1}^K w_S^k P_S^k$ and $P_T = \sum_{k=1}^K w_T^k P_T^k$. For all k, P_S^k / P_T^k are Gaussian distributions with mean m_S^k / m_T^k and covariance Σ_S^k / Σ_T^k. If for $\forall$ k, k′, we assume there exists a small constant $\epsilon > 0$, such that $\max_k(trace(\Sigma_S^k)) \le \epsilon$ and $\max_{k'}(trace(\Sigma_T^{k'})) \le \epsilon$. then:*

$$MW_1(P_S, P_T) \le W_1(P_S, P_T) + 4\sqrt{\epsilon} \tag{35}$$

Proof. Here, we follow the same structure of the proof for Wassertein-2 in Delon & Desolneux (2020). Let $(P_S^n)_n$ and $(P_T^n)_n$ be two sequences of mixtures of Dirac masses respectively converging to P_S and P_T in $\mathcal{P}_1(\mathbb{R}^d)$. Since MW_1 is a distance,

$$\begin{aligned} MW_1(P_S,P_T) &\le MW_1(P_S^n,P_T^n) + MW_1(P_S,P_S^n) + MW_1(P_T,P_T^n) \\ &= W_1(P_S^n,P_T^n) + MW_1(P_S,P_S^n) + MW_1(P_T,P_T^n) \end{aligned}$$

We can study the limits of these three terms when n $\to +\infty$

First, observe that $MW_1(P_S^n,P_T^n) = W_1(P_S^n,P_T^n) \underset{n\to+\infty}{\to} W_1(P_S,P_T)$ since W_1 is continuous on $\mathcal{P}_1(\mathbb{R}^d)$.

Second, based on Lemma A.8, we have that

$$MW_1(P_S,P_S^n) \le W_1(\tilde{P_S},P_S^n) + \sum_{k=1}^K w_S^k \sqrt{\text{ tr}(\Sigma_S^k)} \underset{n\to+\infty}{\to} W_1(\tilde{P_S},P_S) + \sum_{k=1}^K w_S^k \sqrt{\text{ tr}(\Sigma_S^k)}$$

We observe that $x \mapsto \sqrt{x}$ is a concave function, thus by Jensen's inequality, we have that

$$\sum_{k=1}^K w_S^k \sqrt{\text{ tr}(\Sigma_S^k)} \le \sqrt{\sum_{k=1}^K w_S^k \text{ tr}(\Sigma_S^k)}$$

Also By Jensen's inequality, we have that,

$$W_1(\tilde{P_S},P_S) \le W_2(\tilde{P_S},P_S).$$

And from Proposition 6 in (Delon & Desolneux, 2020), we have

$$W_2(\tilde{P_S},P_S) \le \sqrt{\sum_{k=1}^K w_S^k \text{ tr}(\Sigma_S^k)}$$

Similarly for $MW_1(P_T, P_T^n)$ the same argument holds. Therefore we have,

$$\lim_{n\to\infty} MW_1(P_S, P_S^n) \leq 2\sqrt{\sum_{k=1}^{K} w_S^k \ \mathrm{tr}(\Sigma_S^k)}$$

And

$$\lim_{n\to\infty} MW_1(P_T, P_T^n) \leq 2\sqrt{\sum_{k=1}^{K} w_T^k \ \mathrm{tr}(\Sigma_T^k)}$$

We can conclude that:

$$\begin{aligned} MW_1(P_S, P_T) &\leq \lim\inf_{n\to\infty}(W_1(P_S^n, P_T^n) + MW_1(P_S, P_S^n) + MW_1(P_T, P_T^n)) \\ &\leq W_1(P_S, P_T) + 2\sqrt{\sum_{k=1}^{K} w_S^k \ \mathrm{tr}(\Sigma_S^k)} + 2\sqrt{\sum_{k=1}^{K} w_T^k \ \mathrm{tr}(\Sigma_T^k)} \\ &\leq W_1(P_S, P_T) + 4\sqrt{\epsilon} \end{aligned}$$

This concludes the proof. □

Theorem A.11 (Benefits of Sub-domain Alignment)**.** *Under the following assumptions:*

A1. *For all k, P_S^k / P_T^k are Gaussian distributions with mean m_S^k / m_T^k and covariance Σ_S^k / Σ_T^k.*
A2. *Distance between the paired source-target sub-domain is less or equal to distance between the non-paired source-target sub-domain, i.e., $W_1(P_S^k, P_T^k) \leq W_1(P_S^k, P_T^{k'})$ for $k \neq k'$.*
A3. *There exists a small constant $\epsilon > 0$, such that $\max_{1\leq k\leq K}(tr(\Sigma_S^k)) \leq \epsilon$ and $\max_{1\leq k\leq K}(tr(\Sigma_T^{k'})) \leq \epsilon$. Then the following inequality holds:*

$$\textstyle\sum_{k=1}^{K} w_T^k W_1(P_S^k, P_T^k) \leq W_1(P_S, P_T) + \delta_c, \tag{36}$$

where δ_c is $4\sqrt{\epsilon}$. In particular, when $w_S^k = w_T^k$ for all k,

$$\textstyle\sum_{k=1}^{K} w_S^k W_1(P_S^k, P_T^k) \leq W_1(P_S, P_T) + \delta_c. \tag{37}$$

Proof. With $w \in \Pi(\mathbf{w_S}, \mathbf{w_T})$, we can write out w_T^k as $\sum_{k'=1}^{K} w_{k,k'}$, then based on assumption A.2, we have:

$$\begin{aligned} \sum_{k=1}^{K} w_T^k W_1(P_S^k, P_T^k) &= \sum_{k=1}^{K}\sum_{k'=1}^{K} w_{k,k'} W_1(P_S^k, P_T^k) \\ &\leq \sum_{k=1}^{K}\sum_{k'=1}^{K} w_{k,k'} W_1(P_S^k, P_T^{k'}). \end{aligned}$$

Thus we have,

$$\begin{aligned} \sum_{k=1}^{K} w_T^k W_1(P_S^k, P_T^k) &\leq \min_{w\in\Pi(\mathbf{w_S},\mathbf{w_T})} \sum_{k=1}^{K}\sum_{k'=1}^{K} w_{k,k'} W_1(P_S^k, P_T^{k'}) \\ &= MW_1(P_S, P_T). \end{aligned} \tag{38}$$

Also we prove in Theorem A.10 that:

$$MW_1(P_S, P_T) \leq W_1(P_S, P_T) + 4\sqrt{\epsilon}.$$

Then we conclude our proof and show that:

$$\sum_{k=1}^{K} w_T^k W_1(P_S^k, P_T^k) \leq MW_1(P_S, P_T) \leq W_1(P_S, P_T) + 4\sqrt{\epsilon} = W_1(P_S, P_T) + \delta_c. \tag{39}$$

□

Theorem A.12. *Let $\mathbb{H} \doteq \{f | f : \mathcal{X} \to [0,1]\}$ denote a hypothesis space. Under the Assumptions in Theorem 4.7(or Theorem A.11), if the following assumption hold for all $f \in \mathbb{H}$:*

$$\textstyle\sum_{k=1}^K w_T^k \gamma_S^k(f) \le \sum_{k=1}^K w_S^k \gamma_S^k(f), \tag{40}$$

then we have

$$\textstyle\sum_{k=1}^K w_T^k (\gamma^k)^\star \le \gamma^\star.$$

Further, let

$$\epsilon_c(f) \doteq \textstyle\sum_{k=1}^K w_T^k \gamma_S^k(f) + \sum_{k=1}^K w_T^k W_1(P_S^k, P_T^k) + \sum_{k=1}^K w_T^k (\gamma^k)^\star$$

denote the sub-domain-based generalization bound and let

$$\epsilon_g(f) \doteq \gamma_S(f) + W_1(P_S, P_T) + \gamma^\star$$

denote the general generalization bound without any sub-domain information. We have,

$$\epsilon_c(f) \le \epsilon_g(f) + \delta_c.$$

where δ_c is $4\sqrt{\epsilon}$.

Proof. We will proove that $\sum_{k=1}^K w_T^k (\gamma^k)^\star \le \gamma^\star$, where $\gamma^\star = \min\limits_{f\in\mathbb{H}} \gamma_S(f) + \gamma_T(f)$, $\mathbb{H}$, and $(\gamma^k)^\star = \min_{f\in\mathbb{H}} \gamma_S^k(f) + \gamma_T^k(f)$

We have:

$$\begin{aligned}
\gamma^\star &= \min_{f\in\mathbb{H}} \left(\gamma_S(f) + \gamma_T(f)\right) \\
&= \min_{f\in\mathbb{H}} \left(\sum_{k=1}^K w_S^k \gamma_S^k(f) + \sum_{k=1}^K w_T^k \gamma_T^k(f)\right) \\
&= \min_{f\in\mathbb{H}} \left(\sum_{k=1}^K w_S^k \gamma_{P_S^k}(f, g_S) + \sum_{k=1}^K w_T^k \gamma_{P_T^k}(f, g_T)\right) \\
&= \min_{f\in\mathbb{H}} \left(\sum_{k=1}^K w_T^k \gamma_{P_S^k}(f, g_S) + \sum_{k=1}^K w_T^k \gamma_{P_T^k}(f, g_T) + \sum_{k=1}^K w_S^k \gamma_{P_S^k}(f, g_S) - \sum_{k=1}^K w_T^k \gamma_{P_S^k}(f, g_S)\right) \\
&= \min_{f\in\mathbb{H}} \left(\sum_{k=1}^K w_T^k (\gamma_{P_S^k}(f, g_S) + \gamma_{P_T^k}(f, g_T)) + \sum_{k=1}^K (w_S^k - w_T^k) \gamma_{P_S^k}(f, g_S)\right) \\
&\ge \min_{f\in\mathbb{H}} \left(\sum_{k=1}^K w_T^k (\gamma_{P_S^k}(f, g_S) + \gamma_{P_T^k}(f, g_T))\right) \\
&\ge \sum_{k=1}^K \min_{f\in\mathbb{H}} \left(\gamma_S^k(f) + \gamma_T^k(f)\right) \\
&= \sum_{k=1}^K w_T^k (\gamma^k)^\star
\end{aligned} \tag{41}$$

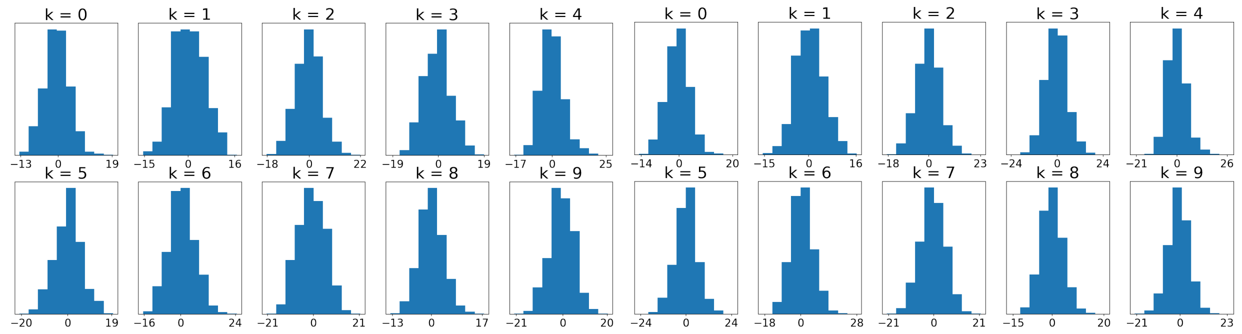
where the first inequality (the 6th line in the equation) is based on the assumption that $\sum_{k=1}^K w_T^k \gamma_S^k(f) \le \sum_{k=1}^K w_S^k \gamma_S^k(f)$. The second inequality (the 7th line in the equation) is based on $\min\{f(x) + g(x)\} \ge \min\{f(x)\} + \min\{g(x)\}$

□

B Empirical Evidence for Assumptions in Theorem 4.7

B.1 Gaussian Distribution

To affirm the Gaussian distribution assumption for latent representations of each sub-domain, we include Figure 4. These plots, derived from the MNIST and MNIST-M datasets for all values of k, reinforce our theorem's practical realization.



(a) MNIST (b) MNIST-M

Figure 4: Distributions of the learned representations

.

B.2 Distance Assumption

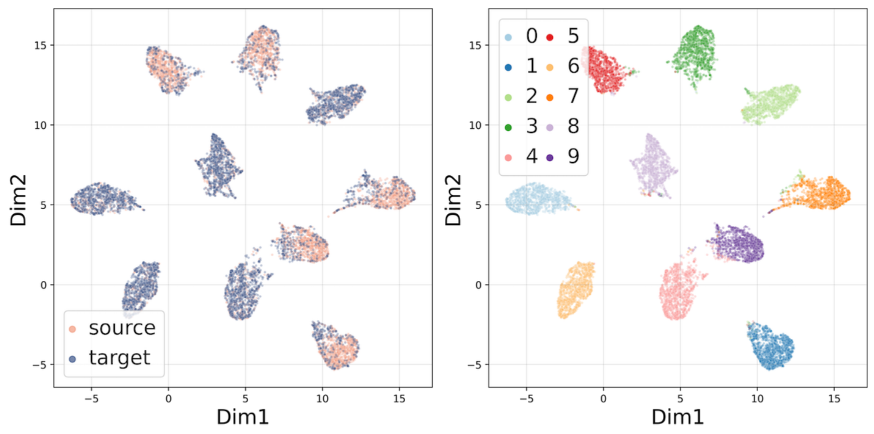
We further present empirical evidence to support the distance relations between paired and non-paired source-target sub-domains on the MNIST to MNIST-M task. Our methodology specifically aims to minimize the distance between paired source-target sub-domains during the training process. Table 5 illustrates that empirically the distance between the paired source-target sub-domain is consistently less than the average distance between non-paired source-target sub-domains.

Table 5: Distance relations between paired and non-paired source-target sub-domains on the MNIST to MNIST-M task. We use Wasserstein-1 (W1) distance as distance metric.

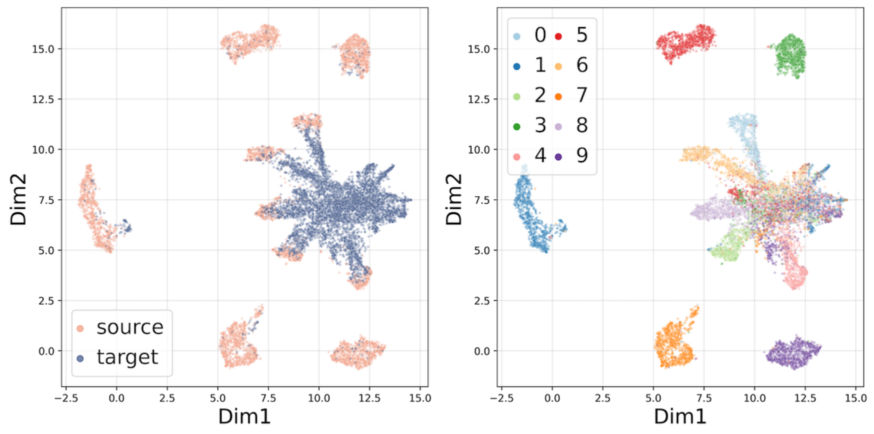
Sub-domain	Paired source-target sub-domain	Non-paired source-target sub-domains
0	5.9	79.4
1	9.5	85.5
2	7.5	85.4
3	11.4	82.6
4	7.7	80.5
5	11.4	84.8
6	7.0	89.5
7	11.9	89.6
8	8.1	72.8
9	9.6	81.2

C Analysis of Feature Space

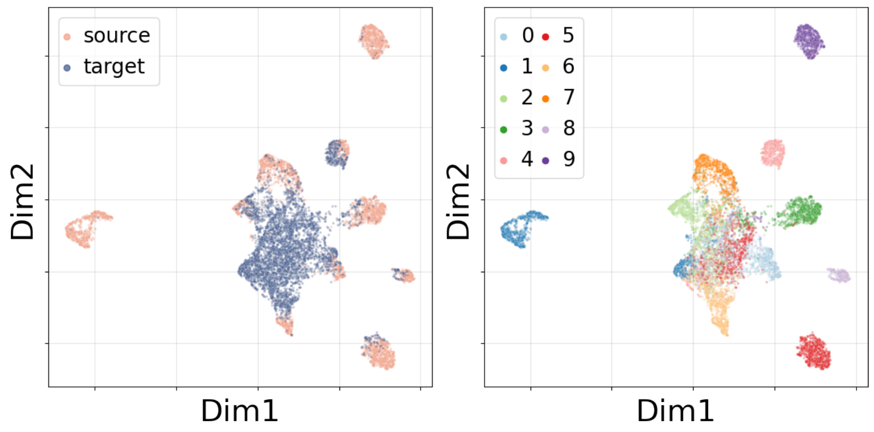
We visualize the feature spaces learned by DANN (Ganin et al., 2016), CAT (Deng et al., 2019) and our method, DARSA, using UMAP (Sainburg et al., 2021). As shown in Figure 5, features learned with DARSA form stronger clusters when the labels are the same, and clusters with different labels are more separated from one another. In contrast, both DANN and CAT fail to learn a good source-target domain alignment in the feature space (shown in Figure 5 (b)(c)) in the presence of label distribution shifts. This confirms that our method, DARSA, can learn a label-conditional feature space that is discriminative and domain-invariant, which improves performance in target domain prediction.



(a) DARSA



(b) DANN



(c) CAT

Figure 5: For MNIST to MNIST-M UDA task with label shifting (a) feature space learned by our method, DARSA. (b) feature space learned by DANN. (c) feature space learned by CAT. Left panel: colored by source/target; Right panel: colored by true label (digits). The features are projected to 2-D using UMAP.

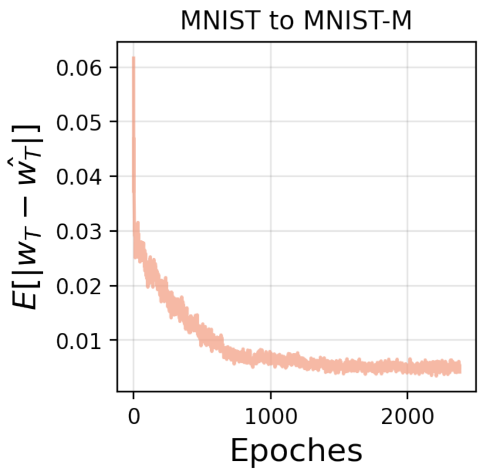


Figure 6: Evolution of the absolute difference between the estimated target weight $\hat{w_T}$ and the actual ground truth w_T across epochs.

D Algorithm

Our framework is outlined in pseudo-code in Algorithm 1.

Algorithm 1 Domain Adaptation via Rebalanced Sub-domain Alignment(DARSA)

```
Input: Source data $X_S$; Source label $y_S$, Target data $X_T$; coefficient $\lambda_Y, \lambda_D, \lambda_c, \lambda_a$; learning rate $\alpha$;
Pretrain feature extractor and classifier with $X_S$ and $y_S$, initialize $\theta_E^S$, $\theta_E^T$, and $\theta_Y$ with pretrained weights.
Initialize $w_T^k$ and $w_T^k$ with 1/K for k = 1,2 ..., K
repeat
    Sample minibatch from $X_S$ and $X_T$
    $\theta_Y \leftarrow \theta_Y - \alpha \nabla_{\theta_Y}(\lambda_Y \mathcal{L}_Y + \lambda_D \mathcal{L}_D + \lambda_c \mathcal{L}_{intra} + \lambda_a \mathcal{L}_{inter})$
    $\theta_E^S \leftarrow \theta_E^S - \alpha \nabla_{\theta_E^S}(\lambda_Y \mathcal{L}_Y + \lambda_D \mathcal{L}_D + \lambda_c \mathcal{L}_{intra} + \lambda_a \mathcal{L}_{inter})$
    $\theta_E^T \leftarrow \theta_E^T - \alpha \nabla_{\theta_E^T}(\lambda_D \mathcal{L}_D + \lambda_c \mathcal{L}_{intra} + \lambda_a \mathcal{L}_{inter})$
until $\theta_E^S$, $\theta_E^T$, and $\theta_Y$ converge
```

E Analysis of Weights

In order to assess the accuracy of our target weight estimation in the DARSA algorithm, we conducted an additional analysis. This analysis focused on the MNIST to MNIST-M experiment under conditions of weak imbalance. Specifically, we compared the difference between the actual ground truth target weight (denoted as w_T) and our estimated target weight (denoted as $\hat{w_T}$) across epochs.

As illustrated in the figure 6, our estimation aligns closely with the ground truth towards the end of the training process. This proximity indicates the effectiveness of our weight estimation approach within the DARSA algorithm.

F Details of Experimental Setup: Digits Datasets with label shifting

In our Digits datasets experiments, we evaluate our performance across four datasets: MNIST (LeCun et al., 1998), MNIST-M (Ganin et al., 2016), USPS, and SVHN, all modified to induce label distribution shifts. Here, the parameter α denotes the class imbalance rate, representing a ratio such as 1:α and α:1 for the odd:even distribution in the source and target datasets, respectively. Weak and strong imbalance correspond to $\alpha = 3$ and $\alpha = 8$. A small subset of the target domain is used for hyperparameter search, serving as an upper performance bound for UDA methods. Results shown in Table 2 demonstrates DARSA's competitiveness in handling label shifting. It is worth noting that many state-of-the-art comparisons are not specifically tailored for shifted label distribution scenarios, potentially affecting their performance. To ensure a fair comparison, we use the Ax platform (Bakshy et al.; Letham et al., 2019) for automated hyperparameter tuning to maximize domain-shifting performance (More details in F.8). Additionally, DARSA performs well with varying imbalance rates (Table 6) and competes favorably in scenarios without label distribution shifts (Table 7).

F.1 Details of the Digits Datasets with Label Distribution Shifts

F.1.1 Weak Imbalance: $\alpha = 3$

MNIST → MNIST-M: For source dataset, we randomly sample 36000 images from MNIST training set with odd digits three times the even digits. For target dataset, we randomly sample 6000 images from MNIST-M constructed from MNIST testing set, with even digits three times the odd digits. To create MNIST-M dataset, we follow the procedure outlined in Ganin et al. (2016) to blend digits from the MNIST over patches randomly extracted from color photos in the BSDS500 dataset (Arbelaez et al., 2010).

MNIST-M → MNIST: For source dataset, we randomly sample 36000 images from MNIST-M constructed from MNIST training set, with even digits three times the odd digits. For target dataset, we randomly sample 5800 images from MNIST testing set, with odd digits three times the even digits.

USPS → MNIST: For source dataset, we randomly sample 3600 images from USPS training set, with even digits three times the odd digits. For target dataset, we randomly sample 5800 images from MNIST testing set, with odd digits three times the even digits.

SVHN → MNIST: For source dataset, we randomly sample 30000 images from SVHN training set, with even digits three times the odd digits. For target dataset, we randomly sample 5800 images from MNIST testing set, with odd digits three times the even digits.

F.1.2 Strong Imbalance: $\alpha = 8$

MNIST → MNIST-M: For source dataset, we randomly sample 27000 images from MNIST training set with odd digits eight times the even digits. For target dataset, we randomly sample 4500 images from MNIST-M constructed from MNIST testing set, with even digits eight times the odd digits. To create MNIST-M dataset, we follow the procedure outlined in Ganin et al. (2016) to blend digits from the MNIST over patches randomly extracted from color photos in the BSDS500 dataset (Arbelaez et al., 2010).

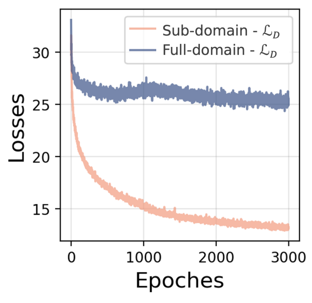
MNIST-M → MNIST: For source dataset, we randomly sample 13500 images from MNIST-M constructed from MNIST training set, with even digits eight times the odd digits. For target dataset, we randomly sample 13500 images from MNIST testing set, with odd digits eight times the even digits.

USPS → MNIST: For source dataset, we randomly sample 2700 images from USPS training set, with even digits eight times the odd digits. For target dataset, we randomly sample 13500 images from MNIST testing set, with odd digits eight times the even digits.

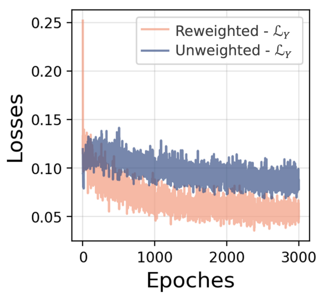
SVHN → MNIST: For source dataset, we randomly sample 27000 images from SVHN training set, with even digits eight times the odd digits. For target dataset, we randomly sample 18000 images from MNIST testing set, with odd digits eight times the even digits.

F.2 Additional Empirical Analysis of our Proposed Generalization Bound

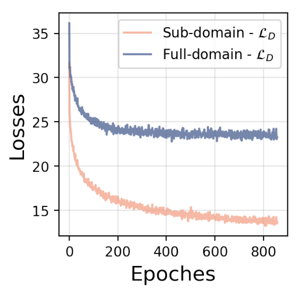
Here we empirically evaluate the proposed generalization bound with weak imbalance ($\alpha = 3$) and strong imbalance ($\alpha = 8$). As shown in Figure 7 and Figure 8, our empirical results demonstrate that the sub-domain-based generalization bound in Theorem 4.5 is empirically much stronger than the non-sub-domain-based bound in Theorem 4.1.



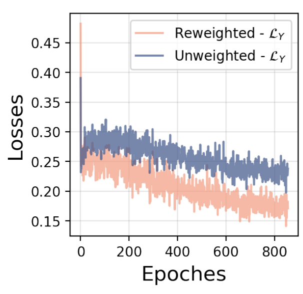
(a) Domain Discrepancy



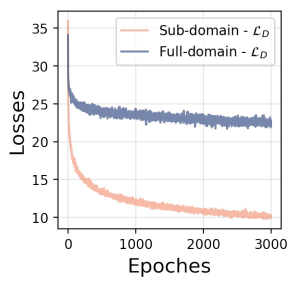
(b) Source Clf Loss



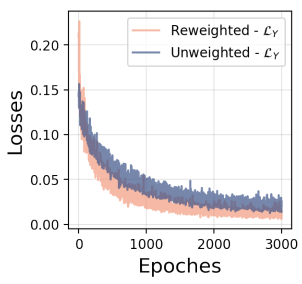
(c) Domain Discrepancy



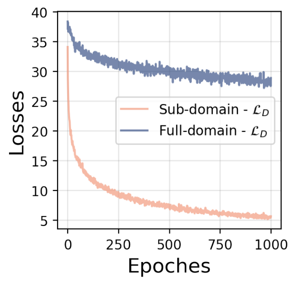
(d) Source Clf Loss



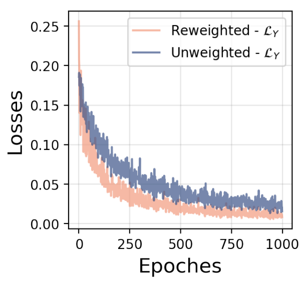
(e) Domain Discrepancy



(f) Source Clf Loss



(g) Domain Discrepancy



(h) Source Clf Loss

Figure 7: For experiments with weak imbalance ($\alpha = 3$), we compare the domain discrepancy term ($\mathcal{L}_D$) and source classification term ($\mathcal{L}_Y$) in our proposed bound to that in Theorem 4.1, respectively. Empirical results for each experiment are demonstrated in the subfigures: (a)(b) MNIST to MNIST-M (c)(d) MNIST-M to MNIST, (e)(f) USPS to MNIST, (g)(h) SVHN to MNIST.

F.3 Evaluate DARSA On Varying Imbalance Rates

We have conducted a study on the USPS to MNIST adaptation to explore the effects of varying imbalance rates. As can be seen from Table 6, the performance of our algorithm is stable across a wide range of imbalance rates.

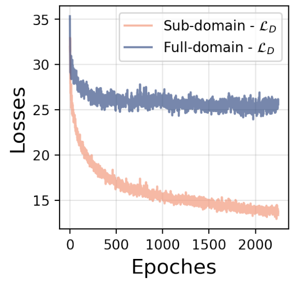
Source: USPS - even : odd digit = 1:α
Target: MNIST - odd : even digit = α:1

Table 6: Summary of UDA results on USPS to MNIST adaptation with varying imbalance rates

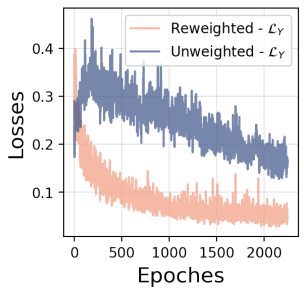
α	1	2	3	4	5	6	7	8	9	10
Accuracy (%)	97.1	92.8	92.6	92.1	85.8	93.3	88.1	87.9	77.4	71.3

F.4 The rationale for competing algorithms choices

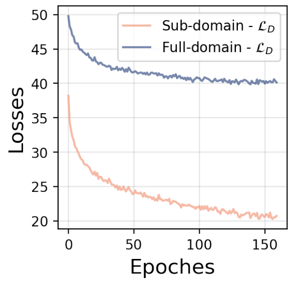
For benchmark methods, we have chosen methods that not only have theoretical underpinnings, but also exhibit impressive performance in transfer learning on digit benchmarks as per the listings on the public competition Papers with Code (2023). Furthermore, our selection includes methods that contemplate sub-domain alignment to facilitate a direct comparison with our approach. It's important to note that our selection of benchmark methods may include some older models. However, these models were chosen not just for their performance but for their theoretical relevance and their ability to provide valuable insights into the effectiveness of our proposed method. Thus, while our experiments primarily serve to validate our theory, the benchmarks also offer meaningful evaluation of our theory's practical impact.



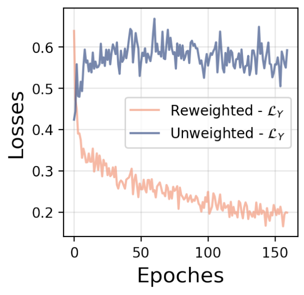
(a) Domain Discrepancy



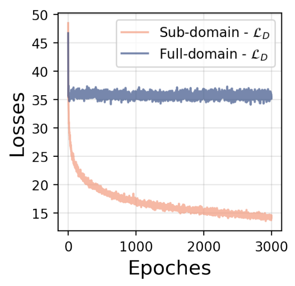
(b) Source Clf Loss



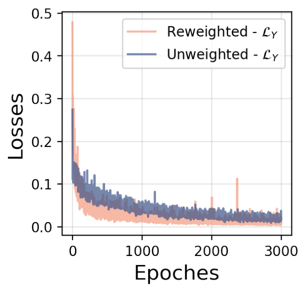
(c) Domain Discrepancy



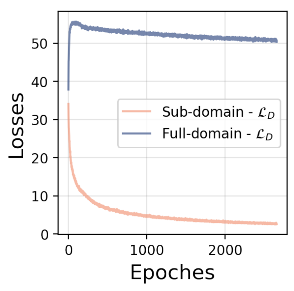
(d) Source Clf Loss



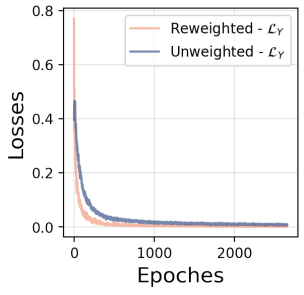
(e) Domain Discrepancy



(f) Source Clf Loss



(g) Domain Discrepancy



(h) Source Clf Loss

Figure 8: For tasks with strong imbalance ($\alpha = 8$), we compare the domain discrepancy term ($\mathcal{L}_D$) and source classification term ($\mathcal{L}_Y$) in our proposed bound to that in Theorem 4.1, respectively. Empirical results for each experiment are demonstrated in the subfigures: (a)(b) MNIST to MNIST-M (c)(d) MNIST-M to MNIST, (e)(f) USPS to MNIST, (g)(h) SVHN to MNIST.

F.5 Model Structures

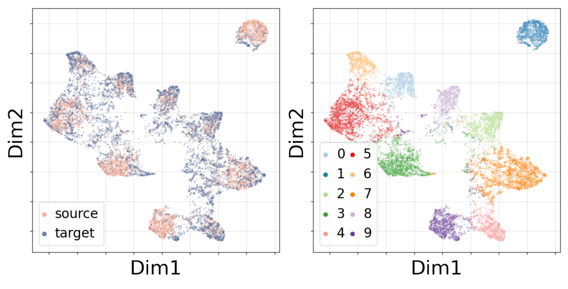
For feature extractor, we employ a network structure similar to LeNet-5 ((LeCun et al., 1998)), but with minor modifications: the first convolutional layer produces 10 feature maps, the second convolutional layer produces 20 feature maps, and we use ReLU as an activation function for the hidden layer. Our feature space has 128 dimensions. For benchmarks, we utilize the network structures provided in the benchmark source code. In cases where experiments are not included in the source code, we use the same network architecture as our model to ensure fair comparisons. For classifier, we use a network structure with three fully connected layers with ReLU activation and a dropout layer with a rate of 0.5. See the included code link for further details of each experiment.

F.6 Ablation study

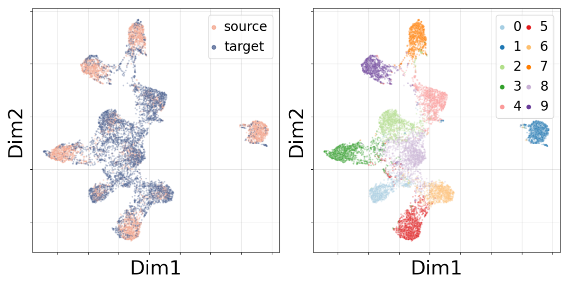
To gain insight into the individual impact of each component within our objective function (Section 5), we performed an ablation study under conditions of weak imbalance. The outcomes of this investigation are detailed in Table 4. The ablation analysis confirms that each component in our objective function contributes to its overall performance. Therefore, we recommend the use of all components for optimal results. In addition, we have included ablation study with feature space visualization in Figure 9. As can be seen in Figure 9, the learned representation of DARSA has improved separation when using all the components, supporting the effectiveness of the proposed objective function.

F.7 Evaluate DARSA On Benchmarks Without Additional Label distribution shifts

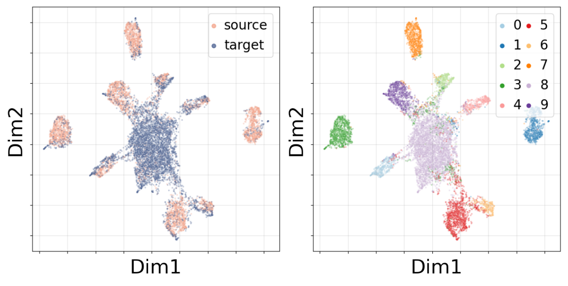
We conduct additional experiments to evaluate how DARSA performs in scenarios without label distribution shifts. Results in Table 7 show DARSA's performance is comparable with the most competitive methods.



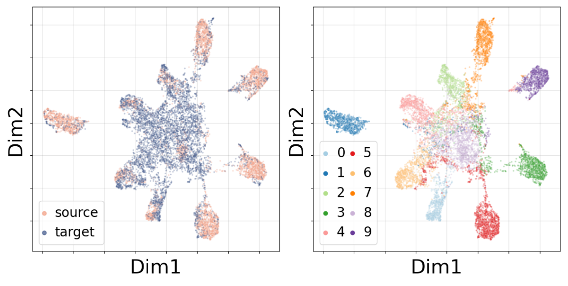
(a) No $\mathcal{L}_Y$



(b) No $\mathcal{L}_D$



(c) No $\mathcal{L}_{intra}$



(d) No $\mathcal{L}_{inter}$

Figure 9: For MNIST to MNIST-M UDA task with shifted label distribution with representations learnt by ablation study. Colored by 1) source/target; 2) predicted label (digit). The features are projected to 2-D using UMAP.

Table 7: Summary of UDA results on the Digits datasets without shifted label distribution, measured in terms of prediction accuracy (%) on the target domain.

	DANN	WDGRL	DSN	ADDA	CAT	CDAN	pixelDA	DRANet	**DARSA**
U $\rightarrow$ M	74.5	84.8	91.0	90.1	80.9	98.0	87.6	97.8	97.4
S $\rightarrow$ M	73.9	59.3	82.7	76.0	98.1	89.2	71.6	59.7	98.6

F.8 Model hyperparameters

We use Adaptive Experimentation (Ax) platform (Bakshy et al.; Letham et al., 2019), an automatic tuning approaches to select hyperparameters to maximize the performance of our method. We use Bayesian optimization supported by Ax with 20 iterations to decide the hyperparameter choice. We note that most of the SOTA comparisons are not specifically designed for shifted label distribution scenarios, and this setting caused issues in several competing methods. We used Ax to maximize their performance in label shifting scenarios. Details on the model hyperparameters used for the Digits datasets with shifted label distribution are provided in Table 8 and Table 9 (If not explicitly stated, we resort to the default hyperparameters from the respective implementations of the benchmark.).

Table 8: Model hyperparameters used for Digits datasets with weak imbalance $\alpha = 3$

	MNIST to MNIST-M	MNIST-M to MNIST	USPS to MNIST	SVHN to MNIST
DARSA	batch size = 512, $\alpha = 0.01$, $\lambda_Y = 0.4$, $\lambda_D = 0.35$, $\lambda_c = 1$, $\lambda_a = 0.9$, m = 30 SGD, momentum = 0.5	batch size = 512, $\alpha = 0.01$, $\lambda_Y = 1$, $\lambda_D = 0.5$, $\lambda_c = 1$, $\lambda_a = 1$, m = 30 SGD, momentum = 0.4	batch size = 256, , $\alpha = 0.01$, $\lambda_Y = 1$, $\lambda_D = 0.5$, $\lambda_c = 1$, $\lambda_a = 1$, m = 30 SGD, momentum = 0.4	batch size = 256, $\alpha = 0.05$, $\lambda_Y = 0.95$, $\lambda_D = 0.11$, $\lambda_c = 0.3$, $\lambda_a = 0.11$, m = 50 SGD, momentum = 0.4
DANN	batch size = 32 Adam, learning rate = 1e-4	batch size = 32 Adam, learning rate = 1e-5	batch size = 32 Adam, learning rate = 1e-4	batch size = 64 Adam, learning rate = 1e-4
WDGRL	batch size = 32 Adam, learning rate = 1e-5, $\gamma = 10$, critic training step: 1, feature extractor and discriminator training step: 3	batch size = 64 Adam, learning rate = 1e-4, $\gamma = 10$, critic training step: 5, feature extractor and discriminator training step: 10	batch size = 32 Adam, learning rate = 1e-4, $\gamma = 10$, critic training step: 1, feature extractor and discriminator training step: 2	batch size = 32 Adam, learning rate = 1e-5, $\gamma = 10$, critic training step: 1, feature extractor and discriminator training step: 3
DSN	batch size = 32 SGD, momentum = 0.8, learning rate = 1e-2, $\alpha = 0.01$, $\beta = 0.075, \gamma = 0.25$	batch size = 32 SGD, momentum = 0.8, learning rate = 0.01, $\alpha = 0.01$, $\beta = 0.075, \gamma = 0.25$	batch size = 32 SGD, momentum = 0.8, learning rate = 0.01, $\alpha = 0.01$, $\beta = 0.075, \gamma = 0.4$	batch size = 512 SGD, momentum = 0.5, learning rate = 1e-5, $\alpha = 0.046$, $\beta = 0.61, \gamma = 0.92$
ADDA	batch size = 64 Adam, learning rate = 1e-3, critic training step: 1, target model training step: 10	batch size =64 Adam, learning rate = 1e-5, critic training step: 1, target model training step: 1	batch size = 64 Adam, learning rate = 1e-3, critic training step: 1, target model training step: 1	batch size = 64 Adam, learning rate = 1e-3, critic training step: 3, target model training step: 2
CAT	batch size = 512 SGD, learning rate = $\frac{0.01}{(1+10p)^{0.75}}$, momentum = 0.9, p = 0.9, m = 30	batch size = 128 SGD, learning rate = $\frac{0.01}{(1+10p)^{0.75}}$, momentum = 0.9, p = 0.9, m = 30	batch size = 128 SGD, learning rate = $\frac{0.01}{(1+10p)^{0.75}}$, momentum = 0.9, p = 0.9, m = 30	batch size = 256 SGD, learning rate = $\frac{0.01}{(1+10p)^{0.75}}$, momentum = 0.9, p = 0.9, m = 30
CDAN	batch size = 64 SGD, momentum = 0.9, learning rate =0.01	batch size = 64 SGD, momentum = 0.9, learning rate =0.01	batch size = 64 SGD, momentum = 0.9, learning rate =0.01	batch size = 64 SGD, momentum = 0.9, learning rate =0.1
pixelDA	batch size = 64 Adam learning rate =0.0002, dim of the noise input: 10	batch size = 64 Adam learning rate =0.0002, dim of the noise input: 10	batch size = 32 Adam, learning rate =0.0001, dim of the noise input: 20	batch size = 32 Adam, learning rate =0.0001, dim of the noise input: 20
DRANet	batch size = 32 Adam	batch size = 32 Adam	batch size = 32 Adam	batch size = 32 Adam
MCD	batch size = 128 learning rate = 0.01	batch size = 128 learning rate = 0.01	batch size = 128 learning rate = 0.01	batch size = 128 learning rate = 0.01
MDD	batch size = 128 learning rate = 0.01	batch size = 128 learning rate = 0.01	batch size = 128 learning rate = 0.01	batch size = 128 learning rate = 0.01

Table 9: Model hyperparameters used for Digits datasets with strong imbalance $\alpha = 8$

	MNIST to MNIST-M	MNIST-M to MNIST	USPS to MNIST	SVHN to MNIST
DARSA	batch size = 1024, $\alpha = 0.01$, $\lambda_Y = 0.8$, $\lambda_D = 0.4$, $\lambda_c = 0.9$, $\lambda_a = 0.9$, m = 30 SGD, momentum = 0.5	batch size = 1024, $\alpha = 0.01$, $\lambda_Y = 1$, $\lambda_D = 0.5$, $\lambda_c = 1$, $\lambda_a = 0.5$, m = 30 SGD, momentum = 0.4	batch size = 1024, , $\alpha = 0.01$, $\lambda_Y = 1$, $\lambda_D = 0.1$, $\lambda_c = 1$, $\lambda_a = 1$, m = 30 SGD, momentum = 0.4	batch size = 1024, $\alpha = 0.01$, $\lambda_Y = 0.95$, $\lambda_D = 0.11$, $\lambda_c = 1$, $\lambda_a = 1$, m = 50 SGD, momentum = 0.4
DANN	batch size = 64 Adam, learning rate = 1e-3	batch size = 256 Adam, learning rate = 1e-6	batch size = 64 Adam, learning rate = 1e-4	batch size = 128 Adam, learning rate = 1e-3
WDGRL	batch size = 64 Adam, learning rate = 1e-4, $\gamma = 10$, critic training step: 5, feature extractor and discriminator training step: 10	batch size = 128 Adam, learning rate = 1e-5, $\gamma = 10$, critic training step: 1, feature extractor and discriminator training step: 5	batch size = 64 Adam, learning rate = 1e-4, $\gamma = 10$, critic training step: 1, feature extractor and discriminator training step: 4	batch size = 64 Adam, learning rate = 1e-6, $\gamma = 10$, critic training step: 1, feature extractor and discriminator training step: 3
DSN	batch size = 32 SGD, momentum = 0.8, learning rate = 1e-2, $\alpha = 0.01$, $\beta = 0.075, \gamma = 0.25$	batch size = 32 SGD, momentum = 0.8, learning rate = 1e-2, $\alpha = 0.01$, $\beta = 0.075, \gamma = 0.25$	batch size = 64 SGD, momentum = 0.8, learning rate = 1e-2, $\alpha = 0.01$, $\beta = 0.075, \gamma = 0.4$	batch size = 256 SGD, momentum = 0.1, learning rate = 1e-4, $\alpha = 0.046$, $\beta = 0.075, \gamma = 0.25$
ADDA	batch size = 128 Adam, learning rate = 1e-5, critic training step: 1, target model training step: 1	batch size =256 Adam, learning rate = 1e-6, critic training step: 2, target model training step: 1	batch size = 128 Adam, learning rate = 1e-5, critic training step: 1, target model training step: 1	batch size = 128 Adam, learning rate = 1e-6, critic training step: 4, target model training step: 1

CAT	batch size = 512
	SGD,
	learning rate = $\frac{0.01}{(1+10p)^{0.75}}$,
	momentum = 0.9,
	p = 0.9,
	m = 30
		batch size = 128
		SGD,
		learning rate = $\frac{0.01}{(1+10p)^{0.75}}$,
		momentum = 0.9,
		p = 0.9,
		m = 30
			batch size = 256
			SGD,
			learning rate = $\frac{0.01}{(1+10p)^{0.75}}$,
			momentum = 0.9,
			p = 0.9,
			m = 30
				batch size = 128
				SGD,
				learning rate = $\frac{0.01}{(1+10p)^{0.75}}$,
				momentum = 0.9,
				p = 0.9,
				m = 30

CDAN	batch size = 64 SGD, momentum = 0.9, learning rate =1e-3	batch size = 128 SGD, momentum = 0.5, learning rate =0.1	batch size = 64 SGD, momentum = 0.9, learning rate =0.01	batch size = 16 SGD, momentum = 0.1, learning rate =1e-4
pixelDA	batch size = 64 Adam learning rate =0.0002, dim of the noise input: 10	batch size = 64 Adam learning rate =0.0002, dim of the noise input: 20	batch size = 32 Adam, learning rate =0.001, dim of the noise input: 20	batch size = 32 Adam, learning rate =0.001, dim of the noise input: 20
DRANet	batch size = 32 Adam	batch size = 32 Adam	batch size = 32 Adam	batch size = 32 Adam
MCD	batch size = 128 learning rate = 0.1	batch size = 128 learning rate = 0.01	batch size = 128 learning rate = 0.01	batch size = 128 learning rate = 0.01
MDD	batch size = 128 learning rate = 0.1	batch size = 128 learning rate = 0.01	batch size = 128 learning rate = 0.01	batch size = 128 learning rate = 0.01

G Details of Experimental Setup: TST Dataset with Shifted Label Distribution

The Tail Suspension Test (TST) dataset (Gallagher et al., 2017) consists of local field potentials (LFPs) recorded from the brains of 26 mice. These mice belong to two genetic backgrounds: a genetic model of bipolar disorder (Clock-Δ19) and wildtype mice. Each mouse is subjected to 3 behavioral assays which are designed to vary stress: home cage (HC), open field (OF), and tail-suspension (TS). We conduct experiments on two domain adaptation tasks using these neural activity data: transferring from wildtype mice to the bipolar mouse model and vice versa. We aim to predict for each one second window which of the 3 conditions (HC, OF, or TS) the mouse is currently experiencing. To create label distribution shifts, we subsample the datasets so that we have 6000 Homecage observations, 3000 Open Field observations, and 6000 Tail Suspension observations in the bipolar genotype dataset and 3000 Homecage observations, 6000 OpenField observations, and 3000 Tail Suspension observations in the wildtype genotype dataset.

Table 10: Summary of UDA results on the TST datasets with shifted label distribution, measured in terms of prediction accuracy (%) on the target domain. Values represent the mean prediction accuracy, and parentheses denote standard deviation across 5 runs

	DANN	WDGRL	DSN	ADDA	CDAN	Source only	**DARSA**
Clock-Δ19 to Wildtype	80.5 (0.91)	80.3 (0.92)	79.7 (0.39)	72.6 (3.29)	74.2 (0.62)	73.6 (0.38)	**86.2 (0.31)**
Wildtype to Clock-Δ19	80.6 (0.51)	80.3 (0.55)	80.7 (0.65)	68.6 (4.85)	74.3 (0.23)	70.6 (0.97)	**84.8 (0.09)**

G.1 Details of the TST Dataset with Shifted Label Distribution

The Tail Suspension Test (TST) dataset (Gallagher et al., 2017) consists of 26 mice recorded from two genetic backgrounds, Clock-Δ19 and wildtype. Clock-Δ19 is a genotype which has been proposed as a model of bipolar disorder while wildtype is considered as a typical or common genotype. Local field potentials (LFPs) are recorded from 11 brain regions and segmented into 1 second windows. For each window, power spectral density, coherence, and granger causality features are derived. Each mouse is placed through 3 behavioral contexts while collecting LFP recordings: home cage, open field, and tail-suspension. Mice spent 5 minutes in the home cage which is considered a baseline or low level of distress behavioral context. Mice spent 5 minutes in the open field context which is considered a moderate level of distress. Mice then spent 10 minutes in the tail suspension test which is a high distress context.

G.2 Model Structures

For feature extractor of the wildtype to bipolar task we use a network structure consisting of: a fully connected layer that maps our data to a feature space of 256 dimensions, with a LeakyReLU activation function; a fully connected layer that maps the feature space to 128 dimensions, and a Softplus activation function. For the bipolar to wildtype task, we use a network structure that includes: a fully connected layer that maps our data to a feature space of 256 dimensions, with a ReLU activation function; a fully connected layer that maps the feature space to 128 dimensions, with another ReLU activation function. For the classifier, we use a network structure that includes: three fully connected layers with ReLU activation and a dropout layer with a rate of 0.5. For benchmarks, we use the same network structures as our model to ensure fair comparisons, with the exception of DSN which has two fully connected layers with ReLU activation. See the included code link for additional details on each experiment.

G.3 Model hyperparameters

Again, we use Adaptive Experimentation (Ax) platform (Bakshy et al.; Letham et al., 2019), an automatic tuning approaches to select hyperparameters to maximize the performance of our method. We use Bayesian optimization supported by Ax with 20 iterations to decide the hyperparameter choice. We note that most of the SOTA comparisons are not specifically designed for shifted label distribution scenarios, and this setting caused issues in several competing. We used Ax to maximize their performance in label shifting scenarios. Details on the model hyperparameters are provided in Table 11 (If not specified, the default hyperparameters from their respective implementations are employed.).

H Details of Experimental Setup: VisDA-2017 Dataset with Shifted Label Distribution

Here, we further explore performance in a more complex scenario. As such, we have also evaluated DARSA on VisDA-2017, a challenging benchmark with significant domain shifts and non-trivial data structures.

VisDA-2017 Dataset and Experimental Setup:

The VisDA-2017 dataset (Peng et al., 2017) is a large-scale domain adaptation benchmark designed to evaluate the performance of algorithms on synthetic-to-real domain transfer. It comprises images from 12 object categories rendered in a virtual environment (source domain) and real-world images (target domain).

Table 11: Model hyperparameters used for the label distribution shifting TST datasets

	Clock-Δ19 to Wildtype	Wildtype to Clock-Δ19
DARSA	batch size = 128, α=1e-4, $\lambda_Y = 1$, $\lambda_D = 0.4$, $\lambda_c = 0.1$, $\lambda_a = 0.9$, m = 50 SGD, momentum = 0.6	batch size = 128, $\alpha = 0.001$, $\lambda_Y = 0.7$, $\lambda_D = 0.1$, $\lambda_c = 0.1$, $\lambda_a = 1$, m = 50 SGD, momentum = 0.3
DANN	batch size = 32 Adam, learning rate = 1e-4	batch size = 32 Adam, learning rate = 1e-4
WDGRL	batch size = 32 Adam, learning rate = 1e-4, $\gamma = 10$, critic training step: 1, feature extractor and discriminator training step: 2	batch size = 32 Adam, learning rate = 1e-5, $\gamma = 10$, critic training step: 1, feature extractor and discriminator training step: 3
DSN	batch size = 64 SGD, momentum = 0.5, learning rate = 0.1, $\alpha = 1$, $\beta = 1, \gamma = 1$	batch size = 32 SGD, momentum = 0.5, learning rate = 0.1, $\alpha = 1$, $\beta = 1, \gamma = 1$
CAT	batch size = 128 SGD, learning rate = $\frac{0.01}{(1+10p)^{0.75}}$, momentum = 0.9, p = 0.9, m = 3	batch size = 128 SGD, learning rate = $\frac{0.01}{(1+10p)^{0.75}}$, momentum = 0.9, p = 0.9, m = 3
CDAN	batch size = 64 SGD, momentum = 0.9, learning rate = 0.1	batch size = 64 SGD, momentum = 0.9, learning rate = 0.1

To evaluate DARSA under strong class imbalance, we follow a similar label shifting strategy as in the MNIST experiments, setting $\alpha = 8$. This implies an eightfold difference in the number of samples per class between the source and target domains. Specifically, for the source dataset, we sample a total of 32,400 images, and for the target dataset, we sample 5,400 images, maintaining the class imbalance.

Table 12 summarizes DARSA's per-class accuracy compared with several strong baselines (all metrics reported with $\alpha = 8$ for label shifting). Notably, we observed that increasing the batch size for sub-domain alignment can further improve DARSA's performance on VisDA-2017, indicating potential benefits from larger-scale training settings.

Table 12: Per-class and average accuracy on VisDA-2017 under a strong class imbalance with $\alpha = 8$.

	airplane	bicycle	bus	car	horse	knife	motorcycle	person	plant	skateboard	train	truck	**Avg. Acc.**
Source Only	30.7	87.0	0.0	88.0	61.4	9.1	1.8	25.3	18.1	91.1	8.8	6.0	35.6
CDAN (Long et al., 2018)	39.8	**95.0**	12.5	72.0	50.4	69.0	25.6	**68.0**	42.0	75.0	40.9	16.0	50.5
CDAN+ELS (Zhang et al., 2023)	94.6	59.0	66.8	74.0	71.5	41.0	**89.3**	39.0	64.5	84.0	**92.3**	21.0	66.4
CDAN+MCC (Jin et al., 2020)	88.5	84.0	40.0	57.0	42.6	19.0	37.9	60.0	**72.9**	92.0	57.0	31.0	56.8
EUDA (Abedi et al., 2024)	44.9	88.0	24.4	44.0	42.9	**72.0**	85.3	11.0	10.3	**99.0**	31.3	**48.0**	50.1
DARSA	**96.3**	75.0	**67.0**	**89.0**	**81.4**	69.0	86.6	49.0	58.8	92.0	91.6	11.0	**72.2**

These findings suggest that DARSA remains robust for more complex, non-Gaussian domains.

I Computational Analysis of DARSA

While our paper primarily focuses on the theoretical analysis of DARSA, we acknowledge the necessity of more extensive computational studies due to the inherent computational cost associated with the Wasserstein distance. Specifically, when dealing with probability measures having at most n supports, the computational complexity of the exact Wasserstein distance is on the order of $\mathcal{O}(n^3 \log(n))$ (Pele & Werman, 2009). In high-dimensional settings, such as images with numerous pixels, the size of these supports can become exceptionally large, significantly exacerbating the computational burden.

I.1 Embedding Network for Dimensionality Reduction

A crucial step in our approach involves utilizing an embedding or representation learning network. By mapping high-dimensional input data into a lower-dimensional latent space, we can substantially reduce the size of the supports for the probability measures. This dimensionality reduction directly translates to a significant decrease in the computational cost associated with calculating the Wasserstein distance or its approximation.

I.2 Sinkhorn Algorithm for Practicality

To further enhance computational efficiency, we employ the Sinkhorn algorithm (Altschuler et al., 2017) to approximate the Wasserstein distance. Directly solving the optimal transport problem can be computationally prohibitive, particularly in high dimensions. The Sinkhorn algorithm offers a more tractable alternative by adding an entropic regularization term to the optimal transport problem. While this introduces an approximation error, the regularization parameter ϵ provides a mechanism to balance the trade-off between accuracy and computational speed. Importantly, the entropic version of the Sinkhorn algorithm can achieve an approximate Wasserstein distance computation with a complexity of $\mathcal{O}(n^2)$ (up to polynomial orders of approximation errors) (Nguyen et al., 2022).

I.3 Empirical Time Analysis

In addition to these theoretical considerations, we conducted an empirical time study for the MNIST $\rightarrow$ MNIST-M domain adaptation task with a strong class imbalance ($\alpha = 8$). These experiments were performed on an NVIDIA GeForce RTX 2080 Ti GPU. Table 13 presents a comparison of total training steps, batch sizes, peak GPU memory usage, and average time per step for our proposed DARSA method against several established domain adaptation baselines.

Table 13: Computational analysis of DARSA

Metric	**DARSA**	DANN	ADDA	WDGRL	DSN	CDAN	CAT	DRANet
Total Steps	20k	4.23k	2.5k	50k	14.1k	42k	10k	100k
Batch size	1024	64	128	64	32	64	512	32
Peak GPU Memory Usage (MB)	54.5	1.3	3.2	2.9	15.8	118.3	469.0	112.1
Average Time Per Step (s)	1.88	0.21	0.45	0.29	0.22	0.17	2.73	0.34
Accuracy (%)	78.8	61.1	47.9	22.3	57.5	37.1	48.9	63.3

The seemingly higher average time per step for DARSA is primarily attributed to its utilization of a significantly larger batch size (1024). While processing these larger batches, even with the efficient Sinkhorn approximation for the Wasserstein distance on embedded representations, inherently demands more computation per step. However, this strategic choice allows DARSA to achieve convergence in a considerably lower number of total steps (20k) compared to many baselines. Consequently, despite the longer time per step, DARSA achieves a notably higher accuracy (78.8%), suggesting a more efficient overall training process than methods that rely on smaller batch sizes.

These results, when considered in conjunction with the theoretical computational complexities, demonstrate the practical feasibility of our proposed DARSA approach when employing the embedding network for dimensionality reduction and the Sinkhorn algorithm for efficient Wasserstein distance approximation.

J Accessibility of the Datasets and Computing Resources

Accessibility of the Datasets The MNIST, BSDS500, USPS, SVHN, and VisDA-2017 datasets are publicly available with an open-access license. The Tail Suspension Test (TST) dataset (Gallagher et al., 2017) is available to download at `https://research.repository.duke.edu/concern/datasets/zc77sr31x?locale=en` for free under a Creative Commons BY-NC Attribution-NonCommercial 4.0 International license.

Computing Resources The experiments are conducted on a computer cluster equipped with a NVIDIA GeForce RTX 2080 Ti that has a memory capacity of 11019MiB.