ADVERSARIAL SUPPORT ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the problem of aligning the supports of distributions. Compared to the existing work on distribution alignment, support alignment does not require the densities to be matched. We propose symmetric support difference as a divergence measure to quantify the mismatch between supports. We show that select discriminators (e.g. discriminator trained for Jensen–Shannon divergence) are able to map support differences as support differences in their one-dimensional output space. Following this result, our method aligns supports by minimizing a symmetrized relaxed optimal transport cost in the discriminator 1D space via an adversarial process. Furthermore, we show that our approach can be viewed as a limit of existing notions of alignment by increasing transportation assignment tolerance. We quantitatively evaluate the method across domain adaptation tasks with shifts in label distributions. Our experiments show that the proposed method is more robust against these shifts than other alignment-based baselines.

1 INTRODUCTION

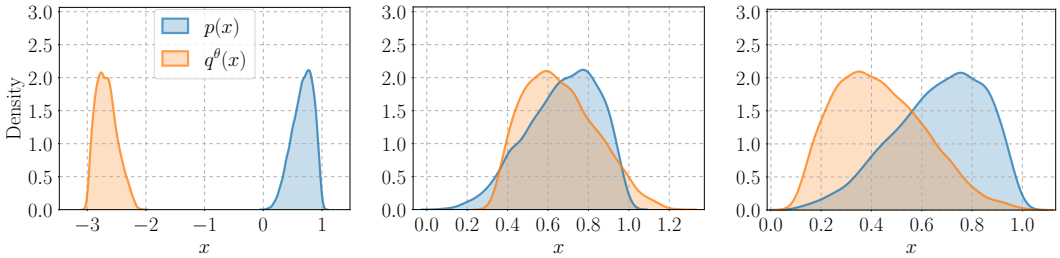
Learning tasks often involve estimating properties of distributions from samples or aligning such characteristics across domains. We can align full distributions (adversarial domain alignment), certain statistics (canonical correlation analysis), or the support of distributions (this paper). Much of the recent work has focused on full distributional alignment, for good reasons. In domain adaptation, motivated by theoretical results (Ben-David et al., 2007; 2010), a series of papers (Ajakan et al., 2014; Ganin & Lempitsky, 2015; Ganin et al., 2016; Tzeng et al., 2017; Shen et al., 2018; Pei et al., 2018; Zhao et al., 2018; Li et al., 2018a; Wang et al., 2021; Kumar et al., 2018) seek to align distributions of representations between domains, and utilize a shared classifier on the aligned representation space.

Alignment in distributions implies alignment in supports. However, when there are additional objectives/constraints to satisfy, the minimizer for a distribution alignment objective does not necessarily minimize a support alignment objective. Example in Figure 1 demonstrates the qualitative distinction between two minimizers when distribution alignment is not achievable. The distribution alignment objective prefers to keep supports unaligned even if support alignment is achievable. Recent works (Zhao et al., 2019; Li et al., 2020; Tan et al., 2020; Wu et al., 2019b; Tachet des Combes et al., 2020) have demonstrated that a shift in label distributions between source and target leads to a characterizable performance drop when the representations are forced into a distribution alignment. The error bound in Johansson et al. (2019) suggests aligning the supports of representations instead.

In this paper, we focus on distribution support as the key characteristic to align. We introduce a support divergence to measure the support mismatch and algorithms to optimize such alignment. We also position our approach in the spectrum of other alignment methods. Our contributions are as follows (all proofs can be found in Appendix A):

1. In Section 2.1, we measure the differences between supports of distributions. Building on the Hausdorff distance, we introduce a novel support divergence better suited for optimization, which we refer to as *symmetric support difference* (SSD) divergence.
2. In Section 2.2, we identify an important property of the discriminator trained for Jensen–Shannon divergence: support differences in the original space of interest are "preserved" as support differences in its one-dimensional output space.
3. In Section 3, we present our practical algorithm for support alignment, *Adversarial Support Alignment* (ASA). Essentially, based on the analysis presented in Section 2.2, our solution is to align supports in the discriminator 1D space, which is computationally efficient.

4. In Section 4, we place different notions of alignment – distribution alignment, relaxed distribution alignment and support alignment – within a coherent spectrum from the point of view of optimal transport, characterizing their relationships, both theoretically in terms of their objectives and practically in terms of their algorithms.
5. In Section 5, we demonstrate the theoretical effectiveness of support alignment in practice for domain adaptation setting. Compared to other alignment-based baselines, our proposed method is more robust against shifts in label distributions.



(a) Initializaion
$\mathcal{D}_W(p, q^\theta) = 11.12$
$\mathcal{D}_\triangle(p, q^\theta) = 14.9$

(b) Distribution alignment
$\mathcal{D}_W(p, q^\theta) = 2 \cdot 10^{-3}$
$\mathcal{D}_\triangle(p, q^\theta) = 6 \cdot 10^{-4}$

(c) Support alignment
$\mathcal{D}_W(p, q^\theta) = 5 \cdot 10^{-2}$
$\mathcal{D}_\triangle(p, q^\theta) < 1 \cdot 10^{-6}$

Figure 1: Illustration of differences between the final configurations of distribution alignment and support alignment procedures. $p(x)$ is a fixed Beta distribution $p(x) = \mathrm{Beta}(x \,|\, 4, 2)$ with support $[0, 1]$; $q^\theta(x)$ is a "shifted" Beta distribution $q^\theta(x) = \mathrm{Beta}(x - \theta \,|\, 2, 4)$ parameterized by θ with support $[\theta, \theta + 1]$. Panel (a) shows the initial configuration with $\theta_{\text{init}} = -3$. Panel (b) shows the result by distribution alignment. Panel (c) shows the results by support alignment. We report Wasserstein distance $\mathcal{D}_W(p, q^\theta)$ (7) and SSD divergence $\mathcal{D}_\triangle(p, q^\theta)$ (1).

2 SSD DIVERGENCE AND SUPPORT ALIGNMENT

Notation. We consider an Euclidean space $\mathcal{X} = \mathbb{R}^n$ equipped with Borel sigma algebra $\mathcal{B}$ and a metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (e.g. Euclidean distance). Let $\mathcal{P}$ be the set of probability measures on $(\mathcal{X}, \mathcal{B})$. For $p \in \mathcal{P}$, the support of p is denoted by $\mathrm{supp}(p)$ and is defined as the smallest closed set $X \subseteq \mathcal{X}$ such that $p(X) = 1$. $f_\sharp p$ denotes the pushforward measure of p induced by a measurable mapping f. The distance between a point $x \in \mathcal{X}$ and a subset $Y \subseteq \mathcal{X}$ is defined as $d(x, Y) = \inf_{y \in Y} d(x, y)$. The symmetric difference of two sets A and B is defined as $A \triangle B = (A \setminus B) \cup (B \setminus A)$.

2.1 DIFFERENCE BETWEEN SUPPORTS

To align the supports of distributions, we first evaluate how different they are. Similar to distribution divergences like Jensen–Shannon divergence and Wasserstein distance, we introduce a notion of support divergence. A *support divergence*[1] between two distributions in $\mathcal{P}$ is a function $\mathcal{D}_S(\cdot, \cdot) : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ satisfying: 1) $\mathcal{D}_S(p, q) \geq 0$ for all $p, q \in \mathcal{P}$; 2) $\mathcal{D}_S(p, q) = 0$ iff $\mathrm{supp}(p) = \mathrm{supp}(q)$.

While a distribution divergence is sensitive to both density and support differences, a support divergence only needs to detect mismatches in supports, which are subsets of the metric space $\mathcal{X}$. An example of a distance between subsets of a metric space is the Hausdorff distance: $d_H(X, Y) = \max\{\sup_{x \in X} d(x, Y), \sup_{y \in Y} d(y, X)\}$. Since it depends only on the greatest distance between a point and a set, minimizing this objective for alignment only provides signal to a single point. To make the optimization less sparse, we consider all points that violate the support alignment criterion and introduce *symmetric support difference* (SSD) divergence:

$$\mathcal{D}_\triangle(p, q) = \mathbb{E}_p \left[d(x, \mathrm{supp}(q)) \right] + \mathbb{E}_q \left[d(x, \mathrm{supp}(p)) \right]. \tag{1}$$

Proposition 2.1. *SSD divergence $\mathcal{D}_\triangle(p, q)$ is a support divergence.*

[1] It is not technically a divergence on the space of distributions, since $\mathcal{D}_S(p, q) = 0$ does not imply $p = q$.

2.2 SUPPORT ALIGNMENT IN ONE-DIMENSIONAL SPACE

Goodfellow et al. (2014) showed that the log-loss discriminator $f : \mathcal{X} \to [0, 1]$, trained to distinguish samples from distributions p and q ($\sup_f \mathbb{E}_{x\sim p}[\log f(x)] + \mathbb{E}_{x\sim q}[\log(1 - f(x))]$) can be used to estimate the Jensen–Shannon divergence between p and q. The closed form maximizer f^* is

$$f^*(x) = \frac{p(x)}{p(x) + q(x)}, \qquad \forall x \in \operatorname{supp}(p) \cup \operatorname{supp}(q). \tag{2}$$

Note that for a point $x \notin \operatorname{supp}(p) \cup \operatorname{supp}(q)$ the value of $f^*(x)$ can be set to an arbitrary value in $[0, 1]$, since the log-loss does not depend on $f(x)$ for such x. Using (2), we can establish a connection between the pushforward distributions $f^*{}_\sharp p$ and $f^*{}_\sharp q$.

Proposition 2.2. *Let f^* be the optimal log-loss discriminator (2) between p and q, and let τ_p and τ_q be the probability density functions of the pushforward measures $f^*{}_\sharp p$ and $f^*{}_\sharp q$ respectively*[2]*. Then,*

$$\frac{\tau_p(a)}{\tau_p(a) + \tau_q(a)} = a, \qquad \forall a \in \operatorname{supp}(\tau_p) \cup \operatorname{supp}(\tau_q). \tag{3}$$

Intuitively this proposition states the following. If the optimal discriminator f^* maps $x \in \mathcal{X}$ to $a \in [0, 1]$, i.e. $f^*(x) = a$, a directly corresponds to the ratio of densities in not only the original space $a = p(x)/(p(x) + q(x))$, but also the 1D discriminator output space $a = \tau_p(a)/(\tau_p(a) + \tau_q(a))$, which leads to the fact that $p(x)/(p(x) + q(x)) = f^*(x) = a = \tau_p(a)/(\tau_p(a) + \tau_q(a))$.

Using Proposition 2.2, we can prove our main theorem, which characterizes the ability of the log-loss discriminator to identify support misalignment.

Theorem 2.1. *Let f^* be the optimal discriminator (2) between p and q. Then, $\mathcal{D}_\triangle(p, q) = 0$ if and only if $\mathcal{D}_\triangle(f^*{}_\sharp p, f^*{}_\sharp q) = 0$.*

Proof sketch. We observe that for any point $x \in \operatorname{supp}(p) \setminus \operatorname{supp}(q)$ we have $f^*(x) = 1$ and, by Proposition 2.2, $\tau_p(1) > 0$ and $\tau_q(1) = 0$. Repeating the same argument in reverse direction we see that $\operatorname{supp}(p) \setminus \operatorname{supp}(q) \neq \varnothing$ if and only if $1 \in \operatorname{supp}(f^*{}_\sharp p) \setminus \operatorname{supp}(f^*{}_\sharp q)$. Similarly, $\operatorname{supp}(q) \setminus \operatorname{supp}(p) \neq \varnothing$ if and only if $0 \in \operatorname{supp}(f^*{}_\sharp q) \setminus \operatorname{supp}(f^*{}_\sharp p)$. In other words, f^* outputs the extreme values 1 and 0 if and only if the supports of p and q are misaligned and each of this extreme value belongs to the support of only one of the pushforward distributions $\operatorname{supp}(f^*{}_\sharp p)$ or $\operatorname{supp}(f^*{}_\sharp q)$.

We conclude this section with two technical remarks on Theorem 2.1.

Remark 2.1.1. The result of Theorem 2.1 does not necessarily hold for other types of discriminator. For instance, the dual Wasserstein discriminator (Arjovsky et al., 2017; Gulrajani et al., 2017) does not always highlight the support difference in the original space as a support difference in the discriminator output space. This observation is formaly stated in the following proposition.

Proposition 2.3. *Let $f^\star_W$ be the optimal solution of $\sup_{f:\operatorname{Lip}(f)\leq 1} \mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{x\sim q}[f(x)]$, where $\operatorname{Lip}(f)$ is the Lipschitz constant of f. There exist distributions p and q such that $\operatorname{supp}(p) \neq \operatorname{supp}(q)$ but $\operatorname{supp}(f^\star_W{}_\sharp p) = \operatorname{supp}(f^\star_W{}_\sharp q)$.*

Remark 2.1.2. We note that in practice the discriminator is typically parameterized as $f(x) = \sigma(g(x))$, where $g : \mathcal{X} \to \mathbb{R}$ is realized by a deep neural network and $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function. The optimization problem for g is

$$\inf_g \ \mathbb{E}_{x\sim p}\left[\log(1 + e^{-g(x)})\right] + \mathbb{E}_{x\sim q}\left[\log(1 + e^{g(x)})\right], \tag{4}$$

and the optimal solution is $g^*(x) = \log p(x) - \log q(x)$. Naturally the result of Theorem 2.1 holds for discriminator g^*, since $g^*(x) = \sigma^{-1}(f^*(x))$ and σ is a bijective mapping from $(-\infty, \infty)$ to $[0, 1]$. The main difference of g^* from f^* is that in the case of $\operatorname{supp}(p) \neq \operatorname{supp}(q)$, optimization of the log-loss makes the values $g^*(x)$ infinite ($\pm\infty$) for $x \in \operatorname{supp}(p) \triangle \operatorname{supp}(q)$.

[2] In order to simplify the argument, we assume that all distributions p, q, $f^*{}_\sharp p$, $f^*{}_\sharp q$ have PDFs.

3 ADVERSARIAL SUPPORT ALIGNMENT

We consider distributions p and q parameterized by θ: p^θ, q^θ. The log-loss discriminator g optimized for (4) is parameterized by ψ: g^ψ. Our analysis in Section 2.2 already suggests an algorithm. Namely, we can optimize θ by minimizing $\mathcal{D}_\triangle({g^\psi}_\sharp p^\theta, {g^\psi}_\sharp q^\theta)$ while optimizing ψ by (4). This adversarial game is analogous to the setup of the existing distribution alignment algorithms[3].

In practice, rather than having direct access to p^θ, q^θ, which is unavailable, we are often given i.i.d. samples $\{x_i^p\}_{i=1}^N, \{x_i^q\}_{i=1}^M$. They form discrete distributions $\hat{p}^\theta(x) = \frac{1}{N}\sum_{i=1}^N \delta(x - x_i^p), \hat{q}^\theta(x) = \frac{1}{M}\sum_{i=1}^M \delta(x - x_i^q)$, and ${g^\psi}_\sharp \hat{p}^\theta(a) = \frac{1}{N}\sum_{i=1}^N \delta(a - g^\psi(x_i^p)), {g^\psi}_\sharp \hat{q}^\theta(a) = \frac{1}{M}\sum_{i=1}^M \delta(a - g^\psi(x_i^q))$. Since ${g^\psi}_\sharp \hat{p}^\theta$ and ${g^\psi}_\sharp \hat{q}^\theta$ are discrete distributions, they have supports $\{g^\psi(x_i^p)\}_{i=1}^N$ and $\{g^\psi(x_i^q)\}_{i=1}^M$ respectively. SSD divergence between discrete distributions ${g^\psi}_\sharp \hat{p}^\theta$ and ${g^\psi}_\sharp \hat{q}^\theta$ is

$$\mathcal{D}_\triangle({g^\psi}_\sharp \hat{p}^\theta, {g^\psi}_\sharp \hat{q}^\theta) = \frac{1}{N}\sum_{i=1}^N d\left(g^\psi(x_i^p), \{g^\psi(x_j^q)\}_{j=1}^M\right) + \frac{1}{M}\sum_{i=1}^M d\left(g^\psi(x_i^q), \{g^\psi(x_j^p)\}_{j=1}^N\right). \quad (5)$$

Effect of mini-batch training. When training on large datasets, we need to rely on stochastic optimization with mini-batches. We denote the mini-batches (of same size, as in common practice) from p^θ and q^θ as $x^p = \{x_i^p\}_{i=1}^m$ and $x^q = \{x_i^q\}_{i=1}^m$ respectively. By minimizing $\mathcal{D}_\triangle(g^\psi(x^p), g^\psi(x^q))$, we only consider the mini-batch transportaion cost rather than the population transportation cost (5). We observe that in practice (details in Appendix D.5) the described algorithm brings the distributions to a state closer to distribution alignment rather than support alignment. The problem is in the typically small batch size. The algorithm actually tries to enforce support alignment for all possible pairs of mini-batches, which is a much stricter constraint than population support alignment.

To address the issue mentioned above, without working with a much larger batch size, we create two "history buffers": h^p, storing the previous 1D discriminator outputs of (at most) n samples from p^θ, and a similar buffer h^q for q^θ. Specifically, $h = \{g^{\psi_{\text{old}_i}}(x_{\text{old}_i})\}_{i=1}^n$ stores the values of the previous n samples mapped by their corresponding past "versions" of the discriminator $g^{\psi_{\text{old}_i}}$. We minimize $\mathcal{D}_\triangle(v^p, v^q)$, where $v^p = \text{concat}(h^p, g^\psi(x^p))$, $v^q = \text{concat}(h^q, g^\psi(x^q))$:

$$\mathcal{D}_\triangle(v^p, v^q) = \frac{1}{n+m}\left(\sum_{i=1}^{n+m} d(v_i^p, v^q) + \sum_{j=1}^{n+m} d(v_j^q, v^p)\right). \quad (6)$$

Note that $\mathcal{D}_\triangle(\cdot,\cdot)$ between two sets of 1D samples can be efficiently calculated since $d(v_i^p, v^q)$ and $d(v_j^q, v^p)$ are simply 1-nearest neighbor distances in 1D. Moreover the history buffers store only the scalar values from the previous batches. These values are only considered in nearest neighbor assignment but do not directly provide gradient signal for optimization. Thus, the computation overhead of including a long history buffer is very light. We present our full algorithm, *Adversarial Support Alignment* (ASA), in Algorithm 1.

4 SPECTRUM OF NOTIONS OF ALIGNMENT

In this section, we take a closer look into our work and different existing notions of alignment that have been proposed in the literature, especially their formulations from the optimal transport perspective. We show that our proposed support alignment framework is a limit of existing notions of alignment, both in terms of theory and algorithm, by increasing transportation assignment tolerance.

4.1 THEORETICAL CONNECTIONS

Distribution alignment. Wasserstein distance is a commonly used objective for distribution alignment. In our analysis, we focus on the Wasserstein-1 distance:

$$\mathcal{D}_W(p, q) = \inf_{\gamma \in \Gamma(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[d(x, y)], \quad (7)$$

[3]Following existing adversarial distribution alignment methods, e.g. (Goodfellow et al., 2014), we use single update of ψ per 1 update of θ. While theoretical analysis for both distribution alignment and support alignment (ours) assume optimal discriminators, training with single update of ψ is computationally cheap and effective.

Algorithm 1 Our proposed ASA algorithm. n (maximum history buffer size), we use $n = 1000$.

1: **for** number of training steps **do**
2: Sample mini-batches $\{x_i^p\}_{i=1}^m \sim p^\theta,\ \{x_i^q\}_{i=1}^m \sim q^\theta$.
3: Perform optimization step on ψ using stochastic gradient
$$\nabla_\psi \left(\frac{1}{m} \sum_{i=1}^{m} \Big[\log(1+\exp(-g^\psi(x_i^p))) + \log(1+\exp(g^\psi(x_i^q))) \Big] \right).$$
4: $v^p \leftarrow \text{concat}(h^p, \{g^\psi(x_i^p)\}_{i=1}^m),\quad v^q \leftarrow \text{concat}(h^q, \{g^\psi(x_i^q)\}_{i=1}^m)$.
5: $\pi^i_{p\to q} \leftarrow \arg\min_j d(v_i^p, v_j^q),\quad \pi^j_{q\to p} \leftarrow \arg\min_i d(v_i^p, v_j^q)$.
6: Perform optimization step on θ using stochastic gradient
$$\nabla_\theta \left(\frac{1}{n+m} \sum_{i=1}^{n+m} \Big[d(v_i^p, v^q_{\pi^i_{p\to q}}) + d(v_i^q, v^p_{\pi^i_{q\to p}}) \Big] \right).$$
7: UPDATEHISTORY$(h^p, \{g^\psi(x_i^p)\}_{i=1}^m)$, UPDATEHISTORY$(h^q, \{g^\psi(x_i^q)\}_{i=1}^m)$.
8: **end for**

where $\Gamma(p,q)$ is the set of all measures on $\mathcal{X} \times \mathcal{X}$ with marginals of p and q, respectively. The value of $\mathcal{D}_W(p,q)$ is the minimal transportation cost for transporting probability mass from p to q. The transportation cost is zero if and only if $p = q$, meaning the distributions are aligned.

Relaxed distribution alignment. Wu et al. (2019b) proposed a modified Wasserstein distance to achieve asymmetrically-relaxed distribution alignment, namely β-admissible Wasserstein distance:

$$\mathcal{D}_W^\beta(p,q) = \inf_{\gamma \in \Gamma_\beta(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[d(x,y)], \tag{8}$$

where $\Gamma_\beta(p,q)$ is the set of all measures γ on $\mathcal{X} \times \mathcal{X}$ such that $\int \gamma(x,y)dy = p(x), \forall x$ and $\int \gamma(x,y)dx \le (1+\beta)q(y), \forall y$. With the relaxed marginal constraints, one could choose a transportation plan γ which transports probability mass from p to a modified distribution q' rather than the original distribution q as long as q' satisfies the constraint $q'(x) \le (1+\beta)q(x), \forall x$. Therefore, $\mathcal{D}_W^\beta(p,q)$ is zero if and only if $p(x) \le (1+\beta)q(x), \forall x$. In (Wu et al., 2019b), β is normally set to a positive finite number to achieve the asymmetric-relaxation of distribution alignment, and it is shown that $\mathcal{D}_W^0(p,q) = \mathcal{D}_W(p,q)$. We can extend $\mathcal{D}_W^\beta(p,q)$ to a symmetric version, which we term β_1, β_2-admissible Wasserstein distance:

$$\mathcal{D}_W^{\beta_1,\beta_2}(p,q) = \mathcal{D}_W^{\beta_1}(p,q) + \mathcal{D}_W^{\beta_2}(q,p). \tag{9}$$

The aforementioned property of β-admissible Wasserstein distance implies that $\mathcal{D}_W^{\beta_1,\beta_2}(p,q) = 0$ if and only if $p(x) \le (1+\beta_1)q(x), \forall x$ and $q(y) \le (1+\beta_2)p(y), \forall y$, in which case we call p and q "(β_1, β_2)-aligned", with β_1 and β_2 controlling the transportation assignment tolerances.

Support alignment. The term $\mathbb{E}_p[d(x, \text{supp}(q))]$ in (1) represents the average distance from samples in p to the support of q. From the optimal transport perspective, this value is the minimal transportation cost of transporting the probability mass of p into the support of q. We can show that SSD divergence can be considered as a transportation cost in the limit of infinite assignment tolerance.

Proposition 4.1. $\mathcal{D}_W^{\infty,\infty}(p,q) \coloneqq \lim_{\beta_1,\beta_2\to\infty} \mathcal{D}_W^{\beta_1,\beta_2}(p,q) = \mathcal{D}_\triangle(p,q)$.

We now have completed the spectrum of alignment objectives defined within the optimal transport framework. The following proposition establishes the relationship within the spectrum.

Proposition 4.2. *Let p and q be two distributions in $\mathcal{P}$. Then,*

1. $\mathcal{D}_W(p,q) = 0$ *implies* $\mathcal{D}_W^{\beta_1,\beta_2}(p,q) = 0$ *for all finite* $\beta_1, \beta_2 > 0$.
2. $\mathcal{D}_W^{\beta_1,\beta_2}(p,q) = 0$ *for some finite* $\beta_1, \beta_2 > 0$ *implies* $\mathcal{D}_\triangle(p,q) = 0$.
3. *The converse of statements 1 and 2 are false.*

In addition to the result presented in Theorem 2.1, we can show that the log-loss discriminator can also "preserve" the existing notions of alignment.

Proposition 4.3. *Let f^* be the optimal discriminator (2) for given distributions p and q. Then,*

1. $\mathcal{D}_W(p,q) = 0$ *iff* $\mathcal{D}_W(f^*{}_\sharp p, f^*{}_\sharp q) = 0$;
2. $\mathcal{D}_W^{\beta_1,\beta_2}(p,q) = 0$ *iff* $\mathcal{D}_W^{\beta_1,\beta_2}(f^*{}_\sharp p, f^*{}_\sharp q) = 0$.

4.2 ALGORITHMIC CONNECTIONS

The result of Proposition 4.3 suggests methods similar to our ASA algorithm presented in Section 3 can achieve different notions of alignment by minimizing objectives discussed in Section 4.1 between the 1D pushforward distributions. We consider the setup used in Section 3 but without history buffers to simplify the analysis, as their usage is orthogonal to our discussion in this section.

Recall that we work with a mini-batch setting, where $\{x_i^p\}_{i=1}^m$ and $\{x_i^q\}_{i=1}^m$ are sampled from p and q respectively, and g is the adversarial log-loss discriminator. We denote the corresponding 1D outputs from the log-loss discriminator by $o^p = \{o_i^p\}_{i=1}^m = \{g(x_i^p)\}_{i=1}^m$ and o^q (defined similarly).

Distribution alignment. We adapt (7) for $\{o_i^p\}_{i=1}^m$ and $\{o_i^q\}_{i=1}^m$:

$$\mathcal{D}_W(o^p, o^q) = \inf_{\gamma \in \Gamma(o^p, o^q)} \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^m \gamma_{ij} d(o_i^q, o_j^q), \tag{10}$$

where $\Gamma(o^p, o^q)$ is the set of $m \times m$ doubly stochastic matrices. Since o^p and o^q are sets of 1D samples with the same size, it can be shown (Rabin et al., 2011) that the optimal γ^* corresponds to an assignment π^*, which pairs points in the sorting order and can be computed efficiently by sorting both sets o^p and o^q. The transportation cost is zero if and only if there exists an invertible 1-to-1 assignment π^* such that $o_i^p = o_{\pi(i)}^q$. GAN training algorithms proposed in (Deshpande et al., 2018; 2019) utilize the above sorting procedure to estimate the maximum sliced Wasserstein distance.

Relaxed distribution alignment. Similarly, we can adapt (8):

$$\mathcal{D}_W^\beta(o^p, o^q) = \inf_{\gamma \in \Gamma_\beta(o^p, o^q)} \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^m \gamma_{ij} d(o_i^q, o_j^q), \tag{11}$$

where $\Gamma_\beta(o^p, o^q)$ is the set of $m \times m$ matrices with non-negative real entries, such that $\sum_{j=1}^m \gamma_{ij} = 1, \forall i$ and $\sum_{i=1}^m \gamma_{ij} \leq 1 + \beta, \forall j$. The optimization goal in (11) is to find a "soft-assignment" γ which describes the transportation of probability mass from points o_i^p in o^p to points o_i^q in o^q. The parameter β controls the set of admissible assignments Γ_β, which is similar to its role discussed in Section 4.1: with transportation assignment tolerance β, the total mass of points in o^p transported to each of the points o_i^q cannot exceed $1 + \beta$. We refer to such assignments as $(\beta+1)$-to-1 assignment. The transportation cost is zero if and only if there exists such an assignment between o^p and o^q.

It can be shown (see Appendix C) that for integer value of β, the set of minimizers of (11) must contain a "hard-assignment" transportation plan, which assigns each point o_i^p to exactly one point o_j^q. Then $(1 + \beta)$ gives the upper bound on the number of points o_i^p that can be transported to given point o_j^q. This hard assignment problem can be solved quasi-linearly with worst case time complexity $\mathcal{O}\left((\beta+1)m^2\right)$ (Bonneel & Coeurjolly, 2019), which, combined with Proposition 4.3, can lead to new algorithms for relaxed distribution alignment besides those proposed in Wu et al. (2019b).

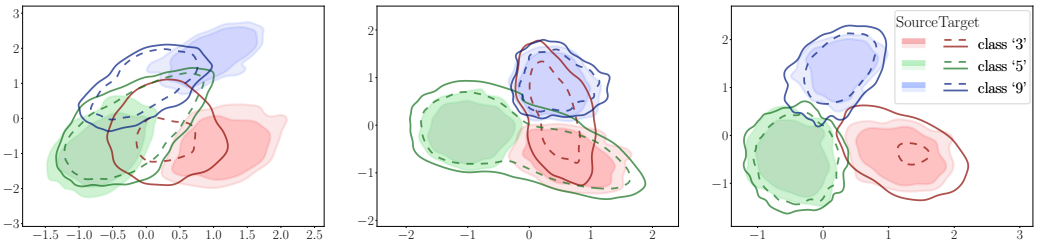
Support alignment. When $\beta = \infty$, there are no constraints on assignment γ in (11), and each point o_i^p can be assigned to any of the points o_j^q. In Section 3, we have shown that this is simply 1-nearest neighbor assignment, or to follow the above terminology, ∞-to-1 assignment.

5 EXPERIMENTS

Problem setting. We evaluate our proposed ASA method in the setting of unsupervised domain adaptation (UDA). The goal of UDA algorithms is to train and "adapt" a classification model $M : \mathcal{X} \to \mathcal{Y}$ from source domain distribution $p_{X,Y}$ to target domain distribution $q_{X,Y}$ given the access to a labeled source dataset $\{x_i^p, y_i^p\}_{i=1}^{N^p} \sim p_{X,Y}$ and an unlabeled target dataset $\{x_i^q\}_{i=1}^{N^q} \sim q_X$.

A common approach for UDA is to represent M as $C^\phi \circ F^\theta$: a classifier $C^\phi : \mathcal{Z} \to \mathcal{Y}$ and a feature extractor $F^\theta : \mathcal{X} \to \mathcal{Z}$, and train C^ϕ and F^θ by minimizing: 1) classification loss ℓ_{cls} on source examples; 2) alignment loss $\mathcal{D}_{\text{align}}$ measuring discrepancy between $p_Z^\theta = {F^\theta}_\sharp p_X$ and $q_Z^\theta = {F^\theta}_\sharp q_X$:

$$\min_{\phi,\theta} \quad \frac{1}{N^p} \sum_{i=1}^{N^p} \ell_{\text{cls}}(C^\phi(F^\theta(x_i^p)), y_i^p) + \lambda \cdot \mathcal{D}_{\text{align}}\left(\{F^\theta(x_i^p)\}_{i=1}^{N^p}, \{F^\theta(x_i^q)\}_{i=1}^{N^q}\right), \tag{12}$$



(a) No DA (avg acc: 63%)
$\mathcal{D}_W(p_Z^\theta, q_Z^\theta) = 0.78$
$\mathcal{D}_\triangle(p_Z^\theta, q_Z^\theta) = 0.10$

(b) DANN (avg acc: 75%)
$\mathcal{D}_W(p_Z^\theta, q_Z^\theta) = 0.07$
$\mathcal{D}_\triangle(p_Z^\theta, q_Z^\theta) = 0.02$

(c) ASA-abs (avg acc: 94%)
$\mathcal{D}_W(p_Z^\theta, q_Z^\theta) = 0.59$
$\mathcal{D}_\triangle(p_Z^\theta, q_Z^\theta) = 0.03$

Figure 2: Visualization of learned 2D embedding spaces on 3-class USPS→MNIST adaptation with label distribution shift. Each panel shows 2 level sets (outer one approximates the support) of the kernel density estimates of embeddings in source (filled regions) and target domains (solid/dashed lines). We report the average class accuracy of the target domain, $\mathcal{D}_W$ and $\mathcal{D}_\triangle$ between embeddings.

In practice $\mathcal{D}_{\text{align}}$ is an estimate of a divergence measure via an adversarial discriminator g^ψ. Choices of $\mathcal{D}_{\text{align}}$ include f-divergences (Ganin et al., 2016; Nowozin et al., 2016) and Wasserstein distance (Arjovsky et al., 2017) to enforce distribution alignment and versions of re-weighted/relaxed distribution divergences (Wu et al., 2019b; Tachet des Combes et al., 2020) to enforce relaxed distribution alignment. For support alignment, we apply the proposed ASA method as the alignment subroutine in (12) with log-loss discriminator g^ψ (4) and $\mathcal{D}_{\text{align}}$ computed as (6).

Task specifications. We consider 3 UDA tasks: USPS→MNIST, STL→CIFAR, and VisDA-2017, and 2 versions of ASA: **ASA-sq**, **ASA-abs** corresponding to squared and absolute distances respectively for $d(\cdot,\cdot)$ in (6). We compare ASA with: **No DA** (no domain adaptation), **DANN** (Ganin et al., 2016) (distribution alignment with JS divergence), **VADA** (Shu et al., 2018) (distribution alignment with virtual adversarial training), **IWDAN**, **IWCDAN** (Tachet des Combes et al., 2020) (relaxed distribution alignment via importance weighting) **sDANN-β** (Wu et al., 2019b) (relaxed/β-admissible JS divergence via re-weighting). Please refer to Appendix D for full experimental details.

To evaluate the robustness of the methods, we simulate label distribution shift by subsampling source and target dataset, so that source has balanced label distribution and target label distribution follows the power law $q_Y(y) \propto \sigma(y)^{-\alpha}$, where σ is a random permutation of class labels $\{1, \dots, K\}$ and α controls the severity of the shift ($\alpha = 0$ means balanced label distribution). For each task, we generate 5 random permutations for 4 different shift levels $\alpha \in \{0, 1, 1.5, 2\}$. Essentially we transform each (source, target) dataset pair to $5 \times 4 = 20$ tasks of different difficulty levels, since classes are not equally difficult and different permutations can give them different weights.

Evaluation metrics. We choose the average (per-)class accuracy and minimum (per-)class accuracy on the target test set as evaluation metrics. Under the average class accuracy metric, all classes are treated as equally important (despite the unequal representation during training for $\alpha > 0$), and the minimum class accuracy focuses on model's worst within-class performance. In order to account for the variability of task difficulties across random permutations of target labels, we report robust statistics, median and a 25-75 percentile interval, across 5 runs.

Illustrative example. First we consider a simplified setting to intuitively understand and directly analyze the behavior of our proposed support alignment method in domain adaptation under label distribution shift. We consider a 3-class USPS→MNIST problem by selecting a subset of examples corresponding to digits '3', '5', and '9', and use a feature extractor network with 2D output space. We introduce label distribution shift as described above with $\alpha = 1.5$, i.e. the probabilities of classes in the target domain are 12%, 23%, and 65%. We compare No DA, DANN, ASA-abs by their average target classification accuracy, Wasserstein distance $\mathcal{D}_W(p_Z^\theta, q_Z^\theta)$ and SSD divergence $\mathcal{D}_\triangle(p_Z^\theta, q_Z^\theta)$ between the learned embeddings of source and target domain. We apply a global affine transformation to each embedding space in order to have comparable distances between different spaces: we center the embeddings so that their average is 0 and re-scale them so that their average norm is 1. The results are shown in Figure 2 and Table D.5. Compared to No DA, both DANN and ASA achieve

Table 1: Average and minimum class accuracy (%) on USPS→MNIST with different levels of shifts in label distributions (higher α implies more severe imbalance). We report median (the main number), and 25 (subscript) and 75 (superscript) percentiles across 5 runs.

	$\alpha = 0.0$		$\alpha = 1.0$		$\alpha = 1.5$		$\alpha = 2.0$	
Algorithm	average	min	average	min	average	min	average	min
No DA	$71.9^{72.9}_{70.4}$	$20.3^{22.9}_{17.6}$	$72.9^{74.7}_{72.0}$	$25.8^{31.8}_{18.3}$	$71.3^{72.5}_{71.2}$	$27.5^{37.3}_{24.2}$	$71.3^{73.0}_{70.6}$	$16.6^{26.8}_{10.8}$
DANN	$97.8^{97.8}_{97.6}$	$96.0^{96.1}_{95.8}$	$83.5^{84.6}_{76.7}$	$25.1^{36.9}_{08.4}$	$70.0^{71.2}_{63.9}$	$01.1^{01.5}_{01.0}$	$57.8^{60.4}_{52.0}$	$00.9^{01.6}_{00.5}$
VADA	$\mathbf{98.0}^{98.0}_{97.9}$	$96.2^{96.3}_{95.9}$	$88.2^{89.9}_{88.1}$	$48.9^{50.0}_{47.8}$	$78.2^{83.1}_{70.7}$	$06.6^{23.5}_{02.4}$	$61.9^{65.4}_{56.3}$	$01.4^{01.5}_{00.8}$
IWDAN	$97.5^{97.5}_{97.4}$	$95.7^{95.9}_{95.7}$	$95.7^{95.8}_{92.6}$	$81.3^{82.3}_{67.1}$	$86.5^{87.8}_{80.2}$	$15.2^{55.0}_{04.2}$	$74.4^{78.6}_{70.0}$	$07.3^{22.4}_{06.3}$
IWCDAN	$\mathbf{98.0}^{98.1}_{97.9}$	$\mathbf{96.6}^{96.9}_{96.4}$	$\mathbf{96.7}^{97.5}_{93.3}$	$85.1^{93.9}_{65.3}$	$91.3^{93.8}_{90.5}$	$66.5^{74.5}_{64.1}$	$77.5^{82.3}_{77.3}$	$22.2^{45.4}_{02.7}$
sDANN-4	$87.4^{95.7}_{87.2}$	$05.6^{90.0}_{05.6}$	$94.9^{94.9}_{94.7}$	$\mathbf{85.7}^{87.7}_{84.4}$	$86.8^{89.1}_{85.5}$	$21.6^{50.3}_{15.4}$	$81.5^{83.1}_{81.3}$	$39.3^{56.2}_{37.9}$
ASA-sq	$93.7^{93.9}_{93.3}$	$89.2^{89.4}_{88.4}$	$92.3^{93.6}_{91.5}$	$83.5^{88.7}_{80.8}$	$90.9^{92.1}_{89.6}$	$69.9^{82.0}_{66.6}$	$87.2^{89.3}_{85.8}$	$62.5^{69.3}_{46.4}$
ASA-abs	$94.1^{94.5}_{93.8}$	$88.9^{91.2}_{87.0}$	$92.8^{93.2}_{89.3}$	$78.9^{82.9}_{65.1}$	$\mathbf{92.5}^{92.9}_{90.9}$	$\mathbf{82.4}^{85.4}_{74.5}$	$\mathbf{90.4}^{90.7}_{89.2}$	$\mathbf{68.4}^{73.0}_{67.5}$

Table 2: Results on STL→CIFAR. Same setup and reporting metrics as Table 1.

	$\alpha = 0.0$		$\alpha = 1.0$		$\alpha = 1.5$		$\alpha = 2.0$	
Algorithm	average	min	average	min	average	min	average	min
No DA	$69.9^{70.0}_{69.8}$	$49.8^{50.6}_{45.3}$	$68.8^{69.3}_{68.3}$	$47.2^{48.2}_{45.3}$	$66.8^{67.2}_{66.4}$	$46.0^{47.0}_{45.8}$	$65.8^{66.7}_{64.8}$	$43.7^{44.6}_{41.6}$
DANN	$75.3^{75.4}_{74.9}$	$54.6^{56.6}_{54.2}$	$69.9^{70.1}_{68.6}$	$44.8^{45.1}_{40.7}$	$64.9^{67.1}_{63.7}$	$34.9^{36.8}_{33.9}$	$63.3^{64.8}_{57.4}$	$27.0^{28.5}_{21.2}$
VADA	$\mathbf{76.7}^{76.7}_{76.6}$	$\mathbf{56.9}^{58.3}_{53.5}$	$70.6^{71.0}_{70.0}$	$47.7^{48.8}_{44.0}$	$66.1^{66.5}_{65.4}$	$35.7^{39.3}_{33.3}$	$63.2^{64.7}_{60.2}$	$25.5^{28.0}_{25.2}$
IWDAN	$69.9^{70.7}_{69.9}$	$50.5^{50.6}_{47.9}$	$68.7^{69.1}_{68.6}$	$45.8^{50.5}_{44.8}$	$67.1^{67.3}_{65.9}$	$44.7^{44.8}_{40.4}$	$64.4^{64.9}_{63.6}$	$36.8^{37.9}_{34.5}$
IWCDAN	$70.1^{70.2}_{70.1}$	$47.8^{49.3}_{42.4}$	$69.4^{69.4}_{69.1}$	$47.1^{51.3}_{46.3}$	$66.1^{67.2}_{65.0}$	$39.9^{40.8}_{37.7}$	$64.5^{65.1}_{63.9}$	$37.0^{40.2}_{35.5}$
sDANN-4	$71.8^{72.1}_{71.7}$	$52.1^{52.8}_{52.1}$	$\mathbf{71.1}^{71.7}_{70.4}$	$49.9^{51.8}_{48.1}$	$69.4^{70.0}_{68.7}$	$\mathbf{48.6}^{49.0}_{43.5}$	$66.4^{67.9}_{66.2}$	$39.0^{47.1}_{33.6}$
ASA-sq	$71.7^{71.9}_{71.7}$	$52.9^{53.4}_{46.7}$	$70.7^{71.0}_{70.4}$	$\mathbf{51.6}^{52.7}_{46.8}$	$69.2^{69.3}_{69.2}$	$45.6^{52.0}_{43.3}$	$\mathbf{68.1}^{68.2}_{67.2}$	$\mathbf{44.7}^{45.9}_{39.8}$
ASA-abs	$71.6^{71.7}_{71.2}$	$49.0^{53.5}_{48.4}$	$70.9^{71.0}_{70.8}$	$49.2^{50.0}_{47.3}$	$\mathbf{69.6}^{69.9}_{69.6}$	$43.2^{49.5}_{42.1}$	$67.8^{68.2}_{66.6}$	$40.9^{49.0}_{35.4}$

support alignment. DANN enforces distribution alignment, and thus places some target embeddings into regions corresponding to the wrong class. In comparison, ASA does not enforce distribution alignment and maintains good class correspondence across the source and target embeddings.

Main results. The results of the main experimental evaluations are shown in Tables 1, 2, 3. Without any alignment, source only training struggles relatively to adapt to the target domain. Nonetheless, its performance across the imbalance levels remains robust, since the training procedure is the same. Agreeing with the observation and theoretical results from previous work (Zhao et al., 2019; Li et al., 2020; Tan et al., 2020; Wu et al., 2019b; Tachet des Combes et al., 2020), distribution alignment methods (DANN and VADA) perform well when there is no shift but suffer otherwise, whereas relaxed distribution alignment methods (IWDAN, IWCDAN and sDANN-β) show more resilience to shifts. On all tasks with positive α, we observe that it is common for the existing methods to achieve good class average accuracies while suffering significantly on some individual classes. These results suggest that the often-ignored but important min-accuracy metric can be very challenging. Finally, our support alignment methods (ASA-sq and ASA-abs) are the most robust ones against the shifts, while still being competitive in the more balanced settings ($\alpha = 0$ or 1). We achieve best results in the more imbalanced and difficult tasks ($\alpha = 1.5$ or 2) for almost all categories on all datasets. Please refer to Appendix D for ablation studies and comparisons with more baselines.

6 Related work

Distribution alignment. Apart from the works, e.g. (Ajakan et al., 2014; Ganin et al., 2016; Ganin & Lempitsky, 2015; Pei et al., 2018; Zhao et al., 2018; Long et al., 2018; Tachet des Combes et al., 2020; Li et al., 2018b; Tzeng et al., 2017; Shen et al., 2018; Kumar et al., 2018; Li et al., 2018a;

Table 3: Results on VisDA17. Same setup and reporting metrics as Table 1.

	$\alpha = 0.0$		$\alpha = 1.0$		$\alpha = 1.5$		$\alpha = 2.0$	
Algorithm	average	min	average	min	average	min	average	min
No DA	$49.5^{50.5}_{49.4}$	$22.2^{24.6}_{22.2}$	$50.2^{50.8}_{49.2}$	$21.2^{21.3}_{20.7}$	$47.1^{47.6}_{46.6}$	$18.6^{22.2}_{18.6}$	$45.3^{46.5}_{45.2}$	$19.5^{19.8}_{14.4}$
DANN	$\mathbf{75.4}^{76.2}_{74.4}$	$36.7^{40.9}_{35.6}$	$64.1^{65.3}_{62.8}$	$25.0^{29.3}_{24.8}$	$52.1^{52.3}_{51.4}$	$11.5^{12.4}_{11.4}$	$43.1^{44.3}_{39.1}$	$03.6^{14.3}_{03.6}$
VADA	$75.3^{76.0}_{74.8}$	$40.5^{41.8}_{39.7}$	$64.6^{65.1}_{61.2}$	$22.8^{28.2}_{21.7}$	$53.0^{54.2}_{51.6}$	$14.8^{21.7}_{13.7}$	$43.9^{44.7}_{40.9}$	$08.5^{11.1}_{05.0}$
IWDAN	$73.2^{73.3}_{72.9}$	$31.7^{34.8}_{22.8}$	$64.4^{64.6}_{61.1}$	$12.1^{24.7}_{05.0}$	$51.3^{56.6}_{51.0}$	$04.6^{10.4}_{02.1}$	$45.1^{48.0}_{41.7}$	$04.6^{13.6}_{01.2}$
IWCDAN	$71.6^{75.2}_{70.6}$	$27.6^{28.0}_{22.8}$	$60.6^{61.0}_{60.2}$	$01.1^{11.3}_{00.7}$	$49.7^{51.9}_{45.6}$	$02.2^{05.7}_{00.2}$	$38.3^{46.2}_{37.3}$	$00.6^{01.7}_{00.3}$
sDANN-4	$72.4^{73.3}_{71.8}$	$37.8^{40.8}_{32.3}$	$\mathbf{68.4}^{68.7}_{66.2}$	$26.6^{29.4}_{26.2}$	$57.2^{57.8}_{56.8}$	$18.6^{23.9}_{16.7}$	$50.7^{51.7}_{49.8}$	$18.6^{20.0}_{17.1}$
ASA-sq	$64.9^{65.0}_{63.7}$	$35.7^{35.8}_{32.1}$	$61.8^{63.2}_{60.6}$	$\mathbf{31.4}^{34.4}_{20.4}$	$\mathbf{57.8}^{58.3}_{55.5}$	$\mathbf{26.7}^{32.1}_{17.3}$	$51.9^{52.0}_{50.8}$	$18.3^{21.2}_{16.9}$
ASA-abs	$64.8^{65.0}_{64.5}$	$\mathbf{40.6}^{41.9}_{36.0}$	$62.0^{62.3}_{60.5}$	$27.3^{29.7}_{16.7}$	$57.1^{58.4}_{56.2}$	$26.0^{31.2}_{13.9}$	$\mathbf{52.5}^{56.6}_{51.9}$	$\mathbf{19.7}^{22.2}_{17.7}$

Wang et al., 2021; Goodfellow et al., 2014; Arjovsky et al., 2017; Gulrajani et al., 2017; Mao et al., 2017; Radford et al., 2015; Salimans et al., 2018; Genevay et al., 2018; Wu et al., 2019a; Deshpande et al., 2018; 2019), that do distribution alignment, there are also papers (Long et al., 2015; 2017; Peng et al., 2019; Sun et al., 2016; Sun & Saenko, 2016) focusing on aligning some characteristics of the distribution, such as first or second moment. Our work is concerned with a different problem, support alignment, which is a novel objective in this line of work. In terms of methodology, our use of the discriminator output space to work with easier optimization in 1D is inspired by a line of work (Salimans et al., 2018; Genevay et al., 2018; Wu et al., 2019a; Deshpande et al., 2018; 2019) on sliced Wasserstein distance based models. Our result in Proposition 4.3 also provides theoretical insight on the practical effectiveness of 1D OT in (Deshpande et al., 2019).

Relaxed distribution alignment. In Section 4, we have already covered in detail the connections between our work and (Wu et al., 2019b). Balaji et al. (2020) introduced a version of relaxed distribution alignment, but their focus is to align distributions while being robust to outliers, which is a different focus of relaxation. Chamfer distance/divergence (CD) is used to compute similarity between images/3D point clouds (Fan et al., 2017; Nguyen et al., 2021). In the context of computing distances between documents, Kusner et al. (2015) presented Relaxed Word Mover's Distance (RWMD) for pruning candidates of similar documents. CD and RWMD are essentially the same as (5) with $d(\cdot, \cdot)$ being the Euclidean distance. They are computed by finding the nearest neighbor in the other set. Our subroutine of calculating the support distance in the 1D discriminator output space is done similarly by finding nearest neighbor assignment within the current batch and history buffers.

Support estimation. There exists a series of work, e.g. (Schölkopf et al., 2001; Hoffmann, 2007; Tax & Duin, 2004; Knorr et al., 2000; Chalapathy et al., 2017; Ruff et al., 2018; Perera et al., 2019; Deecke et al., 2018; Zenati et al., 2018), on novelty/anomaly detection problem, which can be casted as support estimation. We consider a fundamentally different problem setting. Our goal is to align the supports and our approach does not directly estimate the supports. Instead, we implicitly learn the relationships between supports (density ratio to be specific) via a discriminator.

7 Conclusion and Future Work

In this paper, we studied the problem of aligning the supports of distributions. We formalized its theoretical connections with existing alignment notions and demonstrated the effectiveness of the approach in domain adaptation. We believe that our methodology opens possibilities for the design of more nuanced and structured alignment constraints, suitable for various use cases. One natural extension is support containment, achievable with only one term in (1). This approach is fitting for partial domain adaptation, where some source domain classes do not appear in the target domain. Another interesting direction is unsupervised domain transfer, where support alignment is more desired than existing distribution alignment methods due to mode imbalance (Binkowski et al., 2019).

REFERENCES

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014. 1, 8

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017. 3, 7, 9

Yogesh Balaji, Rama Chellappa, and Soheil Feizi. Robust optimal transport with applications in generative modeling and domain adaptation. *Advances in Neural Information Processing Systems Foundation (NeurIPS)*, 2020. 9

Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007. 1

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

Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997. 19

Mikolaj Binkowski, Devon Hjelm, and Aaron Courville. Batch weight for domain adaptation with mass shift. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1844–1853, 2019. 9

Nicolas Bonneel and David Coeurjolly. Spot: sliced partial optimal transport. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019. 6

Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. Implementing random assignments: A generalization of the birkhoff-von neumann theorem. In *2009 Cowles Summer Conference*, 2009. 19

Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 36–51. Springer, 2017. 9

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011. 20

Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image anomaly detection with generative adversarial networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pp. 3–17. Springer, 2018. 9

Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative modeling using the sliced wasserstein distance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3483–3491, 2018. 6, 9, 18

Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G Schwing. Max-sliced wasserstein distance and its use for gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10648–10656, 2019. 6, 9, 18, 21

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017. 9

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015. 1, 8

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 1, 7, 8

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617. PMLR, 2018. 9

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 3, 4, 9

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5769–5779, 2017. 3, 9, 21

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 23

Heiko Hoffmann. Kernel pca for novelty detection. *Pattern recognition*, 40(3):863–874, 2007. 9

Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994. 20

Fredrik D Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 527–536. PMLR, 2019. 1

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 20

Edwin M Knorr, Raymond T Ng, and Vladimir Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8(3):237–253, 2000. 9

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 20

Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, Bill Freeman, and Gregory Wornell. Co-regularized alignment for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 31:9345–9356, 2018. 1, 8

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pp. 957–966. PMLR, 2015. 9

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 20

Bo Li, Yezhen Wang, Tong Che, Shanghang Zhang, Sicheng Zhao, Pengfei Xu, Wei Zhou, Yoshua Bengio, and Kurt Keutzer. Rethinking distributional matching based domain adaptation. *arXiv preprint arXiv:2006.13352*, 2020. 1, 8

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018a. 1, 8

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 624–639, 2018b. 8

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015. 9

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pp. 2208–2217. PMLR, 2017. 9

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 1640–1650, 2018. 8

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2017. 9

Trung Nguyen, Quang-Hieu Pham, Tam Le, Tung Pham, Nhat Ho, and Binh-Son Hua. Point-set distances for learning representations of 3d point clouds. *arXiv preprint arXiv:2102.04014*, 2021. 9

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 271–279, 2016. 7

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1, 8

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

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1406–1415, 2019. 9

Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2898–2906, 2019. 9

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 19

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 435–446. Springer, 2011. 6, 18

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 9

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402. PMLR, 2018. 9

Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving GANs using optimal transport. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkQkBnJAb. 9

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001. 9

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1, 8

Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1q-TM-AW. 7, 20, 22

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016. 9

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. 9

Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoffrey J Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 7, 8, 20

Shuhan Tan, Xingchao Peng, and Kate Saenko. Class-imbalanced domain adaptation: An empirical odyssey. In *European Conference on Computer Vision*, pp. 585–602. Springer, 2020. 1, 8

David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004. 9

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017. 1, 8

Jing Wang, Jiahong Chen, Jianzhe Lin, Leonid Sigal, and Clarence W de Silva. Discriminative feature alignment: Improving transferability of unsupervised domain adaptation by gaussian-guided latent alignment. *Pattern Recognition*, 116:107943, 2021. 1, 9

Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019a. 9, 18

Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International Conference on Machine Learning*, pp. 6872–6881. PMLR, 2019b. 1, 5, 6, 7, 8, 9, 21

Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat, and Vijay Chandrasekhar. Adversarially learned anomaly detection. In *2018 IEEE International conference on data mining (ICDM)*, pp. 727–736. IEEE, 2018. 9

Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 8568–8579, 2018. 1, 8

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pp. 7523–7532. PMLR, 2019. 1, 8

A PROOFS OF THE THEORETICAL RESULTS

A.1 PROOF OF PROPOSITION 2.1

1) $\mathcal{D}_{\triangle}(p, q) \geq 0$ for all $p, q \in \mathcal{P}$:

Since $d(\cdot, \cdot) \geq 0$, for all p, q,

$$\mathrm{SD}(p, q) \coloneqq \mathbb{E}_{x \sim p}[d(x, \mathrm{supp}(q))] = \mathbb{E}_{x \sim p}\left[\inf_{y \sim \mathrm{supp}(q)} d(x, y)\right] \geq 0,$$

which makes $\mathcal{D}_{\triangle}(p, q) = SD(p, q) + SD(q, p) \geq 0$.

2) $\mathcal{D}_{\triangle}(p, q) = 0$ if and only if $\mathrm{supp}(p) = \mathrm{supp}(q)$:

With statement 1, $\mathcal{D}_{\triangle}(p, q) = 0$ if and only if $SD(p, q) = 0$ and $SD(q, p) = 0$.

Then,

$$SD(p, q) = 0 \implies \mathbb{E}_{x \sim p}[d(x, \mathrm{supp}(q))] = 0 \implies p\left(\{x | d(x, \mathrm{supp}(q)) > 0\}\right) = 0.$$

This is equivalent to

$$\forall x \in \mathrm{supp}(p), d(x, \mathrm{supp}(q)) = 0.$$

Thus, $\mathrm{supp}(p) \subseteq \mathrm{supp}(q)$, and similarly, $\mathrm{supp}(q) \subseteq \mathrm{supp}(p)$, which makes $\mathrm{supp}(p) = \mathrm{supp}(q)$.

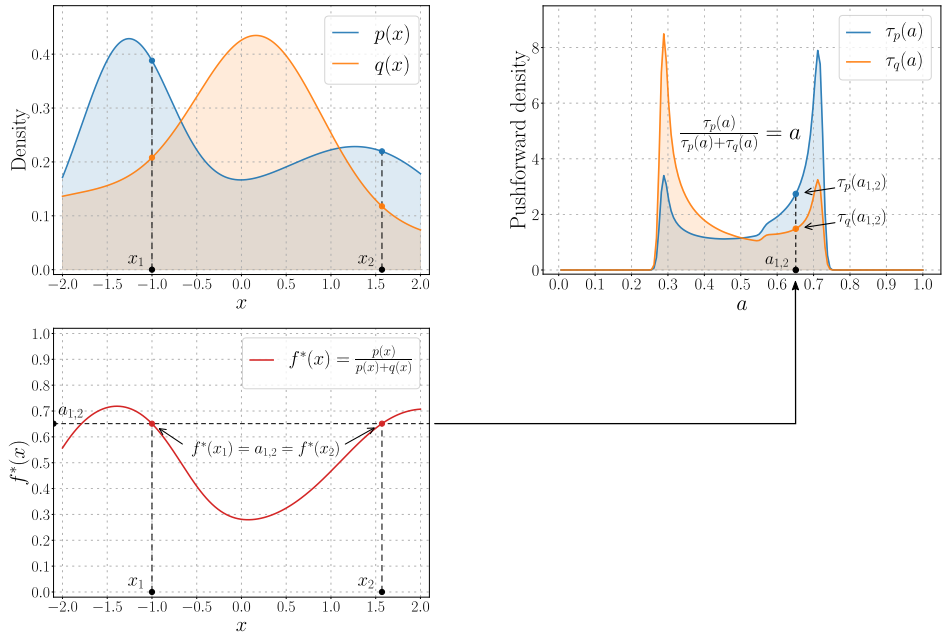


Figure A.1: Visual illustration of the statement of Proposition 2.2 (proof in A.2). Top left panel shows two example PDFs $p(x)$, $q(x)$ on closed interval $[-2, 2]$. Bottom left panel shows the optimal discriminator function $f^*(x) = p(x)/(p(x) + q(x))$ as a function of x on $[-2, 2]$. Top right panel shows the PDFs $\tau_p(a)$, $\tau_q(a)$ of the pushforward distributions $f^*{}_\sharp p$, $f^*{}_\sharp q$ induced by the discriminator mapping f^*. f^* maps $[-2, 2]$ to $[0, 1]$ and the PDFs τ_p, τ_q are defined on $[0, 1]$.
Consider point $x_1 \in [-2, 2]$. The value $f^*(x_1)$ characterizes the ratio of densities $p(x_1)/(p(x_1) + q(x_1))$ at x_1. For another point x_2 mapped to the same value $f^*(x_2) = f^*(x_1) = a_{1,2}$, the ratio of densities $p(x_2)/(p(x_2) + q(x_2))$ is the same as $p(x_1)/(p(x_1) + q(x_1))$. All points x mapped to $a_{1,2}$ share the same ratio of the densities $p(x)/(p(x) + q(x))$. This fact implies that the ratio of the pushforward densities $\tau_p(a_{1,2})/(\tau_p(a_{1,2}) + \tau_q(a_{1,2}))$ at $a_{1,2}$ must be the same as the ratio of densities $p(x_1)/(p(x_1) + q(x_1)) = a_{1,2}$ at x_1 (or x_2). The PDFs $\tau_p(a)$, $\tau_q(a)$ of the pushforward distributions satisfy property $\tau_p(a)/(\tau_p(a) + \tau_q(a)) = a$ for all $a \in \mathrm{supp}(\tau_p) \cup \mathrm{supp}(\tau_q)$.

A.2 Proof of Proposition 2.2

For any point $a \in \operatorname{supp}(\tau_p) \cup \operatorname{supp}(\tau_q)$, the values of the PDFs τ_p and τ_q can be expressed as

$$\tau_p(a) = \lim_{\varepsilon \downarrow 0} \frac{\mathbb{P}_p\left(\{x \mid a-\varepsilon < f^*(x) < a+\varepsilon\}\right)}{2\varepsilon} = \lim_{\varepsilon \downarrow 0} \frac{\int_{\{x \mid a-\varepsilon < f^*(x) < a+\varepsilon\}} p(x)\, dx}{2\varepsilon},$$
$$\tau_q(a) = \lim_{\varepsilon \downarrow 0} \frac{\mathbb{P}_q\left(\{x \mid a-\varepsilon < f^*(x) < a+\varepsilon\}\right)}{2\varepsilon} = \lim_{\varepsilon \downarrow 0} \frac{\int_{\{x \mid a-\varepsilon < f^*(x) < a+\varepsilon\}} q(x)\, dx}{2\varepsilon}.$$

Note that for all $x : a-\varepsilon < f^*(x) < a+\varepsilon$ we have

$$a-\varepsilon < \frac{p(x)}{p(x)+q(x)} < a+\varepsilon,$$

which implies

$$(a-\varepsilon)(p(x)+q(x)) < p(x) < (a+\varepsilon)(p(x)+q(x)).$$

Since these inequalities hold for all $x : a-\varepsilon < f^*(x) < a+\varepsilon$, the similar relationship holds for the integrals:

$$(a-\varepsilon) \int_{\{x \mid a-\varepsilon < f^*(x) < a+\varepsilon\}} (p(x)+q(x))\, dx < \int_{\{x \mid a-\varepsilon < f^*(x) < a+\varepsilon\}} p(x)\, dx < (a+\varepsilon) \int_{\{x \mid a-\varepsilon < f^*(x) < a+\varepsilon\}} (p(x)+q(x))\, dx.$$

The ratio $\tau_p(a)/(\tau_p(a)+\tau_q(a))$ can be expressed as

$$\frac{\tau_p(a)}{\tau_p(a)+\tau_q(a)} = \lim_{\varepsilon \downarrow 0} \frac{\int_{\{x \mid a-\varepsilon < f^*(x) < a+\varepsilon\}} p(x)\, dx}{\int_{\{x \mid a-\varepsilon < f^*(x) < a+\varepsilon\}} (p(x)+q(x))\, dx}.$$

Using the inequality above we observe that

$$a-\varepsilon < \frac{\int_{\{x \mid a-\varepsilon < f^*(x) < a+\varepsilon\}} p(x)\, dx}{\int_{\{x \mid a-\varepsilon < f^*(x) < a+\varepsilon\}} (p(x)+q(x))\, dx} < a+\varepsilon,$$

for all $\varepsilon > 0$, and taking the limit $\varepsilon \downarrow 0$ we obtain

$$\frac{\tau_p(a)}{\tau_p(a)+\tau_q(a)} = a.$$

A.3 Proof of Theorem 2.1 and Proposition 4.3

Note that by (2) and (3), we have

$$f^*(x) = \frac{p(x)}{p(x)+q(x)} = a = \frac{\tau_p(a)}{\tau_p(a)+\tau_q(a)}, \qquad \forall x \in \operatorname{supp}(p) \cup \operatorname{supp}(q).$$

1. $\Longrightarrow$: if $\mathcal{D}_W(p,q) = 0$ then $p = q$, then ${f^*}_\sharp p = {f^*}_\sharp q$. Thus, $\mathcal{D}_W({f^*}_\sharp p, {f^*}_\sharp q) = 0$.
   $\Longleftarrow$: if $\mathcal{D}_W({f^*}_\sharp p, {f^*}_\sharp q) = 0$, then ${f^*}_\sharp p = {f^*}_\sharp q$, which implies $\tau_p(a) = \tau_q(a), \forall a$. Then, we have
   $$p(x) = q(x), \qquad \forall x \in \operatorname{supp}(p) \cup \operatorname{supp}(q).$$
   Thus, $p = q$ and $\mathcal{D}_W(p,q) = 0$.

2. $\Longrightarrow$: if $\mathcal{D}_W^{\beta_1,\beta_2}(p,q)=0$, (13) holds, then

$$\frac{1}{2+\beta_2} \le f^*(x) \le \frac{1+\beta_1}{2+\beta_1}, \qquad \forall x \in \mathrm{supp}(p) \cup \mathrm{supp}(q).$$

This implies that

$$\frac{1}{2+\beta_2} \le \frac{\tau_p(a)}{\tau_p(a)+\tau_q(a)} \le \frac{1+\beta_1}{2+\beta_1}, \qquad \forall a \in \mathrm{supp}(\tau_p) \cup \mathrm{supp}(\tau_q).$$

Thus, we have

$$\frac{1}{1+\beta_2} \le \frac{\tau_p(a)}{\tau_q(a)} \le 1+\beta_1, \qquad \forall a \in \mathrm{supp}(\tau_p) \cup \mathrm{supp}(\tau_q).$$

This leads to $\mathcal{D}_W^{\beta_1,\beta_2}({f^*}_\sharp p, {f^*}_\sharp q)=0$.

$\Longleftarrow$: similarly, when $\mathcal{D}_W^{\beta_1,\beta_2}({f^*}_\sharp p, {f^*}_\sharp q)=0$,

$$\frac{1}{1+\beta_2} \le \frac{\tau_p(a)}{\tau_q(a)} \le 1+\beta_1, \qquad \forall a \in \mathrm{supp}(\tau_p) \cup \mathrm{supp}(\tau_q)$$

$$\Downarrow$$

$$\frac{1}{2+\beta_2} \le f^*(x) \le \frac{1+\beta_1}{2+\beta_1}, \qquad \forall x \in \mathrm{supp}(p) \cup \mathrm{supp}(q)$$

Thus, (13) holds and $\mathcal{D}_W^{\beta_1,\beta_2}(p,q)=0$.

3. Note that $f^*(x)=1$ if and only if $p(x)>0$, $q(x)=0$, and $f^*(x)=0$ if and only if $q(x)>0, p(x)=0$.
We also have $\tau_p(0)=0$ if $0 \in \mathrm{supp}(\tau_p)$ and $\tau_q(1)=0$ if $1 \in \mathrm{supp}(\tau_q)$.
Thus, $f^*(x)=1$ if and only if $\mathrm{supp}(p) \setminus \mathrm{supp}(q) \neq \varnothing$ and $x \in \mathrm{supp}(p) \setminus \mathrm{supp}(q)$, in which case $\tau_p(1)>0$, which makes $\mathrm{supp}(\tau_p) \setminus \mathrm{supp}(\tau_q) \neq \varnothing$.
Similarly, $f^*(x)=0$ if and only if $\mathrm{supp}(q) \setminus \mathrm{supp}(p) \neq \varnothing$ and $x \in \mathrm{supp}(q) \setminus \mathrm{supp}(p)$, in which case $\tau_q(0)>0$, which makes $\mathrm{supp}(\tau_q) \setminus \mathrm{supp}(\tau_p) \neq \varnothing$.
Then, we have $\mathrm{supp}(p) \neq \mathrm{supp}(q)$ if and only if $\mathrm{supp}(\tau_p) \neq \mathrm{supp}(\tau_q)$.
This is equivalent to $\mathcal{D}_\triangle(p,q)=0$ if and only if $\mathcal{D}_\triangle({f^*}_\sharp p, {f^*}_\sharp q)=0$.

A.4 Proof of Proposition 2.3

Consider a 1-dimensional Euclidean space $\mathbb{R}$. Let $\mathrm{supp}(p)=[-\frac{1}{2},\frac{1}{2}] \cup [1,2]$ with $p([-\frac{1}{2},\frac{1}{2}])=\frac{3}{4}$ and $p([1,2])=\frac{1}{4}$. Let $\mathrm{supp}(q)=[-2,-1] \cup [-\frac{1}{2},\frac{1}{2}] \cup [1,2]$ with $q([-2,-1])=\frac{1}{4}, q([-\frac{1}{2},\frac{1}{2}])=\frac{1}{4}$ and $q([1,2])=\frac{1}{2}$. The supports of p and q consist of disjoint closed intervals, and we assume uniform distribution within each of these intervals, i.e. p has density $p(x)=\frac{3}{4}, \forall x \in [-\frac{1}{2},\frac{1}{2}]; p(x)=\frac{1}{4}, \forall x \in [1,2]$ and q has densitiy $q(x)=\frac{1}{4}, \forall x \in [-2,-1]; q(x)=\frac{1}{4}, \forall x \in [-\frac{1}{2},\frac{1}{2}]; q(x)=\frac{1}{2}, \forall x \in [1,2]$. Clearly, $\mathrm{supp}(p) \neq \mathrm{supp}(q)$.

The optimal dual Wasserstein discriminator f_W^* is the maximizer of

$$\sup_{f:\mathrm{Lip}(f)\le 1} \mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{y\sim q}[f(y)].$$

Thus, f_W^* is the maximizer of

$$\sup_{f:\mathrm{Lip}(f)\le 1} \frac{1}{4}\left(3\int_{-\frac{1}{2}}^{\frac{1}{2}} f(x)dx + \int_1^2 f(x)dx - \int_{-2}^{-1} f(x)dx - \int_{-\frac{1}{2}}^{\frac{1}{2}} f(x)dx - 2\int_1^2 f(x)dx\right),$$

which simplifies to

$$\sup_{f:\mathrm{Lip}(f)\le 1} \frac{1}{4}\left(-\int_{-2}^{-1} f(x)dx + 2\int_{-\frac{1}{2}}^{\frac{1}{2}} f(x)dx - \int_1^2 f(x)dx\right).$$

Since the optimization objective and the constraint are invariant to replacing the function $f(x)$ with its symmetric reflection $g(x) = f(-x)$, if f' is a optimal solution, then there exists a symmetric maximizer $f^*_W(x) = \frac{1}{2}f'(x) + \frac{1}{2}f'(-x)$, since $f^*_W(-1) = f^*_W(1)$ and $\mathrm{Lip}(f^*_W) \leq \mathrm{Lip}(f') \leq 1$. Thus, $\mathrm{supp}(f^\star_{W\sharp}p) = \mathrm{supp}(f^\star_{W\sharp}q)$.

Note that one can easily "extend" the above proof to discrete distributions that we encounter in practice, by replacing each of the disjoint segments with points $-1, 0, 1$.

A.5 Proof of Proposition 4.1

From (8), we have

$$\mathcal{D}^\infty_W(p,q) \coloneqq \lim_{\beta\to\infty} \mathcal{D}^\beta_W(p,q) = \lim_{\beta\to\infty} \inf_{\gamma\in\Gamma_\beta(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[d(x,y)],$$

where $\lim_{\beta\to\infty}\Gamma_\beta(p,q)$ is the set of all measures γ on $\mathcal{X}\times\mathcal{X}$ such that $\int\gamma(x,y)dy = p(x), \forall x$ and $\int\gamma(x,y)dx \leq \lim_{\beta\to\infty}(1+\beta)q(y), \forall y$.

The set of inequalities

$$\int \gamma(x,y)dx \leq \lim_{\beta\to\infty}(1+\beta)q(y), \qquad \forall y$$

can be simplified to

$$\int \gamma(x,y)dx = 0, \qquad \forall y \text{ such that } q(y) = 0.$$

To put it together, we have

$$\mathcal{D}^\infty_W(p,q) = \inf_{\gamma\in\Gamma'(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[d(x,y)],$$

where $\Gamma'(p,q)$ is the set of all measures γ on $\mathcal{X}\times\mathcal{X}$ such that $\int\gamma(x,y)dy = p(x), \forall x$ and $\int\gamma(x,y)dx = 0, \forall y$ such that $q(y) = 0$.

Note that

$$\mathbb{E}_{(x,y)\sim\gamma}[d(x,y)] = \int \gamma(x,y)d(x,y)dxdy.$$

Then, we can minimize $\mathbb{E}_{(x,y)\sim\gamma}[d(x,y)]$ pointwise w.r.t. x under $\gamma\in\Gamma'(p,q)$.

With any fixed x, we want to minimize

$$\int q_x(y)d(x,y)dy$$

where $\int q_x(y)dy = p(x)$ and $q_x(y) = 0$ when $q(y) = 0$, which implies $q_x(y) = 0$ if $y \notin \mathrm{supp}(q)$.

Then clearly $q^*_x(y)$ which is equal to $p(x)$ for some y such that $d(x,y) = \inf_{y\in\mathrm{supp}(q)} d(x,y)$ and 0 otherwise is a minimizer.

Then, we arrive at

$$\inf_{\gamma\in\Gamma'(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[d(x,y)] = \mathbb{E}_{(x,y)\sim\gamma^*}[d(x,y)],$$

where $\gamma^*(x,y) = p(x)$, for all $x\in\mathrm{supp}(p)$ and some y such that $(x,y) = \inf_{y\in\mathrm{supp}(q)} d(x,y)$. Otherwise, $\gamma^*(x,y) = 0$.

Thus,

$$\mathbb{E}_{(x,y)\sim\gamma^*}[d(x,y)] = \mathbb{E}_{x\sim p}\left[\inf_{y\sim\mathrm{supp}(q)} d(x,y)\right].$$

The last equation implies $\mathcal{D}^\infty_W(p,q) = \mathrm{SD}(p,q)$.

Then obviously, $\mathcal{D}^{\infty,\infty}_W(p,q) \coloneqq \lim_{\beta_1,\beta_2\to\infty} \mathcal{D}^{\beta_1,\beta_2}_W(p,q) = \mathcal{D}_\triangle(p,q)$.

A.6 Proof of Proposition 4.2

1. $\mathcal{D}_W(p,q)=0$ implies $p=q$, which is equivalent to
$$\frac{p(x)}{q(x)}=1, \qquad \forall x\in \mathrm{supp}(p)\cup \mathrm{supp}(q).$$
Then clearly, for all finite $\beta_1,\beta_2>0$ it satisfies
$$\frac{1}{1+\beta_2}\le \frac{p(x)}{q(x)}\le 1+\beta_1, \qquad \forall x\in \mathrm{supp}(p)\cup \mathrm{supp}(q). \tag{13}$$
Thus, $\mathcal{D}_W^{\beta_1,\beta_2}(p,q)=0$ for all finite $\beta_1,\beta_2>0$.
2. $\mathcal{D}_W^{\beta_1,\beta_2}(p,q)=0$ for some finite $\beta_1,\beta_2>0$ is equivalent to for some finite $\beta_1,\beta_2>0$, (13) is satisfied.
This implies that $\forall x\in \mathrm{supp}(p)$, $x\in \mathrm{supp}(q)$ and $\forall x\in \mathrm{supp}(q)$, $x\in \mathrm{supp}(p)$, which makes $\mathrm{supp}(p)=\mathrm{supp}(q)$. Thus, $\mathcal{D}_\triangle(p,q)=0$.
3. The converse of statements 1 and 2 are false:
   (a) For all finite $\beta_1,\beta_2>0$, let $\mathrm{supp}(p)=\mathrm{supp}(q)=\{x_1,x_2\}$. Let $p(x_1)=p(x_2)=1/2$ and $q(x_1)=(1+\beta')/2$ and $q(x_2)=(1-\beta')/2$ where
$$\beta'=\min\left(\beta_2, 1-\frac{1}{1+\beta_1}\right).$$
   Then, it can be easily checked that (13) is satisfied, which makes $\mathcal{D}_W^{\beta_1,\beta_2}(p,q)=0$. However, since $\beta'\neq 0$, $p\neq q$ and thus $\mathcal{D}_W(p,q)\neq 0$.
   (b) Similar to (a), let $\mathrm{supp}(p)=\mathrm{supp}(q)=\{x_1,x_2\}$. Let $p(x_1)=q(x_2)=\varepsilon$ and $p(x_2)=q(x_1)=1-\varepsilon$ for some $\varepsilon>0$. Since $\mathrm{supp}(p)=\mathrm{supp}(q)$, $\mathcal{D}_\triangle(p,q)=0$. However, for any finite $\beta>0$,
$$\lim_{\varepsilon\downarrow 0}\frac{p(x_1)}{q(x_1)}=\lim_{\varepsilon\downarrow 0}\frac{\varepsilon}{1-\varepsilon}=0<\frac{1}{1+\beta}.$$
   Thus, $\mathcal{D}_W^{\beta_1,\beta_2}(p,q)\neq 0$ for any finite $\beta_1,\beta_2>0$.

B A comment on "sliced" SSD divergence

Recent works (Deshpande et al., 2018; 2019) have proposed to perform optimal transport (OT)-based distribution alignment by reducing the OT problem (7) in the original, potentially high-dimensional space, to that between 1D distributions. Specifically, Deshpande et al. (2018) consider the sliced Wasserstein distance (Rabin et al., 2011):
$$\mathcal{D}_{SW}(p,q)=\int_{\mathbb{S}^{n-1}}\mathcal{D}_W({f^\theta}_\sharp p, {f^\theta}_\sharp q)\,d\theta, \tag{14}$$
where $\mathbb{S}^{n-1}=\{\theta\in\mathbb{R}^n \mid \|\theta\|=1\}$ is a unit sphere in $\mathbb{R}^n$, and f^θ is a 1D linear projection $f^\theta(x)=\langle\theta,x\rangle$. It is known that $\mathcal{D}_{SW}$ is a valid distribution divergence: for any $p\neq q$ there exists a linear slicing function f^θ , $\theta\in\mathbb{S}^{n-1}$ which identifies difference in the distributions, i.e. ${f^\theta}_\sharp p\neq {f^\theta}_\sharp q$ (Cramér–Wold theorem). By reducing the original OT problem to that in a 1D space, Wu et al. (2019a) and Deshpande et al. (2019) develop efficient practical methods for distribution alignment based on fast algorithms for the 1D OT problem.

Unfortunately, the straight-forward extension of SSD divergence (1) to a 1D linearly sliced version does not provide a valid support divergence.

Proposition B.1. *There exist two distributions p and q in $\mathcal{P}$, such that* $\mathrm{supp}(p)\neq\mathrm{supp}(q)$ *but* $\mathrm{supp}({f^\theta}_\sharp p)=\mathrm{supp}({f^\theta}_\sharp q)$, $\forall f^\theta(x)=\langle\theta,x\rangle$ *with* $\theta\in\mathbb{S}^{n-1}$.

Proof. Consider a 2-dimensional Euclidean space $\mathbb{R}^2$ and let $\mathrm{supp}(p)=\{(x,y)|x^2+y^2\le 2\}$ and $\mathrm{supp}(q)=\{(x,y)|1\le x^2+y^2\le 2\}$. Then, $\forall f^\theta(x)=\langle\theta,x\rangle$ with $\theta\in\mathbb{S}^1$,
$$\mathrm{supp}({f^\theta}_\sharp p)=\mathrm{supp}({f^\theta}_\sharp q)=[-2,2].$$
This counterexample is shown in Figure B.1. □

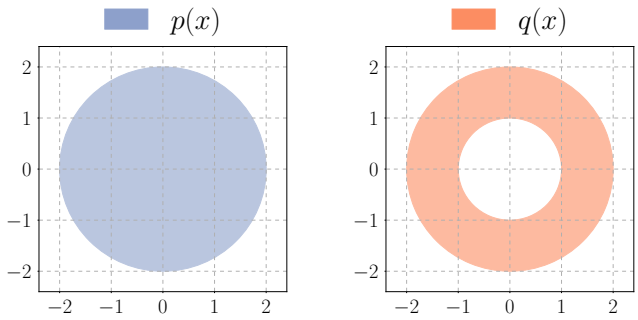


Figure B.1: Visualization of example distributions for Proposition B.1

C Discussion of "soft" and "hard" assignments with 1D discrete distributions

In Section 4.2 we considered the "soft-assignment" relaxed OT problem (11) and claimed that for integer β, the set of minimizers of (11) must contain a "hard-assignment" transportation plan, meaning $\gamma_{ij} \in \{0, 1\}, \forall i, j$. Below we justify this claim.

Note that for $\beta = 0$ the OT problem (11) is the standard OT problem for Wasserstein-1 distance (10), since the inequality constraints $\sum_{i=1}^{m} \gamma_{ij} \leq 1, \forall j$ can only be satisfied as equalities. For this problem, it is known (e.g. see Peyré et al. (2019) Proposition 2.1) that the set of optimal "soft-assignment" contains a "hard-assignment" represented by a normalized permutation matrix. This fact can be proven using the Birkhoff–von Neumann theorem. The Birkhoff–von Neumann theorem states that the set of doubly stochastic matrices

$$P \in \mathbb{R}^{n \times n}: \qquad P_{ij} \geq 0, \forall\, i, j, \qquad \sum_{j=1}^{n} P_{ij} = 1, \forall\, i, \qquad \sum_{i=1}^{n} P_{ij} = 1, \forall\, j$$

is exactly the set of all finite convex combinations of permutation matrices. In the context of the linear program (11) with $\beta = 0$, the Birkhoff–von Neumann theorem means that all extreme points of the polyhedron $\Gamma_\beta(o^p, o^q)$ are hard-assignment matrices. Therefore, by the fundamental theorem of linear programming (Bertsimas & Tsitsiklis, 1997), the minimum of the objective is reached at a "hard-assignment" matrix.

We argue that a similar result holds for the case of integer $\beta > 0$. In this case, the matrices in $\Gamma_\beta(o^p, o^q)$ can not be associated with the doubly stochastic matrices, since constraints on of the marginals of γ are relaxed to inequality constraints. Because of that, the Birkhoff–von Neumann theorem can not be applied. However, Budish et al. (2009) provide a generalization of the Birkhoff–von Neumann theorem (Theorem 1 in (Budish et al., 2009)) which applies to the cases where the equality constraints are replaced with integer-valued inequality constraints (recall that we consider integer β). Using this generalized result, our claim can be proven by performing the following steps.

Clearly, the polyhedron $\Gamma_\beta(o^p, o^q)$ contains all "hard-assignment" matrices and all their finite convex combinations. The result proven in (Budish et al., 2009) implies that each element of $\Gamma_\beta(o^p, o^q)$ can be represented as a finite convex combination of "hard-assignment" matrices. Thus, the polyhedron $\Gamma_\beta(o^p, o^q)$ is exactly the set of all finite convex combinations of "hard-assignment" matrices and all extreme points of the polyhedron are "hard-assignment" matrices. Finally, by analogy with the case of $\beta = 0$, we invoke the fundamental theorem of the linear programming and conclude that the minimum of the objective (11) is reached at γ corresponding to a "hard-assignment" matrix.

D EXPERIMENT DETAILS

D.1 USPS TO MNIST EXPERIMENT SPECIFICATIONS

We use USPS (Hull, 1994) and MNIST (LeCun et al., 1998) datasets for this adaptation problem.

Following Tachet des Combes et al. (2020) we use LeNet-like (LeCun et al., 1998) architecture for the feature extractor with the 500-dimensional feature representation. The classifier consists of a single linear layer. The discriminator is implemented by a 3-layer MLP with 512 hidden units and leaky-ReLU activation.

We train all methods for $65\,000$ steps with batch size 64. We train the feature extractor, the classifier, and the discriminator with SGD (learning rate 0.02, momentum 0.9, weight decay $5 \cdot 10^{-4}$). We perform a single discriminator update per 1 update of the feature extractor and the classifier. After the first $30\,000$ steps we linearly anneal the feature extractor's and clssifier's learning rate for $30\,000$ to the final value $2 \cdot 10^{-5}$.

The feature extractor's loss is given by a weighted combination of the cross-entropy classification loss on the labeled source example and a domain alignment loss computed from the discriminator's signal (recall that different method use different forms of the alignment loss). The weight for the classification term is constant and set to $\lambda_{\text{cls}} = 1$. We introduce schedule for the alignment weight λ_{align}. For all alignment methods we linearly increase λ_{align} from 0 to 1.0 during the first 10000 steps.

For ASA we use history buffers of size 1000.

D.2 STL TO CIFAR EXPERIMENT SPECIFICATIONS

We use STL (Coates et al., 2011) and CIFAR-10 (Krizhevsky, 2009) for this adaptation task. STL and CIFAR-10 are both 10-class classification problems. There are 9 common classes between the two datasets. Following Shu et al. (2018) we create a 9-class classification problem by selecting the subsets of examples of the 9 common classes.

For the feature extractor, we adapt the deep CNN architecture of Shu et al. (2018). The feature representation is 192-dimensional vectors. The classifier consists of a single linear layer. The discriminator is implemented by a 3-layer MLP with 512 hidden units and leaky-ReLU activation.

We train all methods for $40\,000$ steps with batch size 64. We train the feature extractor, the classifier, and the discriminator with ADAM (Kingma & Ba, 2014) (learning rate 0.001, $\beta_1 = 0.5$, $\beta_2 = 0.999$, no weight decay). We perform a single discriminator update per 1 update of the feature extractor and the classifier.

The weight for the classification loss term is constant and set to $\lambda_{\text{cls}} = 1$. For all alignment methods we use constant alignment weight $\lambda_{\text{align}} = 0.1$.

For ASA we use history buffers of size 1000.

Conditional entropy loss. Following (Shu et al., 2018) we use auxiliary conditional entropy loss on target examples for domain adaptation methods. For a classifier $C^\phi : \mathcal{Z} \to \mathcal{Y}$ and a feature extractor $F^\theta : \mathcal{X} \to \mathcal{Z}$ where classifier outputs the distribution over class labels $\{1, \dots, K\}$

$$C^\phi(z) \in \mathbb{R}^K : \quad \left[C^\phi(z)\right]_k \geq 0, \quad \sum_{k=1}^{K} \left[C^\phi(z)\right]_k = 1,$$

the conditional entropy loss on target examples $\{x_i^q\}_{i=1}^{N_q}$ is given by

$$\mathcal{L}_{\text{ent}} = \lambda_{\text{ent}} \cdot \frac{1}{N^q} \sum_{i=1}^{N_q} \left(- \sum_{k=1}^{K} \left[C^\phi(F^\theta(x_i^q))\right]_k \log \left[C^\phi(F^\theta(x_i^q))\right]_k \right). \tag{15}$$

λ_{ent} is the weight of the conditional entropy loss in the total training objective. This loss acts as an additional regularization of the embeddings of the unlabeled target examples: minimization of the conditional entropy pushes target embeddings away from the classifier's decision boundary.

For all domain adapation methods we use the conditional entropy loss (15) on target examples with the weight $\lambda_{\text{ent}} = 0.1$.

D.3 EXTENDED EXPERIMENTAL RESULTS ON STL TO CIFAR

Effect of conditional entropy loss. In order to quantify the improvements of the support alignment objective and the conditional entropy objective in separation, we conduct an ablation study. In addition to the results reported in Section 5, we evaluate all domain adaptation methods (except VADA which uses the conditional entropy in the original implementation) on STL→CIFAR task without the conditional entropy loss ($\lambda_{\text{ent}} = 0$). The results of the ablation study are presented in Table D.1. We observe that the effect of the auxiliary conditional entropy is essentially the same for all methods across all imbalance levels: with $\lambda_{\text{ent}} = 0.1$ the accuracy either improves (especially the average class accuracy) or roughly stays on the same level. The relative ranking of distribution alignment, relaxed distribution alignment, and support alignment methods is the same with both $\lambda_{\text{ent}} = 0$ and $\lambda_{\text{ent}} = 0.1$. The results demonstrate that the benefits of support alignment approach and conditional entropy are orthogonal.

Table D.1: Results of ablation experiments of the effect of auxiliary conditional entropy loss on STL→CIFAR data. Same setup and reporting metrics as Table 1.

		$\alpha = 0.0$		$\alpha = 1.0$		$\alpha = 1.5$		$\alpha = 2.0$	
Algorithm	λ_{ent}	average	min	average	min	average	min	average	min
DANN	0.0	$74.6^{75.1}_{74.1}$	$51.5^{55.0}_{49.9}$	$68.4^{69.2}_{67.0}$	$43.2^{43.7}_{41.2}$	$65.7^{65.9}_{62.8}$	$35.5^{36.2}_{29.6}$	$62.5^{64.6}_{60.0}$	$27.5^{27.5}_{25.7}$
DANN	0.1	$75.3^{75.4}_{74.9}$	$54.6^{56.6}_{54.2}$	$69.9^{70.1}_{68.6}$	$44.8^{45.1}_{40.7}$	$64.9^{67.1}_{63.7}$	$34.9^{36.8}_{33.9}$	$63.3^{64.8}_{57.4}$	$27.0^{28.5}_{21.2}$
IWDAN	0.0	$70.4^{70.7}_{70.2}$	$47.2^{48.0}_{46.8}$	$68.6^{68.8}_{68.4}$	$43.6^{46.3}_{43.2}$	$66.7^{67.9}_{66.0}$	$44.7^{46.2}_{43.3}$	$63.9^{66.1}_{62.9}$	$36.5^{37.3}_{32.7}$
IWDAN	0.1	$69.9^{70.7}_{69.9}$	$50.5^{50.6}_{47.9}$	$68.7^{69.1}_{68.6}$	$45.8^{50.5}_{44.8}$	$67.1^{67.3}_{65.9}$	$44.7^{44.8}_{40.4}$	$64.4^{64.9}_{63.6}$	$36.8^{37.9}_{34.5}$
IWCDAN	0.0	$70.1^{70.8}_{70.0}$	$50.5^{50.8}_{49.1}$	$68.6^{69.4}_{68.2}$	$44.2^{45.8}_{41.2}$	$66.0^{66.0}_{65.9}$	$45.0^{47.8}_{43.7}$	$63.8^{64.1}_{62.3}$	$37.3^{37.7}_{33.6}$
IWCDAN	0.1	$70.1^{70.2}_{70.1}$	$47.8^{49.3}_{42.4}$	$69.4^{69.4}_{69.1}$	$47.1^{51.3}_{46.3}$	$66.1^{67.2}_{65.0}$	$39.9^{40.8}_{37.7}$	$64.5^{65.1}_{63.9}$	$37.0^{40.2}_{35.5}$
sDANN-4	0.0	$69.4^{70.0}_{68.8}$	$46.5^{49.7}_{45.1}$	$69.6^{69.7}_{69.3}$	$49.1^{49.2}_{47.4}$	$68.0^{68.6}_{67.8}$	$48.2^{48.8}_{42.6}$	$66.3^{66.4}_{64.2}$	$40.7^{42.9}_{36.6}$
sDANN-4	0.1	$71.8^{72.1}_{71.7}$	$52.1^{52.8}_{52.1}$	$71.1^{71.7}_{70.4}$	$49.9^{51.8}_{48.1}$	$69.4^{70.0}_{68.7}$	$48.6^{49.0}_{43.5}$	$66.4^{67.9}_{66.2}$	$39.0^{47.1}_{33.6}$
ASA-sq	0.0	$69.9^{70.3}_{69.9}$	$48.0^{50.1}_{46.6}$	$68.8^{68.9}_{68.6}$	$47.3^{49.3}_{45.3}$	$68.1^{68.7}_{67.2}$	$45.4^{47.8}_{45.2}$	$65.7^{66.4}_{65.6}$	$43.6^{45.0}_{41.3}$
ASA-sq	0.1	$71.7^{71.9}_{71.7}$	$52.9^{53.4}_{46.7}$	$70.7^{71.0}_{70.4}$	$51.6^{52.7}_{46.8}$	$69.2^{69.3}_{69.2}$	$45.6^{52.0}_{43.3}$	$68.1^{68.2}_{67.2}$	$44.7^{45.9}_{39.8}$
ASA-abs	0.0	$69.8^{70.0}_{68.9}$	$45.7^{48.0}_{45.4}$	$68.4^{68.6}_{68.4}$	$44.3^{46.8}_{44.0}$	$67.9^{68.1}_{67.0}$	$46.6^{48.4}_{40.4}$	$66.3^{66.9}_{65.7}$	$41.6^{44.9}_{40.3}$
ASA-abs	0.1	$71.6^{71.7}_{71.2}$	$49.0^{53.5}_{48.4}$	$70.9^{71.0}_{70.8}$	$49.2^{50.0}_{47.3}$	$69.6^{69.9}_{69.6}$	$43.2^{49.5}_{42.1}$	$67.8^{68.2}_{66.6}$	$40.9^{49.0}_{35.4}$

Comparison with optimal transport based baselines.

We provide additional experimental results comparing our method with OT-based methods for domain adaptation. We implement two OT-based methods which we describe below.

- The first method is a variant of the max-sliced Wasserstein distance (which was proposed for GAN training by Deshpande et al. (2019)) to domain adaptation. In the table below we refer to this method as DANN-OT. In our implementation DANN-OT minimizes the Wasserstein distance between the pushforward distributions $g^*_\sharp p$, $g^*_\sharp q$ induced by the optimal log-loss discriminator g^*. As discussed in Section 4.2 (paragraph "Distribution alignment") the computation of the Wasserstein distance between 1D distributions can be implemented efficiently via sorting.
- The second method is an OT-based variant of DANN which uses a dual Wasserstein discriminator instead of the log-loss discriminator. In the table below we refer to this method as DANN-WGP. This method minimizes the Wasserstein distance in its dual Kantorovich form. We train the discriminator with the Wasserstein dual objective and a gradient penalty proposed to enforce Lipshitz-norm constraint (Gulrajani et al., 2017).

We present the evaluation results of the OT-based methods on STL→CIFAR domain adaptation task in the Table D.2. Note that the OT-based methods aim to enforce distribution alignment constraints. We observe that the OT-based methods follow the same trend as DANN: they deliver improved accuracy compared to No DA in the balanced setting, but suffer in the imbalanced settings ($\alpha > 0$) due to their distribution alignment nature.

We would also like to make a comment on OT-based relaxed distribution alignment. Wu et al. (2019b) propose method WDANN-β which minimizes the dual form of the asymmetrically-relaxed

Table D.2: Results of comparison with optimal transport based methods on STL→CIFAR data. Same setup and reporting metrics as Table 1.

	$\alpha = 0.0$		$\alpha = 1.0$		$\alpha = 1.5$		$\alpha = 2.0$	
Algorithm	average	min	average	min	average	min	average	min
No DA	$69.9^{70.0}_{69.8}$	$49.8^{50.6}_{45.3}$	$68.8^{69.3}_{68.3}$	$47.2^{48.2}_{45.3}$	$66.8^{67.2}_{66.4}$	$46.0^{47.0}_{45.8}$	$65.8^{66.7}_{64.8}$	$43.7^{44.6}_{41.6}$
DANN	$75.3^{75.4}_{74.9}$	$54.6^{56.6}_{54.2}$	$69.9^{70.1}_{68.6}$	$44.8^{45.1}_{40.7}$	$64.9^{67.1}_{63.7}$	$34.9^{36.8}_{33.9}$	$63.3^{64.8}_{57.4}$	$27.0^{28.5}_{21.2}$
DANN-OT	$76.0^{76.0}_{75.8}$	$55.2^{55.5}_{54.3}$	$67.7^{68.9}_{67.1}$	$43.0^{43.7}_{36.5}$	$64.5^{65.1}_{60.9}$	$34.4^{34.6}_{29.3}$	$61.3^{62.0}_{54.4}$	$24.3^{25.5}_{23.2}$
DANN-WGP	$74.8^{75.1}_{74.7}$	$53.5^{54.4}_{53.3}$	$67.7^{67.9}_{65.3}$	$38.6^{41.0}_{34.4}$	$63.3^{63.4}_{57.1}$	$27.0^{32.4}_{26.3}$	$59.0^{61.8}_{54.3}$	$21.9^{22.5}_{18.6}$
sDANN-4	$71.8^{72.1}_{71.7}$	$52.1^{52.8}_{52.1}$	$71.1^{71.7}_{70.4}$	$49.9^{51.8}_{48.1}$	$69.4^{70.0}_{68.7}$	$48.6^{49.0}_{43.5}$	$66.4^{67.9}_{66.2}$	$39.0^{47.1}_{33.6}$
ASA-sq	$71.7^{71.9}_{71.7}$	$52.9^{53.4}_{46.7}$	$70.7^{71.0}_{70.4}$	$51.6^{52.7}_{46.8}$	$69.2^{69.3}_{69.2}$	$45.6^{52.0}_{43.3}$	$68.1^{68.2}_{67.2}$	$44.7^{45.9}_{39.8}$
ASA-abs	$71.6^{71.7}_{71.2}$	$49.0^{53.5}_{48.4}$	$70.9^{71.0}_{70.8}$	$49.2^{50.0}_{47.3}$	$69.6^{69.9}_{69.6}$	$43.2^{49.5}_{42.1}$	$67.8^{68.2}_{66.6}$	$40.9^{49.0}_{35.4}$

Wasserstein distance. However, they observe that sDANN-β outperforms WDANN-β in experiments. As a result, we use sDANN-β as a relaxed distribution alignment baseline in our experiments.

Effect of alignment weight. We provide additional experimental results comparing ASA with DANN and VADA across different values of the alignment loss weight λ_{align} on STL→CIFAR task. The results are shown in Table D.3. DANN with a higher alignment weight $\lambda_{\text{align}} = 1.0$ performs better in the balanced ($\alpha = 0$) setting and worse in the imbalanced ($\alpha > 0$) setting compared to a lower weight $\lambda_{\text{align}} = 0.1$, as the distribution alignment constraint is enforced stricter. VADA optimizes a combination of distribution alignment + VAT (virtual adversarial training) objectives (Shu et al., 2018), and we observe the same trend: with lower alignment weight $\lambda_{\text{align}} = 0.01$, VADA performs worse in the balanced setting and better in the imbalanced setting compared to a higher weight $\lambda_{\text{align}} = 0.1$. Weight $\lambda_{\text{align}} = 0.1$ is a middle ground between having poor performance in the imbalanced setting ($\lambda_{\text{align}} = 1.0$) and not sufficiently enforcing distribution alignment ($\lambda_{\text{align}} = 0.01$).

The role of VAT (similarly to that of conditional entropy loss) is orthogonal to alignment objectives. Thus, we provide additional evaluations of combining support alignment and VAT (the "ASA-sq + VAT" entry in Table D.3) with alignment weight $\lambda_{\text{align}} = 1.0$. The good performance of such combination shows that:

- One could improve our current support alignment performance by using auxiliary objectives.
- Support alignment based method performs qualitatively different from distribution alignment based method, since the performance holds with stricter support alignment while distribution alignment needs to loosen the constraints considerably to reduce the performance degradation in the imbalanced setting.

Table D.3: Results of comparison of ASA with DANN and VADA across different values of the alignment loss weight λ_{align} on STL→CIFAR data. Same setup and reporting metrics as Table 1.

		$\alpha = 0.0$		$\alpha = 1.0$		$\alpha = 1.5$		$\alpha = 2.0$	
Algorithm	λ_{align}	average	min	average	min	average	min	average	min
DANN	0.01	$72.3^{72.7}_{72.2}$	$49.5^{50.8}_{48.8}$	$70.6^{71.2}_{69.7}$	$48.9^{51.2}_{41.5}$	$68.5^{68.7}_{67.2}$	$46.1^{50.0}_{36.2}$	$65.9^{66.0}_{64.1}$	$36.7^{39.4}_{29.9}$
DANN	0.1	$75.3^{75.4}_{74.9}$	$54.6^{56.6}_{54.2}$	$69.9^{70.1}_{68.6}$	$44.8^{45.1}_{40.7}$	$64.9^{67.1}_{63.7}$	$34.9^{36.8}_{33.9}$	$63.3^{64.8}_{57.4}$	$27.0^{28.5}_{21.2}$
DANN	1.0	$77.2^{77.3}_{76.8}$	$58.5^{59.4}_{56.7}$	$66.3^{66.8}_{64.5}$	$37.9^{41.6}_{37.5}$	$62.8^{63.3}_{56.1}$	$27.5^{28.9}_{24.6}$	$58.7^{59.7}_{52.3}$	$18.5^{20.5}_{17.2}$
VADA	0.01	$74.4^{74.4}_{74.2}$	$54.2^{55.4}_{52.6}$	$71.7^{71.7}_{71.7}$	$51.6^{52.0}_{45.0}$	$69.5^{69.7}_{68.4}$	$47.5^{49.8}_{40.0}$	$65.9^{66.1}_{64.8}$	$37.2^{39.4}_{35.3}$
VADA	0.1	$76.7^{76.7}_{76.6}$	$56.9^{58.3}_{53.5}$	$70.6^{71.0}_{70.0}$	$47.7^{48.8}_{44.0}$	$66.1^{66.5}_{65.4}$	$35.7^{39.3}_{33.3}$	$63.2^{64.7}_{60.2}$	$25.5^{28.0}_{25.2}$
ASA-sq	0.1	$71.7^{71.9}_{71.7}$	$52.9^{53.4}_{46.7}$	$70.7^{71.0}_{70.4}$	$51.6^{52.7}_{46.8}$	$69.2^{69.3}_{69.2}$	$45.6^{52.0}_{43.3}$	$68.1^{68.2}_{67.2}$	$44.7^{45.9}_{39.8}$
ASA-sq + VAR	1.0	$74.2^{74.5}_{74.0}$	$52.2^{52.5}_{51.9}$	$72.2^{72.2}_{71.9}$	$53.5^{53.6}_{45.4}$	$70.6^{70.8}_{70.4}$	$48.9^{52.3}_{45.6}$	$67.4^{67.7}_{66.8}$	$43.0^{46.0}_{39.4}$

D.4 VisDA-17 experiment specifications

We use train and validation sets of the VisDA-17 challenge (Peng et al., 2017).

For the feature extractor we use ResNet-50 He et al. (2016) architecture with modified output size of the final linear layer. The feature representation is 256-dimensional vector. We use the weights from pre-trained ResNet-50 model (`torchvision` model hub) for all layers except the final linear layer. The classifier consists of a single linear layer. The discriminator is implemented by a 3-layer MLP with 1024 hidden units and leaky-ReLU activation.

We train all methods for 50 000 steps with batch size 36. We train the feature extractor, the classifier, and the discriminator with SGD. For the feature extractor we use learning rate 0.001, momentum 0.9, weight decay 0.001. For the classifier we use learning rate 0.01, momentum 0.9, weight decay 0.001. For the discriminator we use learning rate 0.005, momentum 0.9, weight decay 0.001. We perform a single discriminator update per 1 update of the feature extractor and the classifier. We linearly anneal the feature extractor's and classifier's learning rate throughout the training (50 000) steps. By the end of the training the learning rates of the feature extractor and the classifier are decreased by a factor of 0.05.

The weight for the classification term is constant and set to $\lambda_{\text{cls}} = 1$. We introduce schedule for the alignment weight λ_{align}. For all alignment methods we linearly increase λ_{align} from 0 to 0.1 during the first 10000 steps. For all methods we use auxiliary conditional entropy loss on target examples with the weight $\lambda_{\text{ent}} = 0.1$.

For ASA we use history buffers of size 1000.

D.5 History size effect and evaluation of support distance

History size effect. To quantify the effects of mini-batch training mentioned in Section 3, we explore different sizes of history buffers on USPS→MNIST task with the label distribution shift $\alpha = 1.5$. The results are presented in Figure D.1 and Table D.4. Figure D.2 shows the distributions of outputs of the learned discriminator at the end of the training. While without any alignment objectives neither the densities nor the supports of $g^{\psi}{}_{\sharp}p_Z^{\theta}$ and $g^{\psi}{}_{\sharp}q_Z^{\theta}$ are aligned, both alignment methods approximately satisfy their respective alignment constraints. Compared with DANN results, ASA with small history size performs similarly to distribution alignment, while all history sizes are enough for support alignment. We also observe the correlation between distribution distance and target accuracy: under label distribution shifts, the better distribution alignment is achieved, the more target accuracy suffers. Note that with too big history buffers (e.g. $n = 5000$), we observe a sudden drop in performance and increases in distances. We hypothesize that this could be caused by the fact that the history buffer stores discriminator output values from the past steps while the discriminator parameters constantly evolve during training. As a result, for a large history buffer, the older items might no longer accurately represent the current pushforward distribution as they become outdated.

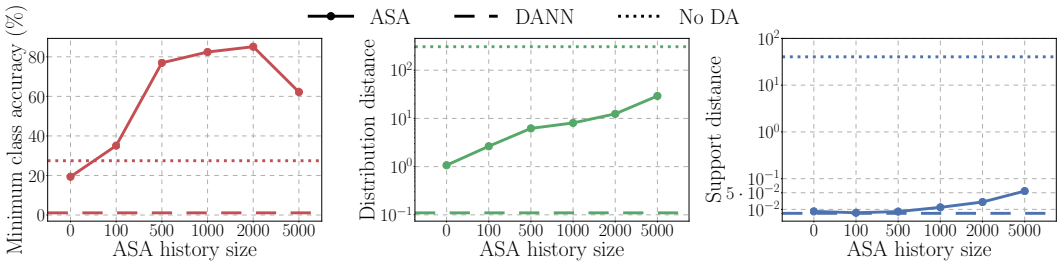
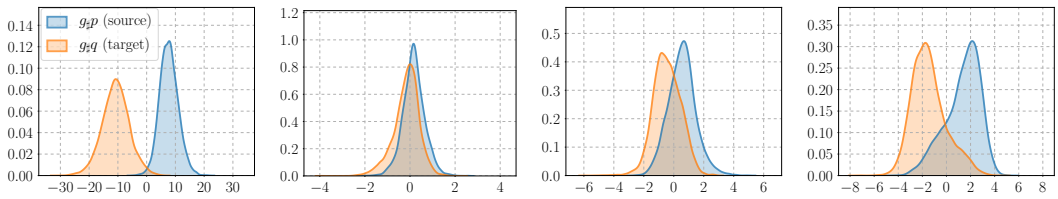


Figure D.1: Evaluation of history size effect for ASA on MNIST→USPS with the label distribution shift ($\alpha = 1.5$). The panels show (left to right): minimum class accuracy on target test set; Wasserstein distance $\mathcal{D}_W(g^{\psi}{}_{\sharp}p_Z^{\theta}, g^{\psi}{}_{\sharp}q_Z^{\theta})$ between the pushforward distributions of source and target representations induced by the discriminator; SSD divergence $\mathcal{D}_{\triangle}(g^{\psi}{}_{\sharp}p_Z^{\theta}, g^{\psi}{}_{\sharp}q_Z^{\theta})$ between the pushforward distributions. In each panel the dashed lines show the respective quantities for "No DA" and DANN methods.

Direct evaluation of support distance. In order to directly evaluate the ability of ASA (with history buffers) to enforce of support alignment, we consider the setting of the illustrative experiment described in Section 5 (3-class USPS→MNIST adaptation with 2D feature extractor, $\alpha = 1.5$). We compare methods No DA, DANN, and ASA-abs (with different history buffer sizes). For each method we consider the embedding space of the learned feature extractor at the end of training and compute Wasserstein distance $\mathcal{D}_W(p_Z^\theta, q_Z^\theta)$ and SSD divergence $\mathcal{D}_\triangle(p_Z^\theta, q_Z^\theta)$ between the embeddings of source and target domain (note that we compute the distances in the original embedding space directly without projecting data to 1D with the discriminator). To ensure meaningful comparison of the distances between different embedding spaces, we apply a global affine transformation for each embedding space: we center the embeddings so that their average is 0 and re-scale them so that their average norm is 1. The results of this evaluation are shown in Table D.5. We observe that, compared to no alignment and distribution alignment (DANN) methods, ASA aligns the supports without necessarily aligning the distributions (in this imbalanced setting, distribution alignment implies low adaptation accuracy).

Table D.4: Analysis of effect history size parameter for ASA on USPS→MNIST with class label distribution shift corresponding to $\alpha = 1.5$. We report distribution and support distances between the pushforward distributions $g^\psi{}_\sharp p_Z^\theta$ and $g^\psi{}_\sharp q_Z^\theta$, as well as the value of discriminator's log-loss.

		Target accuracy (%)		Distribution distances		
Method	History size	average	min	$\mathcal{D}_W(g^\psi{}_\sharp p_Z^\theta, g^\psi{}_\sharp q_Z^\theta)$	$\mathcal{D}_\triangle(g^\psi{}_\sharp p_Z^\theta, g^\psi{}_\sharp q_Z^\theta)$	Log-loss
No DA	—	$71.28^{72.51}_{71.25}$	$27.46^{37.26}_{24.21}$	$307.56^{322.00}_{277.33}$	$40.35^{46.06}_{32.10}$	$00.05^{00.07}_{00.04}$
DANN	—	$69.96^{71.25}_{63.89}$	$01.11^{01.53}_{00.99}$	$00.11^{00.11}_{00.10}$	$00.00^{00.00}_{00.00}$	$00.65^{00.65}_{00.65}$
ASA-abs	0	$62.75^{64.35}_{61.78}$	$19.36^{23.63}_{17.90}$	$01.07^{01.15}_{00.99}$	$00.01^{00.01}_{00.00}$	$00.57^{00.58}_{00.56}$
ASA-abs	100	$80.58^{81.73}_{78.22}$	$35.09^{44.37}_{32.10}$	$02.64^{02.70}_{02.15}$	$00.00^{00.00}_{00.00}$	$00.53^{00.53}_{00.52}$
ASA-abs	500	$92.02^{92.76}_{90.56}$	$76.96^{83.72}_{70.94}$	$06.21^{06.48}_{05.69}$	$00.00^{00.01}_{00.00}$	$00.45^{00.45}_{00.45}$
ASA-abs	1000	$92.54^{92.93}_{90.90}$	$82.41^{85.43}_{74.53}$	$08.06^{08.19}_{07.97}$	$00.01^{00.02}_{00.01}$	$00.41^{00.41}_{00.40}$
ASA-abs	5000	$86.03^{87.50}_{84.86}$	$62.19^{71.62}_{46.98}$	$29.23^{29.63}_{24.54}$	$00.05^{00.08}_{00.05}$	$00.29^{00.30}_{00.29}$



(a) No DA (b) DANN (c) ASA ($n = 0$) (d) ASA ($n = 1000$)

Figure D.2: Kernel density estimates (in the discriminator output space) of $g^\psi{}_\sharp p_Z^\theta$, $g^\psi{}_\sharp q_Z^\theta$ at the end of the training on USPS→MNIST task with $\alpha = 1.5$. n is the size of ASA history buffers.

Table D.5: Results of No DA, DANN, and ASA-abs (with different history sizes) on 3-class USPS→MNIST adaptation with 2D feature extractor and label distribution shift corresponding to $\alpha = 1.5$. We report average and minimum target class accuracy, as well as Wasserstein distance $\mathcal{D}_W$ and support divergence $\mathcal{D}_\triangle$ between source p_Z^θ and target q_Z^θ 2D embedding distributions.

Algorithm	History size	Accuracy (avg)	Accuracy (min)	$\mathcal{D}_W(p_Z^\theta, q_Z^\theta)$	$\mathcal{D}_\triangle(p_Z^\theta, q_Z^\theta)$
No DA	—	$63.0\,{}^{69.6}_{62.3}$	$45.3\,{}^{53.6}_{37.9}$	$0.78\,{}^{00.84}_{00.75}$	$0.10\,{}^{0.10}_{0.10}$
DANN	—	$75.6\,{}^{83.7}_{72.4}$	$54.8\,{}^{55.1}_{49.6}$	$0.07\,{}^{0.08}_{0.06}$	$0.02\,{}^{0.02}_{0.02}$
ASA-abs	0	$73.9\,{}^{84.1}_{73.4}$	$61.8\,{}^{72.4}_{54.6}$	$0.23\,{}^{0.47}_{0.22}$	$0.03\,{}^{0.03}_{0.03}$
ASA-abs	100	$88.5\,{}^{95.1}_{86.8}$	$71.4\,{}^{93.3}_{70.6}$	$0.54\,{}^{0.36}_{0.56}$	$0.03\,{}^{0.03}_{0.03}$
ASA-abs	500	$94.5\,{}^{94.7}_{88.7}$	$89.0\,{}^{90.3}_{83.1}$	$0.59\,{}^{0.64}_{0.55}$	$0.03\,{}^{0.03}_{0.03}$
ASA-abs	1000	$91.1\,{}^{93.0}_{91.1}$	$85.6\,{}^{86.2}_{80.7}$	$0.59\,{}^{0.62}_{0.55}$	$0.03\,{}^{0.03}_{0.03}$
ASA-abs	2000	$94.0\,{}^{94.7}_{91.2}$	$88.6\,{}^{89.4}_{80.2}$	$0.62\,{}^{0.66}_{0.58}$	$0.03\,{}^{0.03}_{0.03}$
ASA-abs	5000	$82.1\,{}^{83.9}_{81.8}$	$68.9\,{}^{70.9}_{65.5}$	$0.64\,{}^{0.67}_{0.63}$	$0.04\,{}^{0.04}_{0.04}$