Learning from Label Proportions and Covariate-shifted Instances

Sagalpreet Singh[1] **Navodita Sharma**[*†1] **Shreyas Havaldar**[*†2] **Rishi Saket**[1] **Aravindan Raghuveer**[1]

[1]Google DeepMind. {sagalpreet,navoditasharma,rishisaket,araghuveer}@google.com
[2]Columbia University, NY, USA. shreyas.havaldar@columbia.edu

Abstract

In many applications, especially due to lack of supervision or privacy concerns, the training data is grouped into bags of instances (feature-vectors) and for each bag we have only an aggregate label derived from the instance-labels in the bag. In learning from label proportions (LLP) the aggregate label is the average of the instance-labels in a bag, and a significant body of work has focused on training models in the LLP setting to predict instance-labels. In practice however, the training data may have fully supervised albeit covariate-shifted *source* data, along with the usual *target* data with bag-labels, and we wish to train a good instance-level predictor on the target domain. We call this the covariate-shifted hybrid LLP problem. Fully supervised covariate shifted data often has useful training signals and the goal is to leverage them for better predictive performance in the hybrid LLP setting. To achieve this, we develop methods for hybrid LLP which naturally incorporate the target bag-labels along with the source instance-labels, in the domain adaptation framework. Apart from proving theoretical guarantees bounding the target generalization error, we also conduct experiments on several publicly available datasets showing that our methods outperform LLP and domain adaptation baselines as well techniques from previous related work.

1 INTRODUCTION

Learning from label proportions (LLP) is a direct generalization of supervised learning where the training instances (i.e., feature-vectors) are partitioned into *bags* and for each bag only the average label of its instances is available as the *bag*-label. Full supervision is equivalent to the special case of unit-sized bags. In LLP, using bags of instances and their bag-labels, the goal is to train a good predictor of the instance-labels. Over the last two decades, LLP has been used in scenarios with lack of fully supervised data due to legal requirements [Rueping, 2010], privacy constraints [Wojtusiak et al., 2011] or coarse supervision [Chen et al., 2004]. Applications of LLP include image classification [Bortsova et al., 2018, Ørting et al., 2016], spam detection [Quadrianto et al., 2009a], IVF prediction [Hernández-González et al., 2018], and high energy physics [Dery et al., 2017]. More recently, restrictions on cross-site tracking of users has led to coarsening of previously available fine-grained signals which have been used to train large-scale models predicting user behavior for e.g. clicks or product preferences. Popular mechanisms (see Apple SKAN [ska] and Chrome Privacy sandbox [san]) aggregate relevant labels for bags of users resulting in LLP training data. Due to revenue criticality of user modeling in advertising, the study of LLP specifically for such applications has gained importance. A popular baseline method to train models using training bags and their bag-labels is to minimize a bag-level loss which for any bag is some suitable loss function between the the average prediction and the bag-label (see Ardehaly and Culotta [2017]). Other methods using different bag-level losses have also been proposed (e.g. Liu et al. [2021], Baručić and Kybic [2022]) for training models in the LLP setting.

One aspect of data in real-world applications is its heterogeneity, which introduces new aspects to the vanilla LLP modeling formulation. In particular, apart from bag-level data from the *target* distribution, the learner may have access to instance labels from a covariate-shifted *source* distribution. For example, in user behavior modeling for online advertising, while bag-level aggregate labels could be available for a target set of (privacy sensitive) users as mentioned above, other users may choose to share browsing and purchase history, which would yield covariate-shifted source data with instance-level labels. This is also mentioned in

[*]Work done while at Google DeepMind.
[†]Equal contribution.

Section 2.1 of O'Brien et al. [2022] which states: ".. *some platforms may continue to allow conversion tracking, and some users may also choose to allow conversion tracking , the training set is likely to contain some examples with individual labels and some examples with only group labels"*. This can also occur when the source originates from geographies which impose less stringent privacy constraints on data corresponding to online activity, medical records or financial transactions, thereby not requiring the aggregation of labels. Recent work has also studied age-dependent privacy, in which *releasing outdated data may lead to less privacy leakage if a user only focuses on protecting its real-time status* (from Section 1 of Zhang et al. [2022b], see also Lin et al. [2024]). Such outdated data could correspond to the source distribution for which instance-labels are available.

Here, we think of covariate-shift as a difference in $p(\mathbf{X})$ i.e. the distribution of feature-vectors, between the source $\mathcal{D}_S$ and target $\mathcal{D}_T$ distributions, with the conditional label distribution $p(Y \mid \mathbf{X})$ being the same on $\mathcal{D}_S$ and $\mathcal{D}_T$. We call this *covariate-shifted hybrid LLP* in which the goal is to leverage the full supervision on the source as well as the bag-level supervision on the target to train better instance-label predictors on the target distribution.

Previous works [Ardehaly and Culotta, 2016, Li and Culotta, 2023a] studied the case where the source training data was aggregated into bags whose bag-labels are available, while the training data from the target distribution is completely unsupervised. The work of Ardehaly and Culotta [2016] gave a *self-training* based approach where the model trained on the source data is used to predict bag-labels on the unsupervised target train-set from which a subset of the most confidently labeled bags are used (along with the source data) to retrain the predictor. The more recent work of Li and Culotta [2023a] proposed solutions directly applying domain adversarial neural-network (DANN) methods in which apart from minimizing the bag-level loss on the source data, an unsupervised domain prediction loss is *maximized* to ensure that the predictor is domain-independent.

The works of Ardehaly and Culotta [2016], and Li and Culotta [2023a] as well as standard domain adaptation methods (e.g. Long et al. [2015]) can be applied to our setting by simply ignoring the bag-labels of the target train-set, and treating the labeled instances in the source data as bags of size 1. Note however that these approaches discard the informative signal from the target bag-labels and are thus likely to degrade the predictive performance.

The main contributions of this paper are a suite of techniques which use the bag-labels from the target training set, not only to minimize the bag-loss i.e., the predictive loss on bags, but also to do better domain adaptation. We focus on regression as the underlying task and propose loss functions which, at a high level, have three components: (i) the instance-level loss on the source data, (ii) a bag-level loss on the target training bags, and (iii) a domain adaptation loss which leverages the instance-labels from source and bag-labels from target. Our main methodological novelty is the third term which leverages bag-labels (unlike previous works) from the target domain for domain adaptation, along with instance-labels from the source domain. Specifically, our BL-WFA method using the BagCSI loss (eqn. (3)) is the first to incorporate the instance-labels from the source along with the target bag-labels into the domain adaptation loss. The design of our BagCSI loss is theoretically justified: we prove generalization error bounds (Section 3.1), and we also generalize this to PL-WFA which can use target-level pseudo-labels instead (see Section 6 for details of BL-WFA and PL-WFA). Complementing these analytical insights, we provide in Section 7 extensive experimental evaluations of our methods showing performance gains, on real as well as synthetic datasets.

2 PREVIOUS RELATED WORK

Learning from Label Proportions (LLP). Early work on LLP by de Freitas and Kück [2005], Hernández-González et al. [2013] applied trained probabilistic models using Monte-Carlo methods, while Musicant et al. [2007], Rueping [2010] provided adaptations of standard supervised learning approaches such as SVM, k-NN and neural nets, and Chen et al. [2009], Stolpe and Morik [2011] developed clustering based methods for LLP. More specialized techniques were proposed by Quadrianto et al. [2009b] and later extended by Patrini et al. [2014] to estimate parameters from label proportions for the exponential generative model assuming well-behaved label distributions of bags. An optimization based approach of Yu et al. [2013] provided a novel $\propto$-SVM method for LLP. Newer methods involve deep learning [Kotzias et al., 2015, Dulac-Arnold et al., 2019, Liu et al., 2019, Nandy et al., 2022] and others leverage characteristics of the distribution of bags [Saket et al., 2022, Zhang et al., 2022a, Chen et al., 2023] while Busa-Fekete et al. [2023] developed model training techniques on derived *surrogate* labels for instances for random bags. Defining the LLP label proportion regression task in the PAC framework, Yu et al. [2014], established bounds on the generalization error bounds for bag-distributions. For the classification setting and specific types of loss functions, bag-to-instance generalization error bounds were shown by Busa-Fekete et al. [2023], Chen et al. [2023].

Domain Adaptation. Many of the domain adaptation techniques try to align the source and target distributions by minimizing a distance-measure between domains. The work of Long et al. [2015] generalized deep convolutional neural networks to the domain adaptation scenario, by matching the task-specific hidden representations for the source and target domains in a reproducing kernel Hilbert space. An extension of this work by Long et al. [2017] proposed a Joint

Adaptation Network which aligns the joint distributions of multiple domain-specific hidden layers using a joint maximum mean discrepancy measure. The technique proposed by Ganin et al. [2016] focuses on learning from features which are indiscriminate with respect to the shift between domains. Recently, Li and Culotta [2023b] applied domain adaptation to the LLP and proposed a model combining domain-adversarial neural network (DANN) and label regularization, to learn from source-domain bags and predict on instances from a target domain.

3 PRELIMINARIES

For a given $d \in \mathbb{Z}^+$, feature-vectors (instances) are d-dimensional reals and labels are real-valued scalars. Let $\mathcal{D}_S$ and $\mathcal{D}_T$ denote respectively the source and target distributions over $\mathbb{R}^d \times [0,1]$.

We denote by $\mathcal{S}(n)$ a source *training* set of n examples $\{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, n\}$ drawn iid from $\mathcal{D}_S$, and analogously define $\mathcal{T}(n)$ as n iid examples from $\mathcal{D}_T$. However, while the source training set is available at the instance-level, the target train-set is aggregated randomly into *bags*. We specify the bag-creation as follows.

Target Training Bags. A bag $B \subseteq \mathbb{R}^d$ is a finite set of instances $\mathbf{x}$ with labels $y_{\mathbf{x}}$ and its *bag-label* $y_B := (1/|B|)\sum_{\mathbf{x}\in B} y_{\mathbf{x}}$ is the average of the instance-labels in the bag. The sample target training bags denoted by $\mathcal{B}(m,k)$ is a random set of m k-sized bags $(B_1, y_{B_1}), \ldots, (B_m, y_{B_m})$ created as follows:

1. Let $\mathcal{T}(mk) := \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, mk\}$ be mk iid examples from $\mathcal{D}_T$.
2. Let $I_j = \{k(j-1)+1, \ldots, kj\}$, $j = 1, \ldots, m$ be a partition of $[mk]$.
3. For each $j = 1, \ldots, m$, let $B_j = \{\mathbf{x}_i \mid i \in I_j\}$ with bag-labels $y_{B_j} = (1/k)\sum_{i\in I_j} y_i$.

Instance and Bag-level losses. Since we focus on regression as the underlying task for an instance-level predictor we shall define our losses using *mean squared-error* (mse). For any function $h : \mathbb{R}^d \to \mathbb{R}$, the loss w.r.t. to a distribution $\mathcal{D}$ over $\mathbb{R}^d \times \mathbb{R}$ is

$$\varepsilon(\mathcal{D}, h) := \mathbb{E}_{(\mathbf{x},y)\leftarrow D}\left[(h(\mathbf{x}) - y)^2\right],$$

where we shall let $\mathcal{D}$ be $\mathcal{D}_S$ or $\mathcal{D}_T$ for our purpose. The loss over a finite sample $\mathcal{U}$ of labeled points is:

$$\hat{\varepsilon}(\mathcal{U}, h) := \frac{1}{|\mathcal{U}|}\sum_{(\mathbf{x},y)\in\mathcal{U}}\left[(h(\mathbf{x}) - y)^2\right]$$

where we shall take $\mathcal{U}$ as the source training-set $\mathcal{S}$ or target training-set $\mathcal{T}$ (we omit the sizes of the train-set for convenience). Finally, we have the loss on sampled bags:

$$\bar{\varepsilon}(\mathcal{B}, h) := \frac{1}{|\mathcal{B}|}\sum_{(B,y_B)\in\mathcal{B}}\left[\left(\left(\frac{1}{|B|}\sum_{\mathbf{x}\in B} h(\mathbf{x})\right) - y_B\right)^2\right]$$

Function Classes and pseudo-dimension. We will consider a class $\mathcal{F}$ of real-valued functions (regressors) mapping $\mathbb{R}^d$ to $[0,1]$. For any $\boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^d$ s.t. $|\boldsymbol{\mathcal{X}}| = N$, let $\mathcal{C}_p(\xi, \mathcal{F}, \boldsymbol{\mathcal{X}})$ denote a minimum cardinality ℓ_p-metric ξ-cover of $\mathcal{F}$ over $\boldsymbol{\mathcal{X}}$, for some $\xi > 0$. Specifically, $\mathcal{C}_p(\xi, \mathcal{F}, \boldsymbol{\mathcal{X}})$ is a minimum sized subset of $\mathcal{F}$ such that for each $f^* \in \mathcal{F}$, there exists $f \in \mathcal{C}_p(\xi, \mathcal{F}, \boldsymbol{\mathcal{X}})$ s.t. $\left(\mathbb{E}_{\mathbf{x}\in\boldsymbol{\mathcal{X}}}\left[|f^*(\mathbf{x}) - f(\mathbf{x})|^p\right]\right)^{1/p} \le \xi$ for $p \in [1,\infty)$, and $\max_{\mathbf{x}\in\boldsymbol{\mathcal{X}}} |f^*(\mathbf{x}) - f(\mathbf{x})| \le \xi$ for $p = \infty$.

As detailed in Sections 10.2-10.4 of Anthony and Bartlett [2009], the largest size of such a cover over all choices of $\boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^d$ s.t. $|\boldsymbol{\mathcal{X}}| = N$ is defined to be $N_p(\xi, \mathcal{F}, N)$.

The *pseudo-dimension* of $\mathcal{F}$, $\mathsf{Pdim}(\mathcal{F})$ (see Section 10.4 and 12.3 of Anthony and Bartlett [2009], Appendix B.2) can be used to bound the size of covers for $\mathcal{F}$ as follows:

$$N_1(\xi, \mathcal{F}, N) \le N_\infty(\xi, \mathcal{F}, N) \le (eN/\xi p)^p \qquad (1)$$

where $p = \mathsf{Pdim}(\mathcal{F})$ and $N \ge d$.

Since the task of our interest is regression, we shall assume that for any $f \in \mathcal{F}$, $f(\mathbf{x}) = \mathbf{r}_f^\intercal \phi(\mathbf{x})$ where ϕ is a mapping to a real-vector in an embedding space and $\mathbf{r}_f$ is the representation of f in that space (see Appendix A for an explanation).

3.1 OUR CONTRIBUTIONS

For $\mathcal{S} = \mathcal{S}(mk) = \{(\mathbf{z}_i, \ell_i)\}_{i=1}^{mk}$, and $\mathcal{B} = \mathcal{B}(m,k) = \{(B_j, y_{B_j})\}_{j=1}^m$ be the bags constructed from $\mathcal{T} = \mathcal{T}(mk)$, we define the following *covariate-shift* loss.

$$\xi(\mathcal{S}, \mathcal{B}) := 2\left\|\frac{1}{m}\sum_{j=1}^m y_{B_j}\left(\frac{1}{k}\sum_{\mathbf{x}\in B_j}\phi(\mathbf{x})\right) - \frac{1}{mk}\sum_{i=1}^{mk} l_i\phi(\mathbf{z}_i)\right\|_2 \qquad (2)$$

Note that the above domain adaptation loss depends on the labels from the source train-set labels as well as the bag-labels of the target training bags. In other words, it leverages the supervision provided on the training data $\mathcal{S}$ and $\mathcal{B}$. We bound the difference of the sample bag-loss on target training bags $\mathcal{B}$ and the sample instance-level loss on the source as follows.

Lemma 3.1. *For any $h \in \mathcal{F}$,*

$$\bar{\varepsilon}(\mathcal{B}, h) - \hat{\varepsilon}(\mathcal{S}, h) \le \xi(\mathcal{S}, \mathcal{B})\,\|\mathbf{r}_h\|_2 + \lambda'(\mathcal{S}, \mathcal{T}) + R(h, \mathcal{S}, \mathcal{T})$$

where $\lambda'(\mathcal{S},\mathcal{T})$ is independent of h and $R(h,\mathcal{S},\mathcal{T})$ is a label-independent regularization on $\mathcal{S}$ and $\mathcal{T}$.

The above lemma whose proof along with the expressions for $\lambda'(\mathcal{S},\mathcal{T})$ and $R(h,\mathcal{S},\mathcal{T})$, is provided in Section 4, shows that minimizing the instance-level loss on the source train-set $\mathcal{S}$ along with the covariate-shift loss training data can upper bound the bag-level loss on the target training bags $\mathcal{B}$. Since our goal is to upper bound the instance-level loss on the target distribution, we bound the latter using the bag-loss on the training bags in the following novel generalization error bound.

Theorem 3.2. *For $m,k \in \mathbb{Z}^+, \nu,\delta > 0$, w.p. $1-\delta$ over choice of $\mathcal{B} = \mathcal{B}(m,k)$, $\varepsilon(\mathcal{D}_T,h) \leq 16k\bar{\varepsilon}(\mathcal{B},h)$ for all $h \in \mathcal{F}$ s.t. $\varepsilon(\mathcal{D}_T,h) \geq \nu$ and $p = \mathsf{Pdim}(\mathcal{F})$, when $m \geq O\left(\left(p\left(\log\left(\frac{k}{\nu}\right) + \log\log\left(\frac{1}{\delta}\right)\right) + \log\frac{1}{\delta}\right)\max\left\{\frac{1}{k\nu^2}, \frac{k^2}{\nu}\right\}\right)$.*

The above is, to the best of our knowledge, the first bag-to-instance generalization error bound for regression tasks in LLP using the pseudo-dimension of the regressor class. Note however that there is a blowup in the error proportional to the bag-size k, which is understandable since, due to convexity, the mse loss between the average prediction in a bag and its bag-label is less than the average loss of the instance-wise predictions and labels. In other words, the error bound from Theorem 3.2 is weaker with increasing bag size, and in Appendix C we demonstrate through an example that this degradation with bag-size is unavoidable.

Lemma 3.1 can, however, be used to mitigate the weakening of the bound in Theorem 3.2. In particular, combining Lemma 3.1 with the implication of Theorem 3.2 we obtain $\varepsilon(\mathcal{D}_T,h) \leq w_1\bar{\varepsilon}(\mathcal{B},h) + w_2\hat{\varepsilon}(\mathcal{S},h) + w_2\left(\bar{\varepsilon}(\mathcal{B},h) - \hat{\varepsilon}(\mathcal{S},h)\right)$ where $w_1 + w_2 \geq 16k$. This can be bounded by $w_1\bar{\varepsilon}(\mathcal{B},h) + w_2\hat{\varepsilon}(\mathcal{S},h) + w_2\left(\xi(\mathcal{S},\mathcal{B})\left\|\mathbf{r}_h\right\|_2 + \lambda' + R(h,\mathcal{S},\mathcal{T})\right)$. Therefore, it makes sense to directly optimize $\bar{\varepsilon}(\mathcal{B},h)$ along with $\xi(\mathcal{S},\mathcal{B})$ and $\hat{\varepsilon}(\mathcal{S},h)$. In this, we can assume a bound on $\left\|\mathbf{r}_h\right\|_2$ since the range of all $h \in \mathcal{F}$ is bounded in $[0,1]$. Further, the term $R(h,\mathcal{S},\mathcal{T})$ is a difference of two unsupervised regularization terms on $\mathcal{S}$ and $\mathcal{T}$, which is expected to be small for reasonable covariate-shift in the datasets, and hence can omitted from the optimization (see Appendix A.2).

With this we formalize the above intuition to propose our loss on bags and covariate-shifted instances.

Bags and covariate-shifted instances loss. For parameters $\lambda_1,\lambda_2,\lambda_3 \geq 0$, the BagCSI loss is defined as:

$$\begin{aligned}&\mathsf{BagCSI}\left(\mathcal{S},\mathcal{B},h,\{\lambda_i\}_{i=1}^3\right)\\ :=\;& \lambda_1\bar{\varepsilon}(\mathcal{B},h) + \lambda_2\hat{\varepsilon}(\mathcal{S},h) + \lambda_3\xi^2(\mathcal{S},\mathcal{B}) \qquad (3)\end{aligned}$$

For practical considerations we use ξ^2 instead of ξ because ξ cannot be summed over mini-batches of the training dataset. We use BagCSI loss to propose model training method in Section 6. We also perform extensive experiments to evaluate our methods and share the outcomes in Section 7.

4 PROOF OF LEMMA 3.1

Using the definitions in Section 3 define $\mathbf{u}_j := (1/k)\sum_{i\in I_j}\phi(\mathbf{x}_i)$ so that $\frac{1}{k}\sum_{i\in I_j}h(\mathbf{x}_i) = \mathbf{r}_h^\mathsf{T}\mathbf{u}_j$.

$$\begin{aligned}\bar{\varepsilon}(\mathcal{B},h) &= \frac{1}{m}\sum_{j=1}^{m}\left[\left(\left(\frac{1}{k}\sum_{i\in I_j}h(\mathbf{x}_i)\right) - y_{B_j}\right)^2\right]\\ &= \frac{1}{m}\sum_{j=1}^{m}\left[\left(\frac{1}{k}\sum_{i\in I_j}h(\mathbf{x}_i)\right)^2 + y_{B_j}^2 - 2y_{B_j}\mathbf{r}_h^\mathsf{T}\mathbf{u}_j\right]\\ &\leq \frac{1}{m}\sum_{j=1}^{m}\left[\frac{1}{k}\sum_{i\in I_j}(h(\mathbf{x}_i)^2 + y_i^2) - 2y_{B_j}\mathbf{r}_h^\mathsf{T}\mathbf{u}_j\right] \qquad (4)\end{aligned}$$

where the last upper bound uses Cauchy-Schwarz inequality. On the other hand,

$$\begin{aligned}\hat{\varepsilon}(\mathcal{S},h) &= \frac{1}{mk}\sum_{i=1}^{mk}\left[\left(h(\mathbf{z}_i) - \ell_i\right)^2\right]\\ &= \frac{1}{mk}\sum_{i=1}^{mk}\left[h(\mathbf{z}_i)^2 + \ell_i^2 - 2\ell_i\mathbf{r}_h^\mathsf{T}\phi(\mathbf{z}_i)\right] \qquad (5)\end{aligned}$$

Using the above along with (4) we obtain,

$$\begin{aligned}&\bar{\varepsilon}(\mathcal{B},h) - \hat{\varepsilon}(\mathcal{S},h)\\ \leq&\frac{1}{mk}\sum_{i=1}^{mk}\left(h(\mathbf{x}_i)^2 - h(\mathbf{z}_i)^2\right)\\ &+ 2\mathbf{r}_h^\mathsf{T}\left(\frac{1}{m}\sum_{j=1}^{m}y_{B_j}\mathbf{u}_j - \frac{1}{mk}\sum_{i=1}^{mk}\ell_i\phi(\mathbf{z}_i)\right)\\ &+ \frac{1}{mk}\sum_{i=1}^{mk}\left(y_i^2 - \ell_i^2\right)\end{aligned}$$

Notice that the second term on the RHS of the above is $\leq \xi(\mathcal{S},\mathcal{B})\left\|\mathbf{r}_h\right\|_2$. Taking $\lambda'(\mathcal{S},\mathcal{T}) = \left|1/(mk)\sum_{i=1}^{mk}\left(y_i^2 - \ell_i^2\right)\right|$ and $R(h,\mathcal{S},\mathcal{T}) = \left|1/(mk)\sum_{i=1}^{mk}\left(h(\mathbf{x}_i)^2 - h(\mathbf{z}_i)^2\right)\right|$ completes the proof of Lemma 3.1.

5 PROOF OF THEOREM 3.2

The proof proceeds by first reformulating the process of sampling the m training bags as: (i) sample $2mk$ examples from $\mathcal{D}_T$, (ii) partition them into m disjoint $(2k)$-sized subsets, and (iii) from each subset randomly choose k points to

include in a bag, to obtain m k-sized bags. First, for a fixed sample of $2mk$ examples and regressor $h \in \mathcal{F}$, we use the randomness in step (iii) along with concentration bounds to show that with high probability the bag-level mse loss of h on the bags is at least an $O(k)$-fraction of its loss on the sampled instances. A union bound over a fine-grained ℓ_∞ cover of $\mathcal{F}$ essentially allows us to restrict ourselves to regressors in the cover. The randomness in step (i) is used along with standard generalization error bounds to show that instance-level sample loss of every $h \in \mathcal{F}$ can be replaced with the distributional loss. The parameter m is chosen to make the error probability arbitrarily small. The rest of this section contains the formal proof.

We first describe the following equivalent way of sampling the target training bags $\mathcal{B} = \mathcal{B}(m,k) = \{(B_j, y_{B_j}) \mid j = 1,\dots,m\}$.

1. Let $\mathcal{Z} := \{(\mathbf{x}_i, y_i) \mid i = 1,\dots,2mk\}$ be $2mk$ iid examples from $\mathcal{D}_T$.
2. Define $\overline{I}_j = \{2k(j-1)+1,\dots,2kj\}$, $j = 1,\dots,m$ be a partition of $[2mk]$ into m disjoint subsets.
3. Independently for each $j = 1,\dots,m$, let I_j be a random subset of $\overline{I}_j$ of exactly k indices.
4. For each $j = 1,\dots,m$, let $B_j = \{\mathbf{x}_i \mid i \in I_j\}$ with bag-labels $y_{B_j} = (1/k)\sum_{i\in I_j} y_i$.

Let us first fix $h \in \mathcal{F}$ and $\mathcal{Z}$ and prove a lower bound on the bag-level loss.

Analysis for fixed h and $\mathcal{Z}$. Let us assume that $\hat{\varepsilon}(\mathcal{Z}, h) = \zeta$, for some $\zeta \geq 0$. For convenience let $z_i = h(\mathbf{x}_i) - y_i$, $i = 1,\dots,2mk$. Note that since $y_i, h(\mathbf{x}_i) \in [0,1]$, $|z_i| \leq 1$. Let $\overline{\mathcal{Z}}^{(j)} = \{(\mathbf{x}_i, y_i) \mid i \in \overline{I}_j\}$ be the restriction of $\mathcal{Z}$ to the indices in $\overline{I}_j$, so that $\sum_{j=1}^m \hat{\varepsilon}(\mathcal{Z}^{(j)}, h) = \hat{\varepsilon}(\mathcal{Z}, h)$. Over the choice of $\{I_j\}_{j=1}^m$ define the random variable $L_j := \left[\left(\frac{1}{k}\sum_{i\in I_j} h(\mathbf{x}_i)\right) - y_{B_j}\right]^2$. Since $y_{B_j} = (1/k)\sum_{i\in I_j} y_i$, $L_j = \left(\frac{1}{k}\sum_{i\in I_j} z_i\right)^2 \leq \left(\frac{1}{k}\sum_{i\in I_j} |z_i|\right)^2$. Since $I_j \subseteq \overline{I}_j$ and $|z_i| \leq 1$ for all i, this implies

$$\begin{aligned} L_j &\leq \min\left\{1, \left(\frac{1}{k}\sum_{i\in\overline{I}_j} |z_i|\right)^2\right\} \\ &\leq \min\left\{1, \frac{2}{k}\sum_{i\in\overline{I}_j} |z_i|^2\right\} =: \gamma_j \end{aligned} \tag{6}$$

since $\sum_{i\in\overline{I}_j} |z_i| \leq \sqrt{2k}\sqrt{\sum_{i\in\overline{I}_j} |z_i|}$ by Cauchy-Schwarz inequality. Note that after fixing $\mathcal{Z}$, the choices of $I_1,\dots,I_m$ are independent of each other, and each L_j only depends of the choice of I_j.

$$\begin{aligned} &\mathbb{E}[L_j] = \mathbb{E}\left[\left(\frac{1}{k}\sum_{i\in I_j} z_i\right)^2\right] \\ = \;& \frac{1}{k^2}\left(\sum_{r\in\overline{I}_j} z_r^2 \Pr[r \in I_j] + \sum_{\substack{r,s\in\overline{I}_j \\ r\neq s}} z_r z_s \Pr[r,s \in I_j]\right) \end{aligned}$$

Since I_j is a random subset of $\overline{I}_j$ of k out of $2k$ indices, $\Pr[r \in I_j \mid r \in \overline{I}_j] = 1/2$ and $\Pr[r,s \in I_j \mid r,s \in \overline{I}_j, r \neq s] = (k-1)/(2(2k-1))$ which simplifies the RHS of the above to:

$$\begin{aligned} &\frac{1}{2k^2}\left[\left(1 - \frac{k-1}{2k-1}\right)\sum_{r\in\overline{I}_j} z_r^2 + \frac{k-1}{2k-1}\sum_{r,s\in\overline{I}_j} z_r z_s\right] \\ \geq\;& \frac{1}{2k^2}\left[\frac{1}{2}\sum_{r\in\overline{I}_j} z_r^2 + \frac{k-1}{2k-1}\left(\sum_{r\in\overline{I}_j} z_r\right)^2\right] \\ \geq\;& \frac{1}{4k^2}\sum_{r\in\overline{I}_j} z_r^2 \end{aligned} \tag{7}$$

Using (6) one can apply Hoeffding's inequality to obtain for any $t \geq 0$ (see Appendix B.1),

$$\begin{aligned} &\Pr\left[\sum_{j=1}^m L_j \leq \mathbb{E}\left[\sum_{j=1}^m L_j\right] - t\right] \\ \leq\;& 2\exp\left(\frac{-2t^2}{\sum_{j=1}^m \gamma_j^2}\right) \\ \leq\;& 2\exp\left(\frac{-2t^2}{(\max\{\gamma_j\}_{j=1}^m)\sum_{j=1}^m \gamma_j}\right) \\ \leq\;& 2\exp\left(\frac{-t^2 k}{\sum_{j=1}^m \sum_{i\in\overline{I}_j} z_i^2}\right) \end{aligned}$$

By definition we have $\sum_{j\in\overline{I}_j} z_i^2 = \sum_{i=1}^{2mk} z_i^2 = 2\zeta mk$. Thus, the above along with (7) yields $\Pr\left[\sum_{j=1}^m L_j \leq \zeta m/(2k) - t\right] \leq 2\exp\left(\frac{-t^2}{2\zeta m}\right)$. Recalling that $\zeta = \hat{\varepsilon}(\mathcal{Z}, h)$, and noting that $\sum_{j=1}^m L_j = m\overline{\varepsilon}(\mathcal{B}, h)$ while taking $t = \zeta m/(4k)$ we obtain

$$\Pr\left[\overline{\varepsilon}(\mathcal{B}, h) \leq \frac{\hat{\varepsilon}(\mathcal{Z}, h)}{4k}\right] \leq 2\exp\left(\frac{-\hat{\varepsilon}(\mathcal{Z}, h)m}{32k^2}\right) \tag{8}$$

High probability bound for $\mathcal{F}$ and $\mathcal{Z}$. Let us fix the parameter ε in the statement of Theorem 3.2. We fix $\mathcal{Z}$ for now and consider the cover $\mathcal{C}_\infty(\xi, \mathcal{F}, \mathcal{Z})$ for some ξ which we will choose later, and $q_\infty = N_\infty(\xi, \mathcal{F}, 2mk)$ be the upper bound on its size. Let $\mathcal{C}_{\text{err}} \subseteq \mathcal{C}_p(\xi, \mathcal{F}, \mathcal{Z})$ s.t.

Table 1: MSE scores for different methods and bag sizes on the IPUMS dataset (averaged over 10 runs). The source instance loss is 1.8714 ± 0.08 and target instance loss is 1.1237 ± 0.00. Lower is better.

Method \ Bag Size	8	32	128	256
Bagged-Target	1.14 ± 0.00	1.16 ± 0.00	1.22 ± 0.0046	1.31 ± 0.01
AF	1.23 ± 0.01	1.31 ± 0.01	1.41 ± 0.02	1.43 ± 0.02
LR	1.15 ± 0.00	1.18 ± 0.00	1.24 ± 0.01	1.29 ± 0.01
AF-DANN	1.25 ± 0.02	1.33 ± 0.07	1.39 ± 0.07	1.39 ± 0.02
LR-DANN	1.16 ± 0.00	1.23 ± 0.02	1.51 ± 0.07	1.61 ± 0.13
DMFA	1.15 ± 0.00	1.18 ± 0.00	1.26 ± 0.01	1.30 ± 0.01
PL-WFA (our)	1.15 ± 0.00	1.18 ± 0.00	1.25 ± 0.01	1.29 ± 0.01
BL-WFA (our)	$\mathbf{1.14 \pm 0.00}$	$\mathbf{1.16 \pm 0.00}$	$\mathbf{1.22 \pm 0.00}$	$\mathbf{1.25 \pm 0.01}$

Table 2: MSE scores for different methods and bag sizes on the Wine dataset (averaged over 20 runs). The source instance loss is 195.5 ± 1.2 and target instance loss is 170.5 ± 0.1. Lower is better.

Method \ Bag Size	8	32	128	256
Bagged-Target	$\mathbf{173.5 \pm 0.4}$	$\mathbf{177.7 \pm 1.2}$	191.0 ± 2.5	206.9 ± 3.5
AF	186.8 ± 2.1	190.3 ± 2.8	191.0 ± 2.4	192.4 ± 1.8
LR	185.9 ± 2.0	191.6 ± 1.6	193.8 ± 0.8	194.5 ± 1.0
AF-DANN	187.6 ± 1.7	190.5 ± 1.7	191.2 ± 2.5	191.9 ± 2.1
LR-DANN	186.2 ± 1.5	192.1 ± 2.0	193.7 ± 2.4	193.8 ± 2.5
DMFA	186.1 ± 1.7	191.8 ± 2.1	193.5 ± 2.4	194.5 ± 0.9
PL-WFA (our)	183.0 ± 0.6	186.6 ± 1.0	189.0 ± 0.8	188.9 ± 1.2
BL-WFA (our)	180.9 ± 0.5	184.6 ± 0.7	$\mathbf{186.0 \pm 0.8}$	$\mathbf{186.4 \pm 0.5}$

$\forall \hat{h} \in \mathcal{C}_{\text{err}}, \hat{\varepsilon}(\mathcal{Z}, \hat{h}) \geq \nu/2$. Taking a union bound of the error in (8) over $\hat{\mathcal{F}}_{\text{err}}$ we obtain that:

$$\Pr\left[\forall \hat{h} \in \mathcal{C}_{\text{err}} : \bar{\varepsilon}(\mathcal{B}, \hat{h}) \geq \frac{\hat{\varepsilon}(\mathcal{Z}, \hat{h})}{4k}\right] \leq 1 - 2q_\infty \exp\left(\frac{-\nu m}{64k^2}\right) \quad (9)$$

Define $\hat{\mathcal{F}}_{\text{err}} := \{h \in \mathcal{F} \mid \hat{\varepsilon}(\mathcal{Z}, h) \geq 3\nu/4\}$. For any $h \in \hat{\mathcal{F}}_{\text{err}}$ there is $\hat{h} \in \mathcal{C}_\infty(\xi, \mathcal{F}, \mathcal{Z})$ s.t. $|\hat{h}(\mathbf{x}) - h(\mathbf{x})| \leq \xi$ for all $(\mathbf{x}, y) \in \mathcal{Z}$. Now, $(\hat{h}(\mathbf{x}) - y)^2 = (h(\mathbf{x}) - y + \hat{h}(\mathbf{x}) - h(\mathbf{x}))^2 \geq (h(\mathbf{x}) - y)^2 - 2|\hat{h}(\mathbf{x}) - h(\mathbf{x})||h(\mathbf{x}) - y| + (\hat{h}(\mathbf{x}) - h(\mathbf{x}))^2 \geq (h(\mathbf{x}) - y)^2 - 2\xi$ since $h(\mathbf{x}), y \in [0, 1]$. Similarly, consider any bag $B \in \mathcal{B}$. Using arguments analogous to above we obtain $\left(\mathbb{E}\left[h(\mathbf{x})\right] - y_B\right)^2 \geq \left(\mathbb{E}\left[\hat{h}(\mathbf{x})\right] - y_B\right)^2 - 2\left|\mathbb{E}[\hat{h}(\mathbf{x}) - h(\mathbf{x})]\right| \left(\mathbb{E}\left[h(\mathbf{x})\right] - y_B\right) \geq \left(\mathbb{E}\left[\hat{h}(\mathbf{x})\right] - y_B\right)^2 - 2\xi$, implying

$$\hat{\varepsilon}(\mathcal{Z}, \hat{h}) \geq \hat{\varepsilon}(\mathcal{Z}, h) - 2\xi, \quad \bar{\varepsilon}(\mathcal{B}, h) \geq \bar{\varepsilon}(\mathcal{B}, \hat{h}) - 2\xi. \quad (10)$$

Therefore, taking $\xi = \nu/(32k)$ we obtain from the first bound above that $\hat{h} \in \mathcal{C}_{\text{err}}$ and further that $\hat{\varepsilon}(\mathcal{Z}, \hat{h}) \geq 2\hat{\varepsilon}(\mathcal{Z}, h)/3 \geq \nu/2 = 16k\xi$. Observe that $\bar{\varepsilon}(\mathcal{B}, \hat{h}) \geq \hat{\varepsilon}(\mathcal{Z}, \hat{h})/(4k)$ implies $\bar{\varepsilon}(\mathcal{B}, \hat{h}) \geq 4\xi$, which in turn implies $\bar{\varepsilon}(\mathcal{B}, h) \geq \bar{\varepsilon}(\mathcal{B}, \hat{h}) - 2\xi \geq \bar{\varepsilon}(\mathcal{B}, \hat{h})/2$. Combining this with (9) and (10) we obtain,

$$\Pr\left[\forall h \in \hat{\mathcal{F}}_{\text{err}} : \bar{\varepsilon}(\mathcal{B}, h) \geq \frac{\hat{\varepsilon}(\mathcal{Z}, h)}{12k}\right] \leq 1 - 2q_\infty \exp\left(\frac{-\nu m}{64k^2}\right) \quad (11)$$

We now unfix $\mathcal{Z}$, and define $\mathcal{F}_{\text{err}} = \{h \in \mathcal{F} \mid \varepsilon(\mathcal{D}_T, \hat{h}) \geq \nu\}$. By Theorem 17.1 of Anthony and Bartlett [2009], we obtain with probability at least $1 - 4q_1 \exp\left(-2\nu^2 mk/512\right)$ over the choice of $\mathcal{Z}$, $h \in \mathcal{F}_{\text{err}} \Rightarrow h \in \hat{\mathcal{F}}_{\text{err}}$ where $q_1 = N_1(\nu/64, \mathcal{F}, 4mk)$. Using this along with (11), we obtain that with probability at least $1 - 2q_\infty \exp\left(-\nu m/(64k^2)\right) - 4q_1 \exp\left(-2\nu^2 mk/512\right)$, $\forall h \in \mathcal{F}_{\text{err}}, \bar{\varepsilon}(\mathcal{B}, h) \geq \frac{\hat{\varepsilon}(\mathcal{Z}, h)}{12k} \geq \frac{3\nu}{48k} = \frac{\nu}{16k}$. Using the upper bounds in (1) we see that the probability is $1 - \delta$ if we choose $m \geq O\left(\left(p\left(\log\left(\frac{k}{\nu}\right) + \log\log\left(\frac{1}{\delta}\right)\right) + \log\frac{1}{\delta}\right) \max\left\{\frac{1}{k\nu^2}, \frac{k^2}{\nu}\right\}\right)$. See Appendix A.3 for more details. This completes the proof of Theorem 3.2.

6 PROPOSED METHODS

We propose two novel methods. The first method uses BagCSI loss as the objective. We have shown above that BagCSI loss is an upper bound over $\varepsilon(\mathcal{D}_T, h)$ loss w.r.t target distribution. We now provide intuitive explanation for why BagCSI loss should work.

Let us assume that the goal is to predict label for an unseen instance $\mathbf{x}$, given feature representations $\phi(\mathbf{x}_i)$ in the embedding space and corresponding labels y_i from training data. A natural prediction would be $\mathbb{E}_i\left[\rho(\phi(\mathbf{x}), \phi(\mathbf{x}_i))y_i\right]$, where ρ is some similarity metric. If we choose the similarity metric to be the inner product, the prediction can be written as $\phi(\mathbf{x})^\mathsf{T}\mathbb{E}_i\left[\phi(\mathbf{x}_i)y_i\right]$. The given feature representations and corresponding labels can come either from the source domain or from the target domain. For learning domain invariant feature representation, the prediction should be similar irrespective of the domain considered. This can be achieved by enforcing the term, $\sum_i y_i\phi(\mathbf{x}_i)$ to be equal for source and target domain. However, this approach requires knowledge of instance-level labels $y_{\mathbf{x}}$ from target domain, which are not available. We can however replace $y_{\mathbf{x}}$ with *pseudo-labels* $\hat{y}_{\mathbf{x}}$, using which we introduce a new domain adaptation loss term in the objective, $\psi^2(\mathcal{S}, \mathcal{B})$ where:

$$\psi(\mathcal{S}, \mathcal{B}) := \frac{1}{mk}\left\|\sum_{j=1}^{m}\sum_{\mathbf{x}\in B_j}\hat{y}_{\mathbf{x}}\phi(\mathbf{x}) - \sum_{i=1}^{mk} y_i\phi(\mathbf{z}_i)\right\|_2 \quad (12)$$

One way is to assign the bag-label as the pseudo-label for all instances withing the bag, in which case $\psi(\mathcal{S}, \mathcal{B})$ essentially reduces to $\xi(\mathcal{S}, \mathcal{B})$. We call this method *Bag Label Weighted Feature Alignment* (**BL-WFA**) which involves training using the BagCSI loss.

Another approach is to use the following process for pseudo-labeling instances in a bag B using hypothesis model h:

1. Compute the predictions $\{h(\mathbf{x})\}_{\mathbf{x}\in B}$.
2. The pseudo-labels are given by adding to each prediction the same $b \in \mathbb{R}$ such that average pseudo-label in the bag equals the bag-label. Note that this is equivalent to the nearest vector of pseudo-labels (in Euclidean distance) to the vector predictions, that satisfies the bag-label constraint.

We call this method *Pseudo-label Weighted Feature Alignment* (**PL-WFA**) in which $\psi(\mathcal{S}, \mathcal{B})$ is used to train the model using the above computed pseudo-labels.

7 EXPERIMENTAL EVALUATIONS

We evaluate our approaches via experiments on both synthetic as well as real-world datasets and compare against the baselines for different bag sizes.

Baseline Methodologies. In Li and Culotta [2023a], authors propose methods for domain adaptation in LLP setting for classification tasks. We adapt these methods for regression tasks and consider those as baselines. In this paper, these baselines are referred to as Average Feature (**AF**), Label Regularization (**LR**), Average Feature DANN (**AF-DANN**) and Label Regularization DANN (**LR-DANN**). See Sections 3.1.2, 3.1.3, 3.2.1, 3.2.2 in Li and Culotta [2023a] for respective methods. In literature on domain adaptation (for non-LLP settings) [Long et al., 2015, 2017], it has been shown that approaches using MMD (maximum mean discrepancy) based objectives work well. Hence, we also define a baseline that uses similar objective adapted for our setting, called Domain Mean Feature Alignment (**DMFA**). We also consider bag level target loss (**Bagged-Target**) as a baseline. Appendix D contains additional details about baseline methods. We evaluate and compare our methods against these baselines.

Our model training uses the above losses in a mini-batch loop. For DMFA and PL-WFA we select equal number of instances from both source and target domain in a mini-batch. For BL-WFA, we select as many instances from source domain as the number of bags selected from target domain in a mini-batch. Such a choice avoids explicit normalization in the objective function and incorporates them into the hyper-parameters. We evaluate all the baselines and proposed methods for different bag sizes and datasets.

Synthetic Dataset. The synthetic dataset has 64 dimensional continuous feature vectors and scalar-valued continuous label. For covariate shifted source and target domain data, the feature vectors are sampled from a multi-dimensional Gaussian distribution with different means and covariance matrices. The labels for both source and target data are computed using the same randomly initialized neural network. We also perform ablation studies to observe the impact of magnitude of covariance shift. The train set comprises 0.2 million instances from both source and target domain. The test set comprises 65 thousand instances from target domain.

Real-world Datasets. We also evaluate methods on three real world datasets: *Wine Ratings* [Dara, 2018, Zackthoutt, 2017], *IPUMS USA* [Ruggles et al., 2024] Census data, and *Criteo Sponsored Search Conversion Logs (SSCL)* [Tallis and Yadav, 2018].

Wine: We use Price column as the label. The source domain comprises of wines from France and the target domain comprises of wines from all countries but France. The train set comprises 0.5 million instances from both source and target domain. The test set comprises 0.2 million instances from target domain.
IPUMS USA: We use INCWAGE column as the label. We consider the data from 1970 as the source domain and data from 2022 as the target domain. The train set comprises 1.3 million instances from source and 9.4 million instances from target domain. The test set comprises 0.3 million instances from target domain.
Criteo SSCL: We use SalesAmountInEuro column as the label. We create a domain split on the basis of the country field (the most frequently occurring country in the dataset as source and rest as the target). The train set comprises 0.5 million instances from source and 0.9 million instances from target domain. The test set comprises 0.2 million instances from target domain.

Appendix E contains details about size and pre-processing for all the datasets.

All datasets are split into two components, source and target domain. For our study, it is important that there is a reasonable covariate shift between these two components. The target domain dataset is split into train (80%) and test (20%) sets. The target domain component of train set is partitioned randomly into bags of equal size. We also perform experiments with correlated bags. To partition the dataset into correlated bags, we select a feature and create bags such that all the samples in that bag have the same value of that feature if the feature is categorical. If the feature is numerical, we sort the dataset on the basis of that feature and use consecutive samples for creating the bags. Further details about creation of correlated bags are provided in Appendix G.2. Additionally, we also perform experiments by partitioning the dataset into bags of mixed (non-uniform) sizes. We do so in two different ways; SBB (Sample Balanced Bagging - equal number of instances for each bag size) and BBB (Bag Balanced Bagging - equal number of bags of each size). Each bag in the resultant dataset is of the size 8, 32, 128 or 256. Further details about partitioning the dataset into mixed bag sizes are provided in Appendix G.1.

Table 3: MSE scores for different methods and bag sizes on the Synthetic dataset (averaged over 20 runs). The source instance loss is 2718.13 ± 2062.32 and target instance loss is 0.19 ± 0.02. Lower is better.

Method \ Bag Size	8	32	128	256
Bagged-Target	0.71 ± 0.05	5.49 ± 0.93	17.87 ± 0.49	19.95 ± 0.34
AF	0.96 ± 0.07	6.22 ± 0.81	18.16 ± 0.50	20.00 ± 0.86
LR	0.71 ± 0.04	5.15 ± 1.06	18.10 ± 0.40	19.92 ± 1.55
AF-DANN	1.23 ± 0.06	8.16 ± 0.54	18.04 ± 0.95	20.15 ± 0.49
LR-DANN	1.02 ± 0.04	7.84 ± 0.87	17.76 ± 0.24	19.72 ± 0.29
DMFA	$\mathbf{0.69 \pm 0.05}$	4.39 ± 0.84	16.50 ± 1.47	19.07 ± 1.16
PL-WFA (our)	0.75 ± 0.06	4.43 ± 0.81	15.60 ± 0.94	18.40 ± 0.74
BL-WFA (our)	0.75 ± 0.05	$\mathbf{2.22 \pm 0.22}$	$\mathbf{10.36 \pm 3.15}$	$\mathbf{13.76 \pm 0.60}$

Table 4: MSE scores for different methods and bag sizes on the Criteo SSCL dataset (averaged over 10 runs). The source instance loss is 293.74 ± 5.1 and target instance loss is 147.79 ± 0.3. Lower is better.

Method \ Bag Size	64	128	256	512
Bagged-Target	208.78 ± 2.7	234.32 ± 3.3	254.78 ± 5.3	264.74 ± 5.3
AF	297.95 ± 6.5	296.51 ± 6.1	294.86 ± 5.3	299.93 ± 6.5
LR	207.78 ± 2.7	232.72 ± 10.	256.68 ± 13.	264.46 ± 5.4
AF-DANN	296.95 ± 6.3	296.35 ± 6.4	295.49 ± 5.2	297.91 ± 7.3
LR-DANN	206.39 ± 2.3	230.84 ± 3.1	243.62 ± 4.5	265.33 ± 4.6
DMFA	207.60 ± 2.7	232.40 ± 9.9	247.66 ± 3.4	264.51 ± 5.5
PL-WFA (our)	204.71 ± 2.6	226.39 ± 2.9	240.55 ± 3.3	254.46 ± 5.5
BL-WFA (our)	**204.62 ± 2.4**	**226.33 ± 2.9**	**240.39 ± 3.2**	**254.36 ± 5.5**

Table 5: MSE scores on IPUMS dataset with correlated bags. Lower is better.

Method \ Bag Size	8	32	128	256
Bagged Target	1.09 ± 0.00	1.12 ± 0.00	1.17 ± 0.00	1.24 ± 0.00
AF	1.31 ± 0.00	1.35 ± 0.01	1.38 ± 0.01	1.41 ± 0.01
LR	1.10 ± 0.00	1.12 ± 0.00	1.18 ± 0.00	1.25 ± 0.00
AFDANN	1.31 ± 0.01	1.35 ± 0.01	1.37 ± 0.01	1.38 ± 0.01
LRDANN	1.09 ± 0.00	1.11 ± 0.00	1.17 ± 0.00	**1.21 ± 0.00**
DMFA	1.10 ± 0.00	1.12 ± 0.00	1.18 ± 0.00	1.22 ± 0.00
PLWFA	**1.08 ± 0.00**	**1.10 ± 0.00**	**1.16 ± 0.00**	**1.21 ± 0.00**
BLWFA	**1.08 ± 0.00**	**1.10 ± 0.00**	**1.16 ± 0.00**	**1.21 ± 0.00**

Table 6: MSE scores on Synthetic dataset with correlated bags. Lower is better.

Method \ Bag Size	8	32	128	256
Bagged-Target	0.65 ± 0.07	2.13 ± 0.35	7.01 ± 0.61	10.35 ± 1.33
AF	1.02 ± 0.17	4.20 ± 0.52	9.65 ± 0.54	12.35 ± 0.89
LR	0.60 ± 0.05	2.01 ± 0.33	6.72 ± 0.78	9.18 ± 0.74
AF-DANN	1.30 ± 0.16	5.14 ± 0.46	10.67 ± 0.43	13.42 ± 0.88
LR-DANN	0.81 ± 0.08	3.26 ± 0.38	8.31 ± 0.63	10.37 ± 0.78
DMFA	0.57 ± 0.05	2.17 ± 0.28	6.35 ± 0.56	8.86 ± 0.69
PL-WFA (our)	**0.56 ± 0.04**	**1.90 ± 0.30**	**6.09 ± 0.69**	8.50 ± 0.60
BL-WFA (our)	0.60 ± 0.05	1.90 ± 0.31	6.11 ± 0.74	**8.48 ± 0.81**

Table 7: MSE scores on Criteo dataset with correlated bags. Lower is better.

Method \ Bag Size	64	128	256	512
Bagged-Target	204.78 ± 2.7	211.12 ± 3.1	226.78 ± 3.8	254.74 ± 5.3
AF	257.92 ± 2.0	266.94 ± 3.4	276.82 ± 3.1	294.13 ± 4.4
LR	179.88 ± 0.6	183.77 ± 1.1	191.25 ± 1.1	207.24 ± 1.2
AF-DANN	257.48 ± 2.1	263.87 ± 0.6	275.14 ± 3.5	292.43 ± 5.0
LR-DANN	179.37 ± 0.5	183.73 ± 1.0	191.17 ± 1.1	207.98 ± 1.6
DMFA	180.89 ± 0.6	183.47 ± 1.2	191.27 ± 1.2	207.18 ± 1.2
PL-WFA (our)	177.76 ± 0.7	181.70 ± 1.2	188.29 ± 1.2	197.23 ± 1.2
BL-WFA (our)	**177.74 ± 0.7**	**181.66 ± 1.1**	**188.19 ± 1.2**	**197.07 ± 1.3**

Training & Evaluation. We use a simple neural network comprising of an input layer followed by two sequential ReLU activated layers (128 nodes) and a final linear layer (1 node). For IPUMS and Criteo SSCL datasets, we additionally include embedding layers after the input layer for all the cardinal and categorical features that were not converted to one-hot representations. For AF-DANN and LR-DANN, we also have a sigmoid activated domain prediction layer in parallel to the final dense layer.

During training, we perform a grid search to find the most optimal set of hyperparameters for each configuration (specific dataset, methodology and bag size). We try out two different optimizers for all experiments mentioned in the main paper - Adam and SGD and report scores corresponding to the best performer. We observed that Adam works better for most of the cases, so we perform experiments described in Appendix with Adam optimizer only. See Appendix F for more details.

For each configuration, we run the same experiment multiple times and report the MSE scores on target domain's test data as the evaluation metric. Note that the instances in target domain are randomly bagged for each run. The final evaluation metric is reported by the mean and standard deviation over these runs. We run 20 trials for each configuration with Wine and Synthetic datasets and 10 trials for each configuration with IPUMS and Criteo SSCL datasets.

Experimental Code and Resources.[1] Our experiments were run on a system with standard 8-core CPU, 256GB of memory with one P100 GPU.

MSE scores on IPUMS, Wine, Synthetic and Criteo SSCL datasets with random bagging for different bag sizes are reported in Tables 1, 2, 3 and 4. MSE scores on Wine, Criteo SSCL and IPUMS datasets with random bagging for BBB-mixed and SBB-mixed bag sizes are reported in Tables 8 and 9. MSE scores with correlated bags for IPUMS, Synthetic and Criteo datasets are reported in Tables 5, 6 and 7 respectively.

Results for more experiments are reported in Appendix G. This includes experiments on Wine dataset with a different domain split (see Table 16), experiments on synthetic dataset with a non-diagonal covariance matrix (see Table 17), and experiments on synthetic dataset by varying the magnitude of covariate shift (see Table 18).

Results & Inferences. For largest bag size, BL-WFA achieves 2.5%, 2.9%, 27.9% and 3.8% improvement over the best baseline method for IPUMS (see Table 1), Wine

[1]The code for our experiments can be found at `www.github.com/google-deepmind/covariate_shifted_llp`.

Table 8: MSE scores for data with BBB mixed bag sizes. BBB is bag balanced bagging i.e. there are equal number of bags of each size. Lower is better.

Method \ Dataset	Wine	Criteo	IPUMS
Bagged Target	219.51 ± 7.96	209.37 ± 1.98	1.26 ± 0.01
AF	210.45 ± 6.58	210.24 ± 2.54	1.27 ± 0.01
LR	211.58 ± 7.67	213.01 ± 3.01	1.25 ± 0.01
AFDANN	289.16 ± 2.69	205.36 ± 2.65	1.26 ± 0.01
LRDANN	190.88 ± 1.34	208.03 ± 2.79	1.28 ± 0.01
DMFA	193.29 ± 3.90	207.80 ± 3.14	1.26 ± 0.01
PLWFA	183.42 ± 3.76	202.94 ± 2.96	1.25 ± 0.01
BLWFA	**183.00 ± 3.13**	**202.75 ± 2.86**	**1.24 ± 0.01**

Table 9: MSE scores for data with SBB mixed bag sizes. SBB is sample balanced bagging i.e. for a particular bag size, there are equal number of samples. Lower is better.

Method \ Dataset	Wine	Criteo	IPUMS
Bagged Target	183.94 ± 2.00	177.69 ± 0.85	1.16 ± 0.01
AF	186.04 ± 1.43	176.99 ± 0.82	1.15 ± 0.01
LR	186.07 ± 1.46	181.89 ± 1.33	1.15 ± 0.01
AFDANN	163.51 ± 0.36	177.13 ± 0.45	1.17 ± 0.01
LRDANN	163.52 ± 0.37	181.22 ± 0.99	1.18 ± 0.01
DMFA	163.29 ± 1.08	181.62 ± 1.25	1.16 ± 0.01
PLWFA	**161.31 ± 1.03**	171.84 ± 1.34	1.16 ± 0.01
BLWFA	161.59 ± 1.11	**171.71 ± 1.31**	**1.14 ± 0.01**

(see Table 2), Synthetic (see Table 3) and Criteo SSCL (see Table 4) respectively. We observe similar improvements with correlated bags (see Tables 5, 6 and 7) and bags of mixed sizes (see Tables 8 and 9). MSE is used as evaluation metric, hence scores cannot be compared across datasets due to different scales. We make the following inferences from the results:

1. PL-WFA and BL-WFA consistently outperform all other baselines for large enough bag sizes. This is expected because with increase in bag size, the information from just the bagged target domain is not rich enough and benefits greatly from inclusion of covariate shifted source domain data. By leveraging not just the features from target domain but also the bagged-labels, PL-WFA and BL-WFA outperform other baseline methods which rely only on features from target domain for domain adaptation.
2. With increase in bag size, the performance drops. This is expected as information is lost with increase in bag size.
3. On synthetic dataset (where we definitely have a reasonable amount of covariate shift), even with bag size as large as 256, we see that the performance of our proposed methods - PL-WFA and BL-WFA is better than the case where we use instance level labeled target data for training (target instance loss). This improvement is achieved despite the fact that performance when just using the source data for training (source instance loss) is poor.
4. On smaller bag sizes, other methods (for example, LR and DMFA on synthetic dataset and Bagged-Target on Wine dataset) seem to outperform our proposed methods. Such behavior is expected when the information from target data is itself sufficient to learn a good enough function approximator. It is worth noting that the objective function in our proposed method reduces to that of LR for $\lambda = 0$. So, in theory PL-WFA and BL-WFA are always better than LR. By decreasing the λ value, our methods can do at least as good as LR.
5. Although the best baseline method is different for different datasets under consideration (AF-DANN on Wine, LR on IPUMS, LR-DANN on Criteo and DMFA on synthetic), BL-WFA consistently beats the best baseline for a large enough bag size.
6. Our methods perform well with bags of mixed sizes and correlated bags. This demonstrates the robustness of proposed methods to different bagging techniques.

8 CONCLUSION

We formally define the problem of learning from label aggregates where source data has instance wise labels while target data has aggregate labels of instances grouped into bags. We also give bag-to-instance generalization error bound for regression tasks in LLP and use it to arrive at BagCSI loss. We propose two new methods, BL-WFA (based on BagCSI) and PL-WFA (based on a variant of BagCSI) that naturally incorporate the knowledge of aggregate labels from target domain in the domain adaptation framework leading to improvement over baseline methods. We also adapt several methods from literature in domain adaptation and LLP to this setting. Through experiments on synthetic and real-world datasets we show that our methods consistently outperform baseline techniques.

References

Private aggregation api of chrome privacy sandbox. https://developer.chrome.com/docs/ privacy-sandbox/aggregation-service/.

Apple storekit ad network. `https://developer.apple.com/documentation/storekit/skadnetwork/`.

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, USA, 1st edition, 2009. ISBN 052111862X.

Ehsan Mohammady Ardehaly and Aron Culotta. Proc. IJCAI. pages 3670–3676, 2016.

Ehsan Mohammady Ardehaly and Aron Culotta. Co-training for demographic classification using deep learning from label proportions. In *ICDM*, pages 1017–1024, 2017.

Denis Baručić and Jan Kybic. Fast learning from label proportions with small bags. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3156–3160, 2022. doi: 10.1109/ICIP46576.2022.9897895.

G. Bortsova, F. Dubost, S. N. Ørting, I. Katramados, L. Hogeweg, L. H. Thomsen, M. M. W. Wille, and M. de Bruijne. Deep learning from label proportions for emphysema quantification. In *MICCAI*, volume 11071 of *Lecture Notes in Computer Science*, pages 768–776. Springer, 2018. URL https://arxiv.org/abs/1807.08601.

Anand Brahmbhatt, Mohith Pokala, Rishi Saket, and Aravindan Raghuveer. Llp-bench: A large scale tabular benchmark for learning from label proportions. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4374–4381, 2024.

Róbert Istvan Busa-Fekete, Heejin Choi, Travis Dick, Claudio Gentile, and Andrés Muñoz Medina. Easy learning from label proportions. *CoRR*, abs/2302.03115, 2023. doi: 10.48550/arXiv.2302.03115. URL https://doi.org/10.48550/arXiv.2302.03115.

L. Chen, Z. Huang, and R. Ramakrishnan. Cost-based labeling of groups of mass spectra. In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 167–178, 2004.

Lin Chen, Thomas Fu, Amin Karbasi, and Vahab Mirrokni. Learning from aggregated data: Curated bags versus random bags. *arXiv*, 2023. URL https://arxiv.org/abs/2305.09557.

Shuo Chen, Bin Liu, Mingjie Qian, and Changshui Zhang. Kernel k-means based framework for aggregate outputs classification. In Yücel Saygin, Jeffrey Xu Yu, Hillol Kargupta, Wei Wang, Sanjay Ranka, Philip S. Yu, and Xindong Wu, editors, *ICDM*, pages 356–361, 2009.

Dara. Wine ratings. https://www.kaggle.com/datasets/dbahri/wine-ratings/data, 2018. Licensed under CC BY-NC-SA 4.0.

Nando de Freitas and Hendrik Kück. Learning about individuals from group statistics. In *UAI*, pages 332–339, 2005.

L. M. Dery, B. Nachman, F. Rubbo, and A. Schwartzman. Weakly supervised classification in high energy physics. *Journal of High Energy Physics*, 2017(5):1–11, 2017.

G. Dulac-Arnold, N. Zeghidour, M. Cuturi, L. Beyer, and J. P. Vert. Deep multi-class learning from label proportions. *CoRR*, abs/1905.12909, 2019. URL http://arxiv.org/abs/1905.12909.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Jerónimo Hernández-González, Iñaki Inza, and José Antonio Lozano. Learning bayesian network classifiers from label proportions. *Pattern Recognit.*, 46(12):3425–3440, 2013.

J. Hernández-González, I. Inza, L. Crisol-Ortíz, M. A. Guembe, M. J. Iñarra, and J. A. Lozano. Fitting the data from embryo implantation prediction: Learning from label proportions. *Statistical methods in medical research*, 27(4):1056–1066, 2018.

D. Kotzias, M. Denil, N. de Freitas, and P. Smyth. From group to individual labels using deep features. In *Proc. SIGKDD*, pages 597–606, 2015.

Xintian Li and Aron Culotta. Domain adaptation for learning from label proportions using domain-adversarial neural network. *SN Comput. Sci.*, 4(5):615, 2023a.

Xintian Li and Aron Culotta. Domain adaptation for learning from label proportions using domain-adversarial neural network. *SN Computer Science*, 4(5):615, 2023b.

Kuan-Yu Lin, Hsuan-Yin Lin, Yu-Pin Hsu, and Yu-Chih Huang. Age aware scheduling for differentially-private federated learning. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 398–403, 2024. doi: 10.1109/ISIT57864.2024.10619208.

J. Liu, B. Wang, Z. Qi, Y. Tian, and Y. Shi. Learning from label proportions with generative adversarial networks. In *Proc. NeurIPS*, pages 7167–7177, 2019.

Jiabin Liu, Bo Wang, Xin Shen, Zhiquan Qi, and Yingjie Tian. Two-stage training for learning from label proportions. In Zhi-Hua Zhou, editor, *Proc. IJCAI*, pages 2737–2743, 2021.

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

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

David R. Musicant, Janara M. Christensen, and Jamie F. Olson. Supervised learning by training on aggregate outputs. In *ICDM*, pages 252–261. IEEE Computer Society, 2007.

J. Nandy, R. Saket, P. Jain, J. Chauhan, B. Ravindran, and A. Raghuveer. Domain-agnostic contrastive representations for learning from label proportions. In *Proc. CIKM*, pages 1542–1551, 2022.

Conor O'Brien, Arvind Thiagarajan, Sourav Das, Rafael Barreto, Chetan Verma, Tim Hsu, James Neufield, and Jonathan J Hunt. Challenges and approaches to privacy preserving post-click conversion prediction, 2022. URL https://arxiv.org/abs/2201.12666.

S. N. Ørting, J. Petersen, M. Wille, L. Thomsen, and M. de Bruijne. Quantifying emphysema extent from weakly labeled ct scans of the lungs using label proportions learning. In *The Sixth International Workshop on Pulmonary Image Analysis*, pages 31–42, 2016.

Giorgio Patrini, Richard Nock, Tibério S. Caetano, and Paul Rivera. (almost) no label no cry. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 190–198, 2014.

N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le. Estimating labels from label proportions. *J. Mach. Learn. Res.*, 10:2349–2374, 2009a.

Novi Quadrianto, Alexander J. Smola, Tibério S. Caetano, and Quoc V. Le. Estimating labels from label proportions. *J. Mach. Learn. Res.*, 10:2349–2374, 2009b.

S. Rueping. SVM classifier estimation from group probabilities. In *Proc. ICML*, pages 911–918, 2010.

Steven Ruggles, Sarah Flood, Matthew Sobek, Daniel Backman, Annie Chen, Grace Cooper, Stephanie Richards, Renae Rodgers, and Megan Schouweiler. IPUMS USA: Version 15.0 [dataset], 2024. URL https://doi.org/10.18128/D010.V15.0.

Rishi Saket, Aravindan Raghuveer, and Balaraman Ravindran. On combining bags to better learn from label proportions. In *AISTATS*, volume 151 of *Proceedings of Machine Learning Research*, pages 5913–5927. PMLR, 2022. URL https://proceedings.mlr.press/v151/saket22a.html.

Marco Stolpe and Katharina Morik. Learning from label proportions by optimizing cluster model selection. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *ECML PKDD Proceedings, Part III*, volume 6913, pages 349–364. Springer, 2011.

Marcelo Tallis and Pranjul Yadav. Reacting to variations in product demand: An application for conversion rate (cr) prediction in sponsored search. *arXiv preprint arXiv:1806.08211*, 2018.

J. Wojtusiak, K. Irvin, A. Birerdinc, and A. V. Baranova. Using published medical results and non-homogenous data in rule learning. In *Proc. International Conference on Machine Learning and Applications and Workshops*, volume 2, pages 84–89. IEEE, 2011.

F. X. Yu, K. Choromanski, S. Kumar, T. Jebara, and S. F. Chang. On learning from label proportions. *CoRR*, abs/1402.5902, 2014. URL http://arxiv.org/abs/1402.5902.

Felix X. Yu, Dong Liu, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang. $\propto$SVM for learning with label proportions. In *ICML*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 504–512, 2013.

Zackthoutt. Wine reviews. https://www.kaggle.com/datasets/zynicide/wine-reviews, 2017. Licensed under CC BY-NC-SA 4.0.

J. Zhang, Y. Wang, and C. Scott. Learning from label proportions by learning with label noise. In *Proc. NeurIPS*, 2022a.

Meng Zhang, Ermin Wei, Randall Berry, and Jianwei Huang. Age-dependent differential privacy. *SIGMETRICS Perform. Eval. Rev.*, 50(1):115–116, 2022b. ISSN 0163-5999. doi: 10.1145/3547353.3526953. URL https://doi.org/10.1145/3547353.3526953.

A USEFUL CONCEPTS

A.1 EMBEDDING SPACE REPRESENTATION

For a Hilbert space $\mathcal{H}$ of real-valued functions defined over $\boldsymbol{\mathcal{X}}$, for every $\mathbf{x} \in \boldsymbol{\mathcal{X}}$ s.t. the mapping $L_{\mathbf{x}} : \mathcal{H} \to \mathbb{R}$ given by $L_{\mathbf{x}}(f) = f(\mathbf{x})$ is bounded i.e., $|L_{\mathbf{x}}(f)| \leq C_{\mathbf{x}}\|f\|_{\mathcal{H}}$, the Riesz Representation Theorem guarantees the existence of $g_{\mathbf{x}} \in \mathcal{H}$ s.t. $L_{\mathbf{x}}(f) = \langle f, g_{\mathbf{x}} \rangle_{\mathcal{H}}$. As we study regression tasks (typically neural regression) in this work, we can assume boundedness and define $f(\mathbf{x}) = \mathbf{r}_f^{\mathsf{T}} \phi(\mathbf{x})$ where ϕ is a mapping to a real-vector in an embedding space, and $\mathbf{r}_f$ the representation of f in that space.

The function class under consideration in our experiments is a neural network with the final layer being a single node (without any activation) as we are studying the scalar regression use-case. In this case, the embedding space is learnt during training. Here, $\phi(\mathbf{x})$ is the output of penultimate layer of neural network and $\mathbf{r}_f$ are the parameters of the final layer (a single node).

A.2 EXCLUDING REGULARIZATION TERM IN LOSS FUNCTION

The regularization term $R(h, \mathcal{S}, \mathcal{T}) = \left| 1/(mk) \sum_{i=1}^{mk} \left(h(\mathbf{x}_i)^2 - h(\mathbf{z}_i)^2 \right) \right|$ enforces that the *average* squared-predictions of h i.e. the squared ℓ_2-norm of h, on the source and the target domains should be similar. However, covariate-shifts often approximately preserve the ℓ_2-norm of predictors for e.g. if they are *rotational* in the embedding space $\{\phi(\mathbf{x})\}$. Therefore, for practical settings the contribution of $R(h, \mathcal{S}, \mathcal{T})$ (for example, to gradient updates in neural networks) can be ignored and the term is omitted from the BagCSI loss.

This claim is empirically validated in Tables 10, 11 and 12 which establish that the magnitude of $R(h, \mathcal{S}, \mathcal{T})$ term is very small compared to BagCSI. We report the average loss values over 5 random partitionings of the training data into bags.

It is also established empirically that adding the regularization term $R(h, \mathcal{S}, \mathcal{T})$ in the loss does not result in significant improvement. This can be observed in the experimental results presented in the Tables 14, 15 and 13 which are obtained by doing a hyperparameter search within a range $W = \{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10 \times -2\}$ of the weight for the regularization term in the overall loss.

A.3 SAMPLE COMPLEXITY ANALYSIS

Given that with probability at least $1 - 2q_\infty \exp\left(-\nu m/(64k^2)\right) - 4q_1 \exp\left(-2\nu^2 mk/512\right)$, $\forall h \in \mathcal{F}_{\text{err}}, \overline{\varepsilon}(\mathcal{B}, h) \geq \frac{\nu}{16k}$, we show that if we chose $m \geq O\left(\left(p \left(\log\left(\frac{k}{\nu}\right) + \log\log\left(\frac{1}{\delta}\right)\right) + \log\frac{1}{\delta} \right) \max\left\{ \frac{1}{k\nu^2}, \frac{k^2}{\nu} \right\} \right)$, then with probability at least $1 - \delta$, $\forall h \in \mathcal{F}_{\text{err}}, \overline{\varepsilon}(\mathcal{B}, h) \geq \frac{\nu}{16k}$.

Note that, $q_1 = N_1(\nu/64, \mathcal{F}, 4mk)$ and $q_\infty = N_\infty(\nu/32k, \mathcal{F}, 2mk)$.

From (1), $N_1(\xi, \mathcal{F}, N) \leq N_\infty(\xi, \mathcal{F}, N) \leq (eN/\xi p)^p$. Hence, $q_1 \leq \left(\frac{256emk}{\nu p}\right)^p$ and $q_\infty \leq \left(\frac{64emk^2}{\nu p}\right)^p$.

Let $R_\infty = q_\infty \exp\left(-\nu m/(64k^2)\right)$ and $R_1 = q_1 \exp\left(-2\nu^2 mk/512\right)$.

Substituting $m = c\left(p \left(log\left(\frac{k}{\nu}\right) + \log\log\left(\frac{1}{\delta}\right)\right) + log\frac{1}{\delta} \right) \max\left\{ \frac{1}{k\nu^2}, \frac{k^2}{\nu} \right\}$, where c is some large constant,

$$\log R_\infty = p \log\left(\frac{64emk^2}{\varepsilon p}\right) - \frac{\nu m}{64k^2} \leq p\left(\log 64em + \log\frac{k^2}{\nu} - \log p\right) - \frac{c}{64}\left(p\log\frac{k}{\epsilon} + p\log\log\frac{1}{\delta} + \log\frac{1}{\delta}\right).$$

As $\log 64em \leq \log 64ec + \log p + \log\log\frac{k}{\nu} + \log\log\log\frac{1}{\delta} + \log\log\frac{1}{\delta} + \log\frac{k^2}{\nu} + \log\frac{1}{k\nu^2}$,

$$\log R_\infty \leq p\left[\log 64ec + \log p + \log\log\frac{k}{\nu} + \log\log\log\frac{1}{\delta} + \log\log\frac{1}{\delta} + \log\frac{k^2}{\nu} + \log\frac{1}{k\nu^2}\log\frac{k^2}{\nu} - \log p\right]$$
$$- \frac{c}{64}\left(p\log\frac{k}{\nu} + p\log\log\frac{1}{\delta} + \log\frac{1}{\delta}\right)$$
$$\leq -\log\left(\frac{4}{\delta}\right),$$

for a large enough constant c and for small enough δ. Hence, $R_\infty \leq \delta/4$. Using a similar analysis, we also obtain that, $R_1 \leq \delta/8$. Thus, $1 - 2R_\infty - 4R_1 \geq 1 - \delta$ follows for a large enough constant c and small enough δ, completing the proof.

B USEFUL ANALYTICAL TOOLS

B.1 HOEFFDING'S INEQUALITY

We use the Hoeffding's inequality which is stated below.

Theorem B.1 (Hoeffding)**.** *Let $X_1, \ldots, X_n$ be independent random variables, s.t. $a_i \leq X_i \leq b_i$, $\Delta_i = b_i - a_i$ for $i = 1, \ldots, n$. Then, for any $t > 0$,*

$$\Pr\left[\left|\sum_{i=1}^n X_i - \sum_{i=1}^n \mathbb{E}[X_i]\right| > t\right] \leq 2 \cdot \exp\left(-\frac{2t^2}{\sum_{i=1}^n \Delta_i^2}\right).$$

B.2 PSEUDO-DIMENSION

As defined in Section 3, $\mathcal{F}$ is a class of real-values functions (regressors) mapping $\mathbb{R}^d$ to $[0, 1]$.

A finite subset $\boldsymbol{\mathcal{X}} = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^d$ is *pseudo-shattered* by $\mathcal{F}$ if there exist $r_1, r_2, \ldots, r_N$ such that for each $b \in \{0, 1\}^m$, there is a function f_b in $\mathcal{F}$ with $sgn(f_b(x_i) - r_i) = b_i$ for $1 \leq i \leq N$.

$\mathcal{F}$ has pseudo-dimension p if p is the cardinality of the largest finite subset of $\mathbb{R}^d$ that is pseudo-shattered by $\mathcal{F}$. If no such largest finite subset exists, $\mathcal{F}$ is said to have infinite pseudo dimension.

C ERROR BOUND DEGRADATION WITH BAG SIZE

The bag-to-instance generalization error bound established in Theorem 3.2 degrades linearly with bag-size. This section provides a justification of why this degradation with bag-size is unavoidable through the example below:

Consider $D_\mathcal{T}$ where each instance-label in is drawn iid from $[0, 1]$. Let $y_1, \ldots, y_k$ be the instance-labels within a random bag B, and by construction each y_i is iid and drawn u.a.r. from $[0, 1]$. Using simple integration we obtain $\mathrm{E}[y_i - 1/2] = 0$ and $\mathrm{E}[(y_i - 1/2)^2] = 1/12$. Consider a regressor h with a constant prediction of $1/2$. The expected loss on a random bag is $\mathrm{E}[((\sum_{i=1}^k y_i)/k - 1/2)^2] = \mathrm{E}[(\sum_{i=1}^k y_i - k/2)^2]/k^2 = 1/(12k)$. Using Chernoff bounds we obtain with high probability, that the average loss on m iid sampled bags $\mathcal{B}$ satisfies $\bar{\varepsilon}(\mathcal{B}, h) \approx 1/(12k)$. On the other hand, the expected distributional instance-level loss is simply $\mathrm{E}[(y - 1/2)^2] = 1/12$ where y is chosen u.a.r. from $[0, 1]$, and thus $\varepsilon(D_\mathcal{T}, h) = 1/12$ and therefore one needs to incur a blowup of a factor linear in bag-size k.

D BASELINE TECHNIQUES

In Li and Culotta [2023a], authors define several baselines and propose new methods for domain adaptation in LLP setting for classification tasks. We adapt these methods for regression tasks and consider those as baselines. These baselines are defined in Sections D.1, D.2, D.3 and D.4. In literature on domain adaptation (for non-LLP settings) [Long et al., 2015, 2017], it has been shown that approaches using MMD (maximum mean discrepancy) based objectives work well. Hence, we also define a baseline that uses similar objective adapted for our setting in Section D.5.

D.1 AVERAGE FEATURE METHOD (AF)

The feature vectors in a bag are averaged and then predictions are made for the bag-averaged feature vectors via a neural network. The L2 loss function is used to compute difference between the predictions and bag level labels for both the source and target domain, the sum of which is used as the objective for optimization.

Let us define average bag feature by $\bar{x}_B$ such that,

$$\bar{x}_B = \frac{\sum_{\mathbf{x} \in B} \mathbf{x}}{|B|}$$

Then, the objective is defined as follows.

$$J(h, \mathcal{S}, \mathcal{B}) = \sum_{B, y_B \in \mathcal{T}} \left(y_B - h\left(\bar{x}_B\right)\right)^2 + \hat{\varepsilon}(\mathcal{S}, h)$$

D.2 LABEL REGULARIZATION METHOD (LR)

This method is similar to Average Input Method with the only difference that predictions are made via neural network for each of the feature vectors in a bag first and then the predictions are averaged.

$$J(h, \mathcal{S}, \mathcal{B}) = \hat{\varepsilon}(\mathcal{S}, h) + \bar{\varepsilon}(\mathcal{B}, h)$$

D.3 AVERAGE FEATURE DANN METHOD (AF-DANN)

In Sections D.1 and D.2, the objective function just aimed to fit the model onto the the data from source and domain data without considering any shift in the distribution of the source and domain datasets. Average Input DANN (Domain Adversarial Neural Network) Method incorporates additional term in the Average Feature Method's objective to learn features invariant to domain and then use those features for making predictions. This is achieved by introducing an adversarial loss in form of domain prediction. The features from penultimate layer of the neural network are used to classify the input feature vector as belonging to the source/target domain. We denote this domain classifier by $h_d : x \rightarrow [0, 1]$ such that $h_d(x) = \sigma(W_{h_d}^T(\phi_h(x)) + b_{h_d})$ where σ denotes the sigmoid function and h is the actual function approximator. If the classifier is not able to correctly classify labels, it means that the feature representations learnt by the network are invariant to the domain shift. The overall objective is given by J as follows.

$$J(h, \mathcal{S}, \mathcal{B}) = \sum_{B, y_B \in \mathcal{T}} \left(y_B - h\left(\bar{x}_B\right)\right)^2 + \hat{\varepsilon}(\mathcal{S}, h) - \lambda(L_D)$$

$$L_D = \sum_{\mathbf{x}, y \in \mathcal{S}} \mathcal{L}(1, h_d(\mathbf{x})) + \sum_{B, y_B \in \mathcal{T}} \sum_{\mathbf{x} \in B} \mathcal{L}(0, h_d(\mathbf{x}))$$

$$\mathcal{L}(y, \hat{y}) = -ylog(\hat{y}) - (1 - y)log(1 - \hat{y})$$

We call L_D the domain loss. This objective is optimized in two steps. In the first step, J is minimized while keeping (W_{h_d} and b_{h_d} fixed. In the second step, J is maximized while keeping everything but (W_{h_d} and b_{h_d} fixed. Essentially, in the first step encourage domain misclassifications so that the model learns feature representation that is invariant to domain shift present in the dataset. In the second step, the domain classifier is learnt for the updated feature representations. It is worth noting that the domain loss neither depends on the instance level labels from source domain nor does it depend on the bag level labels from target domain.

D.4 LABEL REGULARIZATION DANN METHOD (LR-DANN)

This method is similar to AF-DANN method (defined in Section D.3). The only difference comes from using label regularization loss instead of average feature loss in the objective function. The overall objective hence becomes as follows.

$$J(h, \mathcal{S}, \mathcal{T}) = \bar{\epsilon}(\mathcal{B}, h) + \hat{\epsilon}(\mathcal{S}, h) - \lambda(L_D)$$

where L_D is the same as defined in Section D.3.

D.5 DOMAIN MEAN FEATURE ALIGNMENT METHOD (DMFA)

The idea is to make the feature representations domain-invariant by reducing the distance between the mean of feature representations from the source and the target domain. The overall objective is given by J as follows.

$$J(h, \mathcal{S}, \mathcal{T}) = \bar{\epsilon}(h, \mathcal{T}) + \hat{\epsilon}(h, \mathcal{S}) + \lambda(L_{DMFA})$$

$$L_{DMFA} = \left\| \sum_{B, y_B \in \mathcal{T}} \sum_{\mathbf{x} \in B} \frac{\phi(\mathbf{x})}{|B||\mathcal{T}|} - \sum_{\mathbf{x}, y \in D_S} \frac{\phi(\mathbf{x})}{|D_S|} \right\|_2^2$$

Note that just like AF-DANN method (defined in Section D.3) and LR-DANN (defined in Section D.4), this method also doesn't leverage instance level source labels and bag level target labels in the objective function.

E DATASET PREPARATION DETAILS

E.1 SYNTHETIC DATASET

The feature vector comprises of 64 numerical features. The label is a scalar-valued continuous variable. The feature vectors are sampled from a multi-dimensional Gaussian distribution. For the Gaussian distribution, the mean vector is itself sampled from $\mathcal{N}(0, 16)$ for source domain and $\mathcal{N}(50, 16)$ for target domain. For the experiment results presented in main paper, the co-variance matrix is a diagonal matrix where the diagonal elements are sampled from $\mathcal{N}(10, 16)$ for both the source and target domain. However, we also experiment using synthetic dataset generated with non-diagonal covariance matrix, the results for which are reported in appendix. Although the process of generating co-variance matrices is same for source and target domain, the actual covariance matrices are not the same.

As we assume co-variate shift in the source and target distribution, $p(y|x)$ is same for both distributions, hence we initialize a neural network with random weights and use that for obtaining the labels corresponding to feature vectors for both the source and target data.

The train set comprises 0.2 million instances from both source and target domain. The test set comprises 65 thousand instances from target domain.

E.2 WINE DATASET

Wine dataset [Dara, 2018, Zackthoutt, 2017] is a tabular dataset with 39 boolean features indicating whether a particular word was present in the review for that wine. It also has a cardinal feature named points, which ranges between 80 (inclusive) and 100 (exclusive). The label is the price of the wine. We process feature vectors to convert all features to one hot and thus obtain a $39 \times 2 + (100 - 80) = 98$ dimensional boolean-valued multi-hot vector as input feature vector.

The labels in the dataset are skewed. To prevent the outliers from hindering the learning process, we remove the outliers by discarding features with labels in the top 5 percentile.

We split the dataset into two different domains. The source domain comprises of wines from France and the target domain comprises of wines from all countries but France. We select France as the source domain because it has enough number of instances to qualify as a separate domain and not so many that the target domain becomes small. We run another set of experiments where Italy is chosen as the source domain and the target domain comprises of wines from all countries but Italy. The results for the former are presented in the main paper (see Table 2), and those for the later configuration are presented in the appendix (see Table 16).

The train set comprises 0.5 million instances from both source and target domain. The test set comprises 0.2 million instances from target domain.

E.3 IPUMS DATASET

IPUMS [Ruggles et al., 2024] is a large tabular US Census dataset with a huge number of features. For our experiments, we select income (INCWAGE) as the label. We select a subset of feature columns comprising of the following features:

REGION, STATEICP, AGE, IND, GQ, SEX and WKSWORK2. All of these features are categorical except AGE which is cardinal. We convert GQ (5 categories), SEX (2 categories), WKSWORK2 (7 categories) to one-hot representations while keeping others intact as they have large number of categories which makes one-hot representations impractical.

We consider the data from 1970 as the source domain and data from 2022 as the target domain. Since, the labels (INCWAGE) were large in magnitude, we standardized the labels using $y \to (y - \mu_Y)/\sigma_Y$ by estimating the mean and variance using source domain labels and target domain train labels only.

The train set comprises 1.3 million instances from source and 9.4 million instances from target domain. The test set comprises 0.3 million instances from target domain.

E.4 CRITEO SSCL DATASET

Criteo Sponsored Search Conversion Log Dataset [Tallis and Yadav, 2018] comprises of 90 days of Criteo live traffic data. Every row in the dataset corresponds to a click (product related advertisement) that was displayed to a user. The preprocessing of the dataset is the same as done by Brahmbhatt et al. [2024].

We remove all the rows where the label is -1 because these instances indicate no conversion. Further, we remove all the rows where NaN or -1 is present. For our experiments, we select sales_amount_in_euro as the label. The feature representation comprises of 15 categorical (product_age_group, device_type, audience_id, product_gender, product_brand, product_category_1, product_category_2, product_category_3, product_category_4, product_category_5, product_category_6, product_category_7, product_title, partner_id, user_id) and 3 numerical features (time_delay_for_conversion, nb_clicks_1week, product_price). An embedding of dimension 8 is learnt for all the categorical features in the neural network.

The train set comprises 0.5 million instances from source and 0.9 million instances from target domain. The test set comprises 0.2 million instances from target domain.

F HYPERPARAMETER SEARCH

We use grid search for finding optimal values of λ and learning rate. The values used in grid search are on a logarithmic scale. We try out two different optimizers for all experiments mentioned in the main paper - Adam and SGD and report scores corresponding to the best performer. We observed that Adam works better for most of the cases, so we perform experiments described in Appendix with Adam optimizer only.

Note that the magnitude of $\xi^2(\mathcal{S}, \mathcal{B})$ term in BagCSI loss depends on the embedding and hence the initialization of the network. Hence, we scale $\xi^2(\mathcal{S}, \mathcal{B})$ value to match $\bar{\varepsilon}(\mathcal{B}, h)$. Effectively the BagCSI contains $(\kappa \times \lambda_3)\xi^2(\mathcal{S}, \mathcal{B})$, where $\kappa = \frac{\bar{\varepsilon}(\mathcal{B},h)}{\xi^2(\mathcal{S},\mathcal{B})}$. It must be noted that κ is a constant and no gradient flows through it. $(\kappa \times \lambda_3)$ is an adaptive weight for $\xi^2(\mathcal{S}, \mathcal{B})$ term. We do this for all methods (including baselines) that use a λ hyperparameter.

G ADDITIONAL EXPERIMENTS

In addition to the experiments for which the results were shared in the main paper, we conduct a few more experiments and extensive ablation studies. The setup and results for these experiments are shared in the following sub-sections. More precisely, we perform the following experiments:

1. We create a different source-target domain split in Wine dataset by choosing wines from Italy in the source domain partition and wines from all other countries in the target domain partition. The results are reported in Table 16.
2. We create another synthetic dataset where we choose a non-diagonal covariance matrix while keeping all other configurations the same. The results are reported in Table 17.
3. We empirically study the impact of excluding regularization term in the loss function on the performance of proposed methods. The experimental setup and results are detailed in Appendix A.2.
4. We also perform experiments to study the impact on performance of different algorithms by varying the amount of covariate shift in the synthetic dataset. The setup and results are detailed in Appendix G.3.

Table 10: Comparison of magnitude of the regularization term $\mathcal{R}(h, \mathcal{S}, \mathcal{T})$ and the magnitude of BagCSI loss on IPUMS dataset.

Dataset	PLWFA		BLWFA	
Method / Bag Size	$\mathcal{R}(h, \mathcal{S}, \mathcal{T})$	BagCSI	$\mathcal{R}(h, \mathcal{S}, \mathcal{T})$	BagCSI
8	0.01	1.59	0.02	1.59
32	0.03	1.62	0.03	1.62
128	0.04	1.62	0.05	1.62
256	0.06	1.62	0.06	1.63

Table 11: Comparison of magnitude of the regularization term $\mathcal{R}(h, \mathcal{S}, \mathcal{T})$ and the magnitude of BagCSI loss on Criteo dataset.

Dataset	PLWFA		BLWFA	
Method / Bag Size	$\mathcal{R}(h, \mathcal{S}, \mathcal{T})$	BagCSI	$\mathcal{R}(h, \mathcal{S}, \mathcal{T})$	BagCSI
64	17.52	577.50	17.44	569.73
128	17.46	611.63	17.40	602.30
256	17.97	654.50	17.85	644.50
512	18.19	673.34	18.22	662.61

Table 12: Comparison of magnitude of the regularization term $\mathcal{R}(h, \mathcal{S}, \mathcal{T})$ and the magnitude of BagCSI loss on performance for Wine dataset.

Dataset	PLWFA		BLWFA	
Method / Bag Size	$\mathcal{R}(h, \mathcal{S}, \mathcal{T})$	BagCSI	$\mathcal{R}(h, \mathcal{S}, \mathcal{T})$	BagCSI
8	0.66	704.05	0.62	698.44
32	1.04	707.39	1.14	700.56
128	1.32	708.33	1.33	701.18
256	1.73	713.34	1.75	706.06

Table 13: Effect of adding the regularization term $\mathcal{R}(h, \mathcal{S}, \mathcal{T})$ to loss on performance for Wine dataset. For significant impact of the extra regularization term, the hyperparameter search is done within range $W = \{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$.

Dataset	PLWFA		BLWFA	
Method / Bag Size	$w_R = 0$	best $w_R \in W$	$w_R = 0$	best $w_R \in W$
8	183.0 ± 0.6	255.57 ± 3.81	180.9 ± 0.5	255.47 ± 3.91
32	186.6 ± 1.0	260.85 ± 3.76	184.6 ± 0.7	259.80 ± 3.71
128	189.0 ± 0.8	270.23 ± 3.69	186.0 ± 0.8	270.23 ± 3.66
256	188.9 ± 1.2	276.27 ± 3.72	188.9 ± 1.2	276.25 ± 3.73

G.1 EXPERIMENTS WITH MIXED BAG SIZES

We test the performance of all the baselines and proposed methods when using a non-uniform bag size. The dataset is partitioned into bags of different sizes. More specifically, we use 2 different techniques to have mixed size bags:

- *SBB* is sample balanced bagging. For a particular bag size, there are an equal number of samples that belong to a bag of that size. Hence, if there are n_1 bags of size k_1, and n_2 bags of size k_2, then $n_1 k_1 = n_2 k_2$.
- *BBB* is bag balanced bagging. There are equal number of bags of each size. Hence, if there are n_1 bags of size k_1, and n_2 bags of size k_2, then $n_1 = n_2$.

Clearly, SBB will have more bags of smaller sizes compared to BBB. Every bag is of the size 8, 32, 128 or 256. Tables 8 and 9 contain the results for experiments with mixed bag sizes. It can be inferred from the results that the scores with mixed bag sizes are mostly an interpolation (not necessarily linear) of the results with uniform bag sizes. Scores with BBB strategy for mixing bags are worse compared to SBB since SBB has a higher proportion of small sized bags compared to BBB.

G.2 EXPERIMENTS WITH CORRELATED BAGS

We test the performance of all the baselines and proposed methods with correlated bags as opposed to random bags used for all other experiments in this paper. To partition the dataset into correlated bags, we select a feature and create bags such that all the samples in that bag have the same value of that feature if the feature is categorical. If the feature is numerical, we sort the dataset on the basis of that feature and use consecutive samples for creating the bags. Since all the features in Wine dataset are binary, we did not perform experiments for it. We used *REGION* for IPUMS and *product_brand* for Criteo SSCL as the correlated feature. Tables 5, 6 and 7 contain results for experiments with correlated bags on IPUMS, Synthetic datasets and Criteo SSCL respectively. It can be inferred that the standard deviation values are very low. This is expected because across different runs, similar bags would be created unlike experiments for un-correlated bags where the bags created for each run would comprise of a different set of instances.

G.3 SYNTHETIC DATASET WITH VARYING PERTURBATIONS

We also conduct experiments to analyze the impact on performance of different methods by varying the amount of covariate shift in the source and target domains of the synthetic datasets. The covariate shift can be controlled using the mean and standard deviation of the source and target distributions.

Table 14: Effect of adding the regularization term $\mathcal{R}(h, \mathcal{S}, \mathcal{T})$ to loss on performance for IPUMS dataset. For significant impact of the extra regularization term, the hyperparameter search is done within range $W = \{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$.

Dataset	PLWFA		BLWFA	
Method / Bag Size	$w_R = 0$	best $w_R \in W$	$w_R = 0$	best $w_R \in W$
8	1.15 ± 0.00	1.23 ± 0.01	1.14 ± 0.00	1.23 ± 0.01
32	1.18 ± 0.00	1.32 ± 0.02	1.16 ± 0.00	1.32 ± 0.02
128	1.25 ± 0.01	1.38 ± 0.03	1.22 ± 0.00	1.38 ± 0.03
256	1.29 ± 0.01	1.42 ± 0.02	1.25 ± 0.01	1.42 ± 0.02

Table 15: Effect of adding the regularization term $\mathcal{R}(h, \mathcal{S}, \mathcal{T})$ to loss on performance for Criteo dataset. For significant impact of the extra regularization term, the hyperparameter search is done within range $W = \{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$.

Dataset	PLWFA		BLWFA	
Method / Bag Size	$w_R = 0$	best $w_R \in W$	$w_R = 0$	best $w_R \in W$
64	204.71 ± 2.6	256.11 ± 4.45	204.62 ± 2.4	255.23 ± 4.57
128	226.39 ± 2.9	264.97 ± 3.61	226.33 ± 2.9	264.45 ± 3.50
256	240.55 ± 3.3	279.45 ± 2.06	240.39 ± 3.2	279.48 ± 2.06
512	254.46 ± 5.5	291.38 ± 2.44	254.36 ± 5.5	290.96 ± 2.46

Table 16: MSE scores for different methods and bag sizes on the wine dataset (averaged over 20 runs) using wines from Italy as the source domain. The source instance loss is 204.73 ± 2.7 and target instance loss is 173.91 ± 0.2. Lower is better.

Method / Bag Size	8	32	128	256
Bagged Target	**176.2 ± 0.4**	**180.1 ± 0.9**	193.8 ± 4.2	208.0 ± 4.5
AF	199.3 ± 2.3	203.2 ± 2.7	203.1 ± 2.5	203.7 ± 2.1
LR	196.0 ± 1.1	201.0 ± 1.0	203.0 ± 1.1	203.2 ± 0.8
AF-DANN	193.5 ± 3.5	195.6 ± 3.3	196.2 ± 3.1	194.7 ± 3.0
LR-DANN	195.4 ± 2.5	198.6 ± 3.4	199.7 ± 3.4	199.0 ± 4.1
DMFA	195.5 ± 2.2	201.0 ± 1.2	202.5 ± 1.3	203.2 ± 1.1
PL-WFA (our)	186.2 ± 1.0	188.8 ± 0.7	190.0 ± 0.7	190.3 ± 0.8
BL-WFA (our)	184.5 ± 0.6	187.2 ± 1.8	**188.4 ± 1.2**	**188.0 ± 0.8**

Table 17: MSE scores for different methods and bag sizes on the synthetic dataset (averaged over 20 runs). The source instance loss is 558.3179 ± 65.77 and target instance loss is 9.7217 ± 0.40. Lower is better.

Method / Bag Size	8	32	128	256
Bagged Target	29.53 ± 0.94	58.06 ± 1.93	128.45 ± 7.02	195.41 ± 9.34
AF	75.19 ± 3.30	104.36 ± 4.7	146.00 ± 11.7	207.08 ± 15.78
LR	28.36 ± 0.54	54.99 ± 1.74	120.08 ± 5.78	194.86 ± 11.18
AF-DANN	74.09 ± 4.13	107.31 ± 5.7	152.74 ± 24.0	203.54 ± 16.14
LR-DANN	30.40 ± 0.69	60.58 ± 2.42	130.30 ± 7.87	185.38 ± 25.97
DMFA	**28.07 ± 0.63**	**54.65 ± 2.00**	118.71 ± 7.15	175.68 ± 15.55
PL-WFA (our)	33.75 ± 0.67	63.86 ± 2.43	119.87 ± 5.51	174.12 ± 7.47
BL-WFA (our)	39.03 ± 3.73	65.45 ± 3.86	**116.92 ± 17.7**	**159.08 ± 19.62**

The ϵ parameter is a measure of the the perturbation between the mean vectors of the source and target distributions, and δ is that for the perturbation between the covariance matrices. Specifically, a target distribution is given by a 64-dimensional Gaussian where each entry is iid, sampled from $N(50, 8)$ while Σ is a diagonal matrix where each diagonal element is the magnitude of an iid value sampled from $N(10, 8)$. For each (ϵ, δ), the source distribution is $N(\mu', \sigma')$ where $\mu' = \mu - \epsilon\Delta$ and Δ is a vector with iid values samples from $N(50, 8)$. The diagonal matrix Σ' is obtained by adding the magnitude of value sampled from iid $N(0, 8\delta^2)$ to each diagonal entry of Σ.

We perform experiments for different perturbations in the mean vector (using ϵ) and covariance matrix (using δ) of source and target distributions. As expected, with increasing perturbations, the scores become higher. Since MSE scores worsen more consistently with increase in mean perturbation as compared to perturbation in covariance matrix, we infer that the impact of increasing mean perturbation is more prominent compared to the perturbation in covariance matrix. Table 18 contains scores for different combinations of perturbation values.

Table 18: Effect of covariate shift in synthetic data on the performance (MSE scores) of different algorithms for different bag sizes. Lower is better.

ϵ	δ	AF	LR	AFDANN	LRDANN	DMFA	PLWFA	BLWFA
For bag size **8**, Bagged Target scores 0.67 ± 0.06								
0	0.5	4.56 ± 0.83	3.93 ± 0.25	4.85 ± 0.32	4.58 ± 0.32	**3.49 ± 0.45**	3.78 ± 0.24	3.73 ± 0.39
	1	3.85 ± 0.95	4.06 ± 1.55	4.40 ± 0.60	4.25 ± 0.52	**3.36 ± 0.34**	4.14 ± 1.50	4.06 ± 1.52
0.5	0	3.16 ± 0.27	2.95 ± 0.42	3.67 ± 0.21	3.58 ± 0.21	2.85 ± 0.49	2.84 ± 0.45	**2.81 ± 0.48**
	0.5	2.90 ± 0.27	2.60 ± 0.14	3.18 ± 0.15	3.15 ± 0.09	2.58 ± 0.18	**2.57 ± 0.14**	2.57 ± 0.22
	1	2.81 ± 0.43	2.43 ± 0.20	3.32 ± 0.57	2.89 ± 0.21	2.43 ± 0.18	2.44 ± 0.23	**2.36 ± 0.16**
1	0	5.56 ± 0.74	4.56 ± 0.24	7.65 ± 0.63	6.35 ± 0.75	4.55 ± 0.26	4.50 ± 0.30	**4.44 ± 0.23**
	0.5	5.58 ± 0.67	4.61 ± 0.36	7.76 ± 0.80	6.43 ± 0.36	4.63 ± 0.25	**4.55 ± 0.32**	4.58 ± 0.38
	1	5.47 ± 0.79	4.71 ± 0.31	7.90 ± 0.79	6.46 ± 0.57	4.70 ± 0.35	4.70 ± 0.35	**4.59 ± 0.37**
For bag size **32**, Bagged Target scores 16.51 ± 2.17								
0	0.5	10.07 ± 1.51	10.07 ± 1.70	10.18 ± 1.07	9.69 ± 0.94	9.57 ± 1.93	9.22 ± 1.83	**8.21 ± 1.04**
	1	10.47 ± 2.21	10.23 ± 2.09	9.91 ± 1.21	9.65 ± 1.22	9.85 ± 1.82	9.29 ± 2.02	**9.19 ± 2.09**
0.5	0	8.31 ± 2.03	7.58 ± 0.95	8.53 ± 1.63	7.28 ± 0.79	7.49 ± 1.01	6.61 ± 0.96	**6.54 ± 0.98**
	0.5	7.60 ± 1.66	7.12 ± 0.63	7.74 ± 2.06	6.88 ± 0.78	7.07 ± 0.52	**6.05 ± 0.55**	6.07 ± 0.47
	1	8.27 ± 1.82	6.62 ± 0.60	7.87 ± 1.75	6.81 ± 0.78	6.68 ± 0.58	**5.63 ± 0.60**	5.71 ± 0.73
1	0	12.43 ± 1.03	10.18 ± 0.70	14.06 ± 0.80	11.11 ± 0.74	10.04 ± 0.55	**8.97 ± 0.52**	9.16 ± 0.76
	0.5	12.96 ± 1.18	10.50 ± 0.80	14.46 ± 0.56	11.59 ± 0.70	10.33 ± 0.89	**9.22 ± 0.92**	9.33 ± 0.95
	1	12.85 ± 1.52	10.49 ± 1.30	14.45 ± 1.16	12.15 ± 1.00	10.23 ± 1.23	9.27 ± 1.10	**9.15 ± 0.98**
For bag size **128**, Bagged Target scores 18.52 ± 2.01								
0	0.5	15.86 ± 1.35	16.21 ± 2.08	15.54 ± 1.42	14.74 ± 1.45	15.11 ± 1.17	**15.03 ± 1.91**	15.11 ± 1.96
	1	16.28 ± 1.45	15.85 ± 2.16	15.43 ± 1.22	15.13 ± 1.76	15.34 ± 1.43	14.92 ± 2.19	**14.62 ± 1.95**
0.5	0	13.82 ± 1.71	12.39 ± 0.93	11.91 ± 2.78	12.60 ± 0.57	12.34 ± 0.90	**11.32 ± 0.96**	11.41 ± 0.89
	0.5	13.43 ± 3.21	12.08 ± 0.83	12.74 ± 1.42	12.01 ± 0.86	12.06 ± 0.82	11.08 ± 0.83	**10.99 ± 0.84**
	1	14.54 ± 1.87	11.64 ± 0.96	12.39 ± 1.69	11.65 ± 0.84	11.65 ± 0.98	**10.56 ± 0.95**	10.60 ± 0.97
1	0	17.98 ± 1.90	15.15 ± 1.84	17.09 ± 1.27	15.27 ± 0.43	14.76 ± 1.43	**13.74 ± 1.50**	13.86 ± 1.42
	0.5	17.57 ± 2.34	15.33 ± 1.30	17.45 ± 1.15	15.85 ± 1.59	15.38 ± 1.78	**13.91 ± 1.12**	14.45 ± 2.08
	1	17.64 ± 2.21	16.30 ± 2.73	17.23 ± 1.05	15.75 ± 1.17	15.26 ± 1.74	14.42 ± 1.63	**14.38 ± 1.66**
For bag size **256**, Bagged Target scores 19.28 ± 1.91								
0	0.5	17.84 ± 1.72	17.89 ± 1.82	16.97 ± 0.95	16.97 ± 1.25	17.41 ± 1.20	**16.78 ± 1.71**	16.86 ± 1.71
	1	17.95 ± 1.59	17.99 ± 2.18	17.09 ± 1.16	16.44 ± 1.40	18.09 ± 1.83	16.99 ± 2.16	**16.98 ± 2.17**
0.5	0	16.43 ± 1.71	15.06 ± 0.91	14.97 ± 1.14	14.76 ± 0.79	15.08 ± 0.83	14.04 ± 0.86	**13.99 ± 0.96**
	0.5	15.93 ± 2.33	15.02 ± 0.52	15.18 ± 0.88	14.47 ± 1.24	14.97 ± 0.57	14.03 ± 0.56	**13.96 ± 0.51**
	1	16.44 ± 1.87	14.27 ± 0.94	14.58 ± 1.47	14.03 ± 0.79	14.23 ± 0.93	13.29 ± 0.91	**13.18 ± 0.94**
1	0	18.46 ± 2.10	16.58 ± 1.99	17.47 ± 1.43	17.22 ± 1.37	16.83 ± 1.99	15.68 ± 1.90	**15.32 ± 1.48**
	0.5	18.24 ± 2.15	17.57 ± 2.34	17.58 ± 1.13	17.62 ± 1.31	16.86 ± 1.40	**16.02 ± 1.47**	16.06 ± 1.65
	1	18.53 ± 2.16	17.31 ± 2.50	17.95 ± 1.21	17.31 ± 1.20	17.18 ± 2.31	**15.91 ± 1.90**	16.20 ± 2.35