Journal of Machine Learning Research 25 (2024) 1-32 Submitted 7/23; Published 8/24

Label Alignment Regularization for Distribution Shift

Ehsan Imani* IMANI@UALBERTA.CA
University of Alberta, Alberta Machine Intelligence Institute

Guojun Zhang GUOJUN.ZHANG@HUAWEI.COM
Huawei Noah's Ark Lab

Runjia Li RUNJIA@ROBOTS.OX.AC.UK
Department of Engineering Science, University of Oxford

Jun Luo JUN.LUO1@HUAWEI.COM
Huawei Noah's Ark Lab

Pascal Poupart PPOUPART@UWATERLOO.CA
School of Computer Science, University of Waterloo

Philip H.S. Torr PHILIP.TORR@ENG.OX.AC.UK
Yangchen Pan* YANGCHEN.PAN@ENG.OX.AC.UK
Department of Engineering Science, University of Oxford

Editor: Amos Storkey

Abstract

Recent work has highlighted the label alignment property (LAP) in supervised learning, where the vector of all labels in the dataset is mostly in the span of the top few singular vectors of the data matrix. Drawing inspiration from this observation, we propose a regularization method for unsupervised domain adaptation that encourages alignment between the predictions in the target domain and its top singular vectors. Unlike conventional domain adaptation approaches that focus on regularizing representations, we instead regularize the classifier to align with the unsupervised target data, guided by the LAP in both the source and target domains. Theoretical analysis demonstrates that, under certain assumptions, our solution resides within the span of the top right singular vectors of the target domain data and aligns with the optimal solution. By removing the reliance on the commonly used optimal joint risk assumption found in classic domain adaptation theory, we showcase the effectiveness of our method on addressing problems where traditional domain adaptation methods often fall short due to high joint error. Additionally, we report improved performance over domain adaptation baselines in well-known tasks such as MNIST-USPS domain adaptation and cross-lingual sentiment analysis. An implementation is available at `https://github.com/EhsanEI/lar/`.

Keywords: domain adaptation, principal component analysis, regularization

1. Introduction

Unsupervised domain adaptation studies knowledge transfer from a source domain with labeled data, to a target domain with unlabeled data, where the model will be deployed and evaluated (Ben-David et al., 2010; Mansour et al., 2009). This difference between the

∗. Work done during employment at Huawei Noah's Ark Lab.

two domains, called domain shift, arises in many applications. A document classification or sentiment analysis model for an under-resourced language can benefit from a large corpus for a different language. A personal healthcare system is often trained on a group of users different from its target users. A real-world robot's predictions or decision-making can improve through safe and less costly interactions with a simulator (Pires et al., 2019; Ganin et al., 2016; Peng et al., 2018).

There are diverse settings to study domain adaptation problems. In classification problems, closed set domain adaptation assumes the same categories between the two domains while open-set domain adaptation assumes that the two domains only share a subset of their categories (Panareda Busto and Gall, 2017). Unsupervised, semi-supervised, and supervised domain adaptation assume that the data from the target domain is fully unlabeled, partly labeled, and fully labeled respectively (Ganin et al., 2016). Two related problems to domain adaptation are multi-target domain adaptation where there are multiple target domains (Gholami et al., 2020) and domain generalization where several source domains are sampled from a distribution over tasks and the goal is to generalize to a previously unseen domain from this distribution (Blanchard et al., 2011; Gulrajani and Lopez-Paz, 2021). Within those diverse settings, our work specifically addresses unsupervised domain shift problems.

The prevalence of domain shift in machine learning has inspired a large body of algorithmic and theoretical research on domain adaptation. Ben-David et al. (2010) and Zhang et al. (2019) formulated the difference between the source and the target domain with the notion of $\mathcal{H}$-divergence and Margin Disparity Discrepancy and provided generalization bounds that relate performance on the two domains. Acuna et al. (2021) extended these results to a more general notion of f-divergence. Adversarial domain adaptation algorithms are motivated by these theoretical findings and aim to learn representations that achieve high performance in the source domain while being invariant to the shift between the source and the target domain (Ganin et al., 2016; Zhang et al., 2017; Conneau et al., 2018; Long et al., 2015; Pei et al., 2018).

The aforementioned representation-matching approach assumes that the optimal joint risk between the source and target is small. This assumption fails when the conditional distribution of the labels given input is different between source and target domains. An example occurs when labels in the source domain are much more imbalanced than in the target domain. For instance, Zhao et al. (2019) identified that under such label distribution shift, the optimal joint risk can be quite large and they empirically show the failure of domain adaptation methods on MNIST-USPS digit datasets. Johansson et al. (2019) also pointed out the limitation of matching feature representations by showing its inconsistency, and thus the tendency for high target errors.

In this work, we adopt a novel approach to domain adaptation that focuses on label alignment, defined as the alignment of labels with the top left singular vectors of the representation. Instead of striving for an invariant representation, our proposed algorithm fine-tunes the classifier for the target domain. It achieves this by removing the influence of label alignment in the source domain and applying this alignment principle to the target domain. A critical distinction of our approach from existing methodologies is that we adjust the classifier's weight rather than its representation. Consequently, our method can

be applied in settings with linear function approximation and may complement existing approaches.

We describe the label alignment phenomenon in Section 2, and outline the proposed method in Section 3. Section 4 formally justifies our regularization method by showing that it projects the solution onto the span of the top right singular vectors of the target domain. Section 5 reviews related work. In Section 6, we first provide a synthetic example where the proposed regularizer shows a clear advantage. We then experiment with imbalanced MNIST-USPS binary classification tasks and find that our method, unlike the domain-adversarial baseline, is robust to imbalance in one domain. Finally, we evaluate our algorithm on cross-lingual sentiment analysis tasks and observe improved F_1 score on training with our regularization, compared to adversarial domain adaptation baselines.

2. Background: Label Alignment

In this section, we briefly review the standard linear regression problem and define relevant notations to explain the *label alignment property* (LAP, Imani et al., 2022).

2.1 Linear Regression and Notations

We consider a dataset with n samples, (possibly learned and nonlinear) representation matrix $\Phi \in \mathbb{R}^{n\times d}$ and label vector $y \in \mathbb{R}^n$ from a source domain. Denote the model's weights as $w \in \mathbb{R}^d$, we study the linear regression problem:

$$\min_w \|\Phi w - y\|^2. \tag{1}$$

Without loss of generality, we replace the bias unit with a constant feature in the representation matrix to avoid studying the unit separately. The model will be evaluated on a test set sampled from the target domain.

The singular value decomposition (SVD) of a representation matrix Φ is $\Phi = U\Sigma V^\top = \sum_{i=1}^d \sigma_i u_i v_i^\top$, where

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_d \\ & \mathbf{0} & \end{bmatrix} \in \mathbb{R}^{n\times d}$$

is a rectangular diagonal matrix whose main diagonal consists of singular values $\sigma_1, \cdots, \sigma_d$ in descending order with the remaining rows set to zero, and

$$U = [u_1, \ldots, u_n] \in \mathbb{R}^{n\times n} \text{ and } V = [v_1, \cdots, v_d] \in \mathbb{R}^{d\times d}$$

are orthogonal matrices whose columns $u_i \in \mathbb{R}^n$ and $v_j \in \mathbb{R}^d$ are the corresponding left and right singular vectors. In principal component analysis (Pearson, 1901), $v_1, \cdots, v_k$ are also known as the first k principal components. For a vector a and orthonormal basis B, a^B is a shorthand for $B^\top a$, the representation of a in terms of the row vectors of B. We use $r(\cdot)$ to denote the rank of a matrix.

2.2 Label Alignment

Label alignment is specified in terms of the singular vectors of Φ and label vector y. The left singular vectors of Φ, $\{u_1, \cdots, u_n\}$ form an orthonormal basis that spans the n-dimensional space. The label vector $y \in \mathbb{R}^n$ can be decomposed in this basis with:

$$y = Uy^U = y_1^U u_1 + \cdots + y_n^U u_n, \tag{2}$$

where y_i^U is the i^{th} component of vector $y^U \in \mathbb{R}^n$.

Label alignment (Imani et al., 2022) is a relationship between the labels and the representation where the variation in the labels are mostly along the top principal components of the representation. For our purpose we give the following definition and verify that it approximately holds in a number of real-world tasks. A dataset has *label alignment* with rank k if for $k \ll r(\Phi)$ we have $y_i^U = 0, \forall i \in \{k+1, ..., d\}$.

In Table 1 we investigate this property in binary classification tasks (with ± 1 labels) and regression tasks. In this table $k(\epsilon)$ means the smallest k where

$$\sqrt{\sum_{i=k+1}^{d} (y_i^U)^2} < \epsilon \sqrt{\sum_{i=1}^{d} (y_i^U)^2}.$$

If $k(\epsilon)$ is small for a small ϵ then the projection of the label vector on the span of Φ is mostly in the span of the first few singular vectors. In all the ten tasks less than half the singular vectors with nonzero singular values already span $\geq 90\%$ of the norm of the projection of y on the span of Φ. The number $k(0.1)$ is remarkably small, less than 10, in seven out of the ten tasks. Appendix A shows this property in a controlled setting where a large number of features are correlated with the labels.

Task	n	d	$r(\Phi)$	$k(0.1)$	Task	n	d	$r(\Phi)$	$k(0.1)$
CT Scan	10000	385	372	12	CIFAR-10	10000	513	513	7
Song Year	10000	91	91	6	CIFAR-100	1000	513	513	7
Bike Sharing	10000	13	13	4	STL-10	1000	513	513	2
MNIST	12665	785	580	2	XED (En)	6525	769	769	231
USPS	2199	257	257	2	AG News	10000	769	769	40

Table 1: Label alignment in real-world tasks. The table on the left uses the original features in the dataset and the table on the right uses features extracted from neural networks. CT Scan, Song Year, and Bike Sharing are regression tasks and the rest are binary classification. We used the first two classes of multi-class classification datasets to create a binary classification task. Other details about the datasets are in Appendix B. In all of these tasks, a large portion of the label vector is in the span of a relatively small set of top singular vectors (compared to the rank).

Similar label alignment phenomenon has been also observed in a deep learning setting. Recent work in the Neural Tangent Kernel (NTK) literature has observed that in common datasets the label vector is largely within the span of the top eigenvectors of the NTK Gram matrix (Arora et al., 2019). In contrast, a randomized label vector would be more

or less uniformly aligned with all eigenvectors. More recently, Baratin et al. (2021) and Ortiz-Jiménez et al. (2021) noted that training a finite-width NN makes the alignment between the network's kernel and the task even stronger. Imani et al. (2022) observed a similar behavior in NN hidden representations, indicating that training the NN aligns the top singular vectors of the hidden representations to the task.

2.3 Reformulating the Regression Objective

We describe how to reformulate the linear regression objective function with the label alignment property. This reformulation shows that the linear regression objective is implicitly enforcing the LAP on the source domain (i.e., the training data) and this encourages us to further derive our domain adaptation regularization on the target domain.

Objective (1) can be rewritten by the following steps.

$$\begin{aligned}
\min_w \|\Phi w - y\|^2 &= \min_w \|U\Sigma V^\top w - y\|^2 \\
&= \min_w \|\Sigma V^\top w - U^\top y\|^2 \\
&= \min_w \|\Sigma w^V - y^U\|^2, \text{shorthand notation} \\
&= \min_w \sum_{i=1}^{d} (\sigma_i w_i^V - y_i^U)^2 + \sum_{i=d+1}^{n} (y_i^U)^2. \qquad (3)
\end{aligned}$$

In the first line, since U is an orthogonal matrix, we have $UU^\top = \mathbb{I}$ and $\|Ux\| = \|x\|$ for any vector x. Note that the last term $\sum_{i=d+1}^{n}(y_i^U)^2$ can be dropped as it is a constant and does not affect the optimization.

Assume the LAP holds for the first $k < d$ singular vectors. Then $y_i^U = 0, \forall i \in k+1, ..., d$. Hence the first term in (3) can be further decomposed to

$$\sum_{i=1}^{d} (\sigma_i w_i^V - y_i^U)^2 = \sum_{i=1}^{k} (\sigma_i w_i^V - y_i^U)^2 + \sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2.$$

Plugging this decomposition into the above objective (3) and dropping the last term, we get

$$\min_w \sum_{i=1}^{k} (\sigma_i w_i^V - y_i^U)^2 + \sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2. \qquad (4)$$

We can interpret the first term in the rewritten objective (4) as linear regression on a smaller subspace and the second term as a regularization term implicitly enforcing label alignment property on the training data (Φ, y).

The latter is because minimizing the second term has the effect of regularizing the predictions so they likely align with the top singular vectors. This is because:

$$y = \Phi w = U\Sigma V^\top w$$

and therefore $U^\top y = \Sigma V^\top w$, which can be written as

$$y^U = \Sigma w^V$$

by using the shorthand notations. For the ith component in vector y^U, we have ${u_i}^\top y = \sigma_i {v_i}^\top w$. Minimizing w_i^V for $i \in \{k+1, \cdots, d\}$ will reduce the corresponding y_i^U and leave y_i^U for those components $i < k+1$. We call the second term $\sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2$ from (4) *label alignment regularization*.

The derivation above shows that when minimizing the original mean squared error for linear regression, we implicitly use label alignment regularization on the training data (source domain data). In the next section, we introduce this regularization into the target domain.

3. Label Alignment for Domain Adaptation

This section describes our approach to domain adaptation by enforcing the LAP.

In unsupervised domain adaptation, we have a labeled dataset (Φ, y) and an unlabeled dataset $\tilde{\Phi}$ with the corresponding label vector $\tilde{y}$ unknown. From (4), we know that enforcing the LAP does not require knowing the labels $\tilde{y}$. This inspires our key idea of improving the generalization on the target domain: we can use the unlabeled part to enforce the LAP.

Using tilde notation for the SVD of $\tilde{\Phi}$ and assuming $(\tilde{\Phi}, \tilde{y})$ satisfies the LAP with rank $\tilde{k}$, we can put together the supervised part of the source domain and unsupervised part of the target domain to form the objective:

$$\min_w \|\Phi w - y\|^2 + \sum_{i=\tilde{k}+1}^{d} \tilde{\sigma}_i^2 (w_i^{\tilde{V}})^2. \tag{5}$$

The second term $\sum_{i=\tilde{k}+1}^{d} \tilde{\sigma}_i^2 (w_i^{\tilde{V}})^2$ is the *label alignment regularization on the target domain*. As we explained in the previous section, the first term (i.e. the standard regression part) in the above objective implicitly enforces the LAP (with rank k) on the source domain. If we expand (5) by the reformulated linear regression objective (4), we have:

$$\min_w \sum_{i=1}^{k} (\sigma_i w_i^V - y_i^U)^2 + \sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2 + \sum_{i=\tilde{k}+1}^{d} \tilde{\sigma}_i^2 (w_i^{\tilde{V}})^2.$$

Therefore, we have actually done the regularization *twice*: one with the source domain and one with the target domain. We explicitly remove the label alignment regularization on the source domain and arrive at the final objective function:

$$\min_w \|\Phi w - y\|^2 - \sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2 + \lambda \sum_{i=\tilde{k}+1}^{d} \tilde{\sigma}_i^2 (w_i^{\tilde{V}})^2. \tag{6}$$

Algorithm 1 shows the pseudo-code. The objective to be minimized has three terms and the hyperparameter λ controls the relative importance of the regularizer. As we will show in § 4, under certain constraints this hyperparameter does not affect the final solution and only changes the convergence rate. The first term is the loss that uses the labeled data from

the source domain. Following the recent evidence on the viability of the squared error loss for classification (Hui and Belkin, 2020), we use the squared error in both regression tasks and binary classification tasks. We use ± 1 labels in binary classification as these labels showed the label alignment property (LAP) in Table 1. The second term removes implicit regularization from the source domain. The third term is the proposed regularizer that uses the unlabeled data from the target domain. The second and third terms serve as a projection onto the orthogonal complement of $\text{span}(\tilde{v}_{k+1}, \ldots, \tilde{v}_d)$, or namely, $\text{span}(\tilde{v}_1, \ldots, \tilde{v}_k)$, which we show in the next section.

Algorithm 1 Label Alignment Regression

Get data Φ, y, $\tilde{\Phi}$, and hyperparameters t, α, k, $\tilde{k}$, λ
Compute covariance matrices $\Phi^\top\Phi$ and $\tilde{\Phi}^\top\tilde{\Phi}$
Perform eigendecomposition of $\Phi^\top\Phi$ and $\tilde{\Phi}^\top\tilde{\Phi}$ to get $\sigma_{k+1:d}$, $\tilde{\sigma}_{\tilde{k}+1:d}$, $\tilde{v}_{k+1:d}$ and $\tilde{v}_{\tilde{k}+1:d}$
Initialize w to zero
for t iterations **do**
  Perform gradient step with respect to $\|\Phi w - y\|^2 - \sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2 + \lambda \sum_{i=\tilde{k}+1}^{d} \tilde{\sigma}_i^2 (w_i^{\tilde{V}})^2$ with step-size α and update w
end for

4. Label Alignment Regularization as Projection

In this section, we provide theoretical insight into how close the solution acquired by our regularization approach is to the optimal solution on the target domain. First, we use a simple rotated Gaussian example to illustrate that our label alignment can exactly give the optimal target solution (see also § 6). Second, we generalize our conclusion beyond the Gaussian example and present the main theorem, showing that when $k = \tilde{k}$ and under a weak additional assumption our solution 1) lies in the span of the top few singular vectors of the target domain and 2) lies in the same direction as the optimal target domain solution under certain assumptions. All proofs in this section are in Appendix B.

For convenience, we rewrite our objective (6) as:

$$\min_w \|\Phi w - y\|^2 - w^\top (S - S_k) w + \lambda w^\top (\tilde{S} - \tilde{S}_{\tilde{k}}) w,$$

where $S = \Phi^\top\Phi$ is the covariance matrix of Φ, S_k is the covariance matrix truncated to rank k and similar notations hold for $\tilde{S}$ and $\tilde{S}_{\tilde{k}}$. Then the optimal solution for this problem is:

$$\widehat{w^*} = (S_k + \lambda(\tilde{S} - \tilde{S}_{\tilde{k}}))^{-1} \Phi^\top y, \tag{7}$$

if the matrix $S_k + \lambda(\tilde{S} - \tilde{S}_{\tilde{k}})$ is full rank, which requires $k \geq \tilde{k}$. In practice, we can treat k and $\tilde{k}$ as hyper-parameters and choose them as we wish.

4.1 Rotated Gaussian Example

Consider a simple example where the source and target domain data are both two-dimensional Gaussians, but the target domain is acquired by rotating the source domain (Figure 1 pro-

vides a concrete example). Denote the following Gaussian distribution as:

$$\mathcal{N}(0, Q) = \frac{1}{2\pi\sqrt{|Q|}} \exp\left(-\frac{1}{2}x^\top Q^{-1}x\right), \tag{8}$$

where $Q = P\begin{bmatrix} s_1^2 & 0 \\ 0 & s_2^2 \end{bmatrix} P^\top$, and $P = \begin{bmatrix} p_1 & p_2 \end{bmatrix}$. Here we consider the spectral decomposition of the covariance matrix $Q \in \mathbb{R}^{2\times 2}$ with $s_1 > 0$, $s_2 > 0$. Here $P \in \mathbb{R}^{2\times 2}$ is an orthogonal matrix, and p_1, p_2 are its column vectors. Since $x = PP^\top x = x_1^P p_1 + x_2^P p_2$, we can rewrite the distribution as:

$$\mathcal{N}(0, Q) = \frac{1}{2\pi s_1 s_2} \exp\left(-\frac{1}{2s_1^2}(x_1^P)^2 - \frac{1}{2s_2^2}(x_2^P)^2\right).$$

We further define the conditional distributions as follows:

$$p_{\mathcal{S}}(x|y) = 2\mathcal{N}(0, Q)\mathbb{1}(yx_1^P > 0), \tag{9}$$

where $y \in \{1, -1\}$. Similarly, we can define the target distribution by replacing Q, P, s_i, p_i with $\tilde{Q}, \tilde{P}, \tilde{s}_i, \tilde{p}_i$. We now compute different solutions and then compare them. We assume that there is distribution shift and that p_1 is not parallel to $\tilde{p}_2$.

Recall the regression solution on the source domain:

$$w_{\mathcal{S}}^* = (\Phi^\top\Phi)^{-1}\Phi^\top y = S^{-1}\Phi^\top y. \tag{10}$$

Assuming that the sample size is large enough.

$$\frac{1}{n}\Phi^\top y \approx \mathbb{E}_{x,y}[xy] = \sqrt{\frac{2}{\pi}} s_1 p_1, \tag{11}$$

where the computation of the expectation is detailed in Lemma 9, Appendix B. Combining this equation with

$$\Phi^\top y = V\Sigma^\top y^U = \sigma_1 y_1^U v_1 + \sigma_2 y_2^U v_2, \tag{12}$$

we know that $y_2^U = 0$ if we identify $v_1 = p_1$. In other words, the label alignment property holds on the source domain with rank $k = 1$. The covariance matrix is:

$$\frac{1}{n}\Phi^\top\Phi \approx \mathbb{E}_x[xx^\top] = s_1^2 p_1 p_1^\top + s_2^2 p_2 p_2^\top, \tag{13}$$

We can identify $v_i = p_i$, $s_i^2 = \sigma_i^2/n$ from the SVD of Φ. Plugging (12) and (13) back into (10) we get the optimal solution on the source domain:

$$w_{\mathcal{S}}^* = \sqrt{\frac{2}{\pi}}\frac{1}{s_1} v_1, \tag{14}$$

which agrees with our intuition that $w_{\mathcal{S}}^*$ should be in the direction with the largest singular value. Similarly, the optimal solution on the target domain is

$$w_{\mathcal{T}}^* = \sqrt{\frac{2}{\pi}}\frac{1}{\tilde{s}_1} \tilde{v}_1, \tag{15}$$

where the tilde notations are the same type of variables used on the target domain. According to (7), the label alignment solution with the removal of implicit regularization (given that $\tilde{v}_2$ is not parallel to v_1) is:

$$\widehat{w^*} = (S_k + \lambda(\tilde{S} - \tilde{S}_{\tilde{k}}))^{-1}\Phi^\top y \tag{16}$$

$$= (s_1^2 v_1 v_1^\top + \lambda \tilde{s}_2^2 \tilde{v}_2 \tilde{v}_2^\top)^{-1}\sqrt{\frac{2}{\pi}} s_1 v_1, \tag{17}$$

To better understand the solution $\widehat{w^*}$, suppose $\lambda = 1$. Then if we replace $\tilde{s}_2$, $\tilde{v}_2$ by s_2, v_2, we obtain $w_{\mathcal{S}}^*$. If we replace s_1, v_1 by $\tilde{s}_1$, $\tilde{v}_1$, we obtain $w_{\mathcal{T}}^*$.

In fact in this example the effect of label alignment regularization is some kind of projection into the space of $\tilde{v}_1$. Regardless of the hyperparameter λ, we always have the following result:

Proposition 1 *In the example in this section suppose $v_1^\top \tilde{v}_1 \neq 0$. Then the label alignment solution is $\widehat{w^*} = c w_{\mathcal{T}}^* / v_1^\top \tilde{v}_1$ with $c > 0$.*

The proposition shows that given $v_1^\top \tilde{v}_1 > 0$, our solution $\widehat{w^*}$ is exactly in the same direction as the optimal solution $w_{\mathcal{T}}^*$, which is verified in our experiments (Section 6.1). The above discussion also holds in a more generalized setting, as we show below.

4.2 Generalized Setting

This section derives the relation between the solutions $\widehat{w^*}$ and $w_{\mathcal{T}}^*$ in a more general setting, where x is high dimensional, and $k, \tilde{k}$ can be larger than one. We can rewrite

$$w_{\mathcal{S}}^* = \sum_{i \leq k} \sigma_i^{-1} y_i^U v_i,\ w_{\mathcal{T}}^* = \sum_{i \leq \tilde{k}} \tilde{\sigma}_i^{-1} \tilde{y}_i^{\widetilde{U}} \tilde{v}_i. \tag{18}$$

Hence, $w_{\mathcal{S}}^* \in \text{span}(v_1, \ldots, v_k)$, $w_{\mathcal{T}}^* \in \text{span}(\tilde{v}_1, \ldots, \tilde{v}_k)$. We show that our solution is also in the span of the top right singular vectors of the target domain as $w_{\mathcal{T}}^*$:

Theorem 2 *Assume $k = \tilde{k}$ and $(V'_{d-k})^\top \tilde{V}'_{d-k}$ is invertible with $V'_{d-k} = \begin{bmatrix} v_{k+1} & \ldots & v_d \end{bmatrix}$ and $\tilde{V}'_{d-k} = \begin{bmatrix} \tilde{v}_{\tilde{k}+1} & \ldots & \tilde{v}_d \end{bmatrix}$, then $\widehat{w^*} \in \text{span}(\tilde{v}_1, \ldots, \tilde{v}_k)$ holds and $\widehat{w^*}$ is independent of λ.*

This theorem tells us that after label alignment regularization, $\widehat{w^*}$ and $w_{\mathcal{T}}^*$ lie in the same subspace.

We now characterize when our solution can lie in exactly the same direction as the optimal target domain's solution. Denote $V_k = [v_1 \ldots v_k]$, $\widetilde{V}_{\tilde{k}} = [\tilde{v}_1 \ldots \tilde{v}_{\tilde{k}}]$, and

$$\mu_k = (y_1^U/\sigma_1, \ldots, y_k^U/\sigma_k),\ \tilde{\mu}_{\tilde{k}} = (\tilde{y}_1^{\widetilde{U}}/\tilde{\sigma}_1, \ldots, \tilde{y}_{\tilde{k}}^{\widetilde{U}}/\tilde{\sigma}_{\tilde{k}}).$$

We have the following theorem:

Theorem 3 *Given invertible $V_k^\top \widetilde{V}_k$, with $V_k = [v_1 \ldots v_k]$, $\widetilde{V}_{\tilde{k}} = [\tilde{v}_1 \ldots \tilde{v}_{\tilde{k}}]$ and with the same assumptions of Theorem 2, there exists $c > 0$ such that $\widehat{w^*} = c w_{\mathcal{T}}^*$ iff:*

$$\mu_k = c V_k^\top \widetilde{V}_k \tilde{\mu}_k = c V_k^\top w_{\mathcal{T}}^*. \tag{19}$$

In the special case of $k = \tilde{k} = 1$, we obtain the following:

Corollary 4 *Given $k = \tilde{k} = 1$ and $\tilde{y}_1^{\widetilde{U}} y_1^U v_1^\top \tilde{v}_1 > 0$, we have $\widehat{w^*} = c w_{\mathcal{T}}^*$.*

This corollary tells us that if $k = \tilde{k} = 1$ and if, for both domains, the labels can be determined by the principal component (or, in other words, the most significant feature), then our label alignment regularization finds the optimal target solution.

Back in the more general setting of $k = \tilde{k} \geq 1$, we show a sufficient condition for the invertibility assumption in Theorem 2 to hold: V and $\widetilde{V}$ are somehow similar to each other.

Proposition 5 *Suppose $\epsilon < \min\{\frac{1}{k}, \frac{1}{d-k}\}$, $|v_i^\top \tilde{v}_j| \leq \epsilon$ for any $i \neq j$, and $v_i^\top \tilde{v}_i \geq 1 - \epsilon$ for any i, then both $V_k^\top \widetilde{V}_k$ and $(V'_{d-k})^\top \widetilde{V}'_{d-k}$ are invertible.*

We can give a stronger guarantee for the assumption that $V_k^\top \widetilde{V}_k$ is invertible. Note that $\mathbb{S}^{d-1}$ denotes the $(d-1)$-dimensional unit hypersphere in $\mathbb{R}^d$.

Proposition 6 *Suppose the target singular vectors $\tilde{v}_1, \ldots, \tilde{v}_d$ satisfies the following probability distribution:*

$$p(\tilde{v}_1, \ldots, \tilde{v}_d) = p(\tilde{v}_1) p(\tilde{v}_2 | \tilde{v}_1) \ldots p(\tilde{v}_d | \tilde{v}_1, \ldots, \tilde{v}_{d-1}), \tag{20}$$

where $p(\tilde{v}_1)$ is a continuous distribution on $\mathbb{S}^{d-1}$ and each $p(\tilde{v}_i | \tilde{v}_1, \ldots, \tilde{v}_{i-1})$ is a continuous distribution on the manifold $\mathbb{S}^{d-1}$ for $2 \leq i \leq d$. Then $V_k^\top \widetilde{V}_k$ is invertible almost surely.

Note that our result does not depend on any assumption about the optimal joint error, as is commonly required in the domain adaptation literature (e.g. Ben-David et al., 2010; Acuna et al., 2021). Moreover, as pointed out by Zhao et al. (2019), the usual generalization bound would fail in the presence of heavy shift of label distributions, under which our method is still robust (see Section 6).

5. Related Work

The result by Ben-David et al. (2010) provides a general theoretical guidance regarding how to learn the domain-invariant representations. The basic idea is to make the joint error of the best hypothesis on the two domains on the invariant representation small. Low joint error in the domain-adversarial model is crucial to the model's performance on the target domain.

The dominant approach to domain adaptation is learning domain-invariant representations that are similar in some sense between source and target domains (Tzeng et al., 2014; Zhuang et al., 2015; Ghifary et al., 2016; Long et al., 2016, 2017; Benaim and Wolf, 2017; Bousmalis et al., 2017; Courty et al., 2017; Motiian et al., 2017; Rebuffi et al., 2017; Saito et al., 2017; Zhang et al., 2019). Different methods differ in how the invariance property is enforced, which typically includes how the similarity is defined and implemented. Recent work in deep learning encourages this invariance in one or multiple hidden representations of a neural network.

The popular domain-adversarial methods achieve domain-invariant representations based on the idea of adversarial models (Long et al., 2015; Zhuang et al., 2017; Lee et al., 2019;

Damodaran et al., 2018; Acuna et al., 2021). Specifically, Long et al. (2015) adversarially learn representations to distinguish the data points from the source and target domain while minimizing the supervised loss. Conneau et al. (2018) use a domain-adversarial approach to align representations of the source and target domains in a shared space. They transform the source embeddings with a linear mapping that is encouraged to be orthogonal. The domain-adversarial model then generates pseudo-labels on the target domain for additional refinement. The shared representation, which is learned without a parallel corpus, outperforms previous supervised methods in several cross-lingual tasks.

There are various similarity or distance measures to define a loss function for enforcing invariant representations. For example, Zhuang et al. (2017) and Meng et al. (2018) minimize the KL-divergence and Lee et al. (2019) and Damodaran et al. (2018) minimize the Wasserstein distance. Sun and Saenko (2016) minimize the ℓ_2 distance between the covariance matrices of the source and target domain representations. Long et al. (2015) minimize Maximum Mean Discrepancy between source and target domain hidden representations embedded with a kernel.

Despite the flourishing literature on representation-based domain adaptation methods, they have critical limitations. Zhao et al. (2019) and Johansson et al. (2019) have presented synthetic examples in which a domain-adversarial model that minimizes the supervised loss in the source domain, while aiming for an invariant representation, still fails in the target domain. We will demonstrate this failure through our experiments in the next section and show that our proposed algorithm remains robust in such situations.

Domain-adaptation methods that do not rely on representation learning are less studied and can be applied in highly restricted settings. For example, importance sampling (Shimodaira, 2000) assumes label conditional distribution must be the same and target domain is within the support of the source domain. Our work supplements this direction.

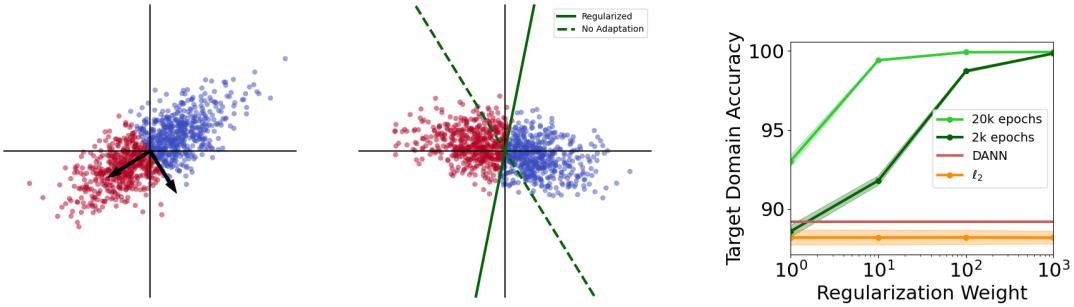
6. Experiments

In this section, we first design a synthetic dataset to verify that our regularizer is indeed beneficial in a distribution shift setting by adjusting the classifier and to perform an ablation study on the role of removing implicit regularization. Then, we demonstrate the effectiveness of our method on a well-known benchmark where classic domain-adversarial methods are known to fail (Zhao et al., 2019). Last, we show our algorithm's practical utility in a cross-lingual sentiment classification task.

6.1 Synthetic Data

We create a distribution shift scenario where the alignment property is present in the labeled data distribution (Figure 1), as theoretically discussed in § 4.1. For the source domain (a), the input is sampled from a two-dimensional Gaussian distribution. The distribution is more spread out in the direction of the first principal component (see the black arrows) which corresponds to a larger singular value. In this task, the two classes are separated along this direction as shown in the figure. The resulting vector of all labels is mostly in the direction of the first singular vector of the representation matrix. We rotate the input by 45° to create the target distribution in (b).

We then run the proposed algorithm with hyperparameters $k = \tilde{k} = 1$ and different values of λ and compared it with the ℓ_2 regularizer and a domain-adversarial baseline DANN (Ganin et al., 2016) with one hidden layer of width 64. Note that the optimal solution should be independent of λ (Prop. 1), but λ may affect the convergence rate. Figures 1 (b) to (c) show the results. Further details are in Appendix B. In Figure 1 (b) we see that the solution without regularization separates the classes as they are separated in the source domain. The proposed algorithm finds a separating hyperplane that matches how the classes are separated in the target domain. Finally, (c) shows that our regularizer surpasses both the ℓ_2 regularizer and the domain-adversarial baseline and achieves a near perfect classification in the target domain in this example. The dark green line in this figure uses 2k epochs and its accuracy is sensitive to λ. Increasing the number of epochs to 20k (green line) reduces this sensitivity, indicating that, as the theory predicted, the final solution is robust to λ and the sensitivity is due to slow optimization.



(a) Source Domain (b) Target Domain (c) Performance

Figure 1: (a) Source domain. The black arrows show principal components. (b) Target domain. The green lines show separating hyperplanes found without using any regularization (dashed) and with our regularizer with $\lambda = 10^3$ (solid). (c) Performance on the target domain. The red line shows the performance of DANN. The x axis is the regularization coefficient for ℓ_2 regularization (orange curve) and λ for the proposed regularizer (green curves). The proposed regularizer achieves near-perfect accuracy on this domain. Shaded areas are standard errors over 10 runs. Variations in target accuracy of DANN are near zero.

We also want to evaluate the effectiveness of removing the implicit regularization term $\sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2$ as described in Equation 6. More specifically, we are interested in when removing the implicit regularization would be effective.

Recall from Section 4 that the vanilla closed form- solution without any regularization is:

$$w = S^{-1} \Phi^T y \tag{21}$$

where $S = \Phi^T \Phi$, and $\Phi \in \mathbb{R}^{n \times d}$ is the feature matrix.

To utilize more specific chracteristics of a dataset, we want to first explore the synthetic data case as described in Section 6. Then the closed-form solution with label alignment

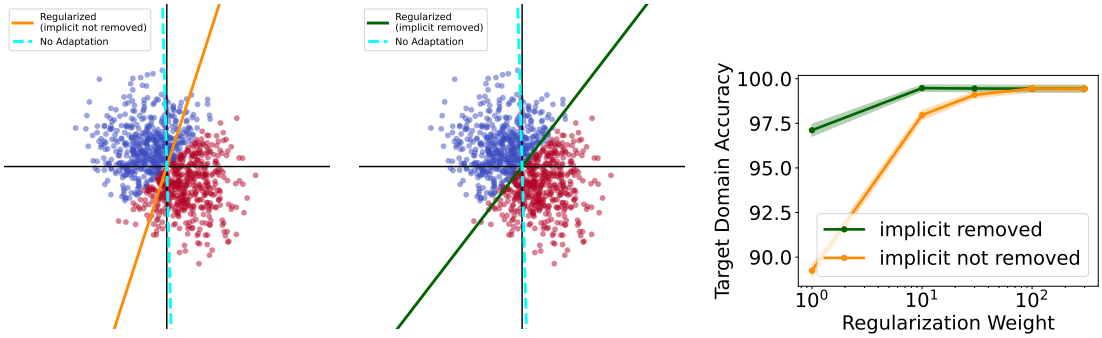
regularization with removal of the implicit regularization term is:

$$\begin{aligned} w &= (S_k + \lambda(\tilde{S} - \tilde{S}_k))^{-1}\Phi^T y \\ &= (s_1^2 p_1 p_1^T + \lambda \tilde{s}_2^2 \tilde{p}_2 \tilde{p}_2^T)^{-1}\sqrt{\frac{2}{\pi}} s_1 p_1 \end{aligned} \tag{22}$$

and the closed-form solution with label alignment regularization without implicit regularization removal is:

$$\begin{aligned} w &= (S + \lambda(\tilde{S} - \tilde{S}_k))^{-1}\Phi^T y \\ &= (s_1^2 p_1 p_1^T + s_2^2 p_2 p_2^T + \lambda \tilde{s}_2^2 \tilde{p}_2 \tilde{p}_2^T)^{-1}\sqrt{\frac{2}{\pi}} s_1 p_1. \end{aligned} \tag{23}$$

The only difference is $s_2^2 p_2 p_2^T$, and therefore the relative magnitude of s_2 and λ should be the deciding factor of the solution w. Experiments conducted on synthetic data corroborated the theoretical conclusion, as illustrated in Figure 2. As seen in the figure, s_2 is larger compared to the previous experiment. The target distribution is a rotation of this new distribution. Removing implicit regularization results in a different performance especially with smaller values of λ.



(a) Without Implicit Removal (b) With Implicit Removal (c) Performance

Figure 2: (a) Without Implicit Removal. The cyan dashed line is the decision boundary without any adaptation. The orange line shows the decision boundary when λ is set to 1 for our proposed regularizer without implicit removal. (b) With Implicit Removal. The green line shows the decision boundary when λ is set to 1 with implicit removal. (d) Performance on the target domain. The horizontal axis is λ for the proposed regularizer. Before λ dominates, the benefits of removing implicit regularization are significant. Shaded areas are standard errors over 10 runs.

6.2 MNIST-USPS

The experiments in Table 2 consider binary classification tasks from the MNIST-USPS domain adaptation benchmark with linear and shallow models. Both MNIST and USPS

are digit classification datasets with 10 classes and therefore 45 binary classification tasks between two digits. In MNIST, the input is a 28×28 grayscale image flattened to a 784-dimensional vector. USPS images are 16×16 and we resize them to 28×28 and flatten each input to a 784 vector.

Each column of the table is the average accuracy over 45 domain adaptation tasks. In the first column, the source domain (fully labeled) is a pair of digits from MNIST and the target domain (fully unlabeled) is the same pair, but from USPS. The datasets for the source and target domains are reversed in the second column. The last three columns are like the second column except that, in binary classification between two digits, only a certain ratio of the lower digit in the source domain, as indicated in the header, is used. This subsampling creates a large degree of imbalance that, as Zhao et al. (2019) observed, poses a challenge to domain-adversarial methods.

We use the train split of the dataset for the source domain and the test split of the other dataset for the target domain. A small set of 100 labeled points from the target domain is used for hyperparameter selection as we have not developed a fully unsupervised hyperparameter selection strategy. However, we give the baseline the same validation set to keep the experiment fair.

The first two rows show the performance of the domain-adversarial method DANN (Ganin et al., 2016) with one hidden layer on these tasks. (Deeper NNs performed worse on the highly imbalanced tasks in our preliminary experiments.) The first row is the average target domain accuracy of a two-layer ReLU NN trained purely on the source domain. In the second row, the domain-adversarial objective is added to reduce domain shift in the hidden representation. DANN improves accuracy in both U→M and M→U. In the cases with subsampling, however, DANN consistently hurts performance. The third and fourth row show the performance of a linear method with or without our regularizer. Using our regularizer improves performance in all columns and outperforms the models in the other rows in the cases with subsampling.

	U → M	M → U	0.3 → U	0.2 → U	0.1 → U
No Adaptation (NN)	77.85	84.88	**83.36**	**72.84**	**53.58**
DANN	**83.93**	**86.69**	78.05	64.2	47.27
No Adaptation (Linear)	78.68	83.84	80.99	79.47	75.41
Label Alignment Regression	**81.97**	**88.96**	**86.99**	**84.84**	**82.71**

Table 2: Accuracies on MNIST-USPS benchmark. Each column is averaged over the 45 binary classification tasks. M and U indicate MNIST and USPS. Ratios indicate MNIST tasks where one digit is subsampled. In tasks with severe subsampling the proposed algorithm improves the accuracy and achieves the highest performance. DANN performs worse than a regular neural network under subsampling.

We then investigate why DANN hurts performance under subsampling. A domain-adversarial network like DANN has three components: a domain classifier (discriminator) that predicts whether a data point is from the source or the target domain, a generator that learns a shared embedding between the two domains, and a label predictor that performs classification on the task of interest using the generator's embedding. The label predictor

uses the labeled source data to increase source accuracy, i.e. the label predictor's accuracy on the source domain. The ultimate goal is to have the label predictor achieve high accuracy on the target domain. The discriminator's accuracy on the other hand shows how successful the discriminator is in recognizing whether a point is from the source or the target domain. In an ideal case this accuracy should be close to that of a random classifier since the data points from the two domains are mapped close to each other in the shared embedding.

Table 3 shows the average source domain accuracy and domain classifier accuracy of DANN. Average source accuracy remains $\geq 95\%$ and average domain classifier remains $\approx 50\%$, indicating that DANN has managed to learn a representation that is suitable for the source domain and maps the points from the source and target domain close to each other. The large drop in DANN's performance can be attributed to the fact that the representation maps positive points in the source domain close to negative points in the target domain and vice-versa and therefore the joint error of the best hypothesis on the two domains (as described in Section 5) is large. We verify this by training a nearest neighbour (1-NN) classifier on the learned representation in the subsampled settings. The 1-NN classifier uses the source domain representations as the training data and the target domain representations as the test data. The accuracy of this classifier will suffer if in the learned representation the source domain points from one class are mapped close to the target domain points from the other class. The third row in the table, which is also averaged over the 45 tasks, shows a noticeable drop in the performance of the 1-NN classifier and indicates that this problem is present in the learned embeddings.

	U $\rightarrow$ M	M $\rightarrow$ U	0.3 $\rightarrow$ U	0.2 $\rightarrow$ U	0.1 $\rightarrow$ U
Source Accuracy	98.06	98.83	98.3	97.56	95.3
Discriminator Accuracy	46.4	50.63	50.42	50.48	50.44
1-NN Accuracy	-	-	77.89	73.22	69.75

Table 3: Source accuracy and domain classifier accuracy of DANN on MNIST-USPS. The drop in source accuracy under severe subsampling is minimal compared to the drop in target accuracy in the previous table. The domain classifier accuracy is near random regardless of the amount of subsampling. The performance of a nearest neighbour classifier trained on the mapped source data points and evaluated on the mapped target data points degrades to a large extent with more subsampling.

6.3 Cross-Lingual Sentiment Classification

This section includes cross-lingual sentiment analysis experiments on deep features. XED (Öhman et al., 2020) is a sentence-level sentiment analysis dataset consisting of 32 languages. We use English as the source domain and another language as the target domain and create 9 binary classification domain adaptation tasks.

There are a total of 1984 language pairs from each of the 32 languages to another. We chose 9 language pairs before running the experiment. The sentences in the dataset are labeled with one or more emotions *anger*, *anticipation*, *disgust*, *fear*, *joy*, *sadness*, *surprise*, and *trust*. Following the authors' guidelines we turn these multi-label classification tasks

to binary classification by labeling data points positive if their original labels only include *anticipation*, *joy*, and *trust*, and negative if the original labels only include *anger*, *disgust*, *fear*, and *sadness*. (*Surprise* is discarded.)

We perform 5 runs and in each one 100 points are randomly sampled from the target domain for validation and the rest are used for evaluation. Similar to the previous experiment, this validation set is used for all algorithms with hyperparameter configurations discussed in Appendix B to have a fair comparison. The representations for the source and target domain are 768-dimensional sentence embeddings obtained with BERT (Devlin et al., 2019) models pre-trained on the corresponding languages. The experiment compares Label Alignment Regression with the following baselines.

Source: This baseline trains a linear regression model with squared error (MSE) or a logistic regression model with crossentropy loss (CE) directly on the source domain and evaluates it on the target domain.

Adversarial-Refine (Conneau et al., 2018): This baseline uses a domain-adversarial approach to learn a linear transformation that maps the source and target domain into a shared space. A refinement step then encourages the transformation to be orthogonal. This approach has shown promising results in several cross-lingual NLP tasks with word embeddings. We train a linear regression model with squared error (MSE) or a logistic regression model with crossentropy loss (CE) on the source data using the learned shared space and then evaluate it on the target domain.

CDAN (Long et al., 2017): Recall that a domain-adversarial method consists of a domain classifier (discriminator) that predicts whether a data point is from the source or the target domain, a generator that learns a shared embedding between the two domains, and a label predictor that performs classification on the task of interest using the generator's embedding. CDAN improves on DANN by conditioning the domain classifier on the label predictor's prediction. The motivation is the improvements observed by incorporating this modification to generative adversarial networks (Goodfellow et al., 2020; Mirza and Osindero, 2014).

f-DAL (Acuna et al., 2021): This approach modifies DANN to use a separate domain classifier for each class which allows minimizing a family of divergence measures between the source and domain embeddings. We use f-DAL to minimize Pearson χ^2 divergence as the authors had observed superior performance with this divergence in previous vision and NLP benchmarks.

IWDAN and IWCDAN (Tachet et al., 2020): These two methods modify DANN and CDAN by incorporating importance sampling to reduce deterioration in performance due to class imbalance. Computing the importance sampling ratios requires access to target domain labels. The authors propose to estimate the ratios and provide the theoretical requirements for the estimation to be accurate.

Table 4 shows F_1 scores for the nine tasks and the average score over the tasks. The first 8 rows are the baselines above and in the last row (LAR) we employ the Label Alignment Regression algorithm. The proposed algorithm achieves the highest F_1 score on seven out of the nine tasks as well as on average over the tasks. Adv - Refine, CDAN, and f-DAL do

not provide a consistent benefit over No Adaptation. The two methods with importance weightings, IWDAN and IWCDAN, find better solutions than No Adaptation as well as the other domain-adversarial methods, suggesting that the reweighting in this algorithm, even if it is an estimate of the true importance weighting, is beneficial.

	en → bg	en → br	en → cn	en → da	en → de
Source (MSE)	55.22 (0.23)	53.51 (0.53)	4.48 (0.27)	64.75 (0.14)	47.95 (0.52)
Source (CE)	51.55 (0.12)	56.94 (0.04)	0.37 (0.00)	64.00 (0.18)	46.55 (0.22)
Adv-R (MSE)	46.88 (0.61)	46.20 (1.56)	53.54 (1.74)	51.98 (1.87)	50.43 (1.07)
Adv-R (CE)	45.12 (0.82)	36.96 (0.95)	49.87 (1.37)	50.99 (1.31)	43.62 (0.66)
CDAN	49.99 (5.43)	31.97 (8.43)	21.93 (12.76)	55.80 (5.12)	33.52 (9.60)
f-DAL	51.95 (0.88)	**57.79 (0.50)**	15.04 (3.65)	64.23 (0.14)	45.74 (0.82)
IWDAN	56.16 (1.05)	55.95 (0.50)	46.78 (3.55)	63.41 (1.20)	46.30 (5.69)
IWCDAN	57.70 (1.32)	54.96 (1.44)	42.08 (5.85)	63.64 (1.49)	41.86 (6.48)
LAR	**59.85 (0.08)**	53.42 (0.33)	**65.10 (0.24)**	**65.58 (0.12)**	**60.46 (0.05)**
	en → es	en → fr	en → he	en → hu	Average
Source (MSE)	39.17 (0.93)	49.89 (0.57)	58.19 (0.23)	59.66 (0.12)	48.09 (0.37)
Source (CE)	47.09 (0.20)	40.90 (0.36)	58.23 (0.10)	55.82 (0.06)	46.83 (0.10)
Adv-R (MSE)	**48.37 (1.38)**	46.45 (0.88)	48.93 (1.03)	47.15 (1.18)	48.88 (0.69)
Adv-R (CE)	41.30 (1.40)	44.18 (2.16)	46.95 (1.39)	44.06 (1.44)	44.78 (0.40)
CDAN	21.26 (6.97)	36.30 (14.82)	41.29 (9.49)	34.95 (6.14)	36.33 (3.73)
f-DAL	48.25 (12.06)	**58.23 (0.62)**	60.10 (0.31)	47.42 (1.28)	49.86 (1.53)
IWDAN	36.21 (2.93)	54.04 (1.72)	58.18 (0.97)	56.00 (1.54)	52.56 (0.84)
IWCDAN	37.63 (2.45)	52.14 (2.39)	61.08 (0.56)	55.82 (1.54)	51.88 (1.24)
LAR	43.47 (0.90)	58.11 (0.24)	**61.24 (0.09)**	**59.68 (0.12)**	**58.55 (0.17)**

Table 4: F_1 score in percents on different XED source-language pairs. The numbers in parentheses are standard errors. Adv-R refers to Adversarial-Refine. MSE and CE denote Mean Squared Error and Cross Entropy loss. LAR (Label Alignment Regression) outperforms the baselines on average and on most of the tasks. For adversarial baselines we verified that the discriminator accuracy is near random in this experiment similar to the MNIST-USPS experiment.

7. Discussion and Limitations

In this work, we proposed a domain adaptation regularization method based on the observation of label alignment property—the label vector of a dataset usually lies in the top left singular vectors of the feature matrix. We show that a regression algorithm in a standard supervised learning task actually contains an implicit regularization method to enforce such a property. Then we demonstrate how we can adapt such a regularization method in a domain adaptation setting. A critical difference between our algorithm and the conventional domain adaptation method is that we do not use regularization to adjust the representation learning. We observe that our algorithm does work well under high imbalance, where the conventional representation-based domain adaptation method fails. We also report improvement over baselines on cross-lingual sentiment analysis tasks.

Immediate next steps are providing an unsupervised hyperparameter selection strategy and extension to multi-class classification. The current method uses a validation set for

choosing the hyperparameters. This validation set is remarkably small and on the NLP tasks we found little benefit from involving this set to train a semi-supervised method.

A better hyperparameter selection strategy can also help with applying the proposed method to multi-class classification problems. In Appendix D we discuss how the regularizer can be extended to multi-class problems using multiple outputs and one-hot labels. In general, the multi-class version would require tuning the hyperparameters separately for each output and the current grid search method would become expensive with large number of classes or fine grids. Using a fixed set of hyperparameter values for all the outputs, we show promising results on the MNIST-USPS benchmark in the same section and leave further exploration to future work.

Other future directions are to investigate the combination of our method and the conventional representation-based domain adaptation method, with the hope that the hybrid method has the advantage of both—it can provide a significant advantage in a broad range of domain-shift settings. It would also be interesting to have a more rigorous theoretical characterization regarding when the label alignment property holds and to what extent the label vector can align with the top singular vectors.

Acknowledgments

The authors would like to thank the members of Huawei Noah's Ark Lab, the Reinforcement Learning and Artificial Intelligence lab at the University of Alberta, and Torr Vision Group at the University of Oxford for helpful discussions. This research was funded and supported by Huawei, National Sciences and Engineering Research Council of Canada (NSERC), Canada CIFAR AI Chairs program, and Digital Research Alliance of Canada.

Table of Contents

Appendix A. Label Alignment Property

In the proposition below we show that label alignment emerges if multiple features are highly correlated with the labels. The following lemma is needed for the proof.

Lemma 7 *If there are $k' < d$ orthonormal vectors $\{\nu_1, \cdots, \nu_{k'}\}$ such that $\|\Phi\nu_i\| < \epsilon$ for all $i \in [k']$ then $\Phi_{n\times d}$ has at most $d - k'$ singular values greater than or equal to $\sqrt{k'}\epsilon$.*

Proof Suppose $\sigma_1, \cdots, \sigma_d$ are the singular values of Φ sorted in descending order. The matrix $N_{d\times k'}$ with orthonormal columns that minimizes $\|\Phi N\|_2$ is the matrix of the last k' right singular vectors, and $\|\Phi N\|_2 = \sqrt{\sum_{i=d-k'+1}^{d} \sigma_i^2} \geq \sigma_{d-k'+1}$ (This easily follows from Section 12.1.2 by Bishop and Nasrabadi (2006)). If $\sigma_{d-k'+1} \geq \sqrt{k'}\epsilon$ then for any N with orthonormal columns we have $\|\Phi N\|_2 \geq \sqrt{k'}\epsilon \implies \|\Phi N\|_\infty \geq \epsilon$ which contradicts the assumption. ■

Proposition 8 *Suppose $\|y\| = 1$ and that columns of Φ are normalized. If $\Phi_{n\times d}$ has $\hat{k} \leq d$ columns $\{\phi_1, \cdots, \phi_{\hat{k}}\}$ where $|\phi_i^\top y| > 1 - \delta$ for all $i \in [\hat{k}]$ and*

- $0 < \delta < 0.2$
- $\hat{k} > 16\delta^2/(-15\delta^2 - 2\delta + 1)$
- $d > 16\delta^2(\hat{k} - 1)$

then the norm of the projection of y on the span of the first $k = d - \hat{k} + 1$ left singular vectors of Φ is greater than

$$\sqrt{\frac{\hat{k}(1-\delta)^2 - 16\delta^2(\hat{k}-1)}{d - 16\delta^2(\hat{k}-1)}}.$$

Proof First suppose the dot products in the statement are positive.

Note that for all $i \in \hat{k}$ we have $\|\phi_i - y\|_2^2 = (\phi_i - y)^\top(\phi_i - y) = \phi_i^\top\phi_i + y^\top y - 2\phi_i^\top y = 2 - 2\phi_i^\top y < 2\delta$. Due to triangle inequality, $\|\phi_i - \phi_j\|_2^2 \leq \|\phi_i - y\|_2^2 + \|\phi_j - y\|_2^2 < 4\delta$.

The span of $\phi_{\hat{k}}, \phi_{\hat{k}+1}, \cdots \phi_d$ has at most $d - \hat{k} + 1$ dimensions. Choose $\hat{k} - 1$ orthonormal vectors $\nu_1, \cdots, \nu_{\hat{k}-1} \in \mathbb{R}^d$ that are perpendicular to this subspace. Then for any $i, j \in [\hat{k} - 1]$

we have $\phi_i^\top \nu_j = (\phi_i - \phi_{\hat{k}} + \phi_{\hat{k}})^\top \nu_j = (\phi_i - \phi_{\hat{k}})^\top \nu_j + 0 \leq \left\|\phi_i - \phi_{\hat{k}}\right\| < 4\delta$. Therefore $\|\Phi\nu_j\| < 4\delta\sqrt{\hat{k}-1}$. Putting this orthonormal basis in the lemma above gives that Φ has at most $d - \hat{k} + 1$ singular values greater than or equal to $4\delta(\hat{k}-1)$.

Now see that $\left\|\Phi^\top y\right\|^2 = \left\|\sum_{i=1}^d \phi_i^\top y\right\|^2$ and is also equal to $\sum_{i=1}^d (\sigma_i y_i^U)^2$. Therefore $\sum_{i=1}^d (\sigma_i y_i^U)^2 \geq \left\|\sum_{i=1}^{\hat{k}} \phi_i^\top y\right\|^2 > \hat{k}(1-\delta)^2$. Since the columns are normalized, $\sum_{i=1}^d \sigma_i^2 = \|\Phi\|_F = d$. In addition, we have shown that the last $\hat{k}-1$ singular values are smaller than $4\delta\sqrt{\hat{k}-1}$. Define $\hat{y}$ as the projection of y on the first $d - \hat{k} + 1$ singular vectors of Φ. Then we have $\hat{k}(1-\delta)^2 < \sum_{i=1}^d (\sigma_i y_i^U)^2 = \sum_{i=1}^{d-\hat{k}+1} (\sigma_i y_i^U)^2 + \sum_{i=d-\hat{k}+2}^d (\sigma_i y_i^U)^2 < d\|\hat{y}\|^2 + 16\delta^2(\hat{k}-1)(1-\|\hat{y}\|^2)$. Rearranging the terms (with the extra conditions in the proposition statement) gives

$$\|\hat{y}\| > \sqrt{\frac{\hat{k}(1-\delta)^2 - 16\delta^2(\hat{k}-1)}{d - 16\delta^2(\hat{k}-1)}}$$

The inequality is tight in the extreme case where $\hat{k} = d$ and $\delta \to 0$ which results in the label vector being fully in the direction of the first left singular vector and all the other singular values tending to zero.

Now suppose some of the dot products in the statement are negative. We can multiply those columns with -1 and prove the result above for this modified matrix. The result holds for the original matrix since this operation only changes the right singular vectors of Φ and does not affect the left singular vectors or the singular values. ■

Let us demonstrate the emergence of alignment and the behavior of the bound when multiple features are highly correlated with the output. In this toy experiment the label vector is sampled from a 1000-dimensional Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbb{I})$ with mean zero and standard deviation 1 and then normalized to norm one. The matrix Φ has 10 columns. The first 9 columns are sampled from $\mathcal{N}(y, s^2\mathbb{I})$ with mean y and a small standard deviation s and the other column is sampled from $\mathcal{N}(\mathbf{0}, \mathbb{I})$. All columns are then normalized to norm one. Note that the proposition above does not assume Gaussian features. Figure 3 shows the norm of the projection of the label vector on the first two singular vectors at different levels of s and its relationship with δ.

Appendix B. Proofs in Section 4

Lemma 9 *In the rotated Gaussian example in Section 4,* $\mathbb{E}_{x,y}[xy] = \sqrt{\frac{2}{\pi}} s_1 p_1$.

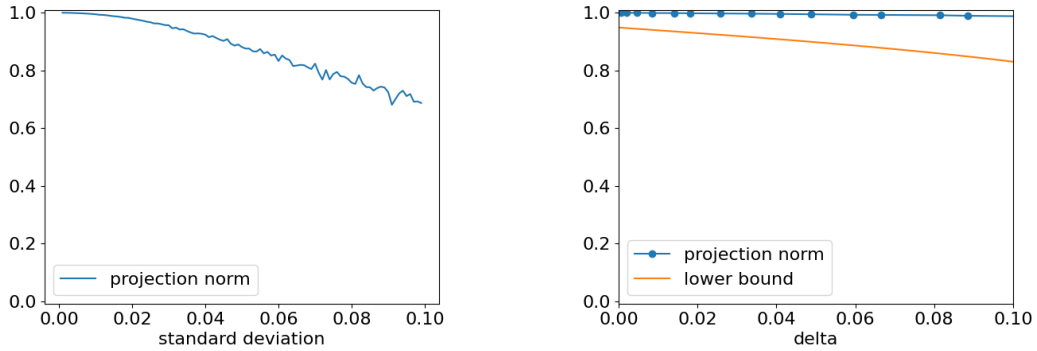


Figure 3: Projection of the label vector on the top two singular vectors in the Gaussian example. For small values of standard deviation (where the labels are highly correlated with the features) and small values of δ, the label vector is mostly in the direction of the top two singular vectors. The lower bound is applicable in this regime and is close to one.

Proof

$$\frac{1}{n}\Phi^\top y \approx \mathbb{E}_{x,y}[xy] \tag{24}$$

$$= \int_x \int_y xyp_{\mathcal{S}}(x|y)p(y)dydx \tag{25}$$

$$= \int_x xp_{\mathcal{S}}(x|y=1)p(y=1) - xp_{\mathcal{S}}(x|y=-1)p(y=-1)dx \tag{26}$$

$$= \int_x x \cdot 2\mathcal{N}(0,Q)\big(\mathbb{1}(x_1^P > 0)p(y=1) - \mathbb{1}(x_1^P < 0)p(y=-1)\big)dx, \tag{27}$$

where we plug into the definition (9) to get the last equality. Further note that $\mathbb{1}(x_1^P < 0) = 1 - \mathbb{1}(x_1^P > 0)$ and plug this into above,

$$(27) = \int_x x \cdot 2\mathcal{N}(0,Q)\big(\mathbb{1}(x_1^P > 0) - p(y=-1)\big)dx \tag{28}$$

$$= \int_x x \cdot 2\mathcal{N}(0,Q)\mathbb{1}(x_1^P > 0)dx \tag{29}$$

$$= 2P^\top \int_z z \cdot \frac{1}{2\pi s_1 s_2} \exp\left(-\frac{1}{2s_1^2}z_1^2 - \frac{1}{2s_2^2}z_2^2\right)\mathbb{1}(z_1 > 0)dz \tag{30}$$

where in the last equality we let $z = Px$, then $x_1^P = x^\top p_1 = z^\top P p_1 = z_1$. The integral above is a vector with two elements because it includes z. The first element is

$$\int_{z_1}\int_{z_2} z_1 \cdot \frac{1}{2\pi s_1 s_2} \exp\left(-\frac{1}{2s_1^2}z_1^2 - \frac{1}{2s_2^2}z_2^2\right) \mathbb{1}(z_1 > 0) dz_1 dz_2 \tag{31}$$

$$= \int_{z_1} z_1 \cdot \frac{1}{\sqrt{2\pi} s_1} \exp\left(-\frac{1}{2s_1^2}z_1^2\right) \mathbb{1}(z_1 > 0) dz_1 \tag{32}$$

$$= \frac{1}{2}\int_0^{+\infty} z_1 \cdot \frac{\sqrt{2}}{\sqrt{\pi} s_1} \exp\left(-\frac{1}{2s_1^2}z_1^2\right) dz_1 \tag{33}$$

$$= \frac{1}{2} s_1 \sqrt{\frac{2}{\pi}} \tag{34}$$

The last equality is because the integration is the mean of half-normal distribution. The second element would become zero as written below and noting that $\mathbb{E}[z_2]$ is the mean of a zero-mean Gaussian random variable:

$$\int_{z_1}\int_{z_2} z_2 \cdot \frac{1}{2\pi s_1 s_2} \exp\left(-\frac{1}{2s_1^2}z_1^2 - \frac{1}{2s_2^2}z_2^2\right) \mathbb{1}(z_1 > 0) dz_1 dz_2 \tag{35}$$

$$= \int_{z_1} \frac{1}{\sqrt{2\pi} s_1} \exp\left(-\frac{1}{2s_1^2}z_1^2\right) \mathbb{1}(z_1 > 0) dz_1 \mathbb{E}[z_2] = 0 \tag{36}$$

Then

$$\mathbb{E}_{x,y}[xy] = 2P^\top \begin{bmatrix} \frac{1}{2} s_1 \sqrt{\frac{2}{\pi}} \\ 0 \end{bmatrix} = \sqrt{\frac{2}{\pi}} s_1 p_1. \tag{37}$$

■

Proposition 1 *In the example in this section suppose $v_1^\top \tilde{v}_1 \neq 0$. Then the label alignment solution is $\widehat{w^*} = c w_{\mathcal{T}}^* / v_1^\top \tilde{v}_1$ with $c > 0$.*

Proof We rewrite (16) as:

$$(s_1^2 v_1 v_1^\top + \lambda \tilde{s}_2^2 \tilde{v}_2 \tilde{v}_2^\top)\widehat{w^*} = \sqrt{\frac{2}{\pi}} s_1 v_1. \tag{38}$$

Suppose $\widehat{w^*} = w_1 \tilde{v}_1 + w_2 \tilde{v}_2$ with $w_1 \in \mathbb{R}, w_2 \in \mathbb{R}$, then the equation above becomes:

$$s_1^2 v_1 (v_1^\top \widehat{w^*}) + \lambda \tilde{s}_2^2 w_2 \tilde{v}_2 = \sqrt{\frac{2}{\pi}} s_1 v_1. \tag{39}$$

Apply $v_2^\top$ on both sides we have:

$$\lambda \tilde{s}_2^2 w_2 \tilde{v}_2^\top v_2 = 0. \tag{40}$$

Since $\tilde{v}_2$ is not parallel to v_1, we must have $\tilde{v}_2^\top v_2 \neq 0$ and thus $w_2 = 0$. To obtain the exactly value of w_1, solve (39) by setting $w_2 = 0$, we have: $s_1^2 (v_1^\top \tilde{v}_1) w_1 = \sqrt{\frac{2}{\pi}} s_1$, $w_1 = \sqrt{\frac{2}{\pi}} \frac{1}{s_1 v_1^\top \tilde{v}_1}$.
■

Theorem 2 *Assume $k = \tilde{k}$ and $(V'_{d-k})^\top \tilde{V}'_{d-k}$ is invertible with $V'_{d-k} = \begin{bmatrix} v_{k+1} & \dots & v_d \end{bmatrix}$ and $\tilde{V}'_{d-k} = \begin{bmatrix} \tilde{v}_{\tilde{k}+1} & \dots & \tilde{v}_d \end{bmatrix}$, then $\widehat{w^*} \in \operatorname{span}(\tilde{v}_1, \dots, \tilde{v}_k)$ holds and $\widehat{w^*}$ is independent of λ.*

Proof From the definition of $\widehat{w^*}$, we see that:

$$(\sum_{i \le k} \sigma_i^2 v_i v_i^\top + \lambda \sum_{j > \tilde{k}} \tilde{\sigma}_j^2 \tilde{v}_j \tilde{v}_j^\top) \widehat{w^*} = \sum_{i \le k} \sigma_i y_i^U v_i. \tag{41}$$

Decompose $\widehat{w^*} = \sum_{i \le d} w_i \tilde{v}_i$. From the equation above we find:

$$\sum_{i \le k} \sigma_i^2 v_i v_i^\top \widehat{w^*} + \lambda \sum_{j > \tilde{k}} \tilde{\sigma}_j^2 \tilde{v}_j w_j = \sum_{i \le k} \sigma_i y_i^U v_i. \tag{42}$$

Applying $v_m^\top$ only both sides with $m > k$, we have:

$$\sum_{j > \tilde{k}} \tilde{\sigma}_j^2 v_m^\top \tilde{v}_j w_j = 0,\ m > k, \tag{43}$$

which can be written as:

$$(V'_{d-k})^\top \tilde{V}'_{d-k} \operatorname{diag}(\tilde{\sigma}_{\tilde{k}+1}^2, \dots, \tilde{\sigma}_d^2) \begin{bmatrix} w_{\tilde{k}+1} \\ \dots \\ w_d \end{bmatrix} = \mathbf{0}. \tag{44}$$

Note that multiplying by $\operatorname{diag}(\tilde{\sigma}_{\tilde{k}+1}^2, \dots, \tilde{\sigma}_d^2)$ does not change the invertibility. By assumption we must have $w_{k+1} = \dots = w_d = 0$, and (42) becomes independent of λ. ■

Theorem 3 *Given invertible $V_k^\top \widetilde{V}_k$, with $V_k = [v_1 \dots v_k]$, $\widetilde{V}_{\tilde{k}} = [\tilde{v}_1 \dots \tilde{v}_{\tilde{k}}]$ and with the same assumptions of Theorem 2, there exists $c > 0$ such that $\widehat{w^*} = c w_{\mathcal{T}}^*$ iff:*

$$\mu_k = c V_k^\top \widetilde{V}_k \tilde{\mu}_k = c V_k^\top w_{\mathcal{T}}^*. \tag{19}$$

Proof With the assumption of Theorem 2, we have: $v_i^\top \widehat{w^*} = y_i^U / \sigma_i$, for $i \le k$. Then we can write down the optimal solutions as $w_{\mathcal{S}}^* = V_k \mu_k$, $w_{\mathcal{T}}^* = \widetilde{V}_{\tilde{k}} \tilde{\mu}_{\tilde{k}}$. Since $V_k^\top \widetilde{V}_k$ is invertible (and under the assumptions of Theorem 2), then we obtain the label alignment regularized result

$$\widehat{w^*} = \widetilde{V}_k (V_k^\top \widetilde{V}_k)^{-1} \mu_k. \tag{45}$$

Note that the solution is independent of the hyperparameter λ.

From $\widehat{w^*} = \widetilde{V}_k (V_k^\top \widetilde{V}_k)^{-1} \mu_k$, and $w_{\mathcal{T}}^* = \widetilde{V}_{\tilde{k}} \tilde{\mu}_{\tilde{k}}$, the equation $\widehat{w^*} = c w_{\mathcal{T}}^*$ holds iff:

$$\widetilde{V}_k ((V_k^\top \widetilde{V}_k)^{-1} \mu_k - c \tilde{\mu}_k) = \mathbf{0}, \tag{46}$$

or in other words, $(V_k^\top \widetilde{V}_k)^{-1} \mu_k = c \tilde{\mu}_k + q$, where $q \in \operatorname{null}(\widetilde{V}_k) = \{\mathbf{0}\}$. ■

Proposition 5 *Suppose $\epsilon < \min\{\frac{1}{k}, \frac{1}{d-k}\}$, $|v_i^\top \tilde{v}_j| \leq \epsilon$ for any $i \neq j$, and $v_i^\top \tilde{v}_i \geq 1 - \epsilon$ for any i, then both $V_k^\top \widetilde{V}_k$ and $(V'_{d-k})^\top \tilde{V}'_{d-k}$ are invertible.*

Proof $V_k^\top \widetilde{V}_k$ can be written as $[v_i^\top \tilde{v}_j]$ with $i, j \in [k]$. From the assumption, $V_k^\top \widetilde{V}_k = \mathbb{I} + \epsilon\Delta$ where $\mathbb{I}$ is the identity matrix and Δ is a $k \times k$ matrix with each element $|\Delta_{ij}| \leq \epsilon$. Suppose $(\mathbb{I} + \epsilon\Delta)x = \mathbf{0}$, then $x = -\epsilon\Delta x$, taking the norm on both sides we have:

$$\|x\| = \epsilon\|\Delta x\| \leq \epsilon\|\Delta\| \cdot \|x\| \tag{47}$$

$$\leq \epsilon\|\Delta\|_F \cdot \|x\| \leq \epsilon k \cdot \|x\| < \|x\|, \tag{48}$$

which gives $x = \mathbf{0}$. Therefore, $\mathbb{I} + \epsilon\Delta$ is invertible. Similarly, $(V'_{d-k})^\top \tilde{V}'_{d-k}$ is also invertible. ■

Proposition 6 *Suppose the target singular vectors $\tilde{v}_1, \ldots, \tilde{v}_d$ satisfies the following probability distribution:*

$$p(\tilde{v}_1, \ldots, \tilde{v}_d) = p(\tilde{v}_1)p(\tilde{v}_2|\tilde{v}_1) \ldots p(\tilde{v}_d|\tilde{v}_1, \ldots, \tilde{v}_{d-1}), \tag{20}$$

where $p(\tilde{v}_1)$ is a continuous distribution on $\mathbb{S}^{d-1}$ and each $p(\tilde{v}_i|\tilde{v}_1, \ldots, \tilde{v}_{i-1})$ is a continuous distribution on the manifold $\mathbb{S}^{d-1}$ for $2 \leq i \leq d$. Then $V_k^\top \widetilde{V}_k$ is invertible almost surely.

Proof It suffices to show that $P(\det(V_k^\top \widetilde{V}_k) = 0) = 0$. Note that $\det(V_k^\top \widetilde{V}_k) = 0$ can be rewritten as:

$$\det(\tilde{v}_1^{V_k} \ldots \tilde{v}_k^{V_k}) = 0, \tag{49}$$

and thus

$$P(\det(V_k^\top \widetilde{V}_k) = 0) \leq p(\tilde{v}_1^{V_k} = 0) + p(\tilde{v}_2^{V_k} \in \text{span}(\tilde{v}_1^{V_k})|\tilde{v}_1) + \cdots + \tag{50}$$

$$p(\tilde{v}_k^{V_k} \in \text{span}(\tilde{v}_1^{V_k}, \ldots, \tilde{v}_{k-1}^{V_k})|\tilde{v}_1, \ldots, \tilde{v}_{k-1}). \tag{51}$$

Since each condition gives a sub-manifold with a smaller dimension and the probability distributions are continuous, from Sard's theorem (e.g. Guillemin and Pollack, 2010), each probability is zero. Therefore, $P(\det(V_k^\top \widetilde{V}_k) = 0) = 0$. ■

Appendix C. Experiment Details

This section outlines dataset details, hyperparameter settings, and other design choices in the experiments.

Label Alignment in Real-World Tasks. We used the following tasks in this experiment:

1. **UCI CT Scan:** A random subset of the CT Position dataset on UCI (Graf et al., 2011). The task is predicting a location of a CT Slice from histogram features.

2. **Song Year:** A random subset of the training portion of the Million Song dataset (Bertin-Mahieux et al., 2011). The task is predicting the release year of a song from audio features.

3. **Bike Sharing:** A random subset of the Bike Sharing dataset on UCI (Fanaee-T and Gama, 2014). The task is predicting the number of rented bikes in an hour based on information about weather, date, and time.

4. **MNIST:** The task is classifying any pair of two digits in MNIST.

5. **USPS:** The task is classifying any pair of two digits in USPS.

6. **CIFAR-10:** The task is classifying airplane and automobile in CIFAR-10 dataset using features from a ResNet-18 pretrained on ImageNet.

7. **CIFAR-100:** The task is classifying beaver and dolphin in CIFAR-100 dataset using features from a ResNet-18 pretrained on ImageNet.

8. **STL-10:** The task is classifying airplane and bird in STL-10 dataset using features from a ResNet-18 pretrained on ImageNet.

9. **XED (English):** The English corpus from XED datsets whose details are discussed in the main paper. The features are sentence embeddings extracted from BERT.

10. **AG News:** A random subset of the first two classes (World and Sports) in AG News document classification dataset. The features are obtained by feeding the document text to BERT.

All datasets have an extra constant 1 feature to account for the bias unit. Rank is computed as the number of singular values larger than $\sigma_1 * \max(n, d) * 1.19209e-07$. This is the default numerical rank computation method in the Numpy package.

MNIST-USPS. DANN uses a one-layer ReLU neural network. This is the Shallow DANN architecture suggested by the original authors (Ganin et al., 2016). We swept over values of 128, 256, 512, and 1024 for the depth of the hidden layer.

The neural network is trained for 10 epochs using SGD with batch size 32, learning rate 0.01, and momentum 0.9. This model already achieves near perfect accuracy on the source domain.

Candidate hyperparameter values for Label Alignment Regularizer were $\{1e-1, 1e+1, 1e+3\}$ for λ and $\{8, 16, 32, \cdots\}$ up to the rank of Φ or $\tilde{\Phi}$ for k and $\tilde{k}$.

Although the number of hyperparameter configurations is greater for our method, this experiment is in favor of DANN if we take runtime into account.

The linear model is trained using full-batch gradient descent for 5000 epochs with learning rate $1/(2\sigma_1)$.

Sentiment Classification. For the domain-adversarial baseline (Conneau et al., 2018) we sweep over values of $\{1e-3, 1e-2, 1e-1\}$ for β. The parameter controls the degree of orhtogonality of the transformation that maps the source and target embeddings into a common space.

The models used in No Adaptation, Adv - Refine, and Label Alignment Regression are linear regression or logistic regression (on the nonlinear extracted representations). These models are trained with learning rate $1/(2\sigma_1)$ (MSE loss) and $1e-2$ (CE loss) and momentum 0.9.

For CDAN and f-DAL we sweep over regularization coefficients $\{1e-4, 1e-2, 1\}$ with a one-hidden-layer ReLU network. This is the architecture suggested by (Ganin et al., 2016) for domain adaptation with a shallow network.

Candidate hyperparameter values for Label Alignment Regularizer were $\{1e-1, 1e+1, 1e+3\}$ for λ and $\{8, 16, 32, \cdots\}$ up to the rank of Φ or $\tilde{\Phi}$ for k and $\tilde{k}$.

Similar to the previous experiment, the hyperparameter grid search in this experiment is in favor of the baselines if we take runtime into account.

Appendix D. Additional Experiments

D.1 Multi-Class Classification

We also try to generalize the formulation of the label alignment regression to a multiclass setting following the derivation in Equation 3 . Given a dataset comprising n samples, each characterized by d features, we denote the feature matrix by $\Phi \in \mathbb{R}^{n\times d}$. In a classification context involving c distinct classes and employing a one-versus-all strategy, the target matrix Y, which adopts a ± 1 style one-hot encoding scheme, is of dimension $\mathbb{R}^{n\times c}$. Consequently, the weight matrix W, which maps the feature space to the class labels, is represented as $\mathbb{R}^{d\times c}$. Then the learning objective can be formulated as:

$$\begin{aligned}
\min_W \|\Phi W - Y\|^2 &= \min_W \|U\Sigma V^T W - Y\|^2 \\
&= \min_W \|\Sigma V^T W - U^T y\|^2 \\
&= \min_W \|\Sigma W^V - Y^U\|^2 \\
&= \min_W \sum_{j=1}^{c}\sum_{i=1}^{d}(\sigma_i W_{ij}^Y - Y_{ij}^U)^2 + \sum_{j=1}^{c}\sum_{i=d+1}^{n}(Y_{ij}^U)^2
\end{aligned} \tag{52}$$

The notation $\|A\|^2$ signifies the 2-norm of matrix A, encapsulating the square root of the sum of the squares of its elements. In this setup, W^V corresponds to a weight matrix with dimensions $\mathbb{R}^{d\times c}$, while Y^U denotes a modified target matrix also of dimension $\mathbb{R}^{n\times c}$. Thus, the expression $(\Sigma W^V - Y^U)$, representing the discrepancy between the projected feature space and the modified target matrix, retains the dimensionality of $\mathbb{R}^{n\times c}$, highlighting the alignment or misalignment of the model predictions with the modified targets in the given multidimensional space.

Assume k is the same for every one vs all setting and $k < d$, we can obtain

$$\begin{aligned}\min_W \|\Phi W - Y\|^2 &= \min_W \sum_{j=1}^{c}\sum_{i=1}^{d}(\sigma_i W_{ij}^Y - Y_{ij}^U)^2 + \sum_{j=1}^{c}\sum_{i=d+1}^{n}(Y_{ij}^U)^2 \\ &= \min_W \sum_{j=1}^{c}\sum_{i=1}^{d}(\sigma_i W_{ij}^V - Y_{ij}^U)^2 \\ &= \min_W \sum_{j=1}^{c}\sum_{i=1}^{k}(\sigma_i W_{ij}^V - Y_{ij}^U)^2 + \sum_{j=1}^{c}\sum_{i=k+1}^{d}(\sigma_i W_{ij}^V)^2.\end{aligned} \tag{53}$$

Therefore the final objective function looks like

$$\min_W \|\Phi W - Y\|^2 - \sum_{j=1}^{c}\sum_{i=k+1}^{d}(\sigma_i W_{ij}^V)^2 + \lambda \sum_{j=1}^{c}\sum_{i=\tilde{k}+1}^{d}(\tilde{\sigma}_i W_{ij}^{\tilde{V}})^2. \tag{54}$$

We also validated the label alignment property by computing $k(0.1)$ for the one-versus-all label vector corresponding to each digit similar to the binary classification case in Table 1. The value of $k(0.1)$ for all the digits was 1.

Then, we compare the classification performance of our label alignment regression to DANN in the multiclass MNIST-USPS classification setting. Our method outperforms DANN by a large margin as shown by the evaluation results in Table 5.

	U → M	M → U	0.3 → U	0.2 → U	0.1 → U
No Adaptation (NN)	35.66	52.46	47.24	45.06	19.48
DANN	41.32	53.46	49.88	43.09	32.01
No Adaptation (Linear)	37.53	54.41	48.71	46.21	41.54
Label Alignment Regression	**42.47**	**63.90**	**54.69**	**51.70**	**47.49**

Table 5: Accuracies on MNIST-USPS multiclass benchmark. M and U indicate MNIST and USPS. Ratios (0.3, 0.2, 0.1) indicate MNIST tasks where 9 out of the 10 digits are subsampled. For the subsampling setting (last three columns), each column is averaged over the 10 subsampling classification tasks. In all tasks, the proposed algorithm improves the accuracy and achieves the highest performance.

D.2 Parameter Sensitivity

The parameter λ indicates the ratio of the loss obtained from the unsupervised information of the target domain. We want to quantitatively evaluate how this ratio influences the performance of our proposed method on the target domain. The sensitivity visualization is shown in Figure 4.

D.3 Regression

We create a regression task similar to the synthetic experiment in the main paper. The aim is to understand if results similar to the synthetic experiment hold in a setting where the

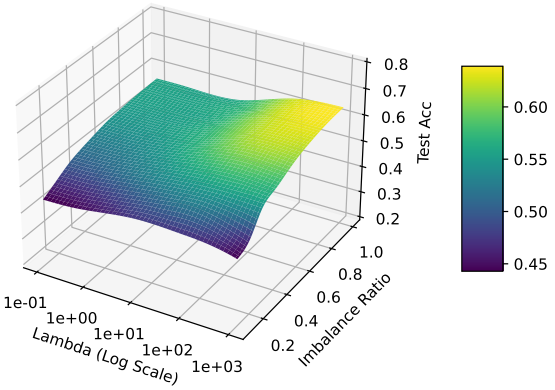
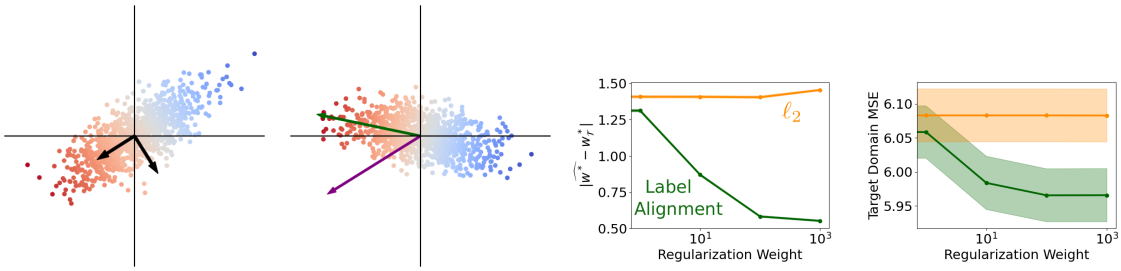


Figure 4: λ sensitivity curves of accuracies on MNIST USPS multiclass benchmark. The performance of the proposed method is relatively invariant over different λ under various imbalance (subsampling) ratios. Generally greater λ comes with better performance in the target domain because more weight and emphasis of loss is put on the information of the target domain.

labels are not restricted to ± 1 and where mean squared error is used for evaluation. All the details are the same as in the classification experiment in the main paper except that the label vector is simply set to the first left singular vector. Figure 5 shows the results and corroborates the findings in the main paper. Note that DANN is not directly applicable here.



(a) Source domain (b) Target Domain (c) Parameter Error (d) Performance

Figure 5: (a) Source domain. The black arrows show principal components. (b) Target domain. The arrows show weights found without using any regularization (purple) and with our regularizer with $\lambda = 10^3$ (green). (c) Distance between the estimated and the optimal weights. The proposed regularizer reduces this distance. (d) Performance on the target domain. The x axis is the regularization coefficient for ℓ_2 regularization and λ for the proposed regularizer. The proposed regularizer achieves lower error on this domain. Shaded areas are standard errors over 10 runs.

References

David Acuna, Guojun Zhang, Marc T Law, and Sanja Fidler. f-domain adversarial learning: Theory and algorithms. In *International Conference on Machine Learning*, 2021.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, 2019.

Aristide Baratin, Thomas George, César Laurent, R Devon Hjelm, Guillaume Lajoie, Pascal Vincent, and Simon Lacoste-Julien. Implicit regularization via neural feature alignment. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 2010.

Sagie Benaim and Lior Wolf. One-sided unsupervised domain mapping. *Advances in neural information processing systems*, 2017.

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 2011.

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *IEEE conference on computer vision and pattern recognition*, 2017.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations*, 2018.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in Neural Information Processing Systems*, 2017.

Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *European Conference on Computer Vision*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Hadi Fanaee-T and Joao Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2014.

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*, 2016.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European conference on computer vision*, 2016.

Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing*, 2020.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020.

Franz Graf, Hans-Peter Kriegel, Matthias Schubert, Sebastian Pölsterl, and Alexander Cavallaro. 2d image registration in ct images using radial image descriptors. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2011.

Victor Guillemin and Alan Pollack. *Differential topology*, volume 370. American Mathematical Soc., 2010.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. In *International Conference on Learning Representations*, 2020.

Ehsan Imani, Wei Hu, and Martha White. Representation alignment in neural networks. *Transactions on Machine Learning Research*, 2022.

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*, 2019.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

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

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*, 2016.

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

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *Annual Conference on Learning Theory*, 2009.

Zhong Meng, Jinyu Li, Yifan Gong, and Biing-Hwang Juang. Adversarial teacher-student learning for unsupervised domain adaptation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv*, 2014.

Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. *Advances in neural information processing systems*, 2017.

Emily Öhman, Marc Pàmies, Kaisla Kajava, and Jörg Tiedemann. XED: A multilingual dataset for sentiment analysis and emotion detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. What can linearized neural networks actually say about generalization? *Advances in Neural Information Processing Systems*, 2021.

Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *IEEE international conference on computer vision*, 2017.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11): 559–572, 1901.

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE international conference on robotics and automation*, 2018.

Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 2017.

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, 2017.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 2000.

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

Remi Tachet, 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*, 2020.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv*, 2014.

Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. Aspect-augmented adversarial networks for domain adaptation. *Transactions of the Association for Computational Linguistics*, 2017.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, 2019.

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

Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. Supervised representation learning: Transfer learning with deep autoencoders. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. Supervised representation learning with double encoding-layer autoencoder for transfer learning. *ACM Transactions on Intelligent Systems and Technology*, 2017.