# Domain Adaptation under Open Set Label Shift

**Anonymous Authors**[1]

## Abstract

We introduce the problem of domain adaptation under Open Set Label Shift (OSLS) where the label distribution can change arbitrarily and a new class may arrive during deployment, but the class-conditional distributions $p(x|y)$ are domain-invariant. The learner's goals here are two-fold: (a) estimate the target label distribution, including the novel class; and (b) learn a target classifier. First, we establish necessary and sufficient conditions for identifying these quantities. Second, we propose practical methods for both tasks. Unlike typical Open Set Domain Adaptation (OSDA) problems, which tend to be ill-posed and amenable only to heuristics, OSLS offers a well-posed problem amenable to more principled machinery. Experiments across numerous semi-synthetic benchmarks on vision, language, and medical datasets demonstrate that our methods consistently outperform OSDA baselines, achieving 10–25% improvements in target domain accuracy. Finally, we analyze the proposed methods, establishing finite-sample convergence to the true label marginal and convergence to optimal classifier for linear models in a Gaussian setup.

## 1. Introduction

Suppose that we wished to deploy a machine learning system to recognize diagnoses based on their clinical manifestations. If the distribution of data were static over time, then we could rely on the standard machinery of statistical prediction. However, disease prevalences are constantly changing, violating the assumption of independent and identically distributed (iid) data. In such scenarios, we might reasonably apply the *label shift* assumption, where prevalences can change but clinical manifestations cannot. When only the relative proportion of previously seen diseases can change, principled methods can detect and correcting for label shift on the fly (Saerens et al., 2002; Zhang et al., 2013; Lipton et al., 2018; Azizzadenesheli et al., 2019; Alexandari et al., 2021; Garg et al., 2020). But what if a new disease, like COVID-19, were to arrive suddenly?

Traditional label shift adaptation techniques break when faced with a previously unseen class. A distinct literature on Open Set Domain Adaptation (OSDA) seeks to handle such cases (Panareda Busto and Gall, 2017; Baktashmotlagh et al., 2019; Tan et al., 2019; You et al., 2019; Saito et al., 2018; 2020; Fu et al., 2020)). Given access to labeled *source* data and unlabeled *target* data, the goal in OSDA is to adapt classifiers in general settings where previous classes can shift in prevalence (and even appearance), and novel classes must be separated out from those previously seen. Most work on OSDA is driven by the creation of and progress on benchmark datasets (e.g., DomainNet, OfficeHome). Existing OSDA methods are heuristic in nature, addressing settings where the right answers seem intuitive but are not identified mathematically. However, absent assumptions on: (i) the nature of distribution shift among source classes and (ii) the relation between source classes and novel class, standard impossibility results for domain adaptation condemn us to guesswork (Ben-David et al., 2010).

In this work, we introduce domain adaptation under Open Set Label Shift (OSLS), a coherent instantiation of OSDA that applies the label shift assumption but allows for a new class to show up in the target distribution. Formally, the label distribution may shift between source and target $p_s(y) \neq p_t(y)$, but the class-conditional distributions among previously seen classes may not (i.e., $\forall y \in \{1, 2, \ldots, k\}, p_s(x|y) = p_t(x|y)$). Moreover, a new class $y = k + 1$ may arrive in the target period. Notably, OSLS subsumes label shift (Saerens et al., 2002; Storkey, 2009) (when $p_t(y = k + 1) = 0$) and learning from Positive and Unlabeled (PU) data (Letouzey et al., 2000; Elkan and Noto, 2008) (when $k = 1$). As with label shift and PU learning, our goals are two-fold. Here, we must (i) estimate the target label distribution $p_t(y)$ (including the novel class); (ii) train a $(k + 1)$-way target-domain classifier.

First, we characterize when the parameters of interest are identified (Sec. 2). Namely, we define a (necessary) *weak positivity* condition, which states that there exists a subset of each label's support that has zero probability mass under the novel class and that the matrix submatrix of $p(x|y)$

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the SCIS workshop.

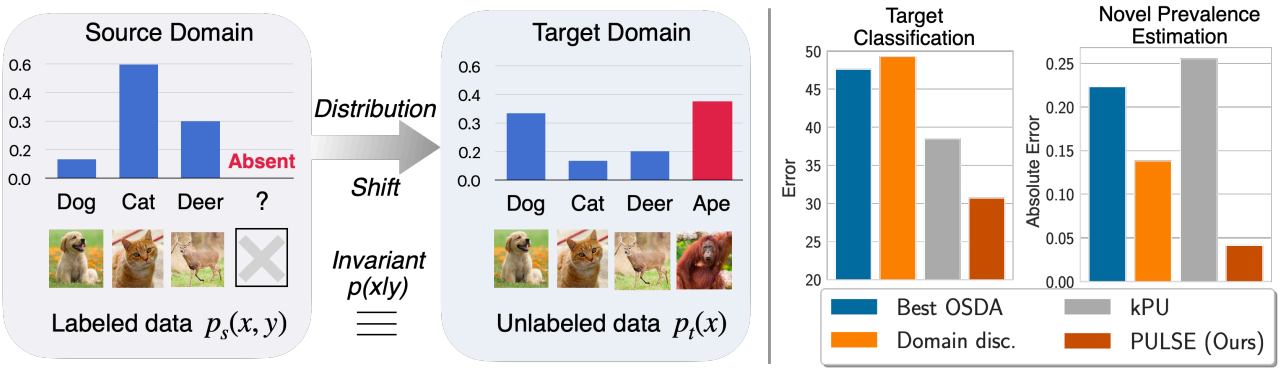



Figure 1: **Left:** *Domain Adaptation under OSLS*. An instantiation of OSDA that applies label shift assumption but allows for a new class to show up in target domain. **Right:** *Aggregated results across seven semi-synthetic benchmark datasets*. For both target classification and novel class prevalence estimation, PULSE significantly outperforms other methods (lower error is better). For brevity, we only include result for the best OSDA method. For detailed comparison, refer Sec. 4.

consisting only of rows outside the novel class's support is full rank. Moreover, we prove that weak positivity alone is not sufficient. Next, we introduce two sufficient conditions: *strong positivity* and *separability*, either of which (independently) ensures identifiability.

Focusing on cases with *strong positivity*, we show that OSLS reduces to $k$ PU learning problems (Sec. 2). However, we demonstrate that straightforward applications of this idea fail because (i) bias accumulates across the $k$ mixture proportion estimates leading to grossly underestimating the novel class's prevalence; and (ii) naive combinations of the $k$ PU classifiers are biased and inaccurate.

Thus motivated, we propose the PULSE framework, which combines methods from Positive and Unlabeled learning and Label Shift Estimation, yielding two-stage techniques for both label marginal estimation and classification (Sec. 3). Our methods build on recent advances in label shift and PU learning that leverage appropriately chosen black-box predictors to avoid the curse of dimensionality. PULSE first estimates the label shift among previously seen classes, and then re-samples the source data to formulate a single PU learning problem between (reweighted) source and target data to estimate fraction of novel class and to learn the target classifier. In particular, our procedure builds on the BBE and CVIR techniques proposed in Garg et al. (2021b).

We conduct extensive semi-synthetic experiments adapting seven benchmark datasets spanning vision (CIFAR10, CIFAR100, Entity30), natural language (Newsgroups-20), biology (Tabula Muris), and medicine (DermNet, BreakHis) (Sec. 4). Across numerous data modalities, draws of the label distributions, and model architectures, PULSE consistently outperforms generic OSDA methods, improving by 10–25% in accuracy on target domain. Moreover, PULSE outperforms methods that naively solve $k$ PU problems on both label distribution estimation and classification.

Finally, we analyze our framework (Sec. 5). First, we extend Garg et al. (2021b)'s analysis of BBE to derive finite-sample error bounds for our estimates of the label marginal. Next, we develop new analyses of the CVIR objective (Garg et al., 2021b) that PULSE relies in the classification stage. Focusing on a Gaussian setup and linear models optimized by gradient descent, we prove that CVIR converges to the true positive versus negative classifier in population. Addressing the overparameterized setting, we conduct an empirical study to elucidate efficacy of CVIR over other consistent objectives, nnPU (Kiryo et al., 2017) and uPU (Du Plessis et al., 2015).

## 2. Open Set Label Shift

For a vector $v \in \mathbb{R}^d$, we use $v_j$ to denote its $j^{\text{th}}$ entry, and for an event $E$, we let $\mathbb{I}[E]$ denote the binary indicator of the event. Let $\mathcal{X}$ be the input space and $\mathcal{Y} = \{1, 2, \ldots, k+1\}$ be the output space for classification. Let $\mathrm{P}_s$ and $\mathrm{P}_t$ be the source and target distributions and let $p_s$ and $p_t$ denote the corresponding probability density (or mass) functions. $\ell(z, y)$ denotes loss incurred by predicting $z$ when the true label is $y$. As in standard unsupervised domain adaptation, we are given independently and identically distributed (iid) samples from labeled source data $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \sim \mathrm{P}_s^n$ and iid samples from unlabeled target data $\{x_{n+1}, \ldots, x_{n+m}\} \sim \mathrm{P}_t^m$. OSLS setting generalizes the label shift and PU learning setting (refer to Sec. C for preliminaries):

**Definition 1** (Open set label shift)**.** *Define $\mathcal{Y}_t = \mathcal{Y}$ and as $\mathcal{Y}_s = \mathcal{Y} \backslash \{k+1\}$. Under OSLS, the label distribution among source classes $\mathcal{Y}_s$ may change but the class conditional $p(x|y)$ for those classes remain invariant between source and target, and the target domain may contain a novel class, i.e., $p_s(x|y = j) = p_t(x|y = j) \quad \forall j \in \mathcal{Y}_s$ and $p_s(y = k+1) = 0$. Additionally, we have non-zero support for all $k$ (previously-seen) labels in the source distribution, i.e., for all $y \in \mathcal{Y}_s$, $p_s(y) \geqslant c$ for some $c > 0$.*

Under OSLS, our goal naturally breaks down into two tasks: (i) estimate the target label marginal $p_t(y)$ (ii) train a classifier $f : \mathcal{X} \to \Delta^k$ to approximate $p_t(y|x)$.

**Identifiablity of OSLS** We assume access to population distribution, i.e., $p_s(x, y)$ and $p_t(x)$. To keep the discussion simple, we assume finite input domain $\mathcal{X}$ which can then be relaxed to continuous inputs. We relegate extension to continuous cases and proofs to App. D.

Given access to $p_t(y)$, the class conditional $p_t(x|y = k+1)$ can be obtained in closed form as $p_t(x) - \sum_{j=1}^{k} p_t(y = j)p_s(x|y = j)$. We can then simply apply Bayes rule to obtain $p_t(y|x)$. Henceforth, we will focus our discussion on identifiability of $p_t(y)$ which implies identifiability of $p_t(y|x)$. In following proposition, we present *weak positivity*, a necessary condition for $p_t(y)$ to be identifiable.

**Proposition 1** (Necessary conditions)**.** *Assume $p_t(y) > 0$ for all $y \in \mathcal{Y}_t$. Then $p_t(y)$ is identified only if $p_t(x|y = k + 1)$ and $p_s(x|y)$ for all $y \in \mathcal{Y}_s$ satisfy weak positivity, i.e., there must exists a subdomain $X_{wp} \subset X$ such that: (i) $p_t(X_{wp}|y = k + 1) = 0$; and (ii) the matrix $[p_s(x|y)]_{x\in X_{wp}, y\in\mathcal{Y}_s}$ is full column-rank.*

Intuitively, Proposition 1 states that if the target marginal doesn't lie on the vertex of the simplex $\Delta^k$, then their must exist a subdomain $X_{\text{wp}}$ where the support of novel class is zero and within $X_{\text{wp}}$, $p_t(y)$ for source classes is identifiable. While it may seem that existence of a subdomain $X_{\text{wp}}$ is enough, we show that for the OSLS problem, existence doesn't imply uniqueness (refer to App. D.1 for counter example). Next, we extend weak positivity to two stronger conditions, either of which (alone) implies identifiability.

**Proposition 2** (Sufficient conditions)**.** *The target marginal $p_t(y)$ is identified if for all $y \in \mathcal{Y}\backslash\{k + 1\}$, $p_t(x|y = k + 1)$ and $p_s(x|y)$ satisfy either: (i) Strong positivity, i.e., there exists $X_{sp} \subset \mathcal{X}$ such that $p_t(X_{sp}|y = k + 1) = 0$ and the matrix $[p_s(x|y)]_{x\in X_{sp}, y\in\mathcal{Y}_s}$ is full-rank and diagonal; or (ii) Separability, i.e., there exists $X_{sep} \subset \mathcal{X}$, such that $p_t(X_{sep}|y = k + 1) = 0$, $p_s(X_{sep}) = 1$, and the matrix $[p_s(x|y)]_{x\in X_{sep}, y\in\mathcal{Y}_s}$ is full column-rank.*

Informally, strong positivity extends weak positivity by making an additional assumption that the matrix formed by $p(x|y)$ on inputs in $X_{\text{wp}}$ is diagonal and the separability assumption extends the weak positivity condition to the full input domain of source classes instead of just $X_{\text{wp}}$.

**Reduction of OSLS to $k$ PU Problems** Under the strong positivity condition, the OSLS problem can be broken down into $k$ PU problems as follows: By treating a given source class $y_j \in \mathcal{Y}_s$ as *positive* and grouping all other classes together as *negative* we observe that the unlabeled target data is then a mixture of data from the positive and negative classes. This yields a PU learning problem and the corresponding mixture proportion gives the fraction $\alpha_j$ of class $y_j$ among the target data. By iterating this process for all source classes, we can solve for the entire target label marginal $p_t(y)$. Hence, a natural choice is to leverage this structure and solve $k$ PU problems to solve the original OSLS problem. We explain this procedure more formally in App. C.1. However, we observe that the naive reduction doesn't scale to datasets with large number of classes because of error accumulation in each of the $k$ MPEs and $k$ one-versus-other PU classifiers.

**Algorithm 1** Positive and Unlabeled learning post Label Shift Estimation (PULSE) framework

**input** : Labeled source data $\{\mathbf{X}^S, \mathbf{y}^S\}$ and unlabeled target samples $\mathbf{X}^T$.

1: Randomly split data into training $\{\mathbf{X}_1^S, \mathbf{y}_1^S\}$, $\mathbf{X}_1^T$ and hold out partition $\{\mathbf{X}_2^S, \mathbf{y}_2^S\}$, $\mathbf{X}_2^T$.

2: Train a classifier $f_s$ on labeled source data $\{\mathbf{X}_1^S, \mathbf{y}_1^S\}$.

3: Estimate label shift $\widehat{p}'_t(y = j) = \frac{\widehat{p}_t(y = j)}{\sum_{k\in\mathcal{Y}_s} \widehat{p}_t(y = k)}$ and hence importance ratios $\widehat{w}(j)$ among source classes $j \in \mathcal{Y}_s$ using Algorithm 2.

4: Re-sample training source data according to label distribution $\widehat{p}'_t$ to get $\{\widetilde{\mathbf{X}}_1^S, \widetilde{\mathbf{y}}_1^S\}$ and $\{\widetilde{\mathbf{X}}_2^S, \widetilde{\mathbf{y}}_2^S\}$.

5: Using Algorithm 3, train a discriminator $f_d$ and estimate novel class fraction $\widehat{p}_t(y = k + 1)$.

6: Assign $[f_t(x)]_j = (f_d(x))\frac{\widehat{w}(j)\cdot[f_s(x)]_j}{\sum_{k\in\mathcal{Y}_s}\widehat{w}(k)\cdot[f_s(x)]_k}$ for all $j \in \mathcal{Y}_s$ and $[f_t(x)]_{k+1} = 1 - f_d(x)$. And for all $j \in \mathcal{Y}_s$, assign $\widehat{p}_t(y = j) = (1 - \widehat{p}_t(y = k + 1)) \cdot \widehat{p}'_t(y = j)$.

**output** : Marginal estimate $\widehat{p}_t \in \Delta^k$ and classifier $f_t(\cdot)$.

## 3. The PULSE Framework for OSLS

Rather than dividing an OSLS instance into $k$ PU problems, we exploit the joint structure of the problem. We note that if only we could apply a *label shift correction* to source, i.e., re-sample source classes according to their relative proportion in the target data, then we could subsequently consider the unlabeled target data as a mixture of (i) the (reweighted) source distribution (say $p'_s(t)$); and (ii) the novel class distribution ($p_t(x|y = k+1)$) obtaining a *single* PU learning problem. We describe this formally in Sec. E.

To estimate label shift among source classes, we build on recently proposed Best Bin Estimation (BBE) technique from Garg et al. (2021b). We tailor BBE to estimate the relative fraction of previously seen classes in the target distribution by exploiting a $k$-way source classifier $f_s$ trained on labeled source data. We describe the procedure in App. E (Algorithm 2). After estimating the fraction of source classes in target, we re-sample the source data to mimic samples from distribution $p'_s(x)$. Thus, obtaining a PU learning problem instance, we resort to PU learning techniques to (i) estimate the fraction of novel class $p_t(y = k + 1)$; and (ii) learn a

binary classifier $f_d(x)$ to discriminate between label shift corrected source $p'_s(x)$ and novel class $p_t(x|y = k+1)$. For PU learning, we employ Conditional Value Ignoring Risk (CVIR) loss proposed in Garg et al. (2021b) which was shown to outperform alternative approaches. We discuss this procedure in detail in App. E.

Finally, to obtain a $(k+1)$-way classifier $f_t(x)$ on target we combine discriminator $f_d$ and source classifier $f_s$ with importance-reweighted label shift correction. Our approach is summarized in Algorithm 1.

## 4. Experiments

Here, we present key experimental findings and present more details in Sec. A. We conduct experiments with seven benchmark classification datasets. For each dataset, we simulate an OSLS problem as described in next paragraph. We compare PULSE with several popular OSDA methods: DANCE (Saito et al., 2020), UAN (You et al., 2019), CMU (Fu et al., 2020), STA (Liu et al., 2019), Backprop-ODA (or BODA) (Saito et al., 2018). For alternative baselines, we experiment with (i) domain discriminator classifier for source versus target; and (ii) $k$ PU classifiers. To evaluate classifiers, we report target accuracy on all classes, seen classes and the novel class. For target marginal, we report absolute difference between true and estimated marginal.

**OSLS Setup** To simulate an OSLS problem, we experiment with different fraction of novel class prevalence, source marginal, and target marginal. We randomly choose classes that constitute the novel target class. Then, we split the training data from each source class randomly into two partitions creating a random label distribution for shared classes among source and target. We then club novel classes to assign them a new class (i.e., $k+1$). Finally, we throw away target labels to obtain an unsupervised DA problem.

Across different datasets, we observe that PULSE consistently outperforms other methods for the target classification and novel prevalence estimation (Table 1). We observe that with default hyperparameters, popular OSDA methods significantly under perform as compared to PULSE. We show that the primary reasons underlying the poor performance of OSDA methods are (i) the heuristics employed to detect novel classes; and (ii) loss functions incorporated to improve alignment between examples from common classes in source and target. In contrast, our two-stage method PULSE, first estimates the fraction of novel class which then guides the classification of novel class versus previously seen classes avoiding the need to guess $\kappa$.

**Ablations** We perform ablations with novel class porportion (App. H.6), contrastive pre-training (App. H.7), alternative PU learning plugin methods (App. H.8) and show that PULSE continues to dominate other methods.

## 5. Theoretical Analysis of PULSE Framework

We analyse key steps of our PULSE procedure for target label marginal estimation (Step 3, 5 Algorithm 1) and learning the domain discriminator classifier (Step 5, Algorithm 1). We relegate formal statements and proofs to App. F.

**Target marginal estimation** Building on BBE results from Garg et al. (2021b), we present finite sample results for target label marginal estimation.

**Theorem 1** (Informal)**.** *Assume that for each class $y \in \mathcal{Y}_s$, there exists a threshold $c_y$ such that for the classifier $f_s$, if $[f_s(x)]_y > c_y$ for any $x$ then the true label for that sample $x$ is $y$. Then, we have $\|\widehat{p}_t - p_t\|_1 \leqslant \mathcal{O}\left(\sqrt{k^3 \log(4k/\delta)/n} + \sqrt{k^2 \log(4k/\delta)/m}\right)$.*

**Analysis of CVIR in population** While the CVIR loss was proposed in Garg et al. (2021b), no analysis was provided for the iterative gradient descent procedure. In our work, we show that in population on a separable Gaussian dataset, CVIR will recover the optimal classifier.

We consider a binary classification problem where we have access to positive distribution (i.e., $p_p$), unlabeled distribution (i.e., $p_u := \alpha p_p + (1-\alpha)p_n$), and mixture coefficient $\alpha$. First we introduce some notation. For a classifier $f$ and loss function $\ell$, define $\mathrm{VIR}_\alpha(f) = \inf\{\tau \in \mathbb{R} : \mathrm{P}_{x\sim p_u}(\ell(x,-1;f) \leqslant \tau) \geqslant 1-\alpha\}$. Intuitively, $\mathrm{VIR}_\alpha(f)$ identifies a threshold $\tau$ to capture bottom $1-\alpha$ fraction of the loss $\ell(x,-1)$ for points $x$ sampled from $p_u$. Additionally, define CVIR loss as $\mathcal{L}(f,w) = \alpha\mathbb{E}_{p_p}[\ell(x,1;f)] + \mathbb{E}_{p_u}[w(x)\ell(x,-1;f)]$ for classifier $f$ and some weights $w(x) \in \{0,1\}$. Recall that given a classifier $f_t$ at an iterate $t$, CVIR procedure proceeds as follows:

$$w_t(x) = \mathbb{I}\left[\ell(x,-1;f_t) \leqslant \mathrm{VIR}_\alpha(f_t)\right] , \tag{1}$$

$$f_{t+1} = f_t - \eta\nabla\mathcal{L}_f(f_t, w_t) . \tag{2}$$

We assume a data generating setup where $x$ are drawn from a multivariate Gaussian, i.e., $x \sim \mathcal{N}(0, I_d)$ and labels are assigned by a random separator $\theta_{\text{opt}}$, i.e., $y = \mathbb{I}\left[\theta_{\text{opt}}^T x > 0\right]$ for some fixed direction $\theta_{\text{opt}} \in \mathbb{R}^d$.

**Theorem 2** (Informal)**.** *In the data setup described above, a linear classifier $f(x;\theta) = \mathbb{I}\left[\theta^T x > 0\right]$ trained with CVIR procedure as in equations (1)-(2) will converge to the optimal classifier in direction, i.e., $\lim_{t\to\infty} \theta_t / \|\theta_t\|_2 \to \theta_{opt}$.*

Additionally, with experiments on toy setup in App. G, we investigate efficacy of CVIR in overparameterized models. We observe that early learning phenomena followed by the self-training procedure crucially helps CVIR.

**Conclusion** In this work, we introduce OSLS a well-posed instantiation of OSDA. We highlight the brittle nature of benchmark driven progress in OSDA and hope that our work can help to stimulate more solid foundations and enable systematic progress in this area.

## References


A. Alexandari, A. Kundaje, and A. Shrikumar. Adapting to label shift with bias-corrected calibration. In *International Conference on Machine Learning (ICML)*, 2021.

M. Z. Alom, C. Yakopcic, M. Nasrin, T. M. Taha, V. K. Asari, et al. Breast cancer classification from histopathological images with inception recurrent residual convolutional neural network. *Journal of digital imaging*, 2019.

S. Arora, S. S. Du, W. Hu, Z. Li, and R. Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations (ICLR)*, 2019.

M. Baktashmotlagh, M. Faraki, T. Drummond, and M. Salzmann. Learning factorized representations for open-set domain adaptation. In *International Conference on Learning Representations (ICLR)*, 2019.

J. Bekker and J. Davis. Estimating the class prior in positive and unlabeled data through decision tree induction. In *Assosication for the Advancement of Artificial Intelligence (AAAI)*, 2018.

J. Bekker and J. Davis. Learning from positive and unlabeled data: a survey. *Machine Learning*, 2020.

S. Ben-David, T. Lu, T. Luu, and D. Pál. Impossibility Theorems for Domain Adaptation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

A. Bendale and T. Boult. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902, 2015.

S. Bucci, M. R. Loghmani, and T. Tommasi. On the effectiveness of image rotation for open set domain adaptation. In *European Conference on Computer Vision*. Springer, 2020.

K. Cao, M. Brbic, and J. Leskovec. Concept learners for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2021.

K. Cao, M. Brbic, and J. Leskovec. Open-world semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

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

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020a.

X. Chen, W. Chen, T. Chen, Y. Yuan, C. Gong, K. Chen, and Z. Wang. Self-pu: Self boosted and calibrated positive-unlabeled training. In *International Conference on Machine Learning*, pages 1510–1519. PMLR, 2020b.

T. M. Consortium et al. A single cell transcriptomic atlas characterizes aging tissues in the mouse. *Nature*, 583 (7817), 2020.

C. Cortes and M. Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519, 2014.

C. Cortes, Y. Mansour, and M. Mohri. Learning Bounds for Importance Weighting. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

M. Du Plessis, G. Niu, and M. Sugiyama. Convex formulation for learning from positive and unlabeled data. In *International conference on machine learning*, pages 1386–1394, 2015.

M. C. Du Plessis and M. Sugiyama. Class prior estimation from positive and unlabeled data. *IEICE TRANSACTIONS on Information and Systems*, 97(5):1358–1362, 2014.

M. C. Du Plessis, G. Niu, and M. Sugiyama. Analysis of learning from positive and unlabeled data. *Advances in neural information processing systems*, 27:703–711, 2014.

C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *International Conference Knowledge Discovery and Data Mining (KDD)*, pages 213–220, 2008.

B. Fu, Z. Cao, M. Long, and J. Wang. Learning to detect open classes for universal domain adaptation. In *European Conference on Computer Vision*. Springer, 2020.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 2016.

S. Garg, Y. Wu, S. Balakrishnan, and Z. Lipton. A unified view of label shift estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

S. Garg, S. Balakrishnan, Z. Kolter, and Z. Lipton. RATT: Leveraging unlabeled data to guarantee generalization. In *International Conference on Machine Learning (ICML)*, 2021a.

S. Garg, Y. Wu, A. Smola, S. Balakrishnan, and Z. Lipton. Mixture proportion estimation and PU learning: A modern approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021b.

S. Garg, S. Balakrishnan, Z. Lipton, B. Neyshabur, and H. Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations (ICLR)*, 2022.

Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. *arXiv preprint arXiv:1705.08500*, 2017.

A. Gretton, A. J. Smola, J. Huang, M. Schmittfull, K. M. Borgwardt, and B. Schölkopf. Covariate Shift by Kernel Mean Matching. *Journal of Machine Learning Research (JMLR)*, 2009.

K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

D. Hendrycks and K. Gimpel. A Baseline for Detecting Misclassified and Out-Of-Distribution Examples in Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2017.

D. Ivanov. DEDPUL: Difference-of-estimated-densities-based positive-unlabeled learning. *arXiv preprint arXiv:1902.06965*, 2019.

S. Jain, M. White, M. W. Trosset, and P. Radivojac. Nonparametric semi-supervised learning of class proportions. *arXiv preprint arXiv:1601.01944*, 2016.

H. Jiang, B. Kim, M. Y. Guan, and M. R. Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5546–5557, 2018.

R. Kiryo, G. Niu, M. C. Du Plessis, and M. Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *Advances in neural information processing systems*, pages 1675–1685, 2017.

P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. A. Earnshaw, I. S. Haque, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021.

A. Krizhevsky and G. Hinton. Learning Multiple Layers of Features from Tiny Images. Technical report, Citeseer, 2009.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

F. Letouzey, F. Denis, and R. Gilleron. Learning from positive and unlabeled examples. In *International Conference on Algorithmic Learning Theory*, pages 71–85. Springer, 2000.

Q. Lian, W. Li, L. Chen, and L. Duan. Known-class aware self-ensemble for open set domain adaptation. *arXiv preprint arXiv:1905.01068*, 2019.

H. Liao. A deep learning approach to universal skin disease classification. *University of Rochester Department of Computer Science, CSC*, 2016.

Z. C. Lipton, Y.-X. Wang, and A. Smola. Detecting and Correcting for Label Shift with Black Box Predictors. In *International Conference on Machine Learning (ICML)*, 2018.

H. Liu, Z. Cao, M. Long, J. Wang, and Q. Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2019.

S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *arXiv preprint arXiv:2007.00151*, 2020.

M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*. PMLR, 2017.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

P. Panareda Busto and J. Gall. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 754–763, 2017.

X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.

H. Ramaswamy, C. Scott, and A. Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *International conference on machine learning*, pages 2052–2060, 2016.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

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

M. Saerens, P. Latinne, and C. Decaestecker. Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure. *Neural Computation*, 2002.

S. Sagawa, P. W. Koh, T. Lee, I. Gao, S. M. Xie, K. Shen, A. Kumar, W. Hu, M. Yasunaga, H. Marklund, S. Beery, E. David, I. Stavness, W. Guo, J. Leskovec, K. Saenko, T. Hashimoto, S. Levine, C. Finn, and P. Liang. Extending the wilds benchmark for unsupervised adaptation. In *NeurIPS Workshop on Distribution Shifts*, 2021.

K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–168, 2018.

K. Saito, D. Kim, S. Sclaroff, and K. Saenko. Universal domain adaptation through self supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

T. Sanderson and C. Scott. Class proportion estimation with application to multiclass anomaly rejection. In *Artificial Intelligence and Statistics (AISTATS)*, pages 850–858, 2014.

S. Santurkar, D. Tsipras, and A. Madry. Breeds: Benchmarks for subpopulation shift. In *International Conference on Learning Representations (ICLR)*, 2021.

W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

C. Scott. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *Artificial Intelligence and Statistics*, pages 838–846, 2015.

K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33, 2020.

D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 2018.

F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte. A dataset for breast cancer histopathological image classification. *Ieee transactions on biomedical engineering*, 63 (7):1455–1462, 2015.

A. Storkey. When Training and Test Sets Are Different: Characterizing Learning Transfer. *Dataset Shift in Machine Learning*, 2009.

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

B. Sun, J. Feng, and K. Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*. Springer, 2017.

S. Tan, J. Jiao, and W.-S. Zheng. Weakly supervised open-set domain adaptation by dual-domain collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5394–5403, 2019.

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

Y. Xu, C. Xu, C. Xu, and D. Tao. Multi-positive and unlabeled learning. In *IJCAI*, pages 3182–3188, 2017.

K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan. Universal domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2720–2729, 2019.

B. Zadrozny. Learning and Evaluating Classifiers Under Sample Selection Bias. In *International Conference on Machine Learning (ICML)*, 2004.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

H. Zhang, A. Li, J. Guo, and Y. Guo. Hybrid models for open set recognition. In *European Conference on Computer Vision*, pages 102–117. Springer, 2020.

J. Zhang, A. Menon, A. Veit, S. Bhojanapalli, S. Kumar, and S. Sra. Coping with label shift via distributionally robust optimisation. In *International Conference on Learning Representations (ICLR)*, 2021.

K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain Adaptation Under Target and Conditional Shift. In *International Conference on Machine Learning (ICML)*, 2013.

W. Zhang, W. Ouyang, W. Li, and D. Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

Y. Zhang, T. Liu, M. Long, and M. Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*. PMLR, 2019.

Table 1: *Comparison of PULSE with other methods.* Across all datasets, PULSE outperforms alternatives for both target classification and novel class prevalence estimation. Acc (All) is target accuracy, Acc (Seen) is target accuracy on examples from previously seen classes, and Acc (Novel) is recall for novel examples. MPE (Novel) is absolute error for novel prevalence estimation. Results reported by averaging across 3 seeds. Detailed results for each dataset with all methods in App. H.4.

| Method | CIFAR-10 | | | | CIFAR-100 | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc (All) | Acc (Seen) | Acc (Novel) | MPE (Novel) | Acc (All) | Acc (Seen) | Acc (Novel) | MPE (Novel) |
| Source-Only | 67.1 | 87.0 | - | - | 46.6 | 66.4 | - | - |
| UAN (You et al., 2019) | 15.4 | 19.7 | 25.2 | 0.214 | 18.1 | 40.6 | 14.8 | 0.133 |
| BODA (Saito et al., 2018) | 63.1 | 66.2 | 42.0 | 0.162 | 36.1 | 17.7 | 81.6 | 0.41 |
| DANCE (Saito et al., 2020) | 70.4 | 85.5 | 14.5 | 0.174 | 47.3 | 66.4 | 1.2 | 0.28 |
| STA (Liu et al., 2019) | 57.9 | 69.6 | 14.9 | 0.124 | 42.6 | 48.5 | 34.8 | 0.14 |
| CMU (Fu et al., 2020) | 62.1 | 77.9 | 41.2 | 0.183 | 35.4 | 46.0 | 15.5 | 0.161 |
| Domain Disc. (Elkan and Noto, 2008) | 47.4 | 87.0 | 30.6 | 0.331 | 45.8 | 66.5 | 39.1 | **0.046** |
| $k$-PU | 83.6 | 79.4 | **98.9** | 0.036 | 36.3 | 22.6 | **99.1** | 0.298 |
| PULSE (Ours) | **86.1** | **91.8** | 88.4 | **0.008** | **63.4** | **67.2** | 63.5 | 0.078 |

| Method | Entity30 | | Newsgroups20 | | Tabula Muris | | BreakHis | | DermNet | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) |
| Source-Only | 32.0 | - | 39.3 | - | 33.8 | - | 70.0 | - | 41.4 | - |
| BODA (Saito et al., 2018) | 42.2 | 0.189 | 43.4 | 0.16 | 76.5 | 0.079 | 71.5 | 0.077 | 43.8 | 0.207 |
| Domain Disc. | 43.2 | 0.135 | 50.9 | 0.176 | 73.0 | 0.071 | 56.5 | 0.091 | 40.6 | 0.083 |
| $k$-PU | 50.7 | 0.394 | 52.1 | 0.373 | 85.9 | 0.307 | 75.6 | **0.059** | 46.0 | 0.313 |
| PULSE (Ours) | **58.0** | **0.054** | **62.2** | **0.061** | **87.8** | **0.058** | **79.1** | **0.054** | **48.9** | **0.043** |

# Supplementary Materials

## A. Summary of Experiments

**Baselines** We compare PULSE with several popular methods from OSDA literature. While these methods are not specifically proposed for OSLS, they are introduced for the more general OSDA problem. In particular, we make comparions with DANCE (Saito et al., 2020), UAN (You et al., 2019), CMU (Fu et al., 2020), STA (Liu et al., 2019), Backprop-ODA (or BODA) (Saito et al., 2018). We use the open source implementation available at `https://github.com/thuml`. For alternative baselines, we experiment with source classifier directly deployed on the target data which may contain novel class and label shift among source classes (referred to as *source-only*). We also train a domain discriminator classifier for source versus target (referred to as *domain disc.*). This is adaptation of PU learning baseline(Elkan and Noto, 2008) which assumes no label shift among source classes. Finally, per the reduction presented in Sec. 2, we train $k$ PU classifiers (referred to as *k-PU*). We include detailed description of each method in App. H.1.

**Datasets** We conduct experiments with seven benchmark classification datasets across vision, natural language, biology and medicine. For each dataset, we simulate an OSLS problem as described in next paragraph. For vision, we use CIFAR10, CIFAR100 (Krizhevsky and Hinton, 2009) and Entity30 (Santurkar et al., 2021). For language, we experiment with Newsgroups-20 (`http://qwone.com/~jason/20Newsgroups/`) dataset. Additionally, inspired by applications of OSLS in biology and medicine, we experiment with Tabula Muris (Consortium et al., 2020) (Gene Ontology prediction), Dermnet (skin disease prediction `https://dermnetnz.org/`), and BreakHis (Spanhol et al., 2015) (tumor cell classification). These datasets span language, image and table modalities. We provide interpretation of OSLS problem for each dataset along with other details in App. H.2.

**OSLS Setup** To simulate an OSLS problem, we experiment with different fraction of novel class prevalence, source label

distribution, and target label distribution. We randomly choose classes that constitute the novel target class. After randomly choosing source and novel classes, we first split the training data from each source class randomly into two partitions. This creates a random label distribution for shared classes among source and target. We then club novel classes to assign them a new class (i.e. $k+1$). Finally, we throw away labels for the target data to obtain an unsupervised DA problem. We repeat the same process on iid hold out data to obtain validation data with no target labels.

**Training and Evaluation** We use Resnet18 (He et al., 2016) for CIFAR10, CIFAR100, and Entity30. For newsgroups, we use a convolutional architecture. For Tabular Muris and MNIST, we use a fully connected MLP. For Dermnet and BreakHis, we use Resnet-50. For all methods, we use the same backbone for discriminator and source classifier. For kPU, we use a separate final layer for each class with the same backbone. We use default hyperparameters for all methods. For OSDA methods, we use default method specific hyperparameters introduced in their works. Since OSDA methods do not estimate the prevalence of novel class explicitly, we use the fraction of examples predicted in class $k+1$ as a surrogate. We train models till the performance on validation source data (labeled) ceases to increase. Unlike OSDA methods, note that we do not use early stopping based on performance on held-out labeled target data. To evaluate classification performance, we report target accuracy on all classes, seen classes and the novel class. For novel class prevalence estimation, we report absolute difference between true and estimated marginal. We open-source our code and by simply changing a single config file, new OSLS setups can be generated and experimented with. We provide precise details about hyperparameters, OSLS setup for each dataset and code in App. H.3.

**Results** Across different datasets, we observe that PULSE consistently outperforms other methods for the target classification and novel prevalence estimation (Table 1). For detection of novel classes (Acc (Novel) column), kPU achieves superior performance as compared to alternative approaches because of its bias to default to $(k+1)^{\text{th}}$ class. This is evident by the sharp decrease in performance on previously seen classes. For each dataset, we plot evolution of performance with training in App. H.4. We observe more stability in performance of PULSE as compared to other methods.

We observe that with default hyperparameters, popular OSDA methods significantly under perform as compared to PULSE. We hypothesize that the primary reasons underlying the poor performance of OSDA methods are (i) the heuristics employed to detect novel classes; and (ii) loss functions incorporated to improve alignment between examples from common classes in source and target. To detect novel classes, a standard heuristic employed popular OSDA methods involves thresholding uncertainty estimates (e.g., prediction entropy, softmax confidence (You et al., 2019; Fu et al., 2020; Saito et al., 2020)) at a predefined threshold $\kappa$. However, a fixed $\kappa$, may not for different datasets and different fractions of the novel class. In App. H.5, we ablate by (i) removing loss function terms incorporated with an aim to improve source target alignment; and (ii) vary threshold $\kappa$ and show improvements in performance of these methods. In contrast, our two-stage method PULSE, first estimates the fraction of novel class which then guides the classification of novel class versus previously seen classes avoiding the need to guess $\kappa$.

**Ablations** Different datasets, in our setup span different fraction of novel class prevalence ranging from $0.22$ (in CIFAR10) to $0.64$ (in Tabula Muris). For each dataset, we perform more ablations on the novel class proportion in App. H.6. For kPU and PULSE, in the main paper, we include results with BBE and CVIR (Garg et al., 2021b). In App. H.8, we perform experiments with alternative PU learning approaches and highlight the superiority of BBE and CVIR over other methods. Moreover, since we have access to unlabeled target data, we experiment with SimCLR (Chen et al., 2020a) pre-training on the mixture of unlabeled source and target dataset. We include setup details and results in App. H.7. While pre-trained backbone architecture improves performance for all methods, PULSE continues to dominate other methods.

## B. Related Work

**(Closed Set) Domain Adaptation (DA)** Under DA, our goal is to adapt a predictor from a source distribution with labeled data to a target distribution from which we observe only unlabeled examples. DA is classically explored under two distribution shift scenarios (Storkey, 2009): (i) Covariate shift (Zhang et al., 2013; Zadrozny, 2004; Cortes et al., 2010; Cortes and Mohri, 2014; Gretton et al., 2009) where $p(y|x)$ remains invariant among source and target; and (ii) Label shift (Saerens et al., 2002; Lipton et al., 2018; Azizzadenesheli et al., 2019; Alexandari et al., 2021; Garg et al., 2020; Zhang et al., 2021) where $p(x|y)$ is shared across source and target. In these settings most theoretical analysis requires that the target distribution's support is a subset of the source support (Ben-David et al., 2010). However, recent empirically work in DA (Long et al., 2015; 2017; Sun and Saenko, 2016; Sun et al., 2017; Zhang et al., 2019; 2018; Ganin et al., 2016; Sohn et al., 2020) focuses on settings motivated by benchmark datasets (e.g., WILDS (Sagawa et al., 2021; Koh et al., 2021), Office-31 (Saenko et al., 2010) OfficeHome (Venkateswara et al., 2017), DomainNet (Peng et al., 2019)) where such overlap assumptions are violated.

Instead, they rely on some intuitive notion of semantic equivalence across domains. These problems are not well-specified and in practice, despite careful hyperparameter tuning, these methods often do not improve over standard empirical risk minimization on source data alone for practical, and importantly, previously unseen datasets (Sagawa et al., 2021).

**Open Set Domain Adaptation (OSDA)** OSDA (Panareda Busto and Gall, 2017; Bendale and Boult, 2015; Scheirer et al., 2013) extends DA to settings where along with distribution shift among previously seen classes, we may observe a novel class in the target data. This setting is also known as *universal domain adaptation* (You et al., 2019; Saito et al., 2020). Rather than making precise assumptions about the nature of shift between source and target, the OSDA literature is primarily governed by semi-synthetic problems on benchmark DA datasets (e.g. DomainNet, Office-31 and OfficeHome). Numerous OSDA methods have been proposed (Baktashmotlagh et al., 2019; Cao et al., 2019; Tan et al., 2019; Lian et al., 2019; You et al., 2019; Saito et al., 2018; 2020; Fu et al., 2020; Bucci et al., 2020). At a high level, most OSDA methods perform two steps: (i) align source and target representation for previously seen classes; and (ii) train a discrimination to reject novel class from previously seen classes. The second step typically uses novelty detection heuristics to identify novel samples.

**PU Learning** Positive and Unlabeled (PU) learning is the base case of OSLS. Here, we observe labeled data a single source class and unlabeled target data contains data from both the novel class and the source class. In PU learning, our goals are: (i) Mixture Proportion Estimation (MPE), i.e., determining the fraction of previously seen class in target ; and (ii) PU classification, i.e., learning to discriminate between the novel and the positive (source) class. Several classical methods were proposed for both MPE (Elkan and Noto, 2008; Du Plessis and Sugiyama, 2014; Scott, 2015; Jain et al., 2016; Bekker and Davis, 2018; 2020) and classification (Elkan and Noto, 2008; Du Plessis et al., 2014; 2015). However, classical MPE methods do not scale to high-dimensional settings (Ramaswamy et al., 2016). More recent methods alleviate these issues by operating in classifier output space (Garg et al., 2021b; Ivanov, 2019). For classification, traditional methods fail when deployed with models classes with high capacity due to their capacity of fitting random labels (Zhang et al., 2017). Recent methods (Garg et al., 2021b; Kiryo et al., 2017; Chen et al., 2020b), avoid over-fitting by employing regularization or self-training techniques.

**Other related work** A separate line of work looks at the problem of Out-Of-Distribution (OOD) detection (Hendrycks and Gimpel, 2017; Geifman and El-Yaniv, 2017; Lakshminarayanan et al., 2016; Jiang et al., 2018; Ovadia et al., 2019; Zhang et al., 2020). Here, the goal is to identify novel examples, i.e., samples that lie out of the support of training distribution. The main different between OOD detection and OSDA is that in OOD detection we do not have access to unlabeled data containing a novel class. Recently, Cao et al. (2022) proposed open-world semi-supervised learning, where the task is to not only identify novel classes in target but also to separate out different novel classes in an unsupervised manner.

Our work takes a step back from the hopelessly general OSDA setup, introducing OSLS, a well-posed OSDA setting where the sought-after parameters can be identified.

## C. Preliminaries

**Domain adaptation under label shift** Under label shift, we observe data from $k$ classes in both source and target where the conditional distribution remain invariant (i.e., $p_s(x|y) = p_t(x|y)$ for all classes $y \in [1, k]$) but the target label marginal may change (i.e., $p_t(y) \neq p_s(y)$). Additionally, for all classes in source have a non-zero support , i.e., for all $y \in [1, k]$, $p_s(y) \geqslant c$, where $c > 0$. Here, given labeled source data and unlabeled target data our tasks are: (i) estimate the shift in label distribution, i.e., $p_t(y)$ for all $y \in [1, k]$; (ii) train a classifier for the target domain $f_t$ to approximate $p_t(y|x)$.

One common approach to label shift involves estimating the importance ratios $p_t(y)/p_s(y)$ by leveraging a blackbox classifier and then employing re-sampling of source data or importance re-weighted ERM on source to obtain a classifier for the target domain (Lipton et al., 2018; Azizzadenesheli et al., 2019; Alexandari et al., 2021).

**PU learning** Under PU learning, we possess labeled source data from a positive class ($p_p$) and unlabeled target data from $p_u = \alpha p_p + (1-\alpha)p_n$ a mixture of positive and negative class ($p_n$). Our goals naturally break down in to two tasks: (i) MPE, determining the fraction of positives $p_p$ in $p_u$ and (ii) PU classification, learning a positive-versus-negative classifier on target.

Note that given access to population of positives and unlabeled, $\alpha$ can be estimated as $\min_x p_u(x)/p_p(x)$. Next, we briefly discuss recent methods for MPE that operate in the classifier output space to avoid curse of dimensionality:

(i) **EN:** Given a domain discriminator classifier $f_d$ trained to discriminate between positive and unlabeled, Elkan and Noto (2008) proposed the following estimator: $\sum_{x_i \in X_p} f_d(x_i) / \sum_{x_i \in X_u} f_d(x_i)$ where $X_p$ is the set of positive examples and $X_u$ is the set of unlabeled examples.

(ii) **DEDPUL:** Given a domain discriminator classifier $f_d$, Ivanov (2019) proposed an estimator that leverages density of the data in the output space of the classifier $f_d$ to directly estimate $\min p_u(f(x))/p_p(f(x))$.

(iii) **BBE:** BBE (Garg et al., 2021b) identifies a threshold on probability scores assigned by the classifier $f_d$ such that by estimating the ratio between the fractions of positive and unlabeled points receiving scores above the threshold, we obtain proportion of positives in unlabeled.

After obtaining an estimate for mixture proportion $\alpha$, following methods can be employed for PU classification:

(i) **Domain Discriminator:** Given positive and unlabeled data, Elkan and Noto (2008) trained a classifier $f_d$ to discriminator between them. To make a prediction on test point from unlabeled data, we can then use Bayes rule to obtain the following transformation on probabilistic output of the domain discriminator: $f = \alpha\left(\frac{m}{n}\right)\left(\frac{f_d(x)}{1-f_d(x)}\right)$, where $n$ and $m$ are the number of positives and unlabeled examples used to train $f_d$ (Elkan and Noto, 2008).

(ii) **uPU:** Du Plessis et al. (2015) proposed an unbiased loss estimator for positive versus negative training. In particular, since $p_u = \alpha p_p + (1-\alpha)p_n$, the loss on negative examples $\mathbb{E}_{p_n}\left[\ell(f(x);-1)\right]$ can be estimated as:

$$\mathbb{E}_{p_n}\left[\ell(f(x);-1)\right] = \frac{1}{1-\alpha}\left[\mathbb{E}_{p_u}\left[\ell(f(x);-1)\right] - \alpha\mathbb{E}_{p_p}\left[\ell(f(x);-1)\right]\right] . \tag{3}$$

Thus, a classifier can be trained with the following uPU loss:

$$\mathcal{L}_{\text{uPU}}(f) = \alpha\mathbb{E}_{p_p}\left[\ell(f(x);+1)\right] + \mathbb{E}_{p_u}\left[\ell(f(x);-1)\right] - \alpha\mathbb{E}_{p_p}\left[\ell(f(x);-1)\right] . \tag{4}$$

(iii) **nnPU:** While unbiased losses exist that estimate the PvN loss given PU data and the mixture proportion $\alpha$, this unbiasedness only holds before the loss is optimized, and becomes ineffective with powerful deep learning models capable of memorization. Kiryo et al. (2017) proposed the following non-negative regularization for unbiased PU learning:

$$\mathcal{L}_{\text{nnPU}}(f) = \alpha\mathbb{E}_{p_p}\left[\ell(f(x);+1)\right] + \max\left\{\mathbb{E}_{p_u}\left[\ell(f(x);-1)\right] - \alpha\mathbb{E}_{p_p}\left[\ell(f(x);-1)\right], 0\right\} . \tag{5}$$

(iv) **CVIR:** Garg et al. (2021b) proposed CVIR objective, which discards the highest loss $\alpha$ fraction of unlabeled examples on each training epoch, removing the incentive to overfit to the unlabeled positive examples. CVIR loss is defined as

$$\mathcal{L}_{\text{CVIR}}(f) = \alpha\mathbb{E}_{p_p}\left[\ell(x,1;f)\right] + \mathbb{E}_{p_u}\left[w(x)\ell(x,-1;f)\right] , \tag{6}$$

where weights $w(x) = \mathbb{I}\left[\ell(x,-1;f) \leqslant \text{VIR}_\alpha(f)\right]$ for $\text{VIR}_\alpha(f)$ defined as $\text{VIR}_\alpha(f) = \inf\{\tau \in \mathbb{R} : \mathrm{P}_{x\sim p_u}(\ell(x,-1;f) \leqslant \tau) \geqslant 1-\alpha\}$. Intuitively, $\text{VIR}_\alpha(f)$ identifies a threshold $\tau$ to capture bottom $1-\alpha$ fraction of the loss $\ell(x,-1)$ for points $x$ sampled from $p_u$.

### C.1. Reduction of OSLS into $k$ PU problems

Under the strong positivity condition, the OSLS problem can be broken down into $k$ PU problems as follows: By treating a given source class $y_j \in \mathcal{Y}_s$ as *positive* and grouping all other classes together as *negative* we observe that the unlabeled target data is then a mixture of data from the positive and negative classes. This yields a PU learning problem and the corresponding mixture proportion gives the fraction $\alpha_j$ of class $y_j$ among the target data. By iterating this process for all source classes, we can solve for the entire target label marginal $p_t(y)$. Thus, OSLS reduces to $k$ instances of PU learning problem. Formally, note that $p_t(x)$ can be written as:

$$p_t(x) = \underbrace{p_s(y=j)}_{\alpha}\underbrace{p_s(x|y=j)}_{p_p} + (1-p_s(y=j))\underbrace{\left(\sum_{i\in\mathcal{Y}\setminus\{j\}}\frac{p_s(y=i)}{1-p_s(y=j)}p_s(x|y=i)\right)}_{p_n} , \tag{7}$$

individually for all $j \in \mathcal{Y}_s$. By repeating this reduction for all classes, we obtain $k$ separate PU learning problems. Hence, a natural choice is to leverage this structure and solve $k$ PU problems to solve the original OSLS problem.

In particular, for each class $j \in \mathcal{Y}_s$, we can first estimate its prevalence $\widehat{\alpha}_j$ in the unlabeled target. Then the target marginal for the novel class is given by $\widehat{\alpha}_{k+1} = 1 - \sum_{i=1}^{k} \widehat{\alpha}_i$. For classification, we can train $k$ PU learning classifiers $f_i$, where $f_i$ is trained to classify a source class $i$ versus others in target. Assuming that each $f_j$ returns a score between $[0, 1]$, during test time, an example $x$ is classified as $f(x)$ given by

$$f(x) = \begin{cases} \arg\max_{j \in \mathcal{Y}_s} f_j(x) & \text{if } \max_{j \in \mathcal{Y}_s} f_j(x) \geqslant 0.5 \\ k+1 & \text{o.w.} \end{cases} \tag{8}$$

That is, if each classifier classifies the example as belonging to other in unlabeled, then we classify the example as belonging to the class $k+1$. This reduction has been mentioned in past work (Sanderson and Scott, 2014; Xu et al., 2017). However, to the best of our knowledge, no previous work has empirically investigated both classification and target label marginal estimation jointly. Sanderson and Scott (2014) focuses only on target marginal estimation for tabular datasets and Xu et al. (2017) assumes that the target marginal is known and only trains $k$ separate PU classifiers.

In our work, we perform the first large scale experiments to evaluate efficacy of the reduction of the OSLS problem to $k$-PU problems. In our main experiments, to estimate $\alpha_j$ and to train $f_j$ classifiers for all $j \in \mathcal{Y}_s$, we use BBE and CVIR as described before which was shown to outperform alternative approaches in Garg et al. (2021b). We ablate with other methods in App. H.8. With plugin state-of-the-art PU learning algorithms (i.e. CVIR and BBE), we observe that this naive reduction doesn't scale to datasets with large number of classes because of error accumulation in each of the $k$ MPEs and $k$ one-versus-other PU classifiers. To mitigate the error accumulation problem, we propose the PULSE framework in the next section.

## D. Proofs for identifiability of OSLS

We now introduce conditions for OSLS, under which the solution is identifiable. For ease, we re-state Proposition 1 and Proposition 2.

**Proposition 1** (Necessary conditions)**.** *Assume $p_t(y) > 0$ for all $y \in \mathcal{Y}_t$. Then $p_t(y)$ is identified only if $p_t(x|y = k+1)$ and $p_s(x|y)$ for all $y \in \mathcal{Y}_s$ satisfy weak positivity, i.e., there must exists a subdomain $X_{wp} \subset X$ such that: (i) $p_t(X_{wp}|y = k+1) = 0$; and (ii) the matrix $[p_s(x|y)]_{x \in X_{wp}, y \in \mathcal{Y}_s}$ is full column-rank.*

*Proof.* We prove this by contradiction. Assume that there exists a unique solution $p_t(y)$. We will obtain contradiction when both (i) and (ii) don't hold.

First, assume for no subset $X_{\text{wp}} \subseteq \mathcal{X}$, we have $[p_s(x|y)]_{x \in X_{\text{wp}}, y \in \mathcal{Y}_s}$ as full-rank. Then in that case, we have vectors $[p_s(x|y = j)]_{x \in \mathcal{X}}$ as linearly dependent for $j \in \mathcal{Y}_s$, i.e., there exists $[\alpha_j]_{j \in \mathcal{Y}_s} \in \mathbb{R}^k$ such that $\sum_j \alpha_j p_s(x|y = j) = 0$ for all $x \in \mathcal{X}$. Thus for small enough $\epsilon > 0$, we have infinite solutions of the form $[p_t(y = j) - \epsilon \cdot a_j]_{j \in \mathcal{Y}_s}$.

Hence, there exists $X_{\text{wp}} \subseteq \mathcal{X}$ for which we have $[p_s(x|y)]_{x \in X_{\text{wp}}, y \in \mathcal{Y}_s}$ as full-rank. Without loss of generality, we assume that $|X_{\text{wp}}| = k$. Assume that $p_t(X_{\text{wp}}|y = k+1) > 0$, i.e., $[p_t(x|y = k+1)]_{x \in X_{\text{wp}}}$ has $l < k$ zero entries. We will now construct another solution for the label marginal $p_t$. For simplicity we denote $A = [p_s(x|y)]_{x \in X_{\text{wp}}, y \in \mathcal{Y}_s}$. Consider the vector $v(\gamma) = [p_t(x) - (p_t(y = k+1) - \gamma) p_t(x|y = k+1)]_{x \in X_{\text{wp}}}$ for some $\gamma > 0$. Intuitively, when $\gamma = 0$, we have $u = A^{-1} v(0)$ where $u = [p_t(y)]_{y \in \mathcal{Y}_s}$, i.e., we recover the true label marginal corresponding to source classes.

However, since the solution is not at vertex, there exists a small enough $\gamma > 0$ such that $u' = A^{-1} v(\gamma)$ with $\sum_j u'_j \leqslant 1$ and $u'_j \geqslant 0$. Since A is full-rank and $v(\gamma) \neq v(0)$, we have $u' \neq u$. Thus we construct a separate solution with $u'$ as $[p_t(y)]_{y \in \mathcal{Y}_s}$ and $p_t(x) - \sum_{j \in \mathcal{Y}_s} u'_j p_s(x|y = j)$ as $p_t(x|y = k+1)$. Hence, when there exists $X_{\text{wp}} \subseteq \mathcal{X}$ for which we have $[p_s(x|y)]_{x \in X_{\text{wp}}, y \in \mathcal{Y}_s}$ as full-rank, for uniqueness we obtain a contradiction on the assumption $p_t(X_{\text{wp}}|y = k+1) > 0$. □

We now make some comments on the assumption $p_t(y) > 0$ for all $y \in \mathcal{Y}_t$ in Proposition 1. Since, $p_t(y)$ needs to satisfy simplex constraints, if the solution is at a vertex of simplex, then OSLS problem may not require weak positivity. For example, there exists contrived scenarios where $p_s(x|y = j) = p_s(x|y = k)$ for all $j, k \in \mathcal{Y}_s$ and $p_t(x|y = k+1) \neq p_s(x|y = j)$ for all $j \in \mathcal{Y}_s$. Then when $p_t(x) = p_t(x|y = k+1)$, we can uniquely identify the OSLS solution even when weak positivity assumption is not satisfied.

**Proposition 2** (Sufficient conditions)**.** *The target marginal $p_t(y)$ is identified if for all $y \in \mathcal{Y} \backslash \{k+1\}$, $p_t(x|y = k+1)$ and $p_s(x|y)$ satisfy either: (i) Strong positivity, i.e., there exists $X_{sp} \subset \mathcal{X}$ such that $p_t(X_{sp}|y = k+1) = 0$ and the*

*matrix* $[p_s(x|y)]_{x\in X_{sp}, y\in\mathcal{Y}_s}$ *is full-rank and diagonal; or (ii) Separability, i.e., there exists* $X_{sep} \subset \mathcal{X}$*, such that* $p_t(X_{sep}|y = k+1) = 0$*,* $p_s(X_{sep}) = 1$*, and the matrix* $[p_s(x|y)]_{x\in X_{sep}, y\in\mathcal{Y}_s}$ *is full column-rank.*

*Proof.* For each condition, we will prove identifiability by constructing the unique solution.

Under strong positivity, for all $j \in \mathcal{Y}_s$ there exists $x \in X_{\text{sp}}$ such that $p_t(x|y = k) = 0$ for all $k \in \mathcal{Y}_t \backslash \{j\}$. Set $\alpha_j = \min_{x\in\mathcal{X}, p_s(x|y=j)>0} \frac{p_t(x)}{p_s(x|y=j)}$ , for all $j \in \mathcal{Y}_s$. For $x \in X_{\text{sp}}$ such that $p_t(x|y = k) = 0$ for all $k \in \mathcal{Y}_t \backslash \{j\}$, we get $\frac{p_t(x)}{p_s(x|y=j)} = p_t(y = j)$ and for all $x' \neq x$, we have $\frac{p_t(x)}{p_s(x|y=j)} \geqslant p_t(y = j)$. Thus, we get $\alpha_j = p_t(y = j)$. Finally, we get $\alpha_{k+1} = 1 - \sum_{j\in\mathcal{Y}_s} \alpha_j$. Plugging in values of the label marginal, we can obtain $p_t(x|y = k+1)$ as $p_t(x) - \sum_{y\in\mathcal{Y}_s} p_t(y = j)p_s(x|y = j)$.

Under separability, we can obtain the label marginal $p_t$ for source classes by simply considering the set $X_{\text{sep}}$. Denote $A = [p(x|y)]_{x\in X_{\text{sep}}, y\in\mathcal{Y}_s}$ and $v = [p_t(x)]_{x\in X_{\text{sep}}}$. Then, since $A$ is full column-rank by assumption, we can define $u = (A^T A)^{-1} A^T v$. For all $x \in X_{\text{sep}}$, we have $p_t(x) = \sum_{y\in\mathcal{Y}_s} p_t(y)p_s(x|y)$ and hence, $u = [p_t(y)]_{y\in\mathcal{Y}_s}$. Having obtained $[p_t(y)]_{y\in\mathcal{Y}_s}$, we recover $p_t(y = k+1) = 1 - \sum_{j\in\mathcal{Y}_s} p_t(y = j)$ and $p_t(x|y = k+1) = p_t(x) - \sum_{j\in\mathcal{Y}_s} p_t(y = j)p_s(x|y = j)$. □

While the two conditions in Proposition 2 overlap, they cover independent set of OSLS problems. Informally, strong positivity extends weak positivity by making an additional assumption that the matrix formed by $p(x|y)$ on inputs in $X_{\text{wp}}$ is diagonal and the separability assumption extends the weak positivity condition to the full input domain of source classes instead of just $X_{\text{wp}}$. Both of these conditions identify a support region of $\mathcal{X}$ which purely belongs to source classes where we can either individually estimate the proportion of each source classes (i.e., under strong positivity) or jointly estimate the proportion (i.e., under separability).

**Extension to continuous inputs** To extend our identifiability conditions for continuous distributions, the linear independence conditions on the matrix $[p_s(x|y)]_{x\in X_{\text{sep}}, y\in\mathcal{Y}_s}$ has the undesirable property of being sensitive to changes on sets of measure zero. In particular, by changing a collection of linearly dependent distributions on a set of measure zero, we can make them linearly independent. Hence, we may introduce stronger notions of linear independence as in Lemma 1 of Garg et al. (2020). We discuss this in App. D.2.

### D.1. Examples illustrating importance of weak positivity condition

In this section, we present two examples, one, to show that weak positivity isn't sufficient for identifiability. Second, we present another example where we show that conditions in Proposition 2 are not necessary for identifiability.

**Example 1** Assume $\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5\}$ and $\mathcal{Y}_t = \{1, 2, 3\}$. Suppose the $p_t(x|y = 1)$, $p_t(x|y = 2)$, and $p_t(x)$ are given as:

| | $p_t(x\|y=1)$ | $p_t(x\|y=2)$ | $p_t(x)$ |
|---|---|---|---|
| $x_1$ | 0.4 | 0.56 | 0.356 |
| $x_2$ | 0.3 | 0.3 | 0.207 |
| $x_3$ | 0.2 | 0.1 | 0.09 |
| $x_4$ | 0.1 | 0.04 | 0.042 |
| $x_5$ | 0.0 | 0.0 | 0.305 |

Here, there exists two separate $p_t(x|y = 3)$ and $p_t(y)$ that are consistent with the given $p_t(x|y = 1)$, $p_t(x|y = 2)$, and $p_t(x)$ and both the solutions satisfy weak positivity for two different $X_{\text{wp}}$ and $X'_{\text{wp}}$.

In particular, notice that $p_t(x|y = 3) = [0.17, 0.0675, 0.0, 0.0, 0.7625]^T$ and $p_t(y) = [0.3, 0.3, 0.4]$ gives us the first solution. $p_t(x|y = 3) = [0.0, 0.0, 0.0645, 0.0096, 0.9839]^T$ and $p_t(y) = [0.19, 0.5, 0.31]$ gives us another solution. For solution 1, $X_{\text{wp}} = \{x_3, x_4\}$ and for solution 2, $X'_{\text{wp}} = \{x_1, x_2\}$. To check consistency of each solution notice that $\sum_{i\in\mathcal{Y}} p_t(y = i)p_t(x|y = i) = p_t(x)$ for each $x \in \mathcal{X}$. □

In the above example, the key is to show that absent knowledge of which $x$'s constitute the set $X_{\text{wp}}$, we might be able to obtain multiple different solutions, each with different $X_{\text{wp}}$ and both $p_t(y)$, $p_t(x|y = k+1)$ satisfying the given information and simplex constraints.

Next, we will show that in certain scenarios weak positivity is enough for identifiability.

**Example 2** Assume $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$ and $\mathcal{Y}_t = \{1, 2, 3\}$. Suppose the $p_t(x|y = 1)$, $p_t(x|y = 2)$, and $p_t(x)$ are given as,

| | $p_t(x\|y = 1)$ | $p_t(x\|y = 2)$ | $p_t(x)$ |
|---|---|---|---|
| $x_1$ | 0.5 | 0.2 | 0.24 |
| $x_2$ | 0.3 | 0.4 | 0.2 |
| $x_3$ | 0.1 | 0.35 | 0.35 |
| $x_4$ | 0.1 | 0.05 | 0.21 |

Here, out of all ${}^4C_2$ possibilities for $X_{\text{wp}}$, only one possibility yields a solution that satisfies weak positivity and simplex constraints. In particular, the solution is given by $p_t(x|y = 3) = [0.0, 0.0, 0.6, 0.4]^T$ and $p_t(y) = [0.4, 0.2, 0.4]$ with $X_{\text{wp}} = \{x_1, x_2\}$. □

In this example, we show that conditions in Proposition 2 are not necessary to ensure identifiability. For discrete domains, this example also highlights that we can check identifiability in exponential time for any OSLS problem given $p_t(x)$ and $p_s(x|y)$ for all $y \in \mathcal{Y}_s$.

### D.2. Extending identifiability conditions to continuous distributions

To extend our identifiability conditions for continuous distributions, the linear independence conditions on the matrix $[p_s(x|y)]_{x \in X_{\text{sep}}, y \in \mathcal{Y}_s}$ has the undesirable property of being sensitive to changes on sets of measure zero. In particular, by changing a collection of linearly dependent distributions on a set of measure zero, we can make them linearly independent. As a consequence, we may impose a *stronger* notion of independence, i.e., the set of distributions $\{p(x|y) : y = 1, ..., k\}$ are such that there does not exist $v \neq 0$ for which $\int_X |\sum_y p(x|y) v_y| dx = 0$ , where $X = X_{\text{wp}}$ for necessary condition and $X = X_{\text{sp}}$ for sufficiency. We refer this condition as *strict linear independence*.

## E. PULSE Framework

We begin with presenting our framework for OSLS problem under strong positivity condition. First, we explain the structure of OSLS that we leverage in PULSE framework and then elaborate design decisions we make to exploit the identified structure.

Rather than simply dividing each OSLS instance into $k$ PU problems, we exploit the joint structure of the problem. To begin, we note that if only we could apply a *label shift correction* to source, i.e., re-sample source classes according to their relative proportion in the target data, then we could subsequently consider the unlabeled target data as a mixture of (i) the (reweighted) source distribution; and (ii) the novel class distribution (i.e., $p_t(x|y = k + 1)$). Formally, we have

$$\begin{aligned} p_t(x) = \sum_{j \in \mathcal{Y}_t} p_t(y = j) p_t(x|y = j) &= \sum_{j \in \mathcal{Y}_s} \frac{p_t(y = j)}{p_s(y = j)} p_s(x, y = j) + p_t(x|y = k + 1) p_t(y = k + 1) \\ &= (1 - p_t(x|y = k + 1)) p'_s(x) + p_t(x|y = k + 1) p_t(y = k + 1)\,, \end{aligned} \tag{9}$$

where $p'_s(x)$ is the label-shift-corrected source distribution, i.e., $p'_s(x) = \sum_{j \in \mathcal{Y}_s} w(j) p_s(x, y = j)$, where $w(j) = (p_t(y=j) / \sum_k p_t(y=k)) / p_s(y = j)$ for all $j \in \mathcal{Y}_s$. Intuitively, $p'_t(j) = p_t(y=j) / \sum_k p_t(y=k)$ is re-normalized label distribution in target among source classes and $w(j)$'s are the importance weights. Hence, after applying a label shift correction to the source distribution $p'_s(x)$, we have reduced the OSLS problem to a *single* PU learning problem, where $p'_s(x)$ plays the part of the positive distribution and $p_t(x|y = k + 1)$ acts as negative distribution with mixture coefficients $1 - p_t(y = k + 1)$ and $p_t(y = k + 1)$ respectively. We now discuss our methods to estimate the importance ratios $w(y)$ and to tackle the PU learning instance obtained from OSLS.

While traditional methods for estimating label shift breakdown in high dimensional settings (Zhang et al., 2013), recent methods exploit black-box classifiers to avoid the curse of dimensionality (Lipton et al., 2018; Azizzadenesheli et al., 2019; Alexandari et al., 2021). However, these recent techniques require overlapping label distributions, and a direct application would require demarcation of samples from $p'_s(x)$ in target, creating a cyclic dependency. Instead, we build on recently

proposed Best Bin Estimation (BBE) technique from Garg et al. (2021b). Given positive and unlabeled data, BBE estimates the fraction of positives in unlabeled by leveraging a black-box classifier. In particular, BBE identifies a threshold on probability scores assigned by the classifier such that by estimating the ratio between the fractions of positive and unlabeled points receiving scores above the threshold, we obtain proportion of positives in unlabeled. We tailor BBE to estimate the relative fraction of previously seen classes in the target distribution by exploiting a $k$-way source classifier $f_s$ trained on labeled source data. We describe the procedure in App. E (Algorithm 2). Compared to the $k$ PU reduction, we do not train $k$ one-versus-other PU classifiers and neither we use estimates of fraction previously seen classes to directly estimate the novel class prevalence.

After estimating the fraction of source classes in target (i.e., $p'_t(j) = {p_t(y=j)}/{\sum_{k\in\mathcal{Y}_s} p_t(y=k)}$ for all $j \in \mathcal{Y}_s$), we re-sample the source data according to $p'_t(y)$ to mimic samples from distribution $p'_s(x)$. Thus, obtaining a PU learning problem instance, we resort to PU learning techniques to (i) estimate the fraction of novel class $p_t(y = k + 1)$; and (ii) learn a binary classifier $f_d(x)$ to discriminate between label shift corrected source $p'_s(x)$ and novel class $p_t(x|y = k + 1)$. Assume that sigmoid output $f_d(x)$ indicates predicted probability of an example $x$ belonging to label shift corrected source $p'_s(x)$. With traditional methods for PU learning involving domain discrimination, over-parameterized models can memorize the positive instances in unlabeled, assigning them confidently to the negative class, which can severely hurt generalization on PN data (Kiryo et al., 2017; Garg et al., 2021b). Rather, we employ Conditional Value Ignoring Risk (CVIR) loss proposed in Garg et al. (2021b) which was shown to outperform alternative approaches. Given an estimate of the fraction of novel class $\widehat{p}_t(y = k + 1)$, CVIR objective discards the highest loss $(1 - \widehat{p}_t(y = k + 1))$ fraction of examples on each training epoch, removing the incentive to overfit to the examples from $p'_s(x)$. In particular, we employ the iterative procedure that alternates between estimating the prevalence of novel class $\widehat{p}_t(y = k + 1)$ (with BBE) and minimizing the CVIR loss with estimated fraction of novel class. We discuss this procedure in detail in App. E.

Finally, to obtain a $(k + 1)$-way classifier $f_t(x)$ on target we combine discriminator $f_d$ and source classifier $f_s$ with importance-reweighted label shift correction. In particular, for all $j \in \mathcal{Y}_s$, $[f_t(x)]_j = (f_d(x))\frac{w(j)\cdot[f_s(x)]_j}{\sum_{k\in\mathcal{Y}_s} w(k)\cdot[f_s(x)]_k}$ and $[f_t(x)]_{k+1} = 1 - f_d(x)$. Overall, our approach proceeds as follows (Algorithm 1): First, we estimate the label shift among previously seen classes. Then we employ importance re-weighting of source data to formulate a single PU learning problem between source and target to estimate fraction of novel class $\widehat{p}_t(y = k+1)$ and to learn a discriminator $f_d$ for the novel class. Combining discriminator and label shift corrected source classifier we get $(k + 1)$-way target classifier. We theoretically analyse crucial steps in PULSE in Sec. 5. Next, we elaborate on Step 3 and 5 in Algorithm 1.

**Extending BBE algorithm to estimate target marginal among previously seen classes** In a PU learning problem, given positive and unlabeled data, BBE estimates the fraction of positives in unlabeled by leveraging a black-box classifier. In particular, BBE identifies a threshold on probability scores assigned by the classifier such that by estimating the ratio between the fractions of positive and unlabeled points receiving scores above the threshold, we obtain proportion of positives in unlabeled. We tailor BBE to estimate the relative fraction of previously seen classes in the target distribution by exploiting a $k$-way source classifier $f_s$ trained on labeled source data. We describe the procedure in Algorithm 2.

For given probability density function $p$ and a scalar output function $f$, define a function $q(z) = \int_{A_z} p(x)dx$, where $A_z = \{x \in \mathcal{X} : f(x) \geqslant z\}$ for all $z \in [0, 1]$. Intuitively, $q(z)$ captures the cumulative density of points in a top bin, the proportion of input domain that is assigned a value larger than $z$ by the function $f$ in the transformed space. We define an empirical estimator $\widehat{q}(z)$ given a set $X = \{x_1, x_2, \ldots, x_n\}$ sampled iid from $p(x)$. Let $Z = f(X)$. Define $\widehat{q}(z) = \sum_{i=1}^{n} \mathbb{I}\left[z_i \geqslant z\right]/n$.

Our procedure proceeds as follows. Given a held-out dataset of source $\{\mathbf{X}_2^S, \mathbf{y}_2^S\}$ and unlabeled target samples $\mathbf{X}_2^T$, we push all examples through the source classifier $f$ to obtain $k$ dimensional outputs. For all $j \in \mathcal{Y}_s$, we repeat the following: Obtain $Z_p = f_j(\mathbf{X}_2^S[\mathrm{id}_j])$ and $Z_u = f_j(\mathbf{X}_2^T)$. Next, with $Z_p$ and $Z_u$, we estimate $\widehat{q}_s$ and $\widehat{q}_t$. Finally, we estimate $[\widehat{p}_t]_j$ as the ratio $\widehat{q}_t(\widehat{c})/\widehat{q}_s(\widehat{c})$ at $\widehat{c}$ that minimizes the upper confidence bound at a pre-specified level $\delta$ and a fixed parameter $\gamma \in (0, 1)$. Our method is summarized in Algorithm 2. Throughout all the experiments, we fix $\delta$ at 0.1 and $\gamma$ at 0.01.

**Extending CVIR to train discriminator $f_d$ and estimate novel class prevalence** After estimating the fraction of source classes in target (i.e., $p'_t(j) = {p_t(y=j)}/{\sum_{k\in\mathcal{Y}_s} p_t(y=k)}$ for all $j \in \mathcal{Y}_s$), we re-sample the source data according to $p'_t(y)$ to mimic samples from distribution $p'_s(x)$. Thus, obtaining a PU learning problem instance, we resort to PU learning techniques to (i) estimate the fraction of novel class $p_t(y = k + 1)$; and (ii) learn a binary classifier $f_d(x)$ to discriminate between label shift corrected source $p'_s(x)$ and novel class $p_t(x|y = k + 1)$. Assume that sigmoid output $f_d(x)$ indicates predicted probability of an example $x$ belonging to label shift corrected source $p'_s(x)$.

**Algorithm 2** Extending Best Bin Estimation (BBE) for Step 3 in Algorithm 1

**input** : Validation source $\{\mathbf{X}_2^S, \mathbf{y}_2^S\}$ and unlabeled target samples $\mathbf{X}_2^T$. Source classifier $f : \mathcal{X} \to \Delta^{k-1}$. Hyperparameter $0 < \delta, \gamma < 1$.

1: $\widehat{p}_t \leftarrow \text{zeros}(size = |\mathcal{Y}_s|)$
2: **for** $j \in \mathcal{Y}_s$ **do**
3: $\quad \text{id}_j \leftarrow \text{where}(\mathbf{y}_2^S = j)$.
4: $\quad Z_s, Z_t \leftarrow \left[f(\mathbf{X}_2^S[\text{id}_j])\right]_j, \left[f(\mathbf{X}_2^T)\right]_j$.
5: $\quad \widehat{q}_s(z), \widehat{q}_t(z) \leftarrow \frac{\sum_{z_i \in Z_s} \mathbb{I}[z_i \geqslant z]}{|\text{id}_j|}, \frac{\sum_{z_i \in Z_t} \mathbb{I}[z_i \geqslant z]}{|\mathbf{X}_2^T|}$ for all $z \in [0, 1]$.
6: $\quad \widehat{c}_j \leftarrow \arg\min_{c \in [0,1]} \left( \frac{\widehat{q}_t(c)}{\widehat{q}_s(c)} + \frac{1+\gamma}{\widehat{q}_s(c)} \left( \sqrt{\frac{\log(4/\delta)}{2|\mathbf{X}_2^T|}} + \sqrt{\frac{\log(4/\delta)}{2|\text{id}_j|}} \right) \right)$.
7: $\quad [\widehat{p}_t]_j \leftarrow \frac{\widehat{q}_t(\widehat{c}_j)}{\widehat{q}_s(\widehat{c}_j)}$.
8: **end for**

**output** : Normalized target marginal among source classes $\widehat{p}'_t \leftarrow \frac{\widehat{p}_t}{\|\widehat{p}_t\|_1}$

Given an estimate of the fraction of novel class $\widehat{p}_t(y = k+1)$, CVIR objective creates a provisional set of novel examples $\mathbf{X}_1^N$ by removing $(1 - \widehat{p}_t(y = k+1))$ fraction of examples from $\mathbf{X}_1^T$ that incur highest loss when predicted as novel class on each training epoch. Next, we update our discriminator $f_d$ by minimizing loss on label shift corrected source $\widetilde{\mathbf{X}}_1^S$ and provisional novel examples $\mathbf{X}_1^N$. This step is aimed to remove any incentive to overfit to the examples from $p'_s(x)$. Consequently, we employ the iterative procedure that alternates between estimating the prevalence of novel class $\widehat{p}_t(y = k+1)$ (with BBE) and minimizing the CVIR loss with estimated fraction of novel class. Algorithm 3 summarizes our approach which is used in Step 3 of Algorithm 1.

Note that we need to warm start with simple domain discrimination training, since in the initial stages mixture proportion estimate is often close to 1 rejecting all the unlabeled examples. In Garg et al. (2021b), it was shown that the procedure is not sensitive to the choice of number of warm start epochs and in a few cases with large datasets, we can even get away without warm start (i.e., $W = 0$) without hurting the performance. In our work, we notice that given an estimate $\widehat{\alpha}$ of prevalence of novel class, we can use unbiased PU error (4) on validation data as a surrogate to identify warm start epochs for domain discriminator training. In particular, we train the domain discriminator classifier for a large number of epochs, say $E(>> W)$, and then choose the discriminator, i.e., warm start epoch $W$ at which $f_d$ achieves minimum unbiased validation loss.

Finally, to obtain a $(k+1)$-way classifier $f_t(x)$ on target we combine discriminator $f_d$ and source classifier $f_s$ with importance-reweighted label shift correction. In particular, for all $j \in \mathcal{Y}_s$, $[f_t(x)]_j = (f_d(x)) \frac{w(j) \cdot [f_s(x)]_j}{\sum_{k \in \mathcal{Y}_s} w(k) \cdot [f_s(x)]_k}$ and $[f_t(x)]_{k+1} = 1 - f_d(x)$. Similarly, to obtain target marginal $p_t$, we re-scale the label shift estimate among previously seen classes with estimate of prevalence of novel examples, i.e., for all $j \in \mathcal{Y}_s$, assign $\widehat{p}_t(y = j) = (1 - \widehat{p}_t(y = k+1)) \cdot \widehat{p}'_t(y = j)$.

Overall, our approach proceeds as follows (Algorithm 1): First, we estimate the label shift among previously seen classes. Then we employ importance re-weighting of source data to formulate a single PU learning problem between source and target to estimate fraction of novel class $\widehat{p}_t(y = k+1)$ and to learn a discriminator $f_d$ for the novel class. Combining discriminator and label shift corrected source classifier we get $(k+1)$-way target classifier.

### E.1. PULSE under separability

Our ideas for PULSE framework can be extended to separability condition since (9) continues to hold. In particular, when OSLS satisfies the separability assumption, we may hope to jointly estimate the label shift among previously seen classes with label shift estimation techniques (Lipton et al., 2018; Alexandari et al., 2021) and learn a domain discriminator classifier. This may be achieved by estimating label shift among examples rejected by domain discriminator classifier as belonging to previously seen classes. However, in our initial experiments, we observe that techniques proposed under strong positivity were empirically stable and outperform methods developed under separability. This is intuitive for many benchmark datasets where it may be more natural to expect that for each class there exists a subdomain that only belongs to that class than assuming separability only between novel class samples and examples from source classes.

**Algorithm 3** Alternating between CVIR and BBE for Step 5 in Algorithm 1

**input** : Re-sampled training source data $\widetilde{\mathbf{X}}_1^S$, validation source data $\widetilde{\mathbf{X}}_2^S$. Training target data $\mathbf{X}_1^T$ and validation data $\mathbf{X}_2^T$. Hyperparameter $W, B, \delta, \gamma$.

1: Initialize a training model $f_\theta$ and an stochastic optimization algorithm $\mathcal{A}$.
2: $\mathbf{X}_1^N \leftarrow \mathbf{X}_1^T$.
{// Warm start with domain discrimination training}
3: **for** $i \leftarrow 1$ to $W$ **do**
4: Shuffle $(\widetilde{\mathbf{X}}_1^S, \mathbf{X}_1^N)$ into $B$ mini-batches. With $(\widetilde{\mathbf{X}}_1^S[i], \mathbf{X}_1^N[i])$ we denote $i^{\text{th}}$ mini-batch.
5: **for** $i \leftarrow 1$ to $B$ **do**
6: Set the gradient $\nabla_\theta \left[\widehat{\mathcal{L}}^+(f_\theta; \widetilde{\mathbf{X}}_1^S[i]) + \widehat{\mathcal{L}}^-(f_\theta; \mathbf{X}_1^N[i])\right]$ and update $\theta$ with algorithm $\mathcal{A}$.
7: **end for**
8: **end for**
9: $\widehat{\alpha} \leftarrow$ BBE$(\widetilde{\mathbf{X}}_2^S, \mathbf{X}_2^T, f_\theta)$ {Algorithm 4}
10: Rank samples $x \in \mathbf{X}_1^T$ according to their loss values $\ell(f_\theta(x), -1)$.
11: $\mathbf{X}_1^N \leftarrow \{\mathbf{X}_1^T\}_{1-\widehat{\alpha}}$ where $\{\mathbf{X}_1^T\}_{1-\widehat{\alpha}}$ denote the lowest ranked $1-\widehat{\alpha}$ fraction of samples.
12: **while** training error $\widehat{\mathcal{E}}^+(f_\theta; \widetilde{\mathbf{X}}_2^S) + \widehat{\mathcal{E}}^-(f_\theta; \mathbf{X}_1^N)$ is not converged **do**
13: Train model $f_\theta$ for one epoch on $(\widetilde{\mathbf{X}}_1^S, \mathbf{X}_1^N)$ as in Lines 4-7.
14: $\widehat{\alpha} \leftarrow$ BBE$(\widetilde{\mathbf{X}}_2^S, \mathbf{X}_2^T, f_\theta)$ {Algorithm 4}
15: Rank samples $x \in \mathbf{X}_1^T$ according to their loss values $\ell(f_\theta(x), -1)$.
16: $\mathbf{X}_1^N \leftarrow \{\mathbf{X}_1^T\}_{1-\widehat{\alpha}}$ where $\{\mathbf{X}_1^T\}_{1-\widehat{\alpha}}$ denote the lowest ranked $1-\widehat{\alpha}$ fraction of samples.
17: **end while**

**output** : Trained discriminator $f_d \leftarrow f_\theta$ and novel class fraction $\widehat{p}_t(y = k+1) \leftarrow 1 - \widehat{\alpha}$.

**Algorithm 4** Best Bin Estimation (BBE)

**input** : Re-sampled source data $\widetilde{\mathbf{X}}^S$ and target samples $\mathbf{X}^T$. Discriminator classifier $\widehat{f} : \mathcal{X} \to [0,1]$. Hyperparameter $0 < \delta, \gamma < 1$.

1: $Z_s, Z_t \leftarrow f(\widetilde{\mathbf{X}}^S), f(\mathbf{X}^T)$.
2: $\widehat{q}_t(z), \widehat{q}_s(z) \leftarrow \frac{\sum_{z_i \in Z_s} \mathbb{I}[z_i \geqslant z]}{|\widetilde{\mathbf{X}}^S|}, \frac{\sum_{z_i \in Z_t} \mathbb{I}[z_i \geqslant z]}{|\mathbf{X}|^T}$ for all $z \in [0,1]$.
3: Estimate $\widehat{c} \leftarrow \arg\min_{c \in [0,1]} \left( \frac{\widehat{q}_t(c)}{\widehat{q}_s(c)} + \frac{1+\gamma}{\widehat{q}_s(c)} \left( \sqrt{\frac{\log(4/\delta)}{2|\widetilde{\mathbf{X}}^S|}} + \sqrt{\frac{\log(4/\delta)}{2|\mathbf{X}^T|}} \right) \right)$.

**output** : $\widehat{\alpha} \leftarrow \frac{\widehat{q}_t(\widehat{c})}{\widehat{q}_s(\widehat{c})}$

## F. Proofs for analysis of OSLS framework

In this section, we provide missing formal statements and proofs for theorems in Sec. 5. This mainly includes analysing key steps of our PULSE procedure for target label marginal estimation (Step 3, 5 Algorithm 1) and learning the domain discriminator classifier (Step 5, Algorithm 1).

### F.1. Formal statement and proof of Theorem 1

When the data satisfies strong positivity, we observe that source classifiers often exhibit a threshold $c_y$ on softmax output of each class $y \in \mathcal{Y}_s$ above which the *top bin* (i.e., $[c_y, 1]$) contains mostly examples from that class $y$. We give empirical evidence to this claim in App. F.1. Then, we show that the existence of (nearly) pure top bin for each class in $f_s$ is sufficient for Step 3 in Algorithm 1 to produce (nearly) consistent estimates.

Before introducing the formal statement, we introduce some additional notation. Given probability density function $p$ and a source classifier $f : \mathcal{X} \to \Delta^{k-1}$, define a function $q(z,j) = \int_{A(z,j)} p(x)dx$, where $A(z,j) = \{x \in \mathcal{X} : [f(x)]_j \geqslant z\}$ for all $z \in [0,1]$. Intuitively, $q(z,j)$ captures the cumulative density of points in a top bin for class $j$, i.e., the proportion of input domain that is assigned a value larger than $z$ by the function $f$ at the index $j$ in the transformed space. We define an empirical estimator $\widehat{q}(z,j)$ given a set $X = \{x_1, x_2, \ldots, x_n\}$ sampled iid from $p(x)$. Let $Z = [f(X)]_j$. Define $\widehat{q}(z,j) = \sum_{i=1}^n \mathbb{I}[z_i \geqslant z]/n$.

For each pdf $p_s$ and $p_t$, we define $q_s$ and $q_t$ respectively. Moreover, for each class $j \in \mathcal{Y}_s$, we define $q_{t,j}$ corresponding to $p_{t,j} := p_t(x|y=j)$ and $q_{t,-j}$ corresponding to $p_{t,-j} := \frac{\sum_{i \in \mathcal{Y}_t \setminus \{j\}} p_t(y=i) p_t(x|y=i)}{\sum_{i \in \mathcal{Y}_t \setminus \{j\}} p_t(y=j)}$. Assume that we have $n$ source examples and $m$ target examples. Now building on BBE results from Garg et al. (2021b), we present finite sample results for target label marginal estimation:

**Theorem 3** (Formal statement of Theorem 1)**.** *Define* $c_j^* = \arg\min_{c \in [0,1]} \left(q_{t,-j}(c,j)/q_{t,j}(c,j)\right)$*, for all* $j \in \mathcal{Y}_s$*. Assume* $\min(n,m) \geqslant \max_{j \in \mathcal{Y}_s} \left( \frac{2\log(4k/\delta)}{q_{t,j}^2(c_j^*,j)} \right)$*. Then, for every* $\delta > 0$*,* $\widehat{p}_t$ *(in Algorithm 2 with* $\delta$ *as* $\delta/k$*) satisfies with probability at least* $1-\delta$*, we have:*

$$\|\widehat{p}_t - p_t\|_1 \leqslant \sum_{j \in \mathcal{Y}_s} (1 - p_t(y=j)) \left( \frac{q_{t,-j}(c_j^*,j)}{q_{t,j}(c_j^*,j)} \right) + \mathcal{O}\left( \sqrt{\frac{k^3 \log(4k/\delta)}{n}} + \sqrt{\frac{k^2 \log(4k/\delta)}{m}} \right).$$

When the data satisfies strong positivity, we observe that source classifiers often exhibit a threshold $c_y$ on softmax output of each class $y \in \mathcal{Y}_s$ above which the *top bin* (i.e., $[c_y, 1]$) contains mostly examples from that class $y$. Formally, as long as there exist a threshold $c_j^* \in (0,1)$ such that $q_{t,j}(c_j^*) \geqslant \epsilon$ and $q_{t,-j}(c_j^*) = 0$ for some constant $\epsilon > 0$ for all $j \in \mathcal{Y}_s$, we show that our estimator $\widehat{\alpha}$ converges to the true $\alpha$ with convergence rate $\min(n,m)^{-1/2}$. The proof technique simply builds on the proof of Theorem 1 in Garg et al. (2021b). First, we state Lemma 1 from Garg et al. (2021b). Next, for completeness we provide the proof for Theorem 3 which extends proof of Theorem 1 (Garg et al., 2021b) for $k$ classes.

**Lemma 1.** *Assume two distributions* $q_p$ *and* $q_u$ *with their empirical estimators denoted by* $\widehat{q}_p$ *and* $\widehat{q}_u$ *respectively. Then for every* $\delta > 0$*, with probability at least* $1-\delta$*, we have for all* $c \in [0,1]$

$$\left| \frac{\widehat{q}_u(c)}{\widehat{q}_p(c)} - \frac{q_u(c)}{q_p(c)} \right| \leqslant \frac{1}{\widehat{q}_p(c)} \left( \sqrt{\frac{\log(4/\delta)}{2n_u}} + \frac{q_u(c)}{q_p(c)} \sqrt{\frac{\log(4/\delta)}{2n_p}} \right).$$

*Proof of Theorem 3.* The main idea of the proof is to use the confidence bound derived in Lemma 1 at $\widehat{c}$ and use the fact that $\widehat{c}$ minimizes the upper confidence bound. The proof is split into two parts. First, we derive a lower bound on $\widehat{q}_{t,j}(\widehat{c}_j)$ for all $j \in \mathcal{Y}_s$ and next, we use the obtained lower bound to derive confidence bound on $\widehat{p}_t(y=j)$. With $\widehat{\alpha}_j$, we denote $\widehat{p}_t(y=j)$ for all $j \in \mathcal{Y}_s$. All the statements in the proof simultaneously hold with probability $1-\delta/k$. We derive the bounds for a single $j \in \mathcal{Y}_s$ and then use union bound to combine bound for all $j \in \mathcal{Y}_s$. When it is clearly from context, we denote $q_{t,j}(c,j)$ with $q_{t,j}(c)$ and $q_t(c,j)$ with $q_t(c)$. Recall,

$$\widehat{c}_j := \arg\min_{c \in [0,1]} \frac{\widehat{q}_t(c)}{\widehat{q}_{t,j}(c)} + \frac{1}{\widehat{q}_{t,j}(c)} \left( \sqrt{\frac{\log(4k/\delta)}{2m}} + (1+\gamma) \sqrt{\frac{\log(4k/\delta)}{2np_s(y=j)}} \right) \quad \text{and} \tag{10}$$

$$\widehat{p}_t(y=j) := \frac{\widehat{q}_t(\widehat{c}_j)}{\widehat{q}_{t,j}(\widehat{c}_j)}. \tag{11}$$

Moreover,

$$c_j^* := \arg\min_{c \in [0,1]} \frac{q_t(c)}{q_{t,j}(c)} \quad \text{and} \quad \alpha_j^* := \frac{q_t(c_j^*)}{q_{t,j}(c_j^*)}. \tag{12}$$

**Part 1:** We establish lower bound on $\widehat{q}_{t,j}(\widehat{c}_j)$. Consider $c_j' \in [0,1]$ such that $\widehat{q}_{t,j}(c_j') = \frac{\gamma}{2+\gamma} \widehat{q}_{t,j}(c_j^*)$. We will now show that Algorithm 2 will select $\widehat{c}_j < c_j'$. For any $c \in [0,1]$, we have with with probability $1-\delta/k$,

$$\widehat{q}_{t,j}(c) - \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \leqslant q_{t,j}(c) \quad \text{and} \quad q_t(c) - \sqrt{\frac{\log(4k/\delta)}{2m}} \leqslant \widehat{q}_t(c). \tag{13}$$

Since $\frac{q_t(c_j^*)}{q_{t,j}(c_j^*)} \leqslant \frac{q_t(c)}{q_{t,j}(c)}$, we have

$$\widehat{q}_t(c) \geqslant q_{t,j}(c) \frac{q_t(c_j^*)}{q_{t,j}(c_j^*)} - \sqrt{\frac{\log(4k/\delta)}{2m}} \geqslant \left( \widehat{q}_{t,j}(c) - \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \right) \frac{q_t(c_j^*)}{q_{t,j}(c_j^*)} - \sqrt{\frac{\log(4k/\delta)}{2m}}. \tag{14}$$

Therefore, at $c$ we have

$$\frac{\widehat{q}_t(c)}{\widehat{q}_{t,j}(c)} \geqslant \alpha_j^* - \frac{1}{\widehat{q}_{t,j}(c)} \left( \sqrt{\frac{\log(4k/\delta)}{2m}} + \frac{q_t(c_j^*)}{q_p(c_j^*)} \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \right) . \tag{15}$$

Using Lemma 1 at $c^*$, we have

$$\frac{\widehat{q}_t(c)}{\widehat{q}_{t,j}(c)} \geqslant \frac{\widehat{q}_t(c_j^*)}{\widehat{q}_{t,j}(c_j^*)} - \left( \frac{1}{\widehat{q}_{t,j}(c_j^*)} + \frac{1}{\widehat{q}_{t,j}(c)} \right) \left( \sqrt{\frac{\log(4k/\delta)}{2m}} + \frac{q_t(c_j^*)}{q_{t,j}(c_j^*)} \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \right) \tag{16}$$

$$\geqslant \frac{\widehat{q}_t(c_j^*)}{\widehat{q}_{t,j}(c_j^*)} - \left( \frac{1}{\widehat{q}_{t,j}(c_j^*)} + \frac{1}{\widehat{q}_{t,j}(c)} \right) \left( \sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \right) , \tag{17}$$

where the last inequality follows from the fact that $\alpha_j^* = \frac{q_t(c_j^*)}{q_{t,j}(c_j^*)} \leqslant 1$. Furthermore, the upper confidence bound at $c$ is lower bound as follows:

$$\frac{\widehat{q}_t(c)}{\widehat{q}_{t,j}(c)} + \frac{1+\gamma}{\widehat{q}_{t,j}(c)} \left( \sqrt{\frac{\log(4l/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \right) \tag{18}$$

$$\geqslant \frac{\widehat{q}_t(c_j^*)}{\widehat{q}_{t,j}(c_j^*)} + \left( \frac{1+\gamma}{\widehat{q}_{t,j}(c)} - \frac{1}{\widehat{q}_{t,j}(c_j^*)} - \frac{1}{\widehat{q}_{t,j}(c)} \right) \left( \sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \right) \tag{19}$$

$$= \frac{\widehat{q}_t(c_j^*)}{\widehat{q}_{t,j}(c_j^*)} + \left( \frac{\gamma}{\widehat{q}_{t,j}(c)} - \frac{1}{\widehat{q}_{t,j}(c_j^*)} \right) \left( \sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \right) \tag{20}$$

Using (20) at $c = c'$, we have the following lower bound on ucb at $c'$:

$$\frac{\widehat{q}_t(c')}{\widehat{q}_{t,j}(c')} + \frac{1+\gamma}{\widehat{q}_{t,j}(c')} \left( \sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \right) \tag{21}$$

$$\geqslant \frac{\widehat{q}_t(c_j^*)}{\widehat{q}_{t,j}(c_j^*)} + \frac{1+\gamma}{\widehat{q}_{t,j}(c_j^*)} \left( \sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \right) , \tag{22}$$

Moreover from (20), we also have that the lower bound on ucb at $c \geqslant c'$ is strictly greater than the lower bound on ucb at $c'$. Using definition of $\widehat{c}$, we have

$$\frac{\widehat{q}_t(c_j^*)}{\widehat{q}_{t,j}(c_j^*)} + \frac{1+\gamma}{\widehat{q}_{t,j}(c_j^*)} \left( \sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \right) \tag{23}$$

$$\geqslant \frac{\widehat{q}_t(\widehat{c})}{\widehat{q}_{t,j}(\widehat{c})} + \frac{1+\gamma}{\widehat{q}_{t,j}(\widehat{c})} \left( \sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n \cdot p_s(y=j)}} \right) , \tag{24}$$

and hence

$$\widehat{c} \leqslant c' . \tag{25}$$

**Part 2:** We now establish an upper and lower bound on $\widehat{\alpha}_j$. We start with upper confidence bound on $\widehat{\alpha}_j$. By definition of $\widehat{c}_j$, we have

$$\frac{\widehat{q}_t(\widehat{c})}{\widehat{q}_{t,j}(\widehat{c})} + \frac{1+\gamma}{\widehat{q}_{t,j}(\widehat{c})}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right) \tag{26}$$

$$\leqslant \min_{c\in[0,1]}\left[\frac{\widehat{q}_t(c)}{\widehat{q}_{t,j}(c)} + \frac{1+\gamma}{\widehat{q}_{t,j}(c)}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right)\right] \tag{27}$$

$$\leqslant \frac{\widehat{q}_t(c_j^*)}{\widehat{q}_{t,j}(c_j^*)} + \frac{1+\gamma}{\widehat{q}_{t,j}(c_j^*)}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right). \tag{28}$$

Using Lemma 1 at $c_j^*$, we get

$$\frac{\widehat{q}_t(c_j^*)}{\widehat{q}_{t,j}(c_j^*)} \leqslant \frac{q_t(c_j^*)}{q_{t,j}(c_j^*)} + \frac{1}{\widehat{q}_{t,j}(c_j^*)}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + \frac{q_t(c_j^*)}{q_{t,j}(c_j^*)}\sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right)$$

$$= \alpha_j^* + \frac{1}{\widehat{q}_{t,j}(c_j^*)}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + \alpha_j^*\sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right). \tag{29}$$

Combining (28) and (29), we get

$$\widehat{\alpha}_j = \frac{\widehat{q}_t(\widehat{c})}{\widehat{q}_{t,j}(\widehat{c})} \leqslant \alpha_j^* + \frac{2+\gamma}{\widehat{q}_{t,j}(c_j^*)}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right). \tag{30}$$

Using DKW inequality on $\widehat{q}_{t,j}(c_j^*)$, we have $\widehat{q}_{t,j}(c_j^*) \geqslant q_{t,j}(c_j^*) - \sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}$. Assuming $n\cdot p_s(y=j) \geqslant \frac{2\log(4k/\delta)}{q_{t,j}^2(c_j^*)}$, we get $\widehat{q}_{t,j}(c_j^*) \leqslant q_{t,j}(c_j^*)/2$ and hence,

$$\widehat{\alpha}_j \leqslant \alpha_j^* + \frac{4+2\gamma}{q_{t,j}(c_j^*)}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right). \tag{31}$$

Finally, we now derive a lower bound on $\widehat{\alpha}_j$. From Lemma 1, we have the following inequality at $\widehat{c}$

$$\frac{q_t(\widehat{c})}{q_{t,j}(\widehat{c})} \leqslant \frac{\widehat{q}_t(\widehat{c})}{\widehat{q}_{t,j}(\widehat{c})} + \frac{1}{\widehat{q}_{t,j}(\widehat{c})}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + \frac{q_t(\widehat{c})}{q_{t,j}(\widehat{c})}\sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right). \tag{32}$$

Since $\alpha_j^* \leqslant \frac{q_t(\widehat{c})}{q_{t,j}(\widehat{c})}$, we have

$$\alpha_j^* \leqslant \frac{q_t(\widehat{c})}{q_{t,j}(\widehat{c})} \leqslant \frac{\widehat{q}_t(\widehat{c})}{\widehat{q}_{t,j}(\widehat{c})} + \frac{1}{\widehat{q}_{t,j}(\widehat{c})}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + \frac{q_t(\widehat{c})}{q_{t,j}(\widehat{c})}\sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right). \tag{33}$$

Using (31), we obtain a very loose upper bound on $\frac{\widehat{q}_t(\widehat{c})}{\widehat{q}_{t,j}(\widehat{c})}$. Assuming $\min(n\cdot p_s(y=j), m) \geqslant \frac{2\log(4k/\delta)}{q_{t,j}^2(c_j^*)}$, we have $\frac{\widehat{q}_t(\widehat{c})}{\widehat{q}_{t,j}(\widehat{c})} \leqslant \alpha_j^* + 4 + 2\gamma \leqslant 5 + 2\gamma$. Using this in (33), we have

$$\alpha_j^* \leqslant \frac{\widehat{q}_t(\widehat{c})}{\widehat{q}_{t,j}(\widehat{c})} + \frac{1}{\widehat{q}_{t,j}(\widehat{c})}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + (5+2\gamma)\sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right). \tag{34}$$

Moreover, as $\widehat{c} \geqslant c'$, we have $\widehat{q}_{t,j}(\widehat{c}) \geqslant \frac{\gamma}{2+\gamma}\widehat{q}_{t,j}(c_j^*)$ and hence,

$$\alpha_j^* - \frac{\gamma+2}{\gamma\widehat{q}_{t,j}(c_j^*)}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + (5+2\gamma)\sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right) \leqslant \frac{\widehat{q}_t(\widehat{c})}{\widehat{q}_{t,j}(\widehat{c})} = \widehat{\alpha}_j\,. \tag{35}$$

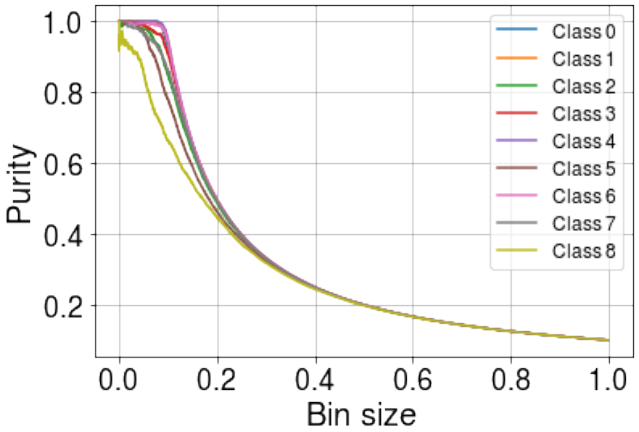



Figure 2: Purity and size (in terms of fraction of unlabeled samples) in the top bin for all classes. Bin size refers to the fraction of examples in the top bin. With purity, we refer to the fraction of examples from a specific class $j$ in the top bin. Results with ResNet-18 on CIFAR10 OSLS setup. Details of the setup in App. H.2. As the bin size increases for all classes the purity decreases.

As we assume $n \cdot p_s(y=j) \geqslant \frac{2\log(4k/\delta)}{q_{t,j}^2(c_j^*)}$, we have $\widehat{q}_{t,j}(c_j^*) \leqslant q_{t,j}(c_j^*)/2$, which implies the following lower bound on $\alpha$:

$$\alpha_j^* - \frac{2\gamma+4}{\gamma q_{t,j}(c_j^*)}\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + (5+2\gamma)\sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right) \leqslant \widehat{\alpha}_j\,. \tag{36}$$

Combining lower bound (36) and upper bound (31), we get

$$\left|\widehat{\alpha}_j - \alpha_j^*\right| \leqslant l_j\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{\log(4k/\delta)}{2n\cdot p_s(y=j)}}\right)\,, \tag{37}$$

for some constant $l_j$. Additionally by our assumption of OSLS problem $p_s(y=j) > c/k$ for some constant $c > 0$, we have

$$\left|\widehat{\alpha}_j - \alpha_j^*\right| \leqslant l'_j\left(\sqrt{\frac{\log(4k/\delta)}{2m}} + \sqrt{\frac{k\log(4k/\delta)}{2n}}\right)\,, \tag{38}$$

for some constant $l'_j$.

Combining the above obtained bound for all $j \in \mathcal{Y}_s$ with union bound, we get with probability at least $1-\delta$,

$$\sum_{j\in\mathcal{Y}_s}\left|\widehat{\alpha}_j - \alpha_j^*\right| \leqslant l'_{\max}\left(\sqrt{\frac{k^2\log(4k/\delta)}{2m}} + \sqrt{\frac{k^3\log(4k/\delta)}{2n}}\right)\,, \tag{39}$$

where $l'_{\max} = \max l'_j$. Now, note that for each $j \in \mathcal{Y}_s$, we have $q_t(c) = p_t(y=j)\cdot q_{t,j}(c) + (1-p_t(y=j))\cdot q_{t,-j}(c)$. Hence $\alpha_j^* = p_t(y=j) + (1-p_t(y=j))\cdot q_{t,-j}(c)/\cdot q_{t,j}(c)$. Plugging this in, we get the desired bound. □

**Empirical evidence of the top bin property** We now empirically validate the positive pure top bin property (Fig. 2). We include results with Resnet-18 trained on the CIFAR10 OSLS setup same as our main experiments. We observe that source classifier approximately satisfies the positive pure top bin property for small enough top bin sizes.

### F.2. Formal statement and proof of Theorem 2

In this section, we show that in population on a separable Gaussian dataset, CVIR will recover the optimal classifier. Note that here we consider a binary classification problem similar to the one in Step 5 in Algorithm 1. Since we are primarily interested in analysing the iterative procedure for obtaining domain discriminator classifier, we assume that $\alpha$ is known.

In population, we have access to positive distribution (i.e., $p_p$), unlabeled distribution (i.e., $p_u := \alpha p_p + (1-\alpha)p_n$), and mixture coefficient $\alpha$. Our goal is to recover the classifier that discriminates $p_p$ versus $p_n$. Making a parallel connection to Step 5 of PULSE, positive distribution $p_p$ here refers to the label shift corrected source distribution $p'_s$ and $p_u$ refers to $p_t = p_t(y=k+1)p_t(x|y=k+1) + (1-p_t(y=k+1))p'_s(x)$. Our goal is to recover the classifier that discriminates $p_p$ versus $p_n$ (parallel $p'_s$ versus $p_t(\cdot|y=k+1)$).

For ease, we re-introduce some notation. For a classifier $f$ and loss function $\ell$, define

$$\text{VIR}_\alpha(f) = \inf\{\tau \in \mathbb{R} : \text{P}_{x\sim p_u}(\ell(x,-1;f) \leqslant \tau) \geqslant 1-\alpha\}\,. \tag{40}$$

Intuitively, $\text{VIR}_\alpha(f)$ identifies a threshold $\tau$ to capture bottom $1-\alpha$ fraction of the loss $\ell(x,-1)$ for points $x$ sampled from $p_u$. Additionally, define CVIR loss as

$$\mathcal{L}(f,w) = \alpha\mathbb{E}_{p_p}\left[\ell(x,1;f)\right] + \mathbb{E}_{p_u}\left[w(x)\ell(x,-1;f)\right]\,, \tag{41}$$

for classifier $f$ and some weights $w(x) \in \{0,1\}$. Recall that given a classifier $f_t$ at an iterate $t$, CVIR procedure proceeds as follows:

$$w_t(x) = \mathbb{I}\left[\ell(x,-1;f_t) \leqslant \text{VIR}_\alpha(f_t)\right]\,, \tag{42}$$

$$f_{t+1} = f_t - \eta\nabla\mathcal{L}_f(f_t,w_t)\,. \tag{43}$$

We assume a data generating setup with where the support of positive and negative data is completely disjoint. We assume that $x$ are drawn from two half multivariate Gaussian with mean zero and identity covariance, i.e.,

$$x \sim p_p \Leftrightarrow x = \gamma_0\theta_{\text{opt}} + z|\,\theta_{\text{opt}}^T z \geqslant 0,\ \text{where } z \sim \mathcal{N}(0,I_d)$$

$$x \sim p_n \Leftrightarrow x = -\gamma_0\theta_{\text{opt}} + z|\,\theta_{\text{opt}}^T z < 0,\ \text{where } z \sim \mathcal{N}(0,I_d)$$

Here $\gamma_0$ is the margin and $\theta_{\text{opt}} \in \mathbb{R}^d$ is the true separator. Here, we have access to distribution $p_p$ and $p_u = \alpha p_p + (1-\alpha)p_n$. Assume $\ell$ as the logistic loss. For simplicity, we will denote $\mathcal{L}(f_{\theta_t},w_t)$ with $\mathcal{L}(\theta_t,w_t)$.

**Theorem 4** (Formal statement of Theorem 2)**.** *In the data setup described above with $\gamma_0 = 0$, a linear classifier $f(x;\theta) = \mathbb{I}\left[\theta^T x > 0\right]$ initialized at some $\theta_0$ such that $\mathcal{L}(\theta_0,w_0) < \log(2)$, trained with CVIR procedure as in equations (42)-(43) will converge to the optimal classifier in direction, i.e., $\lim_{t\to\infty}\theta_t/\left\|\theta_t\right\|_2 \to \theta_{opt}$.*

*Proof of Theorem 4.* The proof uses two key ideas. One, at convergence of the CVIR procedure, the gradient of CVIR loss in (41) converges to zero. Second, for any classifier $\theta$ not in the direction of $\theta_{\text{opt}}$, we show that the CVIR gradient in (41) is non-zero.

**Part 1** We first show that the loss function $\mathcal{L}(\theta,w)$ in (41) is 2-smooth with respect to $\theta$ for fixed $w$. Using gradient descent lemma with the decreasing property of loss in (42)-(43), we show that gradient converges to zero eventually. Considering gradient of $\mathcal{L}$, we have

$$\nabla_\theta\mathcal{L}(\theta,w) = \alpha\mathbb{E}_{p_p}\left[(f(x;\theta)-1)x\right] + \mathbb{E}_{p_u}\left[w(x)(f(x;\theta)-0)x\right]\,. \tag{44}$$

Moreover, $\nabla^2\mathcal{L}$ is given by

$$\nabla^2_\theta\mathcal{L}(\theta,w) = \alpha\mathbb{E}_{p_p}\left[\nabla f(x;\theta)xx^T\right] + \mathbb{E}_{p_u}\left[w(x)\nabla f(x;\theta)xx^T\right]\,. \tag{45}$$

Since $\nabla f(x;\theta) \leqslant 1$, we have $v^T\nabla^2\mathcal{L}v \leqslant 2$ for all unit vector $v \in R^d$. Now, by gradient descent lemma if $\eta \leqslant 1/2$, at any step $t$ we have, $\mathcal{L}(\theta_{t+1},w_t) \leqslant \mathcal{L}(\theta_t,w_t)$. Moreover, by definition of $\text{VIR}_\alpha(\theta)$ in (40) and update (42), we have $\mathcal{L}(\theta_{t+1},w_{t+1}) \leqslant \mathcal{L}(\theta_{t+1},w_t)$. Hence, we have $\mathcal{L}(\theta_{t+1},w_{t+1}) \leqslant \mathcal{L}(\theta_t,w_t)$. Since, the loss is lower bounded from below at 0, for every $\epsilon > 0$, we have for large enough $t$ (depending on $\epsilon$), $\left\|\nabla_\theta\mathcal{L}(\theta_t,w_t)\right\|_2 \leqslant \epsilon$, i.e., $\left\|\nabla_\theta\mathcal{L}(\theta_t,w_t)\right\|_2 \to 0$ as $t\to\infty$.

**Part 2** Consider a general scenario when $\gamma > 0$. Denote the input domain of $p_p$ and $p_n$ as $P$ and $N$ respectively. At any step $t$, for all points $x \in \mathcal{X}$ such that $p_u(x) > 0$ and $w_t(x) = 0$, we say that $x$ is rejected from $p_u$. We denote the incorrectly rejected subdomain of $p_n$ from $p_u$ as $N_r$ and the incorrectly accepted subdomain of $p_p$ from $p_u$ as $P_a$. Formally, $N_r = \{x : p_n(x) > 0 \text{ and } w_t(x) = 0\}$ and $P_a = \{x : p_p(x) > 0 \text{ and } w_t(x) = 1\}$. We will show that $p_p(P_a) \to 0$ as $t\to\infty$, and hence, we will recover the optimal classifier where we reject none of $p_u$ incorrectly.

Observe that at any time $t$, for fixed $w_t$ and $\theta = \theta_t$, the gradient of CVIR loss in (41), can be expressed as:

$$\nabla_\theta \mathcal{L}(\theta, w_t) = \alpha \underbrace{\int_{x \in P \setminus P_a} (f(x;\theta) - 1) x \cdot p_p(x) dx}_{\text{I}} + (1-\alpha) \underbrace{\int_{x \in N \setminus N_r} (f(x;\theta) - 0) x \cdot p_n(x) dx}_{\text{II}}$$
$$+ \alpha \underbrace{\int_{x \in P_a} (2f(x;\theta) - 1) x \cdot p_p(x) dx}_{\text{III}} . \tag{46}$$

Note that for any $x, \theta$, $0 \leqslant f(x;\theta) \leqslant 1$. Now consider inner product of individual terms above with $\theta_{\text{opt}}$, we get

$$\langle \text{I}, \theta_{\text{opt}} \rangle = \int_{x \in P \setminus P_a} (f(x;\theta) - 1) x^T \theta_{\text{opt}} \cdot p_p(x) dx \leqslant -\gamma_0 \int_{x \in P \setminus P_a} (1 - f(x;\theta)) \cdot p_p(x) dx \,, \tag{47}$$

$$\langle \text{II}, \theta_{\text{opt}} \rangle = \int_{x \in N \setminus N_r} (f(x;\theta) - 0) x^T \theta_{\text{opt}} \cdot p_n(x) dx \leqslant -\gamma_0 \int_{x \in N \setminus N_r} (f(x;\theta) - 0) \cdot p_n(x) dx \,, \tag{48}$$

$$\langle \text{III}, \theta_{\text{opt}} \rangle = \int_{x \in P_a} (2f(x;\theta) - 1) x^T \theta_{\text{opt}} \cdot p_p(x) dx \leqslant -\gamma_0 \int_{x \in P_a} (1 - 2f(x;\theta)) \cdot p_p(x) dx \,. \tag{49}$$

Note that in this problem $\text{VIR}_\alpha(\theta) \leqslant 0.5$. Then equation (49) follows by the fact that $f(x;\theta) \leqslant 0.5$ for all $x \in P_a$ since all points in $P_a$ are assigned $w(x) = 1$.

From part 1, for gradient $\|\nabla_\theta \mathcal{L}(\theta_t, w_t)\|_2$ to converge to zero as $t \to \infty$, we must have that LHS in equations (47), (48), and (49) converges to zero individually. Since CVIR loss decreases continuously and $\mathcal{L}(\theta_0, w_0) < \log(2)$, we have that $f(x;\theta_t) \to 1$ for all $x \in P \setminus P_a$, $f(x;\theta_t) \to 0$ for all $x \in N \setminus N_r$, and $p_p(P_a) \to 0$. Now as $\gamma \to 0$, we have $f(x;\theta_t) \to 1$ for all $x \in P$, $f(x;\theta_t) \to 0$ for all $x \in N$ implies that $\frac{\theta_t}{\|\theta_t\|_2} \to \theta_{\text{opt}}$. □

The above analysis can be extended to any margin $\gamma > 0$ by using arguments from Soudry et al. (2018). In particular, as $p_p(P_a) \to 0$, we will have $\theta_t / \|\theta_t\|_2$ converging to the max-margin classifier for $p_p$ versus $p_n$, i.e., $\theta_{\text{opt}}$. Note that we need an assumption that the initialized model $\theta_0$ is strictly better than a model that randomly guesses or initialized at all zeros. This is to avoid convergence to the local minima of $\theta = \mathbf{0}$ with CVIR training. This assumption is satisfied when the classifier is initialized in a way such that $\langle \theta_0, \theta_{\text{opt}} \rangle > 0$. In general, we need a weaker assumption that during training with any randomly initialized classifier, there exists an iterate $t$ during CVIR training such that $\langle \theta_t, \theta_{\text{opt}} \rangle > 0$.

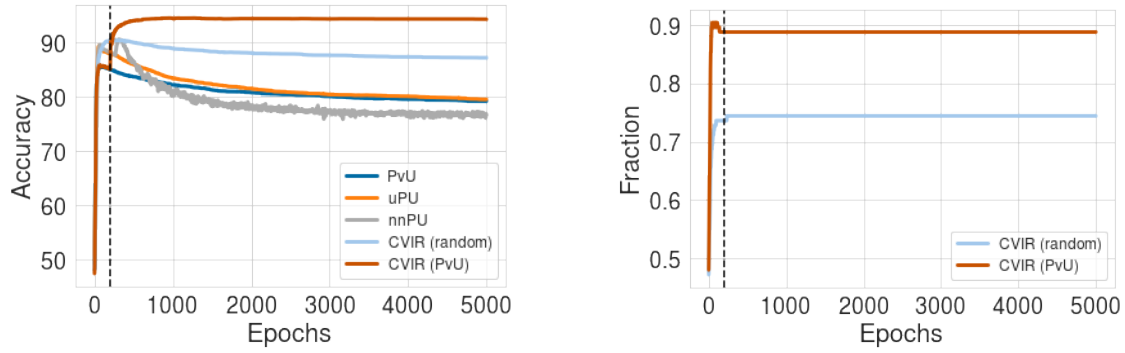
### F.3. Extension of Theorem 1

We also extend the analysis in the proof of Theorem 3 to Step 5 of Algorithm 1 to show convergence of estimate $\widehat{p}_t(y = k+1)$ to true prevalence $p_t(y = k+1)$. In particular, we show that the estimation error for prevalence of the novel class will primarily depend on sum of two terms: (i) error in approximating the label shift corrected source distribution, i.e., $p'_s(x)$; and (ii) purity of the top bin of the domain discriminator classifier.

Before formally introducing the result, we introduce some notation. Similar to before, given probability density function $p$ and a domain discriminator classifier $f : \mathcal{X} \to \Delta$, define a function $q = \int_{A(z)} p(x) dx$, where $A(z) = \{x \in \mathcal{X} : f(x) \geqslant z\}$ for all $z \in [0, 1]$. Intuitively, $q(z)$ captures the cumulative density of points in a top bin, i.e., the proportion of input domain that is assigned a value larger than $z$ by the function $f$ in the transformed space. We denote $p_t(x|y = k+1)$ with $p_{t,k+1}$. For each pdf $p_t$, $p_{t,k+1}$, and $p'_s$, we define $q_t$, $q_{t,k+1}$, and $q'_s$ respectively. Note that since We define an empirical estimator $\widehat{q}(z)$ given a set $X = \{x_1, x_2, \ldots, x_n\}$ sampled iid from $p(x)$. Let $Z = f(X)$. Define $\widehat{q}(z) = \sum_{i=1}^n \mathbb{I}[z_i \geqslant z]/n$.

Recall that in Step 5 of Algorithm 1, to estimate the proportion of novel class, we have access to re-sampled data from approximate label shift corrected source distribution $\widehat{q}'_s(x)$. Assume that we the size of re-sampled dataset is $n$.

**Theorem 5.** *Define* $c^* = \arg\min_{c \in [0,1]} (q_{t,k+1}(c)/\widehat{q}'_s(c))$. *Assume* $\min(n, m) \geqslant \left(\frac{2\log(4/\delta)}{(\widehat{q}'_s(c^*))^2}\right)$. *Then, for every* $\delta > 0$,




(a) Accuracy on validation positive versus negative data

(b) Fraction of correctly rejected examples with CVIR

Figure 3: **Comparison of different methods in overparameterized toy setup.** CVIR (random) denotes CVIR with random initialization and CVIR (PvU) denotes warm start with a positive versus negative classifier. Vertical line denotes the epoch at which we switch from PvU to CVIR in CVIR (PvU) training. (a) We observe that CVIR (PvU) improves significantly even over the best early stopped PvU model. As training proceeds, we observe that accuracy of nnPU, uPU and PvU training drops whereas CVIR (random) and CVIR (PvU) maintains superior and stable performance. (b) We observe that warm start training helps CVIR over randomly initialized model to correctly identity positives among unlabeled for rejection.

$[\widehat{p}_t]_{k+1} := \widehat{p}_t(y = k+1)$ *in Step 5 of Algorithm 1 satisfies with probability at least* $1-\delta$*, we have:*

$$|[\widehat{p}_t]_{k+1} - [p_t]_{k+1}| \leqslant (1 - [p_t]_{k+1}) \underbrace{\frac{|q'_s(c^*) - \widehat{q}'_s(c^*)|}{\widehat{q}'_s(c^*)}}_{\substack{\textit{Error in estimating} \\ \textit{label shift corrected source}}} + [p_t]_{k+1} \underbrace{\left(\frac{q_{t,k+1}(c^*)}{\widehat{q}'_s(c^*)}\right)}_{\substack{\textit{Impurity in} \\ \textit{top bin}}}$$
$$+ \mathcal{O}\left(\sqrt{\frac{\log(4/\delta)}{n}} + \sqrt{\frac{\log(4/\delta)}{m}}\right).$$

*Proof.* We can simply prove this theorem as Corollary of Theorem 1 from Garg et al. (2021b). Note that $q_t(c^*) = (1 - p_t(y = k+1)) \cdot q'_s(c^*) + p_t(y = k+1) \cdot q_{t,k+1}(c^*)$. Adding and subtracting $(1 - p_t(y = k+1)) \cdot \widehat{q}'_s(c^*)$ and dividing by $\widehat{q}'_s$, we get $\frac{q_t(c^*)}{\widehat{q}'_s(c^*)} = (1 - p_t(y = k+1)) \cdot \frac{|q'_s(c^*) - \widehat{q}'_s(c^*)|}{\widehat{q}'_s(c^*)} + (1 - p_t(y = k+1)) + p_t(y = k+1) \cdot \frac{q_{t,k+1}(c^*)}{\widehat{q}'_s(c^*)}$. Plugging in bound for LHS from Theorem 1 in Garg et al. (2021b), we get the desired result. □

### F.4. Extensions of Theorem 2 to general separable datasets

For general separable datasets, CVIR has undesirable property of getting stuck at local optima where gradient in (49) can be zero by maximizing entropy on the subset $P_a$ which is (incorrectly) not-rejected from $p_u$ in CVIR iterations. Intuitively, if the classifier can perfectly separate $P \backslash P_a$ and $N \backslash N_r$ and at the same time maximize the entropy of the region $P_a$, then the classifier trained with CVIR can get stuck in this local minima.

However, we can extend the above analysis with some modifications to the CVIR procedure. Note that when the CVIR classifier maximizes the entropy on $P_a$. it makes an error on points in $P_a$. Since, we have access to the distribution $p_p$, we can add an additional regularization penalty to the CVIR loss that ensures that the converged classifier with CVIR correctly classifies all the points in $p_p$. With a large enough regularization constant for the supervised loss on $p_p$, we can dominate the gradient term in (49) which pushes CVIR classifier to correct decision boundary even on $P_a$ (instead of maximizing entropy). We leave formal analysis of this conjecture for future work. Since we warm start CVIR training with a positive versus unlabeled classifier, if we obtain an initialization close enough to the true positive versus negative decision boundary, by monotonicity property of CVIR iterations, we may never get stuck in such a local minima even without modifications to loss.

## G. Empirical investigation of CVIR in toy setup

As noted in our ablation experiments and in Garg et al. (2021b), domain discriminator trained with CVIR outperforms classifiers trained with other consistent objectives (nnPU (Kiryo et al., 2017) and uPU (Du Plessis et al., 2015)). While the analysis in Sec. 5 highlights consistency of CVIR procedure in population, it doesn't capture the observed empirical efficacy of CVIR over alternative methods in overparameterized models. In the Gaussian setup described in Sec. F.2, we train overparameterized linear models to compare CVIR with other methods (Fig. 3). We fix $d = 1000$ and use $n = 250$ positive and $m = 250$ unlabeled points for training with $\alpha = 0.5$. We set the margin $\gamma$ at $0.05$. We compare CVIR with unbiased losses uPU and nnPU. We also make comparison with a naive positive versus unlabeled classifier (referred to as PvU). For CVIR, we experiment with a randomly initialized classifier and initialized with a PvU classifier trained for 200 epochs.

First, we observe that when a classifier is trained to distinguish positive and unlabeled data, *early learning* happens (Liu et al., 2020; Arora et al., 2019; Garg et al., 2021a), i.e., during the initial phase of learning classifier learns to classify positives in unlabeled correctly as positives achieving high accuracy on validation positive versus negative data. While the early learning happens with all methods, soon in the later phases of training PvU starts overfitting to the unlabeled data as negative hurting its validation performance. For uPU and nnPU, while they improve over PvU training during the initial epochs, the loss soon becomes biased hurting the performance of classifiers trained with uPU and nnPU on validation data.

For CVIR trained from a randomly initialized classifier, we observe that it improves slightly over the best PvU or the best nnPU model. Moreover, it maintains a relatively stable performance throughout the training. CVIR initialized with a PvU classifier significantly improves the performance. In Fig. 3 (b), we show that CVIR initialized with a PvU correctly rejects significantly more fraction of positives from unlabeled than CVIR trained from scratch. Thus, post early learning rejection of large fraction of positives from unlabeled training in equation (1) crucially helps CVIR.

## H. Experimental Details

### H.1. Baselines

We compare PULSE with several popular methods from OSDA literature. While these methods are not specifically proposed for OSLS, they are introduced for the more general OSDA problem. In particular, we make comparions with DANCE (Saito et al., 2020), UAN (You et al., 2019), CMU (Fu et al., 2020), STA (Liu et al., 2019), Backprop-ODA (or BODA) (Saito et al., 2018). We use the open source implementation available at `https://github.com/thuml` and `https://github.com/VisionLearningGroup/DANCE/`. Since OSDA methods do not estimate the prevalence of novel class explicitly, we use the fraction of examples predicted in class $k+1$ as a surrogate. We next briefly describe the main idea for each method:

*Backprob-ODA* Saito et al. (2018) proposed backprob ODA to train a $(k+1)$-way classifier. In particular, the network is trained to correctly classify source samples and for target samples, the classifier (specifically the last layer) is trained to output $0.5$ for the probability of the unknown class. The feature extractor is trained adversarially to move the probability of unknown class away from $0.5$ on target examples by utilizing the gradient reversal layer.

*Separate-To-Adapt (STA)* Liu et al. (2019) trained a network that learns jointly from source and target by learning to separate negative (novel) examples from target. The training is divided into two parts. The first part consists of training a multi-binary $G_c|_{c=1}^{|\mathcal{Y}_s|}$ classifier on labeled source data for each class and a binary classifier $G_b$ which generates the weights $w$ for rejecting target samples in the novel class. The second part consists of feature extractor $G_f$, a classifier $G_y$ and domain discriminator $G_d$ to perform adversarial domain adaptation between source and target data in the source label space. $G_y$ and $G_d$ are trained with incorporating weights $w$ predicted by $G_b$ in the first stage.

*Calibrated Multiple Uncertainties (CMU)* Fu et al. (2020) trained a source classifier and a domain discriminator to discriminate the novel class from previously seen classes in target. To train the discriminator network, CMU uses a weighted binary cross entropy loss where $w(x)$ for each example $x$ in target which is the average of uncertainty estimates, e.g. prediction confidence of source classifier. During test time, target data $x$ with $w(x) \geqslant w_0$ (for some pre-defined threshold $w_0$) is classified as an example from previously seen classes and is given a class prediction with source classifier. Otherwise, the target example is classified as belonging to the novel class.

*DANCE* Saito et al. (2020) proposed DANCE which combines a self-supervised clustering loss to cluster neighboring target examples and an entropy separation loss to consider alignment with source. Similar to CMU, during test time, DANCE uses thresholded prediction entropy of the source classifier to classifier a target example as belonging to the novel class.

*Universal Adaptation Networks (UAN)* You et al. (2019) proposed UAN which also trains a source classifier and a domain discriminator to discriminate the novel class from previously seen classes in target. The objective is similar to CMU where instead of using uncertainty estimates from multiple classifiers, UAN uses prediction confidence of domain discriminator classifier. Similar to CMU, at test time, target data $x$ with $w(x) \leqslant w_0$ (for some pre-defined threshold $w_0$) is classified as an example from previously seen classes and is given a class prediction with source classifier. Otherwise, the target example is classified as belonging to the novel class.

For alternative baselines, we experiment with source classifier directly deployed on the target data which may contain novel class and label shift among source classes (referred to as *source-only*). This naive comparison is included to quantify benefits of label shift correction and identifying novel class over a typical $k$-way classifiers.

We also train a domain discriminator classifier for source versus target (referred to as *domain disc.*). This is an adaptation of PU learning baseline(Elkan and Noto, 2008) which assumes no label shift among source classes. We use simple domain discriminator training to distinguish source versus target. To estimate the fraction of novel examples, we use the EN estimator proposed in Elkan and Noto (2008). For any target input, we make a prediction with the domain discriminator classifier (after re-scaling the sigmoid output with the estimate proportion of novel examples). Any example that is classified as target, we assign it the class $k + 1$. For examples classified as source, we make a prediction for them using the $k$-way source classifier.

Finally, per the reduction presented in Sec. 2, we train $k$ PU classifiers (referred to as *k-PU*). To train each PU learning classifier, we can plugin any method discussed in Sec. C. In the main paper, we included results obtained with plugin state-of-the-art PU learning algorithms. In App. H.8, we present ablations with other PU learning methods.

### H.2. Dataset and OSLS Setup Details

We conduct experiments with seven benchmark classification datasets across vision, natural language, biology and medicine. Our datasets span language, image and table modalities. For each dataset, we simulate an OSLS problem. We experiment with different fraction of novel class prevalence, source label distribution, and target label distribution. We randomly choose classes that constitute the novel target class. After randomly choosing source and novel classes, we first split the training data from each source class randomly into two partitions. This creates a random label distribution for shared classes among source and target. We then club novel classes to assign them a new class (i.e. $k + 1$). Finally, we throw away labels for the target data to obtain an unsupervised DA problem. We repeat the same process on iid hold out data to obtain validation data with no target labels. For main experiments in the paper, we next describe important details for the OSLS setup simulated. All the other details can be found in the code repository.

For vision, we use CIFAR10, CIFAR100 (Krizhevsky and Hinton, 2009) and Entity30 (Santurkar et al., 2021). For language, we experiment with Newsgroups-20 dataset. Additionally, inspired by applications of OSLS in biology and medicine, we experiment with Tabula Muris (Consortium et al., 2020) (Gene Ontology prediction), Dermnet (skin disease prediction), and BreakHis (Spanhol et al., 2015) (tumor cell classification).

**CIFAR10** For CIFAR10, we randomly select 9 classes as the source classes and a novel class formed by the remaining class. After randomly sampling the label marginal for source and target randomly, we get the prevalence for novel class as $0.2152$.

**CIFAR100** For CIFAR100, we randomly select $85$ classes as the source classes and a novel class formed by aggregating the data from $15$ remaining classes. After randomly sampling the label marginal for source and target randomly, we get the prevalence for novel class as $0.2976$.

**Entity30** Entity30 is a subset of ImageNet (Russakovsky et al., 2015) with 30 super classes. For Entity30, we randomly select $24$ classes as the source classes and a novel class formed by aggregating the data from $6$ remaining classes. After randomly sampling the label marginal for source and target randomly, we get the prevalence for novel class as $0.3942$.

**Newgroups-20** For Newsgroups20[1], we randomly select $16$ classes as the source classes and a novel class formed by aggregating the data from $4$ remaining classes. After randomly sampling the label marginal for source and target randomly, we get the prevalence for novel class as $0.3733$. This dataset is motivated by scenarios where novel news categories can appear over time but the distribution of articles given a news category might stay relatively unchanged.

**BreakHis** BreakHis[2] contains $8$ categories of cell types, $4$ types of benign breast tumor and $4$ types malignant tumors

[1] http://qwone.com/~jason/20Newsgroups/

[2] https://web.inf.ufpr.br/vri/databases/breast-cancer-histopathological-database-breakhis/

(breast cancer). Here, we simulate OSLS problem specifically where 6 cell types are observed in the source (3 from each) and a novel class appears in the target with 1 cell type from each category. After randomly sampling the label marginal for source and target randomly, we get the prevalence for novel class as $0.2708$.

**Dermnet** Dermnet data contains images of 23 types of skin diseases taken from Dermnet NZ[3]. We simulate OSLS problem specifically where 18 diseases are observed in the source and a novel class appears in the target with the rest of the 5 diseases. After randomly sampling the label marginal for source and target randomly, we get the prevalence for novel class as $0.3133$.

**Tabula Muris** Tabula Muris dataset (Consortium et al., 2020) comprises of different cell types collected across 23 organs of the mouse model organism. We use the data pre-processing scripts provided in (Cao et al., 2021)[4]. We just use the training set comprising of 57 classes for our experiments. We simulate OSLS problem specifically where 28 cell types are observed in the source and a novel class appears in the target with the rest of the 29 cell types. After randomly sampling the label marginal for source and target randomly, we get the prevalence for novel class as $0.6366$.

### H.3. Details on the Experimental Setup

We use Resnet18 (He et al., 2016) for CIFAR10, CIFAR100, and Entity30. For all three datasets, in our main experiments, we train Resnet-18 from scratch. We use SGD training with momentum of $0.9$ for 200 epochs. We start with learning rate $0.1$ and decay it by multiplying it with $0.1$ every 70 epochs. We use a weight decay of $5 \times 10^{-4}$. For CIFAR100 and CIFAR10, we use batch size of 200. For Entity30, we use a batch size of 32. In App. H.7, we experiment with contrastive pre-training instead of random initialization.

For newsgroups, we use a convolutional architecture[5]. We use glove embeddings to initialize the embedding layer. We use Adam optimizer with a learning rate of $0.0001$ and no weight decay. We use a batch size of 200. We train with constant learning rate for 120 epochs.

For Tabular Muris, we use the fully connected MLP used in Cao et al. (2021). We use the hyperparameters used in Cao et al. (2021). We use Adam optimizer with a learning rate of $0.0001$ and no weight decay. We train with constant learning rate for 40 epochs. We use a batch size of 200.

For Dermnet and BreakHis, we use Resnet-50 pre-trained on Imagenet. We use an initial learning rate of $0.0001$ and decay it by $0.96$ every epoch. We use SGD training with momentum of $0.9$ and weight decay of $5 \times 10^{-4}$. We use a batch size of 32. These are the default hyperparameters used in Alom et al. (2019) and Liao (2016).

For all methods, we use the same backbone for discriminator and source classifier. Additionally, for PULSE and domain disc., we use the exact same set of hyperparameters to train the domain discriminator and source classifier. For kPU, we use a separate final layer for each class with the same backbone. We use the same hyperparameters described above for all three methods. For OSDA methods, we use default method specific hyperparameters introduced in their works. Since we do not have access to labels from the target data, we do not perform hyperparameter tuning but instead use the standard hyperparameters used for training on labeled source data. In future, we may hope to leverage heuristics proposed for accuracy estimation without access to labeled target data (Garg et al., 2022).

We train models till the performance on validation source data (labeled) ceases to increase. Unlike OSDA methods, note that we do not use early stopping based on performance on held-out labeled target data. To evaluate classification performance, we report target accuracy on all classes, seen classes and the novel class. For target marginal, we separately report estimation error for previously seen classes and for the novel class. For the novel class, we report absolute difference between true and estimated marginal. For seen classes, we report average absolute estimation error. We open-source our code at `https://github.com/Neurips2022Anon`. By simply changing a single config file, new OSLS setups can be generated and experimented with.

Note that for our main experiments, for vision datasets (i.e., CIFAR10, CIFAR100, and Entity30) and for language dataset, we do not initialize with a (supervised) pre-trained model to avoid overlap of novel classes with the classes in the dataset used for pre-training. For example, labeled Imagenet-1k is typically used for pre-training. However, Imagenet classes overlaps with all three vision datasets employed and hence, we avoid pre-trained initialization. In App. H.7, we experiment with contrastive pre-training on Entity30 and CIFAR100. In contrast, for medical datasets, we leverage Imagenet pre-trained

[3] `http://www.dermnet.com/dermatology-pictures-skin-disease-pictures`
[4] `https://github.com/snap-stanford/comet`
[5] `https://github.com/mireshghallah/20Newsgroups-Pytorch`

models as there is no overlap between classes in BreakHis and Dermnet with Imagenet.

### H.4. Detailed results from main paper

For completeness, we next include results for all datasets. In particular, for each dataset we tabulate (i) overall accuracy on target; (ii) accuracy on seen classes in target; (iii) accuracy on the novel class; (iv) sum of absolute error in estimating target marginal among previously seen classes, i.e., $\sum_{y \in \mathcal{Y}_s} |\widehat{p}_t(y) - p_t(y)|$; and (v) absolute error for novel fraction estimation, i.e., $|\widehat{p}_t(y = k+1| - p_t(y = k+1)$. Table 5 presents results on all the datasets. Fig. 4 and Fig. 5 presents epoch-wise results.

### H.5. Investigation into OSDA approaches

We observe that with default hyperparameters, popular OSDA methods significantly under perform as compared to PULSE. We hypothesize that the primary reasons underlying the poor performance of OSDA methods are (i) the heuristics employed to detect novel classes; and (ii) loss functions incorporated to improve alignment between examples from common classes in source and target. To detect novel classes, a standard heuristic employed popular OSDA methods involves thresholding uncertainty estimates (e.g., prediction entropy, softmax confidence (You et al., 2019; Fu et al., 2020; Saito et al., 2020)) at a predefined threshold $\kappa$. However, a fixed $\kappa$, may not for different datasets and different fractions of the novel class. Here, we ablate by (i) removing loss function terms incorporated with an aim to improve source target alignment; and (ii) vary threshold $\kappa$ and show improvements in performance of these methods.

For our investigations, we experiment with CIFAR10, with UAN and DANCE methods. For DANCE, we remove the entropy separation loss employed to encourage align target examples with source examples. For UAN, we remove the adversarial domain discriminator training employed to align target examples with source examples. For both the methods, we observe that by removing the corresponding loss function terms we obtain a marginal improvement. For DANCE on CIFAR10, the performance goes up from $70.4$ to $72.5$ (with the same hyperparameters as the default run). FOR UAN, we observe similar minor improvements, where the performance goes up from $15.4$ to $19.6$.

Next, we vary the threshold used for detecting the novel examples. By optimally tuning the threshold for CIFAR10 with UAN, we obtain a substantial increase. In particular, the overall target accuracy increases from $19.6$ to $33.1$. With DANCE on CIFAR10, optimal threshold achieves $75.6$ as compared to the default accuracy $70.4$. In contrast, our two-stage method PULSE avoids the need to guess $\kappa$, by first estimating the fraction of novel class which then guides the classification of novel class versus previously seen classes.

### H.6. Ablation with novel class fraction

In this section, we ablate on novel class proportion on CIFAR10, CIFAR100 and Newsgroups20. For each dataset we experiment with three settings, each obtained by varying the number of classes from the original data that constitutes the novel classes. We tabulate our results in Table 4.

### H.7. Contrastive pre-training on unlabeled data

Here, we experiment with contrastive pre-training to pre-train the backbone networks used for feature extraction. In particular, we initialize the backbone architectures with SimCLR pre-trained weights. We experiment with CIFAR100 and Entity30 datasets. Instead of pre-training on mixture of source and target unlabeled data, we leverage the publicly available pre-trained weights[6]. Table 2 summarizes our results. We observe that pre-training improves over random initialization for all the methods with PULSE continuing to outperform other approaches.

### H.8. Ablation with different PU learning methods

In this section, we experiment with alternative PU learning approaches for PULSE and kPU. In particular, we experiment with the next best alternatives, i.e., nnPU instead of CVIR for classification and DEDPUL instead of BBE for target marginal estimation. We refer to these as kPU (alternative) and PULSE (alternative) in Table 3. We present results on three datasets: CIFAR10, CIFAR100 and Newsgroups20 in the same setting as described in Sec. H.2. We make two key observations: (i) PULSE continues to dominate kPU with alternative choices; (ii) CVIR and BBE significantly outperform alternative choices.

[6]For CIFAR100: https://drive.google.com/file/d/1huW-ChBVvKcx7t8HyDaWTQB5Li1Fht9x/view and for Entity30, we use Imagenet pre-trained weights from here: https://github.com/AndrewAtanov/simclr-pytorch.

Table 2: Comparison with different OSLS approaches with pre-trained feature extractor. We use SimCLR pre-training to initialize the feature extractor for all the methods. All methods improve over random initialization (in Table 1). Note that PULSE continues to outperform other approaches.

| Method | CIFAR100 | | Entity30 | |
|---|---|---|---|---|
| | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) |
| BODA (Saito et al., 2018) | 37.1 | 0.34 | 52.1 | 0.376 |
| Domain Disc. | 49.4 | 0.041 | 57.4 | 0.024 |
| kPU | 37.5 | 0.297 | 70.1 | 0.32 |
| PULSE (Ours) | 67.3 | 0.052 | 72.4 | 0.002 |

Table 3: Comparison with different PU learning approaches. 'Alternative' denotes results with employing nnPU for classification and DEDPUL for target marginal estimation instead of 'default' which uses CVIR and BBE.

| Method | CIFAR10 | | CIFAR100 | | Newsgroups20 | |
|---|---|---|---|---|---|---|
| | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) |
| $k$-PU (alternative) | 53.4 | 0.215 | 12.1 | 0.298 | 14.1 | 0.373 |
| $k$-PU (default) | 83.6 | 0.036 | 36.3 | 0.298 | 52.1 | 0.307 |
| PULSE (alternative) | 80.5 | 0.05 | 30.1 | 0.231 | 39.8 | 0.223 |
| PULSE (default) | 86.1 | 0.008 | 63.4 | 0.078 | 62.2 | 0.061 |

Table 4: Comparison with different OSLS approaches for different novel class prevalence. We observe that for on CIFAR100 and Newsgroups20, PULSE maintains superior performance as compared to other approaches. On CIFAR10, as the proportion of novel class increases, the performance of of kPU improves slightly over PULSE for target accuracy.

| Method | CIFAR10 ($p_t(k+1) = 0.215$) | | CIFAR10 ($p_t(k+1) = 0.406$) | | CIFAR10 ($p_t(k+1) = 0.583$) | |
|---|---|---|---|---|---|---|
| | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) |
| BODA (Saito et al., 2018) | 63.1 | 0.162 | 65.5 | 0.166 | 48.6 | 0.265 |
| Domain Disc. | 47.4 | 0.331 | 57.5 | 0.232 | 68.7 | 0.144 |
| kPU | 83.6 | 0.036 | 87.8 | 0.010 | 89.9 | 0.036 |
| PULSE (Ours) | 86.1 | 0.008 | 87.4 | 0.009 | 83.7 | 0.006 |

| Method | CIFAR100 ($p_t(k+1) = 0.2976$) | | CIFAR100 ($p_t(k+1) = 0.4477$) | | CIFAR100 ($p_t(k+1) = 0.5676$) | |
|---|---|---|---|---|---|---|
| | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) |
| BODA (Saito et al., 2018) | 36.1 | 0.41 | 41.6 | 0.075 | 50.2 | 0.03 |
| Domain Disc. | 45.8 | 0.046 | 52.3 | 0.092 | 58.7 | 0.187 |
| kPU | 36.3 | 0.298 | 52.2 | 0.448 | 63.9 | 0.568 |
| PULSE (Ours) | 63.4 | 0.078 | 66.6 | 0.052 | 68.2 | 0.088 |

| Method | Newsgroups20 ($p_t(k+1) = 0.3733$) | | Newsgroups20 ($p_t(k+1) = 0.6452$) | | Newsgroups20 ($p_t(k+1) = 0.7688$) | |
|---|---|---|---|---|---|---|
| | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) | Acc (All) | MPE (Novel) |
| BODA (Saito et al., 2018) | 43.4 | 0.16 | 25.5 | 0.645 | 17.7 | 0.769 |
| Domain Disc. | 50.9 | 0.176 | 44.8 | 0.085 | 47.8 | 0.064 |
| kPU | 52.1 | 0.373 | 50.2 | 0.645 | 35.5 | 0.769 |
| PULSE (Ours) | 62.2 | 0.061 | 71.7 | 0.044 | 75.73 | 0.179 |

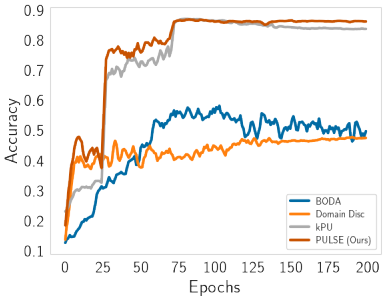
Table 5: *Comparison of PULSE with other methods*. Across all datasets, PULSE outperforms alternatives for both target classification and novel class prevalence estimation. Acc (All) is target accuracy, Acc (Seen) is target accuracy on examples from previously seen classes, and Acc (Novel) is recall for novel examples. MPE (Seen) is sum of absolute error for estimating target marginal among previously seen classes and MPE (Novel) is absolute error for novel prevalence estimation. Results reported by averaging across 3 seeds.

| Method | CIFAR-10 | | | | | CIFAR-100 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Acc (All) | Acc (Seen) | Acc (Novel) | MPE (Seen) | MPE (Novel) | Acc (All) | Acc (Seen) | Acc (Novel) | MPE (Seen) | MPE (Novel) |
| Source-Only | 67.1 | 87.0 | - | - | - | 46.6 | 66.4 | - | - | - |
| UAN (You et al., 2019) | 15.4 | 19.7 | 25.2 | 1.44 | 0.214 | 18.1 | 40.6 | 14.8 | 1.48 | 0.133 |
| BODA (Saito et al., 2018) | 63.1 | 66.2 | 42.0 | 0.541 | 0.162 | 36.1 | 17.7 | 81.6 | 0.564 | 0.41 |
| DANCE (Saito et al., 2020) | 70.4 | 85.5 | 14.5 | 0.784 | 0.174 | 47.3 | 66.4 | 1.2 | 0.702 | 0.28 |
| STA (Liu et al., 2019) | 57.9 | 69.6 | 14.9 | 0.409 | 0.124 | 42.6 | 48.5 | 34.8 | 0.798 | 0.14 |
| CMU (Fu et al., 2020) | 62.1 | 77.9 | 41.2 | 0.443 | 0.183 | 35.4 | 46.0 | 15.5 | 0.695 | 0.161 |
| Domain Disc. | 47.4 | 87.0 | 30.6 | - | 0.331 | 45.8 | 66.5 | 39.1 | - | **0.046** |
| $k$-PU | 83.6 | 79.4 | **98.9** | **0.062** | 0.036 | 36.3 | 22.6 | **99.1** | 6.31 | 0.298 |
| PULSE (Ours) | **86.1** | **91.8** | 88.4 | 0.091 | **0.008** | **63.4** | **67.2** | 63.5 | **0.365** | 0.078 |

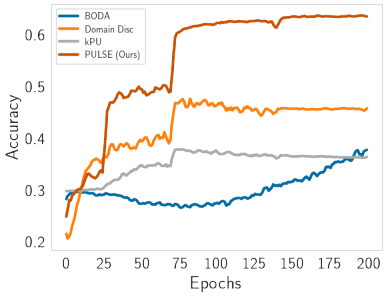
| Method | Entity30 | | | | | Newsgroup20 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Acc (All) | Acc (Seen) | Acc (Novel) | MPE (Seen) | MPE (Novel) | Acc (All) | Acc (Seen) | Acc (Novel) | MPE (Seen) | MPE (Novel) |
| Source-Only | 32.0 | 53.5 | - | - | - | 39.3 | 64.4 | - | - | - |
| BODA (Saito et al., 2018) | 42.22 | 25.9 | 67.2 | 0.367 | 0.189 | 43.4 | 38.0 | 34.1 | 0.550 | 0.167 |
| Domain Disc. | 43.2 | 53.5 | 68.0 | - | 0.135 | 50.9 | 64.4 | **93.2** | - | 0.176 |
| $k$-PU | 50.7 | 22.3 | **94.4** | 0.99 | 0.394 | 52.1 | 57.8 | 42.7 | 0.776 | 0.373 |
| PULSE (Ours) | **58.0** | **54.3** | 72.2 | **0.215** | **0.054** | **62.2** | **65.0** | 83.6 | **0.232** | **0.061** |

| Method | Tabula Muris | | | | | BreakHis | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Acc (All) | Acc (Seen) | Acc (Novel) | MPE (Seen) | MPE (Novel) | Acc (All) | Acc (Seen) | Acc (Novel) | MPE (Seen) | MPE (Novel) |
| Source-Only | 33.8 | 93.3 | - | - | - | 70.0 | 95.8 | - | - | - |
| BODA (Saito et al., 2018) | 76.5 | 59.8 | 87.0 | 0.200 | 0.079 | 71.5 | 81.8 | 44.0 | 0.163 | 0.077 |
| Domain Disc. | 73.0 | 93.3 | **94.7** | - | 0.071 | 56.5 | 95.8 | **90.4** | - | 0.09 |
| $k$-PU | 85.9 | 91.6 | 83.3 | **0.279** | 0.307 | 75.6 | 71.7 | 86.1 | 0.094 | 0.058 |
| PULSE (Ours) | **87.8** | **94.6** | 88.8 | 0.388 | **0.058** | **79.1** | **96.1** | 76.3 | **0.090** | **0.054** |

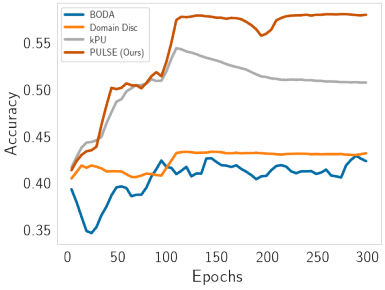
| Method | Dermnet | | | | |
|---|---|---|---|---|---|
| | Acc (All) | Acc (Seen) | Acc (Novel) | MPE (Seen) | MPE (Novel) |
| Source-Only | 41.4 | 53.6 | - | - | - |
| BODA (Saito et al., 2018) | 43.8 | 31.4 | 58.4 | **0.401** | 0.207 |
| Domain Disc. | 40.6 | 53.6 | 82.7 | - | 0.083 |
| $k$-PU | 46.0 | 26.0 | **89.9** | 1.44 | 0.313 |
| PULSE (Ours) | **48.9** | **53.7** | 57.7 | **0.41** | **0.043** |



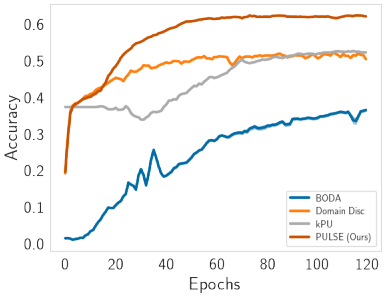

(a) CIFAR10



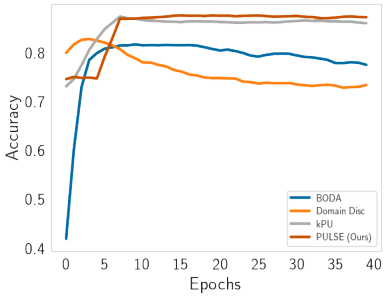

(b) CIFAR100



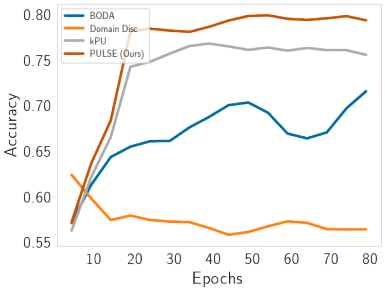

(c) Entity30



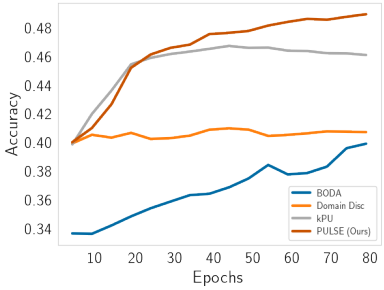

(d) Newsgroups20




(e) Tabula Muris

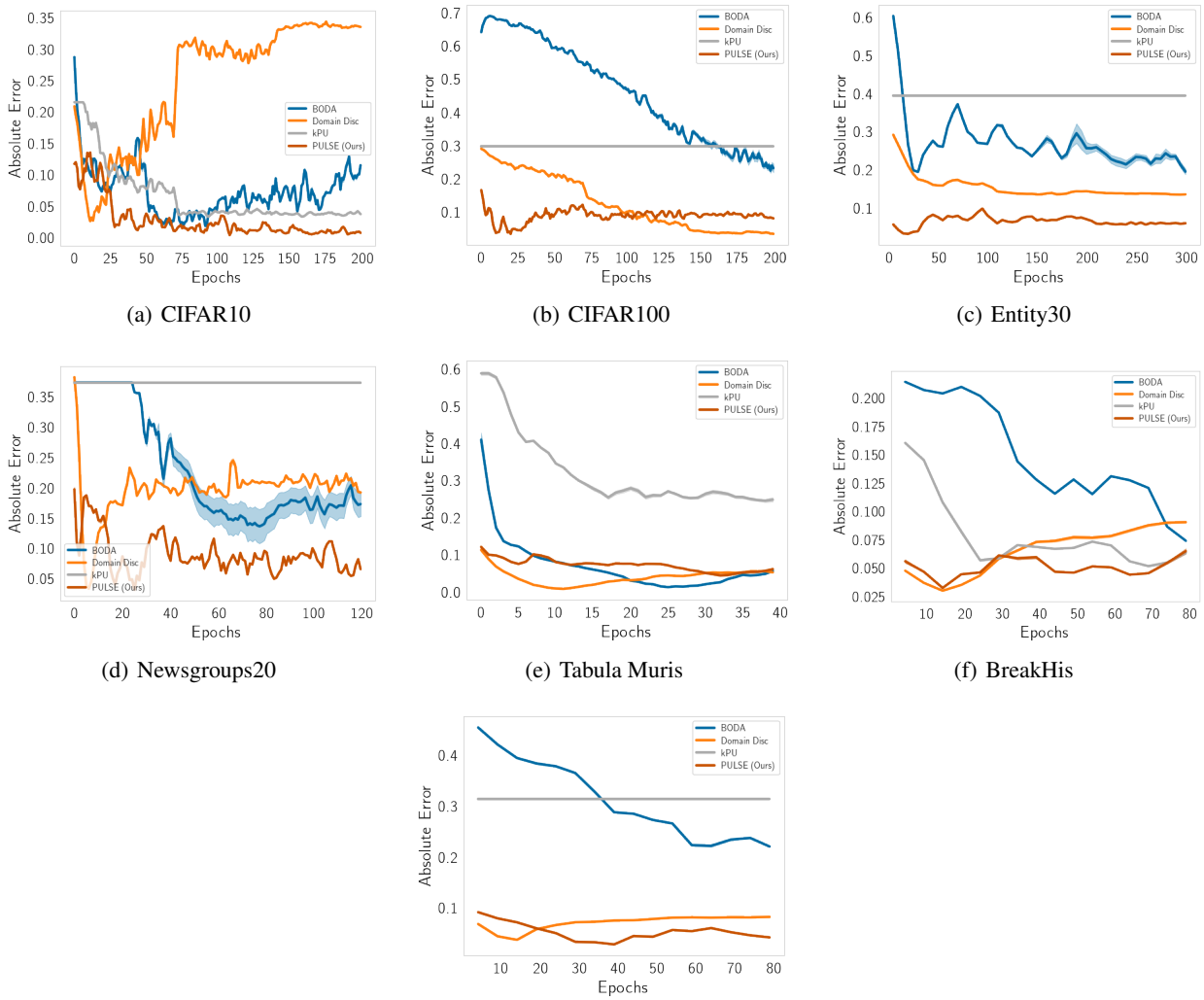



(f) BreakHis




(g) Dermnet

Figure 4: **Epoch wise results for target accuracy.** Results aggregated over 3 seeds. PULSE maintains stable and superior performance when compared to alternative methods.




(a) CIFAR10 (b) CIFAR100 (c) Entity30

(d) Newsgroups20 (e) Tabula Muris (f) BreakHis

(g) Dermnet

Figure 5: **Epoch wise results for novel prevalence estimation.** Results aggregated over 3 seeds. PULSE maintains stable and superior performance when compared to alternative methods.