Thesis for The Degree of Doctor of Philosophy

Adaptation under Distributional Shifts in Centralized and Federated settings

Adam Breitholtz

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg
Gothenburg, Sweden, 2025

Adaptation under Distributional Shifts in Centralized and Federated settings

Adam Breitholtz

ISBN 978-91-8103-349-6
Doktorsavhandlingar vid Chalmers tekniska högskola, Ny serie nr 5806.
ISSN 0346-718X

Department of Computer Science and Engineering
Division of Data Science and AI
Healthy AI research group
Chalmers University of Technology | University of Gothenburg
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2025.

"You can't cross the sea merely by standing staring at the water."

Adaptation under Distributional Shifts in Centralized and Federated settings

Adam Breitholtz

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg

Abstract

Using large datasets to train ever larger models in machine learning have been extremely impactful for many diverse tasks. However, the challenge of ensuring and predicting model generalization under distribution shifts remains an open problem. Such shifts may occur between training and testing environments or even during the training process itself. In real-world applications, these distribution changes can severely degrade model performance, making adaptation a critical concern. This is the focus of domain adaptation (DA), a field dedicated to developing both theoretical frameworks and methods for settings with distribution shift. Domain adaptation primarily operates within the supervised learning paradigm, where access to a large, centralized dataset is assumed. However, such data availability is not always feasible due to privacy concerns or the high costs associated with data collection and storage. The federated learning (FL) setting addresses this by training models across decentralized clients coordinated by a central server. Since clients retain local data, distribution shifts, known as data heterogeneity, can arise between clients. This may potentially degrade model performance. This thesis aims to overcome some of these limitations in both the centralized and federated settings. In particular, this is achieved by (i) questioning how to measure performance under distribution shift in a practical way, (ii) proposing novel assumptions and settings where we expand the amount of information available and (iii) developing competitive methods for these settings. First, we explore the measurement of performance in domain adaptation through evaluating theoretical bounds. We survey the field of available domain adaptation bounds with an eye towards their practicality and, after selecting candidates, make empirical comparisons. Next, we consider a novel set of assumptions based on having access to privileged information which we show is both practical and empirically sound. We continue with expanding on the idea of additional information in the FL setting where we show that access to label marginals can substantially improve performance in cases where clients are meaningfully heterogeneous. Finally, we explore another aspect of heterogeneity in FL where the label sets of clients are non-identical and clients are unwilling to share them.

Keywords

Distribution shift, Domain Adaptation, Privileged Information, Federated Learning, Heterogenous clients

List of Publications

Appended publications

This thesis is based on the following publications:

[**Paper I**] **A. Breitholtz**, F. D. Johansson, *Practicality of generalization guarantees for unsupervised domain adaptation with neural networks*
Transactions on Machine Learning Research (October 2022)
Also presented at The AAAI-22 Workshop on Engineering Dependable and Secure Machine Learning Systems (March, 2022).

[**Paper II**] **A. Breitholtz**, A. Matsson, F. D. Johansson, *Unsupervised Domain Adaptation by Learning using Privileged Information*
Transactions on Machine Learning Research (September 2024)
Also presented at The Second Workshop on Spurious Correlations, Invariance and Stability, ICML (July, 2023).

[**Paper III**] E. Listo Zec, **A. Breitholtz**, F. D. Johansson, *Overcoming label shift in target-aware federated learning*
Submitted, under review. arXiv:2411.03799
Also presented at Tiny Titans: The next wave of On-Device Learning for Foundational Models (TTODLer-FM), ICML (July, 2025).

[**Paper IV**] **A. Breitholtz**, E. Listo Zec, F. D. Johansson, *Federated Learning with Heterogeneous and Private Label Sets*
Springer Workshop Proceedings of ECML-PKDD 2025
Presented at 3rd Workshop on Advancements in Federated Learning (WAFL), ECML-PKDD (September, 2025).

Contribution summary

The author's contributions to the publications included in this thesis is detailed below.

[**Paper I**] Co-designed the study, performed the empirical work, and wrote most of the manuscript.

[**Paper II**] Co-designed the study, performed part of the empirical work, and wrote parts of the manuscript. The first two authors contributed equally.

[**Paper III**] Co-designed the study, performed part of the empirical work, and wrote parts of the manuscript. The first two authors contributed equally.

[**Paper IV**] Co-designed the study, performed part of the empirical work, and wrote parts of the manuscript. The first two authors contributed equally.

Acknowledgment

I would like to start by expressing my deepest gratitude towards my supervisor Fredrik Johansson. Without your seemingly endless patience, optimism, enduring support and belief in our research; this thesis would not exist. Thank you for everything.

I want to also thank my co-supervisor, Devdatt Dubhashi, and my examiner, Dag Wedelin for their feedback, support and encouragement throughout this process.

To everyone working at the DSAI division, I thank you for contributing to making the division a great place to be. Over the years, I have had the privilege of getting to know many great people ranging from professors to the administrative staff. In particular, I want to extend a special thanks the original members of the Healthy AI Lab Anton, Lena and Newton. I wish you all the best going forward. I also want to thank my collaborators Anton and Edvin. Thank you both for your great contributions to our work.

Also, I want to thank my friends that have put up with me throughout this journey. It has been an arduous march and I hope that in future I will be able to give back to you for the grace you have given me during this time. I want to particularly thank Samuel Håkansson for taking the time to proof-read a draft of this thesis.

Finally, I want to thank my family for their support. Without you I would not be writing these words here at all. A special thank you to my girlfriend Nathalie for her love and support when I needed it most.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Chalmers Centre for Computational Science and Engineering (C3SE) partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

Contents

Part I

Overview

Chapter 1

Introduction

Machine learning (ML) has emerged as both a popular and useful way to harness information for beneficial aims in diverse applications such as climate modeling, predicting healthcare outcomes and natural language processing. A central trend in ML is the focus on deep learning techniques, which has contributed to the explosive growth of the field. These models allow for learning intricate patterns from high-dimensional data sets through optimization methods like stochastic gradient descent. This has led to impressive results, such as large language models generating human-like text (Bubeck et al., 2023; DeepSeek-AI et al., 2025), mastering games like Go (Silver et al., 2017), and predicting medical outcomes (Ding et al., 2024). However, deep learning comes with significant limitations. It generally needs access to a large sample of data to be successful, which may not be available in all situations. In particular, sensitive settings such as healthcare may lack the large data sets which are routinely used when optimizing deep learning models. Furthermore, if the data differs materially between modeling and the target domain where it is applied, the models may fail to perform. Moreover, this can also arise *during* training where there are several distinct sources of data, e.g., in distributed training environments. This issue arises due to a shift between data distributions, which limits the generalizability of deep learning models.

To remedy this, the field of research called domain adaptation (DA) aims to offer a theoretical foundation of how to produce more generalizable methods. In DA we consider the case where we apply our model to a specific task but the underlying data distribution changes between training and deployment. The distribution over the input features and corresponding labels is called a *domain*. In real-world applications, a domain shift could be the result of e.g. a change in location where the data is collected. For example, we collect wildlife image data in Europe to predict animal species but apply the model in Africa, where the underlying data distribution is likely different. We generally refer to the data we train on as the *source* domain and the *target* domain for testing data. One of the main challenges in DA is that the underlying distribution of samples is not known, and thus a data set may be distributed differently than a test set due to several factors e.g. a limitation in the number of samples,

biased sampling or external factors changing the sampled data over time.

In scenarios where we observe distribution shift there may be further constraints that arise naturally, an important case being that information may be unavailable. An example of this is in the healthcare setting when we are trying to classify pathologies using chest X-ray images. We can train a model from historical registry data collected at Hospital A. Naturally, we may then want to apply this model at Hospital B. However, the patient cohort at Hospital B can be substantially different than the one at A. In this setting it is common that the labels corresponding to some features might be hard or even impossible to access, e.g. predicting outcomes which have yet to occur. For example, we probably do not know if a patient will develop COPD within a year from when features are collected. Therefore we can assume that we have access to features (chest X-rays) from hospital B but not the corresponding labels. This specific setting is called the *unsupervised domain adaptation* (UDA) setting, where labels are unavailable for the target domain. This setting is common in healthcare and other domains where labeling is expensive, time-consuming, or impossible.

Foundational theoretical work in UDA, such as the seminal work by Ben-David et al. (2007), and subsequent contributions (Cortes et al., 2010; Mansour et al., 2009b), aim to quantify how distribution shifts affect model performance. Since there is a lack of labeled information it does not allow us to adjust our model based on this knowledge. Furthermore, it is known that we may find guarantees of consistent learning if we do have this information (Blitzer et al., 2008). Moreover, adaptation in this setting is impossible without additional assumptions. (Ben-David & Urner, 2012) However, despite many theoretical and methodological advances, the theory of UDA often relies on assumptions which are implausible in real-world application or metrics which are impossible to quantify with the observed data. Furthermore, it is not clear what amount of information to assume access to for realistic and useful generalization guarantees. Moreover, there still remains a gap in performance compared to models that has been given access to target labels, which is not fully explained theoretically.

Distributional shifts and their impacts *during* training arises in DA when we consider multiple sources which contain their own distinct datasets. These multiple sources may give rise to heterogeneity in the classifiers learned on these disparate data sets and has been studied extensively (Mansour et al., 2008; Ben-David et al., 2010a; Hoffman et al., 2018; Peng et al., 2019).

An example of this is the *Federated* learning (FL) setting which also contains multiple sources of information with an additional constraint on communication. Federated learning is a form of decentralized learning where several clients collaborate, coordinated by a central server, to build a stronger model than they could individually. This training is based on iteratively transmitting model parameters or gradient updates to the central server, which then aggregates these into one central model which is transmitted back to the clients. This setup preserves privacy but introduces challenges due to data heterogeneity across clients.

Originally proposed as a method for distributed training under limited computational and bandwidth resources (McMahan et al., 2017), FL has

evolved into a broader paradigm focused on training models without sharing data across clients, maintaining privacy while still benefiting from all client datasets. FL thus embodies a trade-off between performance and privacy, with some works emphasizing privacy preservation, and others focusing on maximizing performance. However, distribution shifts among clients remain a key open challenge (Kairouz et al., 2021). The heterogeneity these shifts produce can degrade model performance and adversely effect convergence.

There are myriad approaches to handling different types of heterogeneity in FL based on adjusting either server-side (Reddi et al., 2021; Zeng et al., 2023; Li et al., 2023; Zhou et al., 2023) or on the client-side (Li et al., 2020; 2021; Chen et al., 2018; Jiang et al., 2019; Fallah et al., 2020; Li & Zhan, 2021) to account for the heterogeneity. However, since we generally do not know the particulars of all datasets at neither the server nor the clients we cannot make use of the theoretical approaches found in DA. Therefore, there is a need to find methods and assumptions which allow us to learn useful models in FL, even under distribution shift.

In this thesis we will investigate the issues detailed above in several ways, both in the centralized UDA and federated learning settings. We consider the following questions (i) how to predict performance under distribution shift in a practical way, (ii) what benefits are gained by introducing settings where we expand the amount of information available during training and (iii) how to develop competitive methods for these settings.

First, we explore the measurement of performance in unsupervised domain adaptation by evaluating theoretical bounds. We survey the field of available generalization bounds with an particular focus towards their practicality and, after selecting candidates, make empirical comparisons. Next, we consider a novel set of assumptions for UDA based on the having access to auxiliary privileged information which we show is both practical and empirically well-performing. We continue with expanding on the idea of additional information in the FL setting where we show that access to label marginals can substantially improve performance in cases where clients are heterogeneous with respect to their label distributions. Finally, we explore another aspect of heterogeneity in FL where the label sets of clients are non-identical and clients are unwilling to share them for privacy reasons. We propose adaptations of common FL methods and find that they incur only a small performance cost for the increased privacy. We also adapt methods from classifier coupling to align client models which achieves some promising results.

1.1 Thesis contributions

The main contributions of this thesis is summarized as follows:

- Paper I (Breitholtz & Johansson, 2022) explores whether it is possible to use the available bounds in the UDA literature for the explicit purpose of accurate performance prediction. We postulate a set of desiderata which such a bound should fulfill: that it should be tight; i.e. close to the observed performance of our model, possible to estimate from observed

data and tractably computable for realistic model classes and datasets. We survey the field of available domain adaptation bounds with regards to these desiderata and make empirical comparisons of how they perform on these three attributes. We find that PAC-Bayesian style bounds are a good fit for this and show two new such bounds.

- Paper II (Breitholtz et al., 2024) proposes a novel set of assumptions based on the concept of *privileged information* (PI), information which is auxiliary and only available during training. We argue that this set is more plausible than the more common set of assumptions used in the literature. Further, we show that these assumptions ensure consistent learning and provide a generalization bound. We then construct two model architectures, one which more closely follows our theory and a more practical end-to-end method. Extensive empirical investigation, across several datasets, show that these methods perform well compared to baselines which does not make use of PI.

- Paper III (Zec et al., 2025) investigates a similar concept to Paper II in the federated learning setting with heterogenous clients. It introduces FedPALS, a method which makes use of label marginals from clients and the target domain. It seeks to find a trade-off between having the aggregation of models be faithful to the target distribution and maintaining maximal sample efficiency. We show that our method resolves to the common federated averaging algorithm when focus on sample efficiency dominates. Empirically, we find that FedPALS performs very well with substantial performance gains compared to baselines which lack access to label marginals.

- Paper IV (Breitholtz et al., 2025) considers a more private FL setting with increased heterogeneity compared to Paper III. We consider that label sets are non-identical across clients. In addition, the clients are unwilling to share their label sets for privacy reasons; we call this the private label set setting. We adapt common FL methods to this setting as well as a tuning approach inspired by classifier coupling literature. We find that these methods perform quite well in this restricted setting with small performance impacts compared to the non-private setting.

1.2 Thesis outline

This thesis is outlined as follows: In Chapter 2 we detail background of distributional shifts, domain adaptation, federated learning and other concepts which are used in the appended papers.

Chapter 3 contain summaries of the appended papers. Chapter 4 contains a summary of the main contributions of the thesis. Further, it discusses some limitations of the methods and theory presented in the thesis and outlines several directions for future work.

Chapter 2

Background

Modeling relationships existent in large sets of data is an extremely powerful tool. This is the main objective in supervised machine learning (ML) where we aim to model a mechanism, h, which produces some desired answer Y, given an input X. These inputs are observed through a collection of samples $\{x_i, y_i\}_{i=1}^n$ which is called the *training set*. From this training set we wish to model the probability distribution $P(Y|X)$ using the data. To measure the models performance, it is subsequently applied to some unseen dataset; which we call a *test set*. This framework is widely applicable to a huge swathe of problems, and has been used to great effect in many fields such as the healthcare sector and several other industrial applications.

We formulate the core problem as a risk minimization where the quantity we seek to minimize is the *expected risk*. The risk denotes how much erroneous behavior we expect our model to have over some data distribution $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$, as measured by a *loss function*, ℓ. The loss function is specified such that it measures discrepancy between the model prediction $h(X)$ and the desired outcome, Y.

The problem we solve in supervised ML can thus be stated as follows

$$\min_{h \in \mathcal{H}} R(h), \quad R(h) := \mathop{\mathbb{E}}_{(X,Y)\in\mathcal{D}} \ell(h(X), Y) \quad . \tag{2.1}$$

In general, the distribution $\mathcal{D}$ is unknown and as such we need to find a way to estimate the risk using known quantities. The general approach to this is to use *empirical risk minimization* (Vapnik, 1991). This is simply to use the samples we have observed, that are drawn from $\mathcal{D}$, and minimize the loss based on these.

$$\min_{h \in \mathcal{H}} \hat{R}(h), \quad \hat{R}(h) := \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i) \tag{2.2}$$

The result of this optimization problem will be a model from the family of models, $\mathcal{H}$, that produces the lowest empirical risk. However, it is not guaranteed that this will be a unique solution or that this is the model which will generalize well on unseen data. To be able to say something regarding this we need to have additional assumptions on the model or the data which we

observe. A fundamental assumption which is central to the success of ML is the assumption that the data points that one is trying to model are similar to the ones that the model will encounter when applied. This assumption states that the data points are independent and identically distributed (i.i.d.) according to the same distribution $\mathcal{D}$ for both the training and test sets. This assumption is in general quite unrealistic as in real-world applications it is quite common for the distribution of data to undergo shifts of some sort. This can come up either through intentional action or inadvertently. Intentionally changing geographical location to apply models, changes in testing protocols or equipment variation are examples of the former. Unintentional shifts may arise due to time passing e.g. seasonal variation or the test population evolving. The kinds of shifts that are possible and the effects that they can have is what we will consider next.

2.1 Distributional shifts

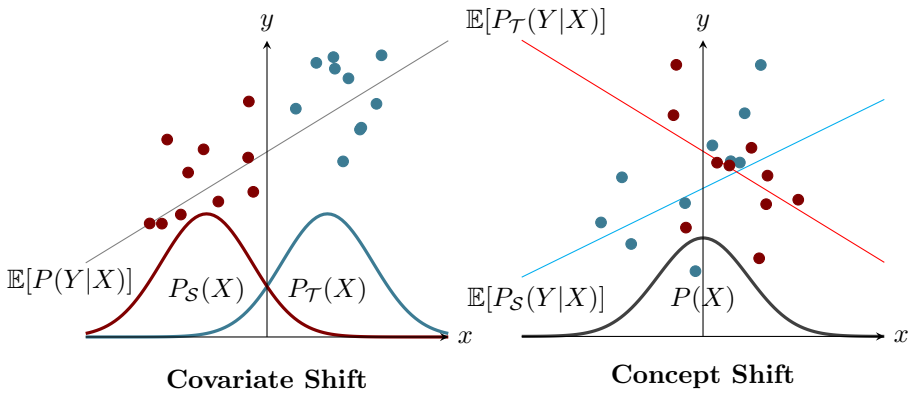


Figure 2.1: Illustration of covariate shift and concept shift when the underlying functions are linear. Label shift and concept drift can be illustrated similarly but with X taken as the dependent variable, instead of Y. Domain shift is when both shifts occur simultaneously.

In cases where the i.i.d. assumption does not hold there is some shift between the distributions that the training set and test set are drawn from. This can be due to several different types of shift. To detail the types of shift that are possible, we should first consider the decomposition of the joint probability distribution,

$$P(X,Y) = P(Y|X)P(X) = P(X|Y)P(Y) \ . \tag{2.3}$$

From this expression we can detail shifts which occur in any combination of the components of the joint distribution between the two distributions. We call the distributions $P_{\mathcal{S}}$ and $P_{\mathcal{T}}$. Some shifts are illustrated in Figure 2.1

1. **Covariate shift**: A shift where the mechanism $P(Y|X)$ is assumed identical across domains and the distribution over features $P(X)$ varies. For example, if our task is to classify animals in images but the training and test sets are taken in different locations. If the background content of images change between the data sets but the images still depict the same types of animals, then there has been covariate shift.

2. **Label shift**: Here we assume that $P(Y)$ can vary but the $P(X|Y)$ is the same across distributions. Consider the example of sales statistics where labels are product categories. Label shift means, in this context, that the proportion of sales across product categories varies between two different retailers and the target, but that the pattern of customers who purchase items in each category ($P(X|Y)$) remain consistent.

3. **Domain shift**: This denotes a more general shift where both the conditional term, $P(Y|X)$, and marginal term, $P(X)$, can vary. Of course this is very challenging and could be impossible to model, absent other relationships between distributions. (Ben-David et al., 2010b;c) This can be the case when we model a problem where both the labeling mechanism and input distributions change. E.g. interpretation of liability from legal texts between countries. The texts may be both written in a different way ($P(X)$ changes), and their interpretation is different depending on countries legal precedent ($P(Y|X)$ changes).

4. **Concept shift**: This refers to when the function which maps inputs to outputs, $P(Y|X)$, change but the input distribution stays the same. For example, in the diagnosis of medical conditions, a set of patient features (e.g. symptoms) may indicate different diseases (labels) in different populations.

5. **Concept drift**: Similar to the shift above, with the change occurring in the mapping $P(X|Y)$ while $P(Y)$ is constant. This can be the due to a shift in appearance of labeled data caused by e.g. seasonal variation. An example is store purchase volumes increasing in some parts of the year than others, which could be driven by (possibly unmeasured) factors such as sales campaigns or stronger consumers due to tax refunds.

As we can see in the list above, there are many different possible shifts in distribution and some are more common than others. The fact remains that to solve the ML problem effectively under a distributional shift we need to consider strategies to deal with the shift. However, we first need to ask the question of how to quantify the existence and extent of a shift.

Measuring differences between distributions is not a perfectly well-defined concept, and many different metrics can be constructed. One may measure discrepancy between distributions in a specific way, and this measure may be more or less useful depending on the data and assumptions that are being made. We will detail some common choices of such metrics next.

2.1.1 Distributional discrepancy metrics

The notion of having a measure of the difference between distributions is not novel. It hails from the statistical problem to discriminate between two populations (Welch, 1939; Brown, 1950), in particular where one makes limited assumptions about their properties. Early works such as the techniques of Fisher (1936) treated this for distributions which are partly known, and more generally using the notion of hypothesis testing for unknown distributions (Neyman & Pearson, 1933).

We can define a specific metric which provides a notion of distance between distributions in many ways. However, they can be sorted into two main categories f-divergences and integral probability metrics. The f-divergences were first introduced in Rényi (1961) and the f-divergence of P from Q can be written as

$$D_f(P \parallel Q) = \int_\Omega f\left(\frac{dP}{dQ}\right) dQ, \tag{2.4}$$

where Ω is some space and f a convex function. An early example of these divergences is that of Kullback & Leibler (1951) which generalized a notion from information theory to form the Kullback-Liebler (KL) divergence. The KL-divergence from a distribution Q to P is defined as

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{x \in \mathcal{X}} \log \frac{P(dx)}{Q(dx)} P(dx), \tag{2.5}$$

for some measurable space $\mathcal{X}$. I.e., it is the expectation of the logarithmic difference between the probabilities P and Q, where the expectation is taken using the probabilities P. One should note that this is not a metric as it is not symmetric, nor is it bounded.

Integral probability metrics (IPMs) is the second type of distances between probability distributions. Initially termed by Müller (1997), IPMs are distances on the space of distributions over a set $\mathcal{X}$ defined by a class $\mathcal{F}$ of real-valued functions on the same set. We define them as

$$D_{\mathcal{F}}(P, Q) = \sup_{f \in \mathcal{F}} \left| \mathbb{E}_{X \sim P} f(X) - \mathbb{E}_{Y \sim Q} f(Y) \right| = \sup_{f \in \mathcal{F}} \left| Pf - Qf \right|. \tag{2.6}$$

A recent example of IPMs is one called the maximum mean discrepancy (MMD). This IPM was used to discriminate based on the moments of the distributions using a kernel function (Gretton et al., 2012). The MMD between two probability distributions P and Q is defined as

$$\mathrm{MMD}(\mathcal{F}, P, Q) = \sup_{f \in \mathcal{F}} (\mathop{\mathbb{E}}_{x \sim Q}[f(x)] - \mathop{\mathbb{E}}_{y \sim P}[f(y)]), \tag{2.7}$$

for some kernel function class $\mathcal{F}$ from a reproducible kernel hilbert space.

Importantly, there is a metric which resides in the intersection between these two classes of metrics, which is the Total variation metric (Jordan, 1881). It is defined on a measurable space $(\Omega, \mathcal{F})$ as

$$\mathrm{TV}(P, Q) = \sup_{A \in \mathcal{F}} |P(A) - Q(A)| \ .$$

I.e., the metric yields the largest absolute difference between the probabilities that the two distributions assign to the same event, A. One can note that the TV metric is quite a pessimistic comparison of distributions. As an example, consider two distributions which agree completely on the probability of all events except one where they disagree with some amount Δ. Compare this to two distributions which do not agree for any events but never more than $\Delta - \epsilon$, which leads to the latter being assigned a lower metric value than the former. This highlights the need to choose metrics for comparison carefully depending upon the application and the notion of "closeness" one wishes to promote. Moreover, it is also salient to consider whether a metric can be accurately estimated from the available data during training. As we will see, this is relevant for achieving guarantees on performance under distribution shift which are not only theoretically correct, but also practically evaluable.

2.2 Unsupervised Domain adaptation

When we expect the i.i.d. assumption to be violated, we need to change our theoretical treatment to account for this. This is the focus in the field of unsupervised domain adaptation (UDA). The training samples that we have access to is assumed have been drawn from an underlying distribution $\mathcal{S}$. This distribution over the product space $\mathcal{X} \times \mathcal{Y}$ is referred to as a *domain*. In UDA, we assume that we have access to $(X, Y) \sim \mathcal{S}$ and $\tilde{X} \sim \mathcal{T}_X$; where $\mathcal{S}$ and $\mathcal{T}$ are called the source and target domains respectively and $\mathcal{T}_X$ is the marginal distribution of features in the target. These quantities will be observed through samples $S = \{x_i, y_i\}_{i=1}^n \sim (\mathcal{S})^n$ and $S'_x = \{\tilde{x}_i\}_{i=1}^m \sim (\mathcal{T}_x)^m$, where $(\mathcal{D})^N$ denotes the distribution of a sample of N datapoints drawn i.i.d. from the domain $\mathcal{D}$. The goal of UDA is to learn a model h, such that it minimizes the risk on data from the target domain. More formally, we write

$$\min_{h \in \mathcal{H}} R_{\mathcal{T}}(h), \quad R_{\mathcal{T}}(h) := \mathbb{E}_{X,Y \sim \mathcal{T}} \left[\ell(h(X), Y) \right]. \tag{2.8}$$

Note that we now have two distributions, $\mathcal{S}$ and $\mathcal{T}$, which are materially different from each other. We also do not observe full samples from the target domain, where we want to estimate model performance, which then precludes the use of ERM as in regular ML. We can think of the distributions as being somehow shifted relative to each other, and understanding this discrepancy is central to estimating $R_{\mathcal{T}}(h)$ well.

The UDA problem was initially considered in natural language processing (Hwa, 1999; Chelba & Acero, 2006; Blitzer et al., 2006) where it was observed to naturally arise. An early theoretical treatment based on maximum entropy models was done by Daumé III & Marcu (2006). The first general treatment is due to Ben-David et al. (2007) where the idea of a discrepancy metric between the source and target domains was introduced. Defining other such metrics has been a focus in several subsequent works in UDA such as e.g. Mansour et al. (2009b); Cortes & Mohri (2014). The proposal of different metrics, such as those detailed in 2.1.1, is motivated by the implicit assumptions made by their use and the fact that there are many ways to construct them.

In summary, the DA framework is a general framework which can describe the process of learning under distribution shift. DA can also lend itself fairly well to variations of different kinds; extending the setting to situations with varying amounts of information available. We will detail some of these extensions next.

2.2.1 Multiple source domain adaptation and federated learning

A natural extension of the UDA setting is to include the possibility of several different source domains. The motivation for such an approach would be that any one source domain would have a lesser probability of accurately representing the target domain. As such, the use of multiple domains would more likely lead to a higher likelihood of the target being covered in a distributional sense. This setting is called multi-source domain adaptation (MSDA) (Mansour et al., 2008; Ben-David et al., 2010a) and occurs in many real-world situations such as sentiment analysis and image classification where data originate from several different sources. Formally, MSDA assumes access to labeled data from n source domains and unlabeled data from one target domain.

In contrast to regular UDA we need to reconcile discrepancies not only between the sources and the target but also among the source domains themselves. This is commonly tackled with representational alignment of features, either by discrepancy based methods (Hoffman et al., 2018; Peng et al., 2019; Guo et al., 2018) or adversarial methods (Xu et al., 2018; Zhao et al., 2020). This type of problem generally require the use of large datasets to achieve high performance. However, to collect and store such datasets may be challenging for many reasons, privacy being a central one. Increased regulation of personal data collection as well as changing user preferences make privacy-preserving methods a valuable direction of study. In the machine learning context this has taken the form of the study of differential privacy (Dwork et al., 2006) and federated learning (McMahan et al., 2017) (FL) methods. Of course, there have been many works studying distributed machine learning before FL was introduced by McMahan et al. (2017). However, we will not provide a comprehensive survey of them in this thesis. Federated Learning is a decentralized machine learning paradigm where model training occurs across multiple clients (e.g., mobile devices, hospitals, organizations), each holding local data that remains private. I.e., the data is not transmitted outside the client to preserve the clients privacy but instead model parameters θ_i are transmitted. See Figure 2.2 for a schematic overview. A central server orchestrates the training process, which we will detail next.

FL is generally formulated as follows: assume that there are K clients, each with a dataset, $\mathcal{D}_i = \{x_k^i, y_k^i\}_{k=1}^{n_k}$, which only they have access to. For each client which takes part in the federation, they solve a local optimization problem by minimizing their risk locally

$$\min_{h \in \mathcal{H}} R_i(\theta_i), \quad R_i(\theta_i) = \mathbb{E}_{D_i}[\ell(h_i(\theta_i; x), y)] \ , \tag{2.9}$$

where θ_i are the local model parameters.

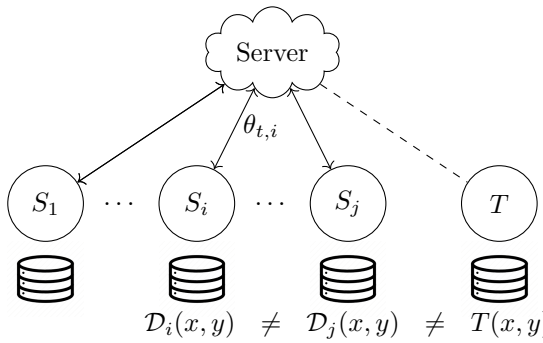


Figure 2.2: Schematic overview of federated learning (FL). Note the likeness with MSDA as each client can be viewed as its own source domain S_i where the datasets $\mathcal{D}_i$ are not necessarily distrubuted equally. The key difference between the two settings being the decentralized nature of FL, where only model parameters are shared with a central, coordinating server.

The larger objective of FL is then to train a single global model that minimizes the risk across all clients, i.e. minimizing

$$R_{fed} = \frac{1}{K} \sum_{i=1}^{K} R_i(\theta_i) \ . \tag{2.10}$$

One of the most widely used algorithms for solving the FL problem in (2.10) is federated averaging (FedAvg) introduced by McMahan et al. (2017). This algorithm first lets the clients take one or several local steps of stochastic gradient descent solving (2.9). Then the clients update their local models which are then transmitted to a central server. This server then aggregates the model parameters to create a global model. The aggregation is done by performing a weighted average of the local models, which is applied layer-wise. The weights in the aggregation is based on the sizes of the local datasets, which for some timestep t is calculated as

$$\theta^t = \sum_{i=1}^{M} \frac{n_i}{n} \theta_i^t \ , \tag{2.11}$$

where n is the total amount of samples in all clients and M is the amount of clients participating in the last round of updates. This is not necessarily equal to K as we allow for some fraction of clients to be inactive. This is motivated by applications with a very large amount of clients, where it is unlikely that all clients will participate at all times. We present an algorithmic overview of FedAvg in Algorithm 1.

The connection between MSDA and FL arises when we interpret the source domains in MSDA as clients in a federated system. In this view, each client represents a distinct data distribution (i.e., a source domain), and the system seeks to train a model that generalizes well to a new, unseen distribution (i.e.,

Algorithm 1: Federated Averaging

Data: Client set $\mathcal{S}$, learning rate η and local datasets D_i indices $\{I_k\}$
Server executes:
Initialize central parameters θ^0
for *each round* $t = 0, ..., T-1$ **do**
  Sample M clients from $\mathcal{S}$
  for *each client* $k = 1, ..., M$ **do**
    Distribute θ^t to client i
    Receive client update $\theta_i^{t+1} = \text{Client\_Update}(i, \theta_t)$
  end
  $\theta^{t+1} = \sum_{i=1}^{M} \theta_i^t \frac{n_i}{n}$ where $n = \sum_{i=1}^{M} n_i$
end
return $h(\theta^{t+1})$

Function `Client_Update(`i, θ`):`
  for *local step* $j = 1, ..., E$ **do**
    $\theta = \theta - \eta \nabla \ell(h(\theta; x), y)$ for $x, y \sim D_i$
  end
  return θ

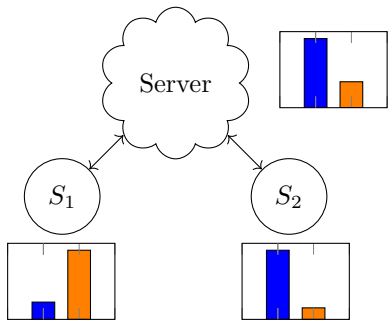


Figure 2.3: A simple example of client heterogeneity in terms of the label set distributions in clients being materially shifted w.r.t each other. Note that the aggregate of clients does not necessarily equal the server side test set distribution.

a target domain). Naturally, heterogeneity can arise in the FL clients due to distribution shifts between them, this is what we will treat next.

2.2.2 Heterogeneity in federated learning

As mentioned above, issues may arise with non-i.i.d. data across domains/clients in FL. When the client datasets differ significantly in distribution, we should not expect averaging of parameter to produce a better classifier than the individual models. Furthermore, in more adverse cases of heterogeneity, the training may

converge slower or the resulting classifier may perform worse than it would have in an i.i.d. setting. This is related to the fact that there often is an assumption that the target is some aggregate of the clients which in general may not be the case. Consider the example in Figure 2.3, where we have two clients with different label distributions over their respective datasets. Moreover, the aggregate of these clients does not conform to the distribution which is observed in the test set on the central server. This can prove challenging as the models may have differing notions of how to discriminate between the two classes and it is not clear whether aggregating the two would have a beneficial effect, especially if the aggregate of the clients is not identical to the test set. This would then lead to the problems we highlighted earlier, e.g. slow convergence or suboptimal performance.

While the problem remains open generally, there have been approaches which consider alternative objectives, e.g maximizing worst-group performance (Mohri et al., 2019). Other efforts to mitigate the effects of distributional shifts in federated learning can generally be categorized into client-side and server-side approaches. Client-side methods use techniques such as regularization techniques that penalize large deviations in client updates (Li et al., 2020; 2021), client clustering (Ghosh et al., 2020; Sattler et al., 2020; Vardhan et al., 2024) which makes models for each client cluster and meta-learning (Chen et al., 2018; Jiang et al., 2019; Fallah et al., 2020). Server-side methods focus on improving model aggregation or adjusting post-aggregation. These include optimizing aggregation weights (Reddi et al., 2021), increasing gradient diversity during updates (Zeng et al., 2023), learning adaptive aggregation weights (Li et al., 2023) and using iterative moving averages to refine the global model (Zhou et al., 2023). Another related area is personalized federated learning, which focuses on fine-tuning models to optimize performance on each client's specific local data (Collins et al., 2022; Boroujeni et al., 2024; Mclaughlin & Su, 2024). This setting, while interesting, will not be a focus in this thesis.

2.3 Guarantees and generalization bounds

In many settings, ensuring good performance of a model is critical to successful deployment. High-stakes settings have this attribute, e.g. autonomous driving and making treatment decision in healthcare settings. If the aim is to guarantee a specific level of model performance we need a bound on the target risk as specified in the previous section. Simply showing acceptable performance on held-out datasets is not a guarantee that the performance will not degrade when applied in other settings. Such a degradation in performance has been observed in e.g. the healthcare setting (Zech et al., 2018), and natural language processing (Jia & Liang, 2017).

As the quantity in (2.8) is written as an expectation over the a priori unknown distribution $\mathcal{T}$ we will have to estimate the risk somehow. Therefore, we can substitute the expectation by computing an approximation of this using

the sample average which we write as

$$\hat{R}_{\mathcal{T}}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(h(\tilde{x}_i), \tilde{y}_i) \tag{2.12}$$

for some sample $\{\tilde{x}_i, \tilde{y}_i\}_{i=1}^{n} \sim (\mathcal{T})^n$. However, as we assume that we do not have access to target labels we have to estimate the target risk with something that we actually can calculate. Therefore, to deal with this complication we use theory to connect the expected target risk to the expected source risk.

Further, we assumed that the distributions of the source and target data were different from each other. Therefore, we need to account for the discrepancy between the two distributions. This can be done in several different ways; to illustrate, we show a simple example of what we want to achieve. We want some way to bound the target risk with quantities which we have some hope of estimating with the available information, i.e. we want to find a bound on the following form:

$$R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + \mathcal{D}(\mathcal{S}, \mathcal{T}) \ .$$

The last term, $\mathcal{D}(\mathcal{S}, \mathcal{T})$, is measuring some distance/discrepancy between the source and target domains. We call this discrepancy the *domain shift* term which figures in some form in all domain adaptation bounds. Of course, we have to ensure that the domain shift term does not depend on knowledge of the target distribution.

After this step, we wish to bound the expected source risk with a sample average like in (2.12). However, we still need to account for the error between the expected and empirical risk. This means that a sample generalization term must be added, i.e. a term accounting for the error which we see due to our sample size being limited. One approach to estimating the sample generalization error of a classifier is to use statistical learning theory which we will detail in the next section.

Thus, if we have the tools to both relate the source risk to the target risk and connect quantities in expectation to their empirical counterparts, we can express generalization bounds on the following form:

$R_{\mathcal{T}} \leq f(\text{Empirical source risk, Domain shift, Sample generalization error}).$

The specific form of f and the terms it depends on is decided by the theoretical approach taken to the steps detailed above. However, the two main forms are whether the domain shift and sample generalization terms are related through addition or multiplication. We will now go into the main theoretical tools used to solve the two challenges detailed above.

2.3.1 The PAC learning framework

The most prevalent theoretical framework for reasoning about the generalization performance of non-deterministic statistical models is statistical learning theory, more specifically, a theory called Probably Approximately Correct (PAC)

learning (Valiant, 1984). This theory allows us to through assumptions on the model, task and data show that a certain task is learnable, specifically as understood through the PAC lens.

This means is that for a certain task it can be shown to be PAC learnable if we can show that given an algorithm $\mathcal{A}$ and a sample of size n, the algorithm $\mathcal{A}$ returns a model from the model class, $\mathcal{H}$, which has a small average error, ϵ, with high probability, $1-\delta$. This then amounts to that we can show that the risk for a specific model on the given data is smaller than some value $\epsilon > 0$ with confidence level $1-\delta$, where $\delta < 1$, or more formally,

$$\Pr[\mathbb{E}[\ell(h(x), y)] \leq \epsilon] \geq 1-\delta, \ \forall h \in \mathcal{H}. \tag{2.13}$$

With a formulation on this form we can then use some well known results, often based on concentration inequalities, such as e.g. Hoeffding's inequality to move from an expectation form to an empirical form. This is due to the inequality providing an upper bound on the probability that the loss deviates from its expected value by more than a certain amount. Using these kinds of techniques we can, using application of standard theory (Vapnik, 1998), get bounds like the following. For an i.i.d. sample of size m we have that the following holds with probability at least $1-\delta$ for every $h \in \mathcal{H}$:

$$R_{\mathcal{S}}(h) \leq \hat{R}_{\mathcal{S}}(h) + \sqrt{\frac{4}{m}\left(d \log \frac{2em}{d} + \log \frac{4}{\delta}\right)}. \tag{2.14}$$

The quantity d in the above expression is the so-called Vapnik-Chervonenkis (VC) dimension. The VC dimension is a measure of how complex the family of functions, $\mathcal{H}$, are. An early example of a bound on the target risk that is achieved using this framework is the following one from Ben-David et al. (2007)

$$R_{\mathcal{T}}(h) \leq \underbrace{\hat{R}_{\mathcal{S}}(h)}_{\substack{\text{Empirical} \\ \text{risk}}} + \underbrace{\sqrt{\frac{4(d \log \frac{2em}{d} + \log \frac{4}{\delta})}{m}}}_{\text{Sample generalization}} + \underbrace{d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda}_{\text{Domain shift}}, \tag{2.15}$$

where d is the VC dimension of the $\mathcal{H}$, λ is the sum of the errors on both domains of the best performing classifier $h^* = \arg\min_{h \in \mathcal{H}}(R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h))$, and $d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) = 2 \sup_{A \in \{\{x : h(x) = 1\} : h \in \mathcal{H}\}} |\Pr_{\mathcal{S}}[A] - \Pr_{\mathcal{T}}[A]|$ is the $\mathcal{A}$-distance for the characteristic sets of hypotheses in $\mathcal{H}$. There are many subsequent works which use this approach also use the PAC style with different choices of discrepancy terms and methods of bounding sample generalization. (Cortes et al., 2015; Acuna et al., 2021; Liu et al., 2025) Using the PAC approach we get a bound which holds uniformly over the class of hypotheses $\mathcal{H}$. This is one of the features of PAC learning, the bounds hold for all classifiers in the considered class. However, this can also be a weakness as this produces bounds which must, by definition, hold for the worst classifier imaginable from the class. Depending on the richness of the class this can be arbitrarily limiting. In response to this issue, there is an extension to the PAC framework which does not suffer the same fate which we will detail next.

2.3.2 The PAC-Bayes framework

PAC-Bayes theory is an extension of PAC theory based on using the PAC framework to understand Bayesian classifiers. This way of analyzing classifiers was initially proposed by Shawe-Taylor & Williamson (1997), with the first generalization guarantee being proved by McAllester (1998). The framework studies generalization of a posterior distribution ρ over hypotheses in $\mathcal{H}$, learned from data, in the context of a prior distribution over hypotheses, π. The generalization error in ρ may be bounded using a divergence between ρ and π as seen in the following classical result due to McAllester (2013).

For a prior π and posterior ρ on $\mathcal{H}$, a bounded loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ and any fixed $\gamma, \delta \in (0, 1)$, we have w.p. at least $1 - \delta$ over the draw of m samples from $\mathcal{D}$, with $\mathrm{KL}(p\|q)$ denoting the Kullback-Liebler (KL) divergence between p and q,

$$\mathop{\mathbb{E}}_{h\sim\rho} R_{\mathcal{D}}(h) \leq \frac{1}{\gamma} \mathop{\mathbb{E}}_{h\sim\rho} \hat{R}_{\mathcal{D}}(h) + \frac{\mathrm{KL}(\rho\|\pi) + \ln(\frac{1}{\delta})}{2\gamma(1-\gamma)m} \ . \tag{2.16}$$

As we can see in (2.16), we now have an expression which is stated as an expectation over the posterior distribution ρ. This is in addition to the expectation over the distribution of the data. Thus the bound holds, on average, for classifiers drawn from the posterior ρ. We can interpret ρ as a distribution over the parameters of our trained classifier. The posterior would then be a distribution around the classifier, effectively covering those classifiers which are close in parameter space to our trained classifier. A prominent feature of the framework then, is that we restrict our prediction to a smaller part of the model space. The additional complexity incurred by this is by adding an expectation over the distribution ρ, i.e, the models which are close to the learned classifier. We consider an additional expectation to be an expense, as it increases computational complexity to a prospective evaluation of the bound. There are some key things to note with this formulation that are advantageous if we want to estimate the quantities in the bound. First, the shape of the prior and posterior distributions are not explicitly stated and can be chosen at will; no matter the choice, the bound will still hold. Further, the posterior ρ is something which we learn from the training data. We will next detail another way in which the PAC-Bayes formulation is preferable when the aim is to achieve tighter bounds.

2.3.3 Data-dependent priors in PAC-Bayes

The sample complexity term in (2.16), based on the KL-divergence, grows as the prior π and posterior ρ become more dissimilar. This may happen if the posterior is very sensitive to the training data or the prior is poorly chosen. The posterior should be sensitive to the data in some respect, otherwise the training of it would be ineffective. To prevent a radical divergence between π and ρ we can inform our choice of prior with some of the data we have available. This is called a data-dependent prior and was developed by the work of Ambroladze

et al. (2007) and Parrado-Hernández et al. (2012), with an extension to neural networks by Dziugaite et al. (2021).

When we use this type of prior, given that enough data has been used to inform the prior, we will observe a tightening of the resulting bound. This will be due to the KL term being smaller since the prior and posterior are now closer to each other. It is important to note that any data which is used to learn a prior must be independent of the data used to evaluate the bound. If this is not ensured the bound will not hold. However, we should also note that this restriction does not affect which data is used to learn the posterior, ρ.

2.3.4 Bounds in MSDA and federated learning

The theoretical frameworks discussed previously can be used to analyze both MSDA and FL as well. For instance, Crammer et al. (2008) and Mansour et al. (2009a) both contribute theoretical PAC-style bounds based on Rademacher complexity and Renyi Divergence respectively. More recent approaches to bounding error in MSDA from Shui et al. (2021) and Chen & Marchand (2023) make use of some labeled target data to achieve guarantees on performance. The main difference between the bounds which we obtain in regular DA, and those in MSDA is that there usually is some term which relates the target risk to some combination of the source risks and also potentially the pairwise discrepancy between sources or between some combination of sources and target.

In federated learning there are similar lines of work which show bounds on performance. The communication constraints inherent to the setting does complicate matters, and treatments which take heterogeneity of clients into account is rare. The works of Mohri et al. (2019), Fallah et al. (2021), Chen et al. (2023) and Boroujeni et al. (2025) all present bounds in the FL setting where heterogeneity is considered. However, they all make different assumptions on the boundedness of the loss, convexity of the problem or on the level of client-wise heterogeneity. This highlights the complex nature of constructing theory in such a difficult and constrained setting, as well as the importance of assumptions. The impacts, trade-offs and limitations inherent to making assumptions on the tasks we want to solve is what we will discuss next.

2.4 Assumptions and their impact

Current theoretical approaches have produced general guarantees for performance, but these are limited in their ability to provide realistic guarantees on UDA and similar problems. It is known that general adaptation is impossible and thus the question then becomes: What do we need to assume to guarantee consistent learning?

Consistent learning implies that our model will learn to solve the task at hand in the limit of infinite samples and that we will do so every time. That is, as the sample size n increases the estimates converge in probability to the value that the estimator is designed to estimate. To ensure this, a large

swathe of works make assumptions that are quite similar. First, one makes some assumption regarding a conditional probability distribution related to either One can assume that the underlying function which generates outcomes is the same between the domains. This is called the covariate shift assumption (Shimodaira, 2000), meaning that the data is allowed to change, but not the function labeling the data. We write this as follows:

Assumption 1 (Covariate shift)**.** *For domains $\mathcal{S}, \mathcal{T}$ on $\mathcal{X} \times \mathcal{Y}$, we say that* covariate shift *holds with respect to $X \in supp(\mathcal{S}, \mathcal{T})$ if*

$$\exists x : \mathcal{T}(x) \neq \mathcal{S}(x) \;\; and \;\; \forall x : \mathcal{T}(Y \mid x) = \mathcal{S}(Y \mid x) \;\; ,$$

where $\text{supp}(\mathcal{S}, \mathcal{T})$ is the common support of the domains. This assumption is often made and can hold in many different settings, we often have little reason to believe that the underlying labeling function will change just because the domain has done so.

Another similar approach is what we call the *label shift* assumption. (Saerens et al., 2002; Lipton et al., 2018) This is similar in nature to covariate shift, but instead focused on the conditional $\mathcal{D}(X \mid Y)$. So here, in contrast to the former, the distribution of inputs given labels is the same across domains. It can be described as follows:

Assumption 2 (Label shift)**.** *For domains $\mathcal{S}, \mathcal{T}$ on $\mathcal{X} \times \mathcal{Y}$, we say that* label shift *holds with respect to $Y \in supp(\mathcal{S}, \mathcal{T})$ if*

$$\exists y : \mathcal{T}(y) \neq \mathcal{S}(y) \;\; and \;\; \forall x : \mathcal{T}(X \mid y) = \mathcal{S}(X \mid y) \;\; .$$

One might be tempted to think that any of the two assumptions above would be enough, however, this is unfortunately not the case. As shown in Ben-David et al. (2010b) we also need a assumption of coverage of the target domain to have a guarantee of consistent learning. Therefore, we also need the overlapping support assumption to be able to guarantee consistent learning.

Assumption 3 (Domain overlap)**.** *A domain $\mathcal{S}$ overlaps another domain $\mathcal{T}$ with respect to a variable Z on $\mathcal{Z}$ if*

$$\forall z \in \mathcal{Z} : \mathcal{T}(Z = z) > 0 \implies \mathcal{S}(Z = z) > 0 \;\; .$$

This assumption states that if some datapoint is possible to observe in the target domain we also have a non-zero probability to observe it in the source domain. As should be evident, this is quite a strong assumption that may not hold in realistic scenarios. We illustrate this phenomenon in figure 2.4.

To exemplify how assumptions result in limitations on theory we will present some examples from the literature. We start with the following bound due to Cortes et al. (2010)

$$R_{\mathcal{T}} \leq \hat{R}^{w}_{\mathcal{S}} + 2^{5/4} \sqrt{d_2(\mathcal{T} \| \mathcal{S})} \sqrt[3/8]{\frac{d \log \frac{2ne}{d} + \log \frac{4}{\delta}}{n}}.$$

This bound is an example of an importance weighting bound which bounds the target risk using a weighted empirical source risk, $\hat{R}^{w}_{\mathcal{S}}$. In this term we re-weight

the loss function according to the density ratio $w(x) = \frac{\mathcal{T}(x)}{\mathcal{S}(x)}$ of each sample. For this style of bound we run into issues when the overlap assumption does not hold. Consider the density ratio above; if there is a lack of overlap we may have a data point which only has non-zero density in the target domain. This leads to a division by zero in w and the bound immediately becomes vacuous. The issue is that it is very simple to violate overlap in practice, e.g. learning from black and white images and applying to color images. This inability to handle the non-overlapping case is a weakness we would like to avoid.

So we might come to the conclusion that we should avoid the importance weighting type bounds but still keep the assumptions. This often yields something akin to the bound we stated in (2.15). We will state a similar bound here due to Ben-David et al. (2010a):

$$R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + 4\sqrt{\frac{2(d \log 2m + \log \frac{2}{\delta})}{m}} + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda, \tag{2.17}$$

where

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) = 2\left(1 - \min_{h,h' \in \mathcal{H}} \left[\frac{1}{m} \sum_{\substack{x \sim (\mathcal{T}_X)^m: \\ h(x) \neq h'(x)}} \mathbb{1}[x] - \frac{1}{m} \sum_{\substack{x \sim (\mathcal{S}_X)^m: \\ h(x) \neq h'(x)}} \mathbb{1}[x]\right]\right).$$

This bound has some qualities that we might take issue with; these are mainly related to the way domain shift is measured. First, the $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ term measures the discrepancy between how much two distinct hypotheses will disagree on the source and target. Intuitively this accounts for the difference between the source and target, but this quantity is not easy to calculate as it requires a minimization over the hypothesis class, $\mathcal{H}$. This is difficult to evaluate as the class can be very large, which is the case for neural network classifiers. This type of of quantity figures in many other works. (Ben-David et al., 2007; Blitzer et al., 2008; Ben-David et al., 2010a; Morvant et al., 2012; Mansour et al., 2009b; Redko et al., 2019; Cortes & Mohri, 2014; Cortes et al., 2015; Yin et al., 2024; Koc et al., 2025).

Second, the λ term, which accounts for the joint optimal error of the best classifier, is not possible to observe with the data available. So if we want a bound that is tight we have to assume that this quantity is small. This non-observable quantity or ones like it is quite common in the literature. (Kuroki et al., 2019; Redko, 2015; Long et al., 2015; Redko et al., 2017; Johansson et al., 2019; Zhang et al., 2019; Dhouib et al., 2020; Shen et al., 2018; Courty et al., 2017; Germain et al., 2013; Dhouib & Redko, 2018; Acuna et al., 2021; Nguyen et al., 2022; Huang et al., 2025; Koc et al., 2025; Liu et al., 2025) As we have seen in this section, the limitations of the current literature are not insubstantial. We can therefore see that there is a need to develop theoretical results which do not suffer from these limitations.

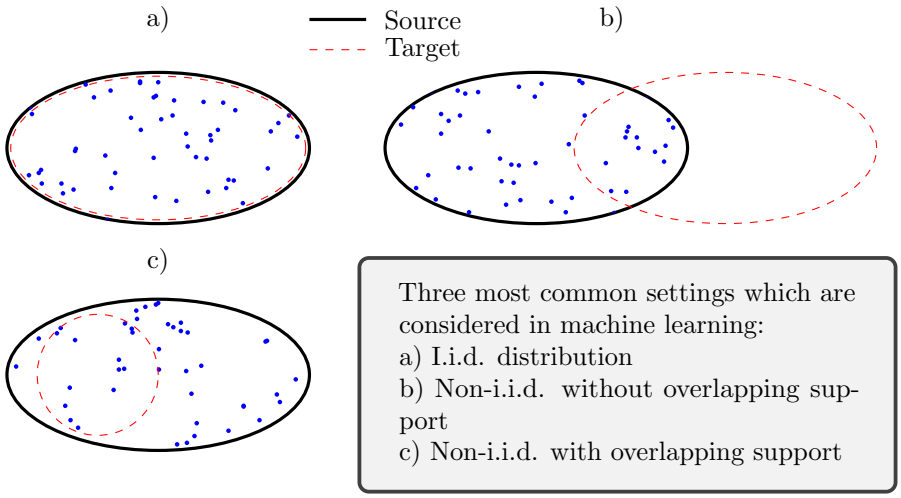


Figure 2.4: Illustration of different settings including shifts in distribution and how this relates to overlapping support of the distributions.

2.5 Privileged information

Learning using privileged information (LUPI) is a framework that was first introduced by Vapnik & Vashist (2009). In this setting we assume we have access to some additional information when training our model. This auxiliary data is in addition to the features and labels available in regular supervised learning. Furthermore, a key assumption is that we only have access to this information at the training stage and not when performing inference.

This additional information is called *privileged information* (PI) and can be many different things: medical journals, object segmentations, bounding boxes etc. The main goal in this framework is to accelerate the pace of learning. The motivation for why this would be achieved is that in real-world learning we often have students being taught by a teacher. This teacher has better knowledge about what material, and how it should be presented to the students in order for them to learn the concepts faster. This may be specific explanations or interjections made to facilitate the students learning. It is this process that the PI framework seeks to imitate. So, using this analogy further there would be some data (x, y) generated by the world and the privileged information would then be generated by the teacher using the conditional distribution $P(w|x)$ which is assumed to be unknown. A related field is that of knowledge distillation (Hinton et al., 2015) which has been noted to be compatible with PI (Lopez-Paz et al., 2016).

Formally, we assume the existence of some PI, $W \in \mathcal{W}$, which is related to the data through $P(w|x)$. Thus the problem is the following: Given tuples $\{(x_i, w_i, y_i)\}_{i=1}^N$ we seek to learn a function f which predicts the outcomes y_i given the data x_i. In contrast to regular supervised learning, we only have access to the PI w_i during training. Since we do not assume that we have access

to this data at test time, the resulting model, f, cannot explicitly depend on the PI, w, as an input. Note, we do not assume any specific form or other properties of the PI. Since explicit model dependence on privileged information is prohibited, one approach is that PI is used in the loss function to learn the model. (Vu et al., 2019) Thereby guiding the optimization based on the information available in the PI. Another is to construct a particular architecture which makes use of the PI. (Xie et al., 2020)

The LUPI paradigm have been applied in settings such as image recognition (Vu et al., 2019; Hoffman et al., 2016), healthcare (Shaikh et al., 2020), finance (Silva et al., 2010), clustering (Feyereisl & Aickelin, 2012). In the context of distribution shifts we have approaches which study the use of PI with SVMs (Li et al., 2022; Sarafianos et al., 2017). Vu et al. (2019) considered using scene depth as PI in semantic segmentation using neural networks. Regarding theory in the DA setting, Xie et al. (2020) provide some theoretical results for linear classifiers and Motiian (2019) investigated using PI for DA using the information bottleneck method.

Chapter 3

Summary of Included Papers

The general problem of adaptation under distributional shift is impossible to guarantee a solution for. (Ben-David & Urner, 2012) However, with the use of assumptions on problem structure it is possible to successfully adapt to such shifts. In this thesis, we search for ways in which we might achieve successful domain adaptation while *still* having useful guarantees on performance. Both of these goals are attainable in some form for the neural network model class, but seldom together. We have shown in section 2.4 that there are limitations of the current theory that inhibit us from achieving this goal at present. Our first paper deals with illustrating the issue with achieving tractably computable and tight generalization bounds for neural network classifiers. One can often find a neural network classifier that performs well on a UDA task, however, there are no realistic guarantees on performance. The second paper focuses on the second and third questions from Chapter 1 and considers practical ways to use privileged information to achieve consistent learning in DA where such information is available. As part of this we propose a novel set of assumptions which we show lead to consistent learning and extensively empirically validate this. In paper III we consider the federated learning setting with heterogeneous clients, where we have a fixed target domain. We show that allowing the server access to label marginals from clients as well as the intended target leads to large potential gains in performance. This performance comes at the cost of some privacy as clients are required to share this information with the central server. Paper IV considers FL with increased privacy characteristics, where we assume that the individual label sets of clients are considered sensitive. In this setting, which we call the private label set setting, we show a way to adapt common FL methods to this setting. We also present a method of tuning the global model for representational alignment, which we show some theoretical guarantees for.

3.1 Paper I - Practicality of generalization guarantees for unsupervised domain adaptation with neural networks

In high-stakes scenarios, like the healthcare setting, we would ideally want to have some guarantees on how well our models are going to perform before deployment. The most straightforward way of achieving this is by upper bounding the error on the target domain. This can be achieved theoretically in many different ways with varying degrees of utility. We can trivially state that the risk is less than or equal to 1 for any task, you cannot be more wrong than all of the time. However, this bound is not particularly informative so we would like to find better bounds which give us further insights into the expected performance of models. Ideally, we would like to have guarantees which are tightly linked with the performance we observe in application while also being possible to evaluate a priori. In Paper I we search the domain adaptation literature for existing bounds which have three main properties:

1. Tightness – Is the bound a poor approximation of the true risk?
2. Estimability – Can we estimate the bound from the observed data?
3. Computability – Can we tractably compute the bound for real-world data sets and hypothesis classes?

After conducting an extensive search we arrive at the conclusion that most available bounds do not fulfill these desiderata. Our final selection contains three types of bounds: importance weighting (IW) bounds, bounds containing integral probability metrics (IPMs) and PAC-Bayesian bounds. To enable easier comparison we adapt the IW and IPM methods to the PAC-Bayesian framework, thereby creating two novel corollaries to a theorem due to McAllester (2013). These bounds, along with two existing ones due to Germain et al. (2020), are the ones we choose to computationally evaluate. One of them requires access to target labels to compute and is included for comparison.

We find that without further modification our evaluation results in vacuous bounds due to the sample generalization terms being too large. To remedy this we apply the practice of learning data-dependent priors which entails sacrificing a part of the sample to inform the choice of prior. This tightens the bounds as the sample generalization term, which measures the difference between the prior and posterior distributions, is smaller using this.

We then compute the four different bounds for two image classification tasks, one based on digit classification and one based on X-ray classification. We find that the bound which requires target labels is the tightest, followed by our IW bound which is computable without such information. The other bounds struggle to remain tight even for the simpler digit classification task. We conclude that in cases where our assumptions hold, an importance weighting strategy works well for bounding the error tightly. Further, we conjecture that changing current assumptions will be a way towards a more complete theory explaining out-of-distribution generalization.

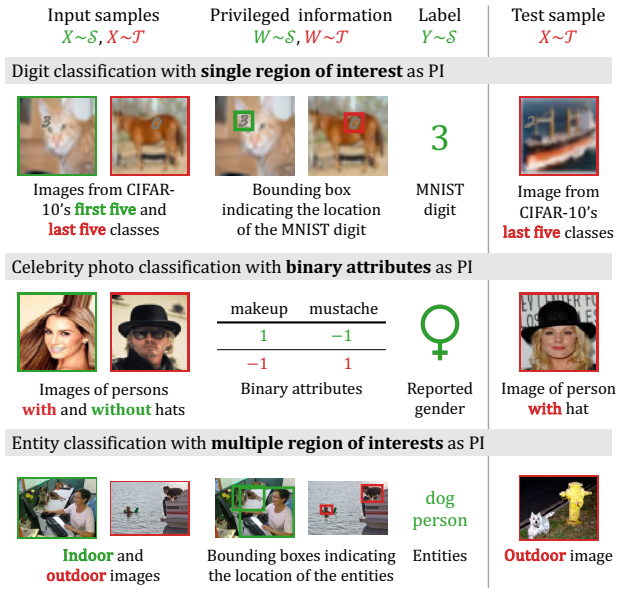
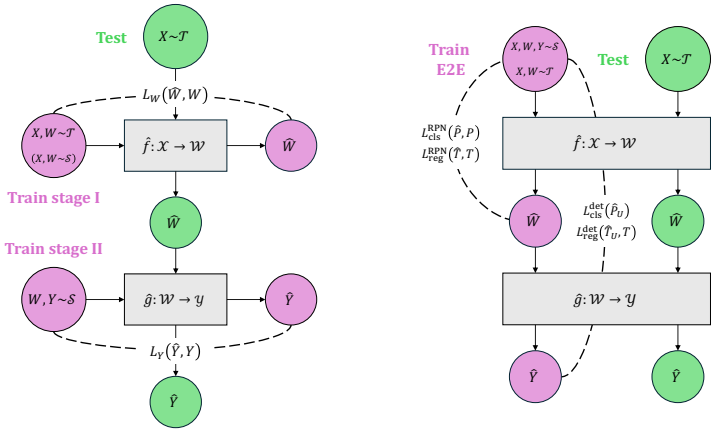


Figure 3.1: An illustration of the data available in the domain adaptation by learning using privileged information (DALUPI) setting. During training, input samples X and privileged information W are available from both source and target domains. Labels Y are only available for inputs from the source domain. At test time, a target sample X is observed.

3.2 Paper II - Unsupervised domain adaptation by learning using privileged information

In this paper we propose some changes to one of the standard sets of assumptions commonly used in UDA, which we do by using the concept of privileged information (PI). We put forward a version of UDA where we assume access to some privileged information, where the PI is assumed to be available in both the source and target domains during training time. Furthermore, at test time, we only assume that we have access to target features as in regular UDA. We call this setup Domain Adaptation by Learning Using Privileged Information (DALUPI). The general structure of our setting is illustrated in 3.1 with some examples. We construct theory based on this novel structure which ensures consistent learning without the reliance on the overlapping support assumption in the input space. The overlap assumption is often violated in practice and as such we would prefer not to build UDA theory using it. The DALUPI setting enables a novel and straightforward way of transferring the model from the source to the target. We simply learn two separate mappings, one from input features to PI and one from PI to the label or outcome. This also gives us a simple way to make theory which conforms to this structure, we just learn one mapping after the other. To avoid the overlap assumption in the input



(a) Two-stage estimator. (b) End-to-end estimator.

Figure 3.2: A schematic representation of the train and test flow for DALUPI using (a) the two-stage estimator and (b) an end-to-end architecture based on Faster R-CNN.

space we instead assume overlap w.r.t the PI, which we argue is more plausible. Additionally, we assume covariate shift w.r.t the PI similar to what is used in other assumption sets. If we add the additional assumption that PI is sufficient for predicting the outcome we show that will have consistent learning. We also propose a bound on the target risk for this setting.

We conduct experiments on four different tasks; a synthetic experiment where we investigate how well a model, which is the composition of two separate mappings learned independently, performs when the amount of overlap is varied. The dataset is constructed by inserting a digit in a larger image and having the bounding box around the digit as PI. It shows that a model based on our framework outperforms all other models. We also adapt an experiment from another work (Xie et al., 2021), based on the CelebA dataset, to compare our method to theirs. Upon comparison, We find that we achieve similar performance to the method proposed by Xie et al. (2021).We then perform two other experiments, one entity classification task based on the MS-COCO dataset and pathology classification from chest X-rays. For these experiments we also propose an end-to-end model, based on the Faster R-CNN architecture (Ren et al., 2015). A schematic overview of the two model architectures we use are presented in Figure 3.2.

In all experiments, except the one based on CelebA, the PI we consider are bounding boxes around the region(s) which are informative for the labeling. In the CelebA experiment, the PI used is a vector of binary appearance attributes present in the pictures. See Figure 3.1 for clarification of the settings we consider.

From the entity classification task we learn that our method outperforms both the UDA baseline well as performs on par with the model which has been given access to target labels. From the X-ray classification task we learn

that the use of PI can yield increased sample efficiency, in-line with previous observations. However, the sufficiency of the PI in this task is not obvious, nor guaranteed. We conclude therefore that the DALUPI setting can be beneficial, even when our assumptions are unlikely to hold. In addition, we note that a domain expert will likely be needed to be able to judge the sufficiency of PI for tasks like the X-ray classification task we considered.

3.3 Paper III - Overcoming label shift in target-aware federated learning

Distributional shifts are not only a cause for concern in centralized settings such as regular DA. Many of the same issues arise in federated learning where we consider a collaborative effort between several clients which contribute to the production of a global model. This training is administrated by a central server which aggregates the model weights which are transmitted from each client to construct the global model. When the clients datasets differ in some way between them we may need to account for this. Furthermore, it is not generally guaranteed that the aggregate of clients is equal to the test data distribution.

In this paper we aim to find a method to remedy this when the clients label distributions differ, but the labels conditioned on inputs is static, i.e . we assume label shift. In this setting there are methods to account for such a shift, however, these do not consider the case that the test domain may not be i.i.d. to the aggregate of all clients. This is an implicit assumption made in most works which propose to mitigate label distribution heterogeneity between clients; we will not make this assumption. Instead, we will consider a target-aware setting where the central server has access to label marginals from the clients as well as the fixed target domain. If we allowed for this information to be shared across clients one could make use of an importance weighting strategy, however, this is not feasible here due to the client's privacy.

We propose to use this information to aggregate client parameters according to the solution of the following simple optimization problem

$$\alpha^{\lambda} = \underset{\alpha \in \Delta^{M-1}}{\arg\min} \|T(Y) - \sum_{i=1}^{M} \alpha_i S_i(Y)\|_2^2 + \lambda \sum_i \frac{\alpha_i^2}{n_i} \ , \tag{3.1}$$

for a given parameter λ and aggregate client parameters as $\theta_{t+1}^{\lambda} = \sum_{i=1}^{M} \alpha_i^{\lambda} \theta_{i,t}$. We refer to this strategy as Federated learning with Parameter Aggregation for Label Shift (FedPALS).

This strategy finds the optimal trade-off between the model aggregation closely aligning with the target label distribution, and minimizing the variance due to weighting using the inverse of the effective sample size (ESS) Kong (1992).

FedPALS also has the nice property that in the limit of $\lambda \to \infty$ we recover the regular FedAvg aggregation, which is equivalent to maximizing the effective sample size.

We perform experiments on several different datasets where we show that FedPALS perform quite well with its awareness of label marginals. We perform classification experiments where we on CIFAR-10 and Fashion-MNIST perform both sparsity sampling and dirichlet sampling. We also construct an experiment based on PACS that modify the dataset to consist of three clients, each missing a label that is present in the other two. Additionally, one client is reduced to one-tenth the size of the others, and the target distribution is made sparse in the same label as that of the small client. As a more real-world experiment we

Table 3.1: Comparison of mean accuracy and standard deviation (±) across different algorithms. The reported values are over 8 independent random seeds for the CIFAR-10 and Fashion-MNIST tasks, and 3 for PACS. C indicates the number of labels per client and β the Dirichlet concentration parameter. M is the number of clients. The *Oracle* method refers to a FedAvg model trained on clients whose distributions are identical to the target.

Data set	Label split	M	FedPALS	FedAvg	FedProx	SCAFFOLD	AFL	FedRS	Oracle
Fashion-MNIST	$C = 3$	10	$\mathbf{92.4 \pm 2.1}$	67.1 ± 22.0	66.9 ± 20.8	69.5 ± 19.3	78.9 ± 14.7	85.3 ± 13.5	97.6 ± 2.1
	$C = 2$		$\mathbf{80.6 \pm 23.7}$	53.9 ± 36.2	52.9 ± 35.7	54.9 ± 36.8	78.6 ± 20.0	63.14 ± 20.2	97.5 ± 4.0
CIFAR-10	$C = 3$		$\mathbf{65.6 \pm 10.1}$	44.0 ± 8.4	43.5 ± 7.2	43.3 ± 7.4	53.2 ± 0.9	44.0 ± 8.0	85.5 ± 5.0
	$C = 2$	10	$\mathbf{72.8 \pm 17.4}$	46.7 ± 15.8	47.7 ± 15.6	46.7 ± 14.9	54.7 ± 0.1	49.4 ± 9.5	89.2 ± 3.9
	$\beta = 0.1$		$\mathbf{62.6 \pm 17.9}$	40.8 ± 9.2	41.9 ± 9.7	43.5 ± 10.5	53.4 ± 11.5	57.1 ± 11.2	79.2 ± 3.7
PACS	$C = 6$	3	$\mathbf{86.0 \pm 2.9}$	73.4 ± 1.6	75.3 ± 1.3	73.9 ± 0.3	74.5 ± 0.9	76.1 ± 1.6	90.5 ± 0.3

evaluate performance on the iWildCam dataset which is sampled sparsely as detailed in the benchmark from Bai et al. (2024).

We compare the performance of FedPALS to similar approaches which are designed to combat client heterogeneity: FedProx, SCAFFOLD and AFL. We also compare to regular FedAvg and an oracle classifier for which the i.i.d. assumption is true to understand how much the introduction of label marginals help performance.

We show the results of these experiments in Table 3.1. We clearly see that the awareness of label marginals of FedPALS is beneficial and see quite substantial gains in many tasks. We also observe that there seems to be a bias-variance trade-off as the FedPALS results experiencing increased variance. This indicates that the addition of target label distribution faithfulness increases the variance of the resulting estimator. Further, we show in an ablation on the PACS task, that increasing the noise level in the target label marginal decreases the performance we observe from FedPALS which is expected.

Overall, we find that FedPALS balances the trade-off between matching the target label distribution and minimizing variance in the model updates. In particular, FedPALS perform well in challenging scenarios where label sparsity and client heterogeneity hinder the performance of other methods.

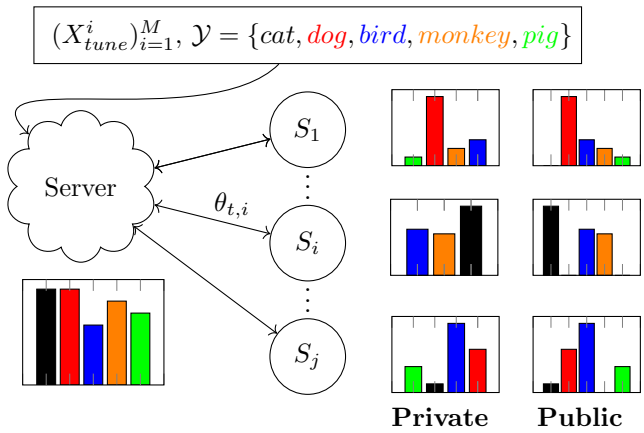


Figure 3.3: A schematic view of the two settings which we consider in Paper IV. The private setting where the clients are unaware of the full label set and the public setting where this is known. Note that the server has access to the information about which clients have which label sets.

3.4 Paper IV - Federated learning with heterogenous and private label sets

Considering the interest in handling heterogeneity in client datasets, it is reasonable to investigate how to handle this in cases where client label sets are considered sensitive. With current methods not sharing this would be unavoidable as the clients need to all agree on the full label set which they are constructing a classifier for. If the label set includes sensitive information, clients will be unwilling to share which labels they have available to other clients.

To ameliorate this issue, we propose the *private label set* setting. We adapt the regular FL setup to allow for client label sets being kept private. This is done by trusting only the server with the information about which labels a client has in its dataset. We illustrate the setup and how it compares to the regular setup, which we call *public*, in Figure 3.3.

Considering this settings challenges, We propose two main approaches to learning a model. First we propose an adaptation of FedAvg which is possible to extend to other FL approaches. This consists of two main modifications, in each round t, each client k is sent the full set of current encoder parameters θ^t_ϕ and the subset of current classifier parameters corresponding to their label set, $\theta^t_\psi[\mathcal{Y}_k] \coloneqq [\theta^t_\psi(y) : y \in \mathcal{Y}_k]^\top \in \mathbb{R}^{|\mathcal{Y}_k|}$. Clients then proceed with local updates as normal.

On the server side, the server receives parameter updates every round $(\theta^t_{\phi,k}, \theta^t_{\psi,k})$ from each client k and averages the classifier parameters for each label y based on models of clients which have the label in their label set, weighted according to their sample size (see Algorithm 2). Encoder parameter updates $\theta^t_{\phi,k}$ are averaged as we would with regular FedAvg. Second, we

Algorithm 2: FedAvg with private label sets

Data: Client label sets $\{\mathcal{Y}_k\}$ and reverse indices $\{I_k\}$
Result: Classifier $h(x) = \sigma(\theta_\psi^\top \phi(x))$
Initialize central parameters $\theta^0 = (\theta_\phi^0, \theta_\psi^0)$
for *each round* $t = 0, ..., T-1$ **do**
  for *each client* $k = 1, ..., m$ **do**
    Distribute $(\theta_\phi^t, \theta_\psi^t[\mathcal{Y}_k])$ to client k
    Receive client update $(\theta_{\phi,k}^t, \theta_{\psi,k}^t)$
  end
  $\theta_\phi^{t+1} = \sum_{k=1}^m \theta_{\phi,k}^t \frac{n_k}{n}$ where $n = \sum_{k=1}^m n_k$
  for *each label* $y \in \mathcal{Y}$ **do**
    $\theta_\psi^{t+1}(y) = \sum_{k:y\in\mathcal{Y}_k} \theta_{\psi,k}^t(I_k(y)) \frac{n_k}{n'_y}$ where $n'_y = \sum_{k:y\in\mathcal{Y}_k} n_k$
  end
end
Return classifier with parameters $\theta = (\theta_\phi^T, \theta_\psi^T)$

propose a tuning strategy in conjunction with the above aggregation. This is motivated by earlier work in the classifier coupling literature (Hastie & Tibshirani, 1997; Wu & Weng, 2004). This is motivated by the fact that the first strategy will not be guaranteed to work well when the representation is suboptimal for one or more clients.

In classifier coupling approaches, methods were developed to combine several *binary* classifiers, e.g., support vector machines, on different pairs of labels into a single multi-class classifier. This technique not commonly used as multi-class classifiers are trained routinely using neural networks with softmax outputs or similar models. However, in federated learning with heterogenous and private label sets, we face a similar problem since no client nor the server has access to labeled data from all classes.

Taking inspiration from these approaches we show that a tuning approach using an unlabeled dataset on the central server is a viable option. One which, under the right conditions, could produce an optimal classifier given optimal client models. However, while this particular special case is unlikely to occur, it is not unreasonable to believe that such a method could be empirically useful. We use a tuning approach with two main loss functions:

$$\underset{h\in\mathcal{H}}{\text{minimize}} \sum_{k=1}^{m} w_k \sum_{\substack{y,y'\in\mathcal{Y}_k \\ y\neq y'}} \mathbb{E}_{\boldsymbol{X}} \left[\big(h_k(y \mid \boldsymbol{X}) h(y' \mid \boldsymbol{X}) - h_k(y' \mid \boldsymbol{X}) h(y \mid \boldsymbol{X}) \big)^2 \right], \tag{3.2}$$

which we call the *pairwise* loss which align model prediction in a pairwise manner. We also consider the direct MSE loss

$$\underset{h\in\mathcal{H}}{\text{minimize}} \sum_{k=1}^{m} w_k \sum_{y\in\mathcal{Y}_k} \mathbb{E}_{\boldsymbol{X}} \left[\big(h_k(y \mid \boldsymbol{X}) - h(y \mid \boldsymbol{X}) \big)^2 \right]. \tag{3.3}$$

We perform classification experiments on both Fashion-MNIST and CIFAR-10 where we vary the number of labels that a client has access to from 2-10. We choose a random set of labels for each client which they then get distributed from the dataset equally. To combat the the heterogeneous amount of examples in clients resulting from this we perform a subsampling step, which results in each client having an equal amount of samples.

We observe from our experiments that our adaptation of regular FL methods to the private setting perform quite well. This suggests that the simple per-class aggregation is sufficient to achieve adequate performance. Further, this indicates that alignment of representation may not always be necessary. However, the pairwise tuning outperforms our adaptations of FedAvg and FedProx on CIFAR-10 in the public setting. This suggests that the tuning of the representation can be of use in this setting also.

In summary, paper IV shows that federated learning is still possible in a private label set scenario. Furthermore, our proposed approaches successfully enable learning in this setting. We observe empirically that these methods work well with a minimal performance drop compared to performance in the public setting.

Chapter 4

Concluding remarks and future directions

In this thesis we have presented the challenges posed by distributional shifts in machine learning. We have explored how to both predict performance as well as adapt to these shift in both centralized and federated scenarios. We performed a survey of generalization bounds with the express purpose to find ones which can practically measure and assess the target performance of models. We also proposed a method for leveraging the structure inherent in privileged information which can be available during training. In the federated learning context, we introduced the concept of target-aware federated learning and proposed a method to leverage the server-side knowledge of the target. Finally, we consider a setting where label sets are too sensitive to share across clients. We have formalized two novel approaches to performing federated learning with both heterogeneous clients and private label sets. Overall, we have shown that there are significant benefits from exploring novel settings and modifying the assumptions of what data is available or their statistical properties.

The methods and theory we propose all come with their own strengths and limitations. These are all discussed in each paper which is appended in Part II of the thesis, however, some are worth discussing here. Most methods and settings we investigate assume that we have access to more information or that the underlying data has certain structure. We have also generally focused on image classification datasets in our work. Of course, this is limiting as there are many other types of data which we could have considered, e.g tabular or time-series data, each with their own distinct properties. However, this is motivated by the large dimensionality of images, and the models used to process them. This is a complicating factor in most theoretical treatments and why we chose to work with these.

In future, the focus should be to further attempt to solve the distributional shift problem for specific scenarios where different amounts of information is available. We have made several inroads in these areas, but several promising avenues still remain to explore further in this respect. Similar to the setting we propose in paper II one might explore how the addition of other modalities

of data could impact learning both on a theoretical and empirical level. Furthermore, in the context of paper III there are many interesting settings to consider. One could attempt to further explore to what extent the information in target-aware federated learning can be inferred from model updates from clients. If this inference is both plausible to conduct and accurate enough for guiding learning one could lessen the amount of information clients would need to explicitly share. An alternative would be that we could transmit the information in a compressed form. For instance, we could achieve this through sharing some representation of the label distribution similar to the approach in Mclaughlin & Su (2024).

Building on paper IV, one could easily envision a setting with even further privacy gains. One would not share the model parameters at all but allow there to be a querying system between the clients and the server. Perhaps they could be answers to how the models would classify certain, publicly available, inputs. The global and local models could then be updated based on the responses to these queries, making use of a similar alignment method as we introduce in our work.

Furthermore, it is of great interest to develop novel theoretical perspectives to make use of the underlying structure of problems explicitly. This may be done by presupposing a specific underlying structure of the data and treating the problem based on this assumption. For example, there have been works that incorporate the notion of a taxonomy in their data and tailor their architecture to exploit this hierarchical structure (Liu et al., 2023). This raises the compelling question of which structures are most beneficial, and how they can be effectively integrated into learning frameworks.

Historically, architectural innovations such as Convolutional Neural Networks (Lecun, 1998) and Transformers (Vaswani et al., 2017) have achieved remarkable success by embedding strong inductive biases directly into the model design. These approaches represent embedding structural assumptions at the architectural level. However, it remains an open question whether similar benefits can be achieved through purely methodological approaches, i.e., without altering model architecture. Crucially, any such assumptions must be justifiable and, ideally, verifiable a priori. If the assumed structure cannot be assessed from the available data, its practical utility is limited. Although, it may be permissible if the empirical gains derived from making the assumption is substantial enough. Whether these structural assumptions can be made general across tasks or must be tailored to specific applications remains an open and important question. If the latter holds, considerable effort will be required to identify and formalize appropriate structures for each application or task.

On a related note, the recent success of large pretrained foundation models suggests that transferability can, to some extent, be achieved simply through access to vast and diverse datasets e.g. LAION (Schuhmann et al., 2022). This brings to light a salient question: in such large-scale settings, is the overlapping support assumption, central to many theoretical generalization guarantees, more likely to hold due to the breadth and diversity of the training data?

Taken together, the work presented in this thesis highlights the importance

of revisiting core assumptions in machine learning. By proposing novel methods and theoretical perspectives in both centralized and federated settings, we have shown that addressing these issues directly can lead to more robust and adaptable models. Future work should continue by investigating how structure in data sets can be leveraged in both model design and method development. Additionally, further theoretical grounding is needed to understand the implications of such assumptions and to formalize generalizable principles across tasks and domains.

As machine learning increasingly moves toward real-world deployment, addressing distributional shift will remain a key concern. This work has provided some initial steps toward this goal, and with future efforts we will hopefully be able to produce powerful, but also reliable and adaptable models, in the face of real-world complexity.

Bibliography

David Acuna, Guojun Zhang, Marc T. Law, and Sanja Fidler. f-domain adversarial learning: Theory and algorithms. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 66–75. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/acuna21a.html`.

Amiran Ambroladze, Emilio Parrado-hernández, and John Shawe-taylor. Tighter pac-bayes bounds. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

Ruqi Bai, Saurabh Bagchi, and David I Inouye. Benchmarking algorithms for federated domain generalization. In *The Twelfth International Conference on Learning Representations*, 2024.

Shai Ben-David and Ruth Urner. On the hardness of domain adaptation and the utility of unlabeled target samples. In *International Conference on Algorithmic Learning Theory*, pp. 139–153. Springer, 2012.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, May 2010a.

Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility Theorems for Domain Adaptation. In *International Conference on Artificial Intelligence and Statistics*, 2010b.

Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. *Journal of Machine Learning Research*, 9:129–136, 2010c. URL `http://www.jmlr.org/proceedings/papers/v9/david10a.html`.

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing - EMNLP '06*, pp. 120, Sydney, Australia, 2006. Association for Computational Linguistics. ISBN 978-1-932432-73-2.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning Bounds for Domain Adaptation. *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pp. 8, 2008.

Mahrokh Ghoddousi Boroujeni, Andreas Krause, and Giancarlo Ferrari-Trecate. Personalized federated learning of probabilistic models: A pac-bayesian approach. *ArXiv*, abs/2401.08351, 2024.

Mahrokh Ghoddousi Boroujeni, Andreas Krause, and Giancarlo Ferrari-Trecate. Personalized federated learning of probabilistic models: A PAC-bayesian approach. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=ZMliWjMCor.

Adam Breitholtz and Fredrik Daniel Johansson. Practicality of generalization guarantees for unsupervised domain adaptation with neural networks. *Transactions on Machine Learning Research*, 2022.

Adam Breitholtz, Anton Matsson, and Fredrik D. Johansson. Unsupervised domain adaptation by learning using privileged information. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=saV3MPH0kw.

Adam Breitholtz, Edvin Listo Zec, and Fredrik D. Johansson. Federated learning with heterogenous and private label sets. In *3rd Workshop on Advancements in Federated Learning (WAFL), ECML-PKDD*, 2025.

George W. Brown. Basic principles for construction and application of discriminators. *Journal of Clinical Psychology*, 6(1):58–61, 1950. doi: https://doi.org/10.1002/1097-4679(195001)6:1⟨58::AID-JCLP2270060112⟩3.0.CO;2-B. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-4679%28195001%296%3A1%3C58%3A%3AAID-JCLP2270060112%3E3.0.CO%3B2-B.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4. March 2023. URL https://www.microsoft.com/en-us/research/publication/sparks-of-artificial-general-intelligence-early-experiments-with-gpt-4/.

Ciprian Chelba and Alex Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech & Language*, 20(4):382–399, 2006. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2005.05.005. URL https://www.sciencedirect.com/science/article/pii/S0885230805000276.

Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv:1802.07876*, 2018.

Qi Chen and Mario Marchand. Algorithm-dependent bounds for representation learning of multi-source domain adaptation. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 10368–10394. PMLR, 25–27 Apr 2023. URL `https://proceedings.mlr.press/v206/chen23h.html`.

Shuxiao Chen, Qinqing Zheng, Qi Long, and Weijie J. Su. Minimax estimation for personalized federated learning: an alternative between fedavg and local training? *J. Mach. Learn. Res.*, 24(1), January 2023. ISSN 1532-4435.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Fedavg with fine tuning: Local updates lead to representation learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, January 2014. ISSN 0304-3975.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning Bounds for Importance Weighting. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 23, pp. 442–450. Curran Associates, Inc., 2010.

Corinna Cortes, Mehryar Mohri, and Andrés Muñoz Medina. Adaptation Algorithm and Theory Based on Generalized Discrepancy. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 169–178, Sydney NSW Australia, August 2015. ACM. ISBN 978-1-4503-3664-2.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint Distribution Optimal Transportation for Domain Adaptation. *arXiv:1705.08848 [cs, stat]*, October 2017. arXiv: 1705.08848.

Koby Crammer, Michael Kearns, and Jennifer Wortman. Learning from multiple sources. *Journal of Machine Learning Research*, 9(57):1757–1774, 2008. URL `http://jmlr.org/papers/v9/crammer08a.html`.

Hal Daumé III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research (JAIR)*, 26:101–126, 2006. URL `http://hal3.name/docs/#daume06megam`.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui

Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL `https://arxiv.org/abs/2412.19437`.

Sofien Dhouib, Ievgen Redko, and Carole Lartizien. Margin-aware adversarial domain adaptation with optimal transport. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2514–2524. PMLR, 13–18 Jul 2020.

Sofien Dhouib and Ievgen Redko. Revisiting (ϵ , γ, τ)-similarity learning for domain adaptation. *NeurIPS*, pp. 7408–7417, 2018.

Tong Ding, Sophia J. Wagner, Andrew H. Song, Richard J. Chen, Ming Y. Lu, Andrew Zhang, Anurag J. Vaidya, Guillaume Jaume, Muhammad Shaban, Ahrong Kim, Drew F. K. Williamson, Bowen Chen, Cristina Almagro-Perez, Paul Doucet, Sharifa Sahai, Chengkuan Chen, Daisuke Komura, Akihiro Kawabe, Shumpei Ishikawa, Georg Gerber, Tingying Peng, Long Phi Le, and Faisal Mahmood. Multimodal whole slide foundation model for pathology, 2024. URL `https://arxiv.org/abs/2411.19666`.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin (eds.), *Theory of Cryptography*, pp. 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel M. Roy. On the role of data in PAC-Bayes bounds. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, October 2021.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in neural information processing systems*, 33:3557–3568, 2020.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Generalization of model-agnostic meta-learning algorithms: recurring and unseen tasks. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Jan Feyereisl and Uwe Aickelin. Privileged information for data clustering. *Information Sciences*, 194:4–23, 2012.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. doi: https://doi.org/10.1111/j.1469-1809.1936.tb02137.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x`.

Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. A PAC-Bayesian Approach for Domain Adaptation with Specialization to Linear Classifiers. In *International Conference on Machine Learning*, pp. 738–746. PMLR, May 2013. ISSN: 1938-7228.

Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. PAC-Bayes and Domain Adaptation. *Neurocomputing*, 379:379–397, February 2020. ISSN 09252312. arXiv: 1707.05712.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL `http://jmlr.org/papers/v13/gretton12a.html`.

Jiang Guo, Darsh Shah, and Regina Barzilay. Multi-source domain adaptation with mixture of experts. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4694–4703, Brussels, Belgium,

October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1498. URL https://aclanthology.org/D18-1498/.

Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In *Proceedings of the 11th International Conference on Neural Information Processing Systems*, NIPS'97, pp. 507–513, Cambridge, MA, USA, 1997. MIT Press.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with Side Information through Modality Hallucination. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 826–834, Las Vegas, NV, USA, June 2016. IEEE.

Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. Algorithms and theory for multiple-source adaptation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 8256–8266, Red Hook, NY, USA, 2018. Curran Associates Inc.

Min Huang, Zifeng Xie, Bo Sun, and Ning Wang. Multi-source unsupervised domain adaptation with prototype aggregation. *Mathematics*, 13(4), 2025. ISSN 2227-7390. doi: 10.3390/math13040579. URL https://www.mdpi.com/2227-7390/13/4/579.

Rebecca Hwa. Supervised grammar induction using training data with limited constituent information. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pp. 73–79, 1999.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL https://aclanthology.org/D17-1215.

Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

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

C. Jordan. On Fourier series. *C. R. Acad. Sci., Paris*, 92:228–230, 1881. ISSN 0001-4036.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner,

Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Far inaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, M ariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Y ang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. ISSN 1935-8237. doi: 10.1561/2200000083. URL `http://dx.doi.org/10.1561/2200000083`.

Okan Koc, Alexander Soen, Chao-Kai Chiang, and Masashi Sugiyama. Domain adaptation and entanglement: an optimal transport perspective. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025. URL `https://openreview.net/forum?id=ZDyi1BeTu7`.

Augustine Kong. A note on importance sampling using standardized weights. *University of Chicago, Dept. of Statistics, Tech. Rep*, 348:14, 1992.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. ISSN 00034851. URL `http://www.jstor.org/stable/2236703`.

Seiichi Kuroki, Nontawat Charoenphakdee, Han Bao, Junya Honda, Issei Sato, and Masashi Sugiyama. Unsupervised Domain Adaptation Based on Source-Guided Discrepancy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4122–4129, July 2019.

Yann Lecun. Gradient-Based Learning Applied to Document Recognition. *proceedings of the IEEE*, 86(11):47, 1998.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, 2020.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pp. 6357–6368. PMLR, 2021.

Xin-Chun Li and De-Chuan Zhan. Fedrs: Federated learning with restricted softmax for label distribution non-iid data. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021.

Yanmeng Li, Huaijiang Sun, and Wenzhu Yan. Domain adaptive twin support vector machine learning using privileged information. *Neurocomputing*, 469: 13–27, 2022.

Zexi Li, Tao Lin, Xinyi Shang, and Chao Wu. Revisiting weighted aggregation in federated learning with neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pp. 3122–3130. PMLR, 2018.

Tianyi Liu, Zihao Xu, Hao He, Guang-Yuan Hao, Guang-He Lee, and Hao Wang. Taxonomy-structured domain adaptation. In *International Conference on on Machine Learning*, 2023.

Yiling Liu, Juncheng Dong, Ziyang Jiang, Ahmed Aloui, Keyu Li, Michael Hunter Klein, Vahid Tarokh, and David Carlson. Understanding and robustifying sub-domain alignment for domain adaptation. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL `https://openreview.net/forum?id=oAzu0gzUUb`.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning Transferable Features with Deep Adaptation Networks. *arXiv:1502.02791 [cs]*, May 2015. arXiv: 1502.02791.

David Lopez-Paz, Leon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR 2016)*, 2016.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain Adaptation with Multiple Sources. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pp. 1041–1048, January 2008.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Multiple source adaptation and the rényi divergence. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009*, Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009, pp. 367–374. AUAI Press, 2009a.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain Adaptation: Learning Bounds and Algorithms. In *Proceedings of the Conference on Learning Theory*, February 2009b.

David A. McAllester. Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, COLT' 98, pp. 230–234, New York, NY, USA, July 1998. Association for Computing Machinery. ISBN 978-1-58113-057-7.

David A. McAllester. A PAC-Bayesian Tutorial with A Dropout Bound. *arXiv e-prints*, 1307:arXiv:1307.2118, July 2013.

Connor Mclaughlin and Lili Su. Personalized federated learning via feature distribution adaptation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=Wl2optQcng`.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, 2017.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4615–4625. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/mohri19a.html.

Emilie Morvant, Amaury Habrard, and Stéphane Ayache. Parsimonious unsupervised and semi-supervised domain adaptation with good similarity functions. *Knowledge and Information Systems*, 33(2):309–349, November 2012. ISSN 0219-1377, 0219-3116.

Saeid Motiian. *Domain Adaptation and Privileged Information for Visual Recognition*. phdthesis, https://researchrepository.wvu.edu/etd/6271, 2019. Graduate Theses, Dissertations, and Problem Reports. 6271.

Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997. ISSN 00018678. URL http://www.jstor.org/stable/1428011.

J Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY OF LONDON SERIES A-CONTAINING PAPERS OF A MATHEMATICAL OR PHYSICAL CHARACTER*, 231:289–337, MAR 1933. ISSN 0264-3952. doi: 10.1098/rsta.1933.0009.

A. Tuan Nguyen, Toan Tran, Yarin Gal, Philip Torr, and Atilim Gunes Baydin. KL guided domain adaptation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0JzqUlIVVDd.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. Pac-bayes bounds with data dependent priors. *J. Mach. Learn. Res.*, 13(1):3507–3531, dec 2012. ISSN 1532-4435.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1406–1415, 2019. doi: 10.1109/ICCV.2019.00149.

Sashank Reddi, Zachary Burr Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Brendan McMahan (eds.). *Adaptive Federated Optimization*, 2021.

Ievgen Redko. *Nonnegative matrix factorization for transfer learning*. PhD thesis, Paris North University, 2015.

Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical Analysis of Domain Adaptation with Optimal Transport. *arXiv:1610.04420 [cs, stat]*, July 2017. arXiv: 1610.04420.

Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia. Optimal transport for multi-source domain adaptation under target shift. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 849–858. PMLR, 16–18 Apr 2019.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics*, volume 4, pp. 547–562. University of California Press, 1961.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new aprioriprobabilities: A simple procedure. *Neural Computation*, 14(1):21–41, January 2002. doi: 10.1162/089976602753284446. URL `http://dx.doi.org/10.1162/089976602753284446`.

Nikolaos Sarafianos, Michalis Vrigkas, and Ioannis A. Kakadiaris. Adaptive svm+: Learning with privileged information for domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32 (8):3710–3722, 2020.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: an open large-scale dataset for training next generation image-text models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Tawseef Ayoub Shaikh, Rashid Ali, and M. M. Sufyan Beg. Transfer learning privileged information fuels CAD diagnosis of breast cancer. *Machine Vision and Applications*, 31(1):9, February 2020.

John Shawe-Taylor and Robert C. Williamson. A pac analysis of a bayesian estimator. In *Proceedings of the Tenth Annual Conference on Computational*

Learning Theory, COLT '97, pp. 2–9, New York, NY, USA, 1997. Association for Computing Machinery. ISBN 0897918916.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein Distance Guided Representation Learning for Domain Adaptation. *arXiv:1707.01217 [cs, stat]*, March 2018. arXiv: 1707.01217.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, October 2000. ISSN 0378-3758.

Changjian Shui, Zijian Li, Jiaqi Li, Christian Gagné, Charles X Ling, and Boyu Wang. Aggregating from multiple target-shifted sources. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9638–9648. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/shui21a.html`.

Catarina Silva, Armando Vieira, Antonio Gaspar-Cunha, and Joao Carvalho das Neves. Financial distress model prediction using SVM+. In *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–7, July 2010.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, October 2017. doi: 10.1038/nature24270. URL `http://dx.doi.org/10.1038/nature24270`.

L. G. Valiant. A theory of the learnable. *Communications of the ACM*, pp. 1134–1142, 1984.

V. Vapnik. Principles of risk minimization for learning theory. In *Proceedings of the 5th International Conference on Neural Information Processing Systems*, NIPS'91, pp. 831–838, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc. ISBN 1558602224.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, July 2009.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 9 1998. ISBN 0471030031.

Harsh Vardhan, Avishek Ghosh, and Arya Mazumdar. An improved federated clustering algorithm with model-based clustering. *Transactions on Machine Learning Research*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7363–7372, 2019.

B. L. Welch. (ii) note on discriminant functions. *Biometrika*, 31(1-2):218–218, 07 1939. ISSN 0006-3444. doi: 10.1093/biomet/31.1-2.218. URL https://doi.org/10.1093/biomet/31.1-2.218.

Ting-Fan Wu and Ruby C Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5: 975–1005, 2004.

Sang Michael Xie, Ananya Kumar, Robbie Jones, Fereshte Khani, Tengyu Ma, and Percy Liang. In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. *arXiv preprint arXiv:2012.04550*, 2020.

Sang Michael Xie, Ananya Kumar, Robbie Jones, Fereshte Khani, Tengyu Ma, and Percy Liang. In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. In *International Conference on Learning Representations (ICLR)*, 2021.

Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Wenzhe Yin, Shujian Yu, Yicong Lin, Jie Li, Jan-Jakob Sonke, and Efstratios Gavves. Domain adaptation with cauchy-schwarz divergence. In *Conference on Uncertainty in Artificial Intelligence*, 2024. URL https://api.semanticscholar.org/CorpusID:270123453.

Edvin Listo Zec, Adam Breitholtz, and Fredrik D. Johansson. Overcoming label shift in targeted federated learning. In *Tiny Titans: The next wave of On-Device Learning for Foundational Models (TTODLer-FM)*, 2025. URL https://openreview.net/forum?id=ZUlrLeW82p.

John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683, 2018.

Dun Zeng, Zenglin Xu, Yu Pan, Qifan Wang, and Xiaoying Tang. Tackling hybrid heterogeneity on federated optimization via gradient diversity maximization. *arXiv preprint arXiv:2310.02702*, 2023.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I. Jordan. Bridging Theory and Algorithm for Domain Adaptation. *arXiv:1904.05801 [cs, stat]*, April 2019. arXiv: 1904.05801 version: 1.

Sicheng Zhao, Guangzhi Wang, Shanghang Zhang, Yang Gu, Yaxian Li, Zhichao Song, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source distilling domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):12975–12983, April 2020. doi: 10.1609/aaai.v34i07.6997.

Tailin Zhou, Zehong Lin, Jinchao Zhang, and Danny H. K. Tsang. Understanding and improving model averaging in federated learning on heterogeneous data. *IEEE Transactions on Mobile Computing*, 2023.

Part II

Appended Publications

Paper I

Practicality of generalization guarantees for unsupervised domain adaptation with neural networks

A. Breitholtz, F. D. Johansson

Transactions of Machine Learning Research (October 2022)

Practicality of generalization guarantees for unsupervised domain adaptation with neural networks

Adam Breitholtz *adambre@chalmers.se*
Department of Computer Science
Chalmers University of Technology

Fredrik D. Johansson *fredrik.johansson@chalmers.se*
Department of Computer Science
Chalmers University of Technology

Reviewed on OpenReview: *https://openreview.net/forum?id=vUuHPRrWs2*

Abstract

Understanding generalization is crucial to confidently engineer and deploy machine learning models, especially when deployment implies a shift in the data domain. For such domain adaptation problems, we seek generalization bounds which are tractably computable and tight. If these desiderata can be reached, the bounds can serve as guarantees for adequate performance in deployment. However, in applications where deep neural networks are the models of choice, deriving results which fulfill these remains an unresolved challenge; most existing bounds are either vacuous or has non-estimable terms, even in favorable conditions. In this work, we evaluate existing bounds from the literature with potential to satisfy our desiderata on domain adaptation image classification tasks, where deep neural networks are preferred. We find that all bounds are vacuous and that sample generalization terms account for much of the observed looseness, especially when these terms interact with measures of domain shift. To overcome this and arrive at the tightest possible results, we combine each bound with recent data-dependent PAC-Bayes analysis, greatly improving the guarantees. We find that, when domain overlap can be assumed, a simple importance weighting extension of previous work provides the tightest estimable bound. Finally, we study which terms dominate the bounds and identify possible directions for further improvement.

1 Introduction

Successful deployment of machine learning systems relies on generalization to inputs never seen in training. In many cases, training and in-deployment inputs differ systematically; these are domain adaptation (DA) problems. An example of a setting where these problems arise is healthcare. Learning a classifier from data from one hospital and applying to samples from another is an example of tasks that machine learning often fail at (AlBadawy et al., 2018; Perone et al., 2019; Castro et al., 2020). In high-stakes settings like healthcare, guarantees on model performance would be required before meaningful deployment is accepted. Modern machine learning models, especially neural networks, perform well on diverse and challenging tasks on which conventional models have had only modest success. However, due to the high flexibility and opaque nature of neural networks, it is often hard to quantify how well we can expect them to perform in practice.

Performance guarantees for machine learning models are typically expressed as generalization bounds. Bounds for unsupervised domain adaptation (UDA), where no labels are available from the target domain, have been explored in a litany of papers using both the PAC (Ben-David et al., 2007; Mansour et al., 2009) and PAC-Bayes (Germain et al., 2020) frameworks; see Redko et al. (2020) for an extensive survey. Despite great interest in this problem, very few works actually compute or report the *value* of the proposed bounds. Instead, the results are used only to guide optimization or algorithm development. Moreover, the bounds

presented often contain terms which are non-estimable without labeled data from the target domain even under favourable conditions (Johansson et al., 2019; Zhao et al., 2019).

For deployment in sensitive settings we wish to find bounds which are: a) Amenable to calculation; they do not contain non-estimable terms and are tractable to compute. b) Tight; they are close to the error in deployment (or at least non-vacuous). How do existing bounds fare in solving this problem? As we will see, for realistic problems under favorable conditions, most, if not all, bounds in the literature struggle to satisfy one or both of these goals to varying degrees.

In this work, we examine the practical usefulness of current UDA bounds as performance guarantees in the context of learning with neural networks. Examining the literature with respect to our desiderata, we identify bounds which show promise in being estimable and tractably computable (Section 2.1). We find that terms related to sample generalization dominate existing bounds for neural networks, prohibiting tight guarantees. To remedy this, we apply PAC-Bayes analysis (McAllester, 1999) with data-dependent priors (Dziugaite & Roy, 2019) in four diverse bounds (Sections 2.3–2.4). Two are existing PAC-Bayes bounds from the UDA literature and two are PAC-Bayes adaptations of bounds based on importance weighting (IW) and integral probability metrics (IPM). We evaluate the bounds empirically under favorable conditions in two tasks which fulfill the covariate shift and domain overlap assumptions; one task concerns digit image classification and the second X-ray classification (Section 3). Our results show that all four bounds are vacuous on both tasks without data-dependent priors, but some can be made tight with them (Section 4). Furthermore, we find that the simple extension of applying importance weights to previous work outperforms the best fully observable bound from the literature in tightness. This result highlights amplification of bound looseness due to interactions between domain adaptation and sample generalization terms. We conclude by offering insights into achieving the tightest bounds possible given the current state of the literature (Section 5).

2 Background

In this section, we introduce the unsupervised domain adaptation (UDA) problem and give a survey of existing generalization bounds through the lens of practicality: do the bounds contain non-estimable terms and are they tractably computable? We go on to select a handful of promising bounds and combine them with data-dependent PAC-Bayes analysis to arrive at the tightest guarantees available.

We study UDA for binary classification, in the context of an input space $\mathcal{X} \subseteq \mathbb{R}^d$ and a label space $\mathcal{Y} = \{-1, 1\}$. While our arguments are general, we use as running example the case where $\mathcal{X}$ is a set of black-and-white images. Let $\mathcal{S}$ and $\mathcal{T}$, where $\mathcal{S} \neq \mathcal{T}$, be two distributions, or *domains*, over the product space $\mathcal{X} \times \mathcal{Y}$, called the source domain and target domain respectively. The source domain is observed through a labeled sample $S = \{x_i, y_i\}_{i=1}^n \sim (\mathcal{S})^n$ and the target domain through a sample $S'_x = \{x'_i\}_{i=1}^m \sim (\mathcal{T}_x)^m$ which lacks labels, where $\mathcal{T}_x$ is the marginal distribution on $\mathcal{X}$ under $\mathcal{T}$. Throughout, $(\mathcal{D})^N$ denotes the distribution of a sample of N datapoints drawn i.i.d. from the domain $\mathcal{D}$.

The UDA problem is to learn hypotheses h from a hypothesis class $\mathcal{H}$, by training on S and S'_x, such that the hypotheses perform well on unseen data drawn from $\mathcal{T}_x$. In the Bayesian setting, we learn posterior distributions ρ over $\mathcal{H}$ from which sampled hypotheses perform well on average. We measure performance using the expected *target risk* $R_\mathcal{T}$ of a single hypothesis h or posterior ρ,

$$\underbrace{R_\mathcal{T}(h) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{T}}[\ell(h(x), y)]}_{\text{Risk for single hypothesis } h} \quad \text{or} \quad \underbrace{\mathop{\mathbb{E}}_{h\sim\rho} R_\mathcal{T}(h)}_{\text{Gibbs risk of posterior } \rho}, \tag{1}$$

for a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$. In this work, we study the zero-one loss, $\ell(y, y') = \mathbb{1}[y \neq y']$. The Gibbs risk is used in the PAC-Bayes guarantees (Shawe-Taylor & Williamson, 1997; McAllester, 1998), a generalization of the PAC framework (Valiant, 1984; Vapnik, 1998).

When learning from samples, the empirical risk $\hat{R}_\mathcal{D}$ can be used as an observable measure of performance,

$$\hat{R}_\mathcal{D}(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h(x_i), y_i), \tag{2}$$

for a sample $\{(x_i, y_i)\}_{i=1}^n \sim (\mathcal{D})^n$, with the empirical risk of the Gibbs classifier defined analogously. However, since no labeled sample from $\mathcal{T}$ is available, the risk of interest is not directly observable. Hence, the most common way to approximate this quantity is to derive an upper bound on the target risk. We refer to these as *UDA bounds*. Crucially, *any practical performance guarantee must be made using only observed data and assumptions on how $\mathcal{S}$ and $\mathcal{T}$ relate*. Throughout, we make the following common assumptions.

Assumption 1 (Covariate shift & overlap)**.** The source domain $\mathcal{S}$ and target domain $\mathcal{T}$ satisfy for all x, y

$$\begin{aligned} &\textbf{Covariate shift:} && \mathcal{T}_y(Y \mid X = x) = \mathcal{S}_y(Y \mid X = x) \quad \text{and} \quad \mathcal{T}_x(X = x) \neq \mathcal{S}_x(X = x) \\ &\textbf{Overlap:} && \mathcal{T}_x(X = x) > 0 \Rightarrow \mathcal{S}_x(X = x) > 0 \ . \end{aligned}$$

These are strong assumptions and covariate shift cannot be verified statistically. They are not required by every bound in the literature but together they are sufficient to guarantee identification and consistent estimation of the target risk (Shimodaira, 2000). More importantly, the generalization guarantees we study more closely are not fully observable without them unless target labels are available. Finally, this setting is among the most simple and favorable ones for UDA which should make for an interesting benchmark—if existing bounds are vacuous also here, significant challenges remain.

2.1 Overview of existing UDA bounds

Most existing UDA bounds on the target risk share a common structure due to their derivation. The typical process starts by bounding the *expected* target risk using the *expected* source risk and measures of domain shift. Thereafter, terms are added which bound the sample generalization error, the difference between the expected source risk and its empirical estimate. The results can be summarized conceptually, with f and arguments variously defined, as expressions on the form

$$R_{\mathcal{T}} \leq f(\text{Empirical source risk, Measures of domain shift, Sample generalization error}) \ .$$

There are two main forms taken by this function; one in which sample generalization terms are related to domain shift terms through addition and one where they are multiplied. We call these additive and multiplicative bounds respectively. One example is the classical result due to Ben-David et al. (2007) which uses the so-called $\mathcal{A}$-distance to bound the target risk of $h \in \mathcal{H}$ with probability $\geq 1 - \delta$,

$$R_{\mathcal{T}}(h) \leq \underbrace{\hat{R}_{\mathcal{S}}(h)}_{\text{Emp. risk}} + \underbrace{\sqrt{\frac{4(d \log \frac{2em}{d} + \log \frac{4}{\delta})}{m}}}_{\text{Sample generalization}} + \underbrace{d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda}_{\text{Domain shift}}, \tag{3}$$

where d is the VC dimension of the $\mathcal{H}$, λ is the sum of the errors on both domains of the best performing classifier $h^* = \arg\min_{h \in \mathcal{H}}(R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h))$, and $d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) = 2 \sup_{A \in \{\{x : h(x) = 1\} : h \in \mathcal{H}\}} |\Pr_{\mathcal{S}}[A] - \Pr_{\mathcal{T}}[A]|$ is the $\mathcal{A}$-distance for the characteristic sets of hypotheses in $\mathcal{H}$.

Three challenges limits the practicality of this bound: i) λ is not directly estimable without target labels and must be assumed small for an informative bound. This is a pattern in UDA theory which illustrates a fundamental link between estimability and assumption. ii) The VC dimension can easily lead to a vacuous result for modern neural networks. For example, the VC dimension of piecewise polynomial networks is $\Omega(pl \log \frac{p}{l})$ where p is the number of parameters and l is the number of layers (Bartlett et al., 2019). iii) The $\mathcal{A}$-distance can be tractably computed only for restricted hypothesis classes. These issues are not unique the bound above, they are exhibited to varying degrees by any UDA bound.

In response to concerns for practical generalization bounds in deep learning, Valle-Pérez & Louis (2020) put forth seven desiderata for predictive bounds. While these are of interest also for UDA, we concern ourselves primarily with the fifth (non-vacuity of the bound) and sixth (efficient computability) desiderata as they are of paramount importance to achieving practically useful guarantees. In this work, we study UDA bounds with emphasis on how each term influences the following properties when learning with deep neural networks.

1. **Tightness.** Is the term a poor approximation? Is it likely to lead to a loose bound?

Table 1: Overview of existing UDA bounds with respect to a) measures of domain divergence, b) whether domain and sample generalization terms add or multiply, c) non-estimable terms, d) whether the bounds were computed empirically, e) computational tractability. The highlighted rows represent a selection of bounds which are possible to estimate under the assumptions we make and which hold promise in fulfilling our other desiderata. X* denotes that under the assumptions made in this work we do not have non-estimable terms.

Paper/reference	Divergence	Add/Mult	Non-est. terms	Evaluated bound	Tractable
Ben-David et al. (2007)	$\mathcal{A}$-distance	Add	✓	X	X
Blitzer et al. (2008)	$H\Delta H$	Add	✓	X	X
Ben-David et al. (2010)	$H\Delta H$	Add	✓	X	X
Morvant et al. (2012)	$H\Delta H$	Add	✓	X	X
Mansour et al. (2009)	Discrepancy dist.	Add	✓	X	X
Redko et al. (2019)	Discrepancy dist.	Add	✓	X	X
Kuroki et al. (2019)	S-discrepancy	Add	✓	X	X
Cortes & Mohri (2014)	Gen. discrepancy	Add	✓	X	X
Cortes et al. (2015)	Gen. discrepancy	Add	✓	X	X
Zhang et al. (2012)	IPM	Add	X	X	✓
Redko (2015)	IPM, MMD	Add	✓	X	✓
Long et al. (2015)	MMD	Add	✓	X	✓
Redko et al. (2017)	IPM	Add	✓	X	✓
Johansson et al. (2019)	IPM	Add	✓	X	✓
Zhang et al. (2019)	Margin disparity	Add	✓	X	X
Dhouib et al. (2020)	Wasserstein	Add	✓	X	✓
Shen et al. (2018)	Wasserstein	Add	✓	X	✓
Courty et al. (2017b)	Wasserstein	Add	✓	X	✓
Germain et al. (2013)	Domain disagreement	Add	✓	X	X
Zhang et al. (2020)	Localized discrepancy	Add	✓	X	X
Cortes et al. (2019)	Localized discrepancy	Add	✓	X	X
Acuna et al. (2021)	f-divergences	Add	✓	X	X
Germain et al. (2016)	β-divergence	Mult	X*	X	✓
Cortes et al. (2010)	Rényi	Mult	X*	X	X
Dhouib & Redko (2018)	L^1, χ^2	Mult	✓	X	X

2. **Estimability.** Is the term something which we can estimate from observed data?

3. **Computability.** Can we tractably compute it for real-world data sets and hypothesis classes?

Next, we give a short summary of existing UDA bounds, with the excellent survey by Redko et al. (2019) as starting point, while evaluating whether they contain non-estimable terms, if the bound was computed in the paper[1] and if the bound is computationally tractable for neural networks. We have listed considered bounds in Table 1.

We will now reason about the bounds' potential to reach our stated desiderata, starting with tractability. We begin by noting that several divergence measures (e.g., $\mathcal{A}$-distance, $H\Delta H$-distance, Discrepancy distance) are defined as suprema over the hypothesis class $\mathcal{H}$. Typically, these are intractable to compute for neural nets due to the richness of the class, and approximations would yield lower bounds rather than upper bounds. Several works fail to yield practically computable bounds for neural networks for this reason (Ben-David et al., 2007; Blitzer et al., 2008; Ben-David et al., 2010; Morvant et al., 2012; Mansour et al., 2009; Redko et al., 2019; Cortes & Mohri, 2014; Cortes et al., 2015). There are also some works which deal with so called localized discrepancy which depends on finding a subset of promising classifiers and bounding their performance instead. (Zhang et al., 2020; Cortes et al., 2019) However, this subset is not easy to find in general and as such we view these approaches as intractable also.

[1]By this we mean the whole bound. Some of the works listed have computed one or several parts of their bound. However, the computation is generally done for simpler model classes than neural networks.

Continuing with estimability, we may remove from consideration also those whose non-estimable term cannot be dealt with without assuming that they are small—an untestable assumption which does not follow from overlap and covariate shift. This immediately disqualifies a large swathe of bounds which all include the joint error of the optimal hypothesis on both domains, or some version thereof, a very common non-estimable term in DA bounds (Kuroki et al., 2019; Redko, 2015; Long et al., 2015; Redko et al., 2017; Johansson et al., 2019; Zhang et al., 2019; Dhouib et al., 2020; Shen et al., 2018; Courty et al., 2017b; Germain et al., 2013; Dhouib & Redko, 2018; Acuna et al., 2021). In principle, we might be able to approximate this quantity under overlap, e.g. by using importance sampling. However, this would entail solving a new optimization problem to find a hypothesis which has a low joint error (see discussion after equation 3). If we instead wish to upper bound the term, we must solve an equivalent problem to the one we are trying to solve in the first place.

Three bounds remain: Zhang et al. (2012) use integral probability metrics (IPM) between source and target domains to account for covariate shift in their bound, which are tractable to compute under assumptions on the hypothesis class; Cortes et al. (2010) use importance weights (IW) and Renyi divergences which are well-defined and easily computed under overlap; Germain et al. (2016) use a related metric based on the norm of the density ratio between domains with similar properties. Respectively, the first two results bound sample generalization error using the uniform entropy number and the covering number. These measures are intractable to compute for neural networks, and while they may be upper bounded by the VC dimension (Wainwright, 2019), this is typically large enough to yield uninformative guarantees (Zhang et al., 2021). In contrast, Germain et al. (2016) use a PAC-Bayes analysis which we can apply to neural networks by specifying prior and posterior distributions over network weights. Using Gaussian distributions for both priors and posteriors, the PAC-Bayes bound can be readily computed. Thus, to enable closer comparison and tractable computation, in the coming sections, we unify each bounds' dependence on sample generalization by adapting IW and IPM bounds to the PAC-Bayes framework. For completeness, we include an additional PAC-Bayes bound from the UDA literature, which has a non-estimable term, namely Germain et al. (2013).

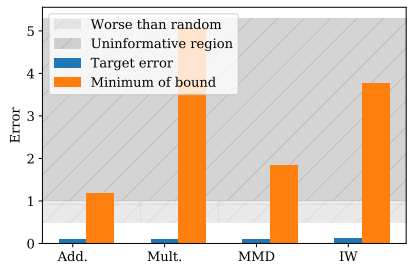


Figure 1: Best bounds achieved without data-dependent priors on the MNIST/MNIST-M task using LeNet-5 as well as the target error for the same posterior hypothesis. Note that all the bounds are vacuous, i.e. they are above one.

The selected bounds are given in Sections 2.3–2.4. Germain et al. (2016) will be referred to as the multiplicative bound (**Mult**), Germain et al. (2013) as the additive bound (**Add**), the adaptation of importance weighting to PAC-Bayes as **IW**; and the adaptation of IPM bounds as **MMD**. Unfortunately, also the resulting PAC-Bayes bounds are uninformative for even simple image classification UDA tasks; see Figure 1. Consistent with our understanding of standard learning with neural networks, *we find both classical PAC and PAC-Bayes bounds vacuous in the tasks most frequently used as empirical benchmarks in papers deriving UDA bounds.* Next, we detail how to make use of data-dependent priors to get the tightest possible bounds.

2.2 Tighter sample generalization guarantees using PAC-Bayes with data-dependent priors

PAC-Bayes theory studies generalization of a posterior distribution ρ over hypotheses in $\mathcal{H}$, learned from data, in the context of a prior distribution over hypotheses, π. The generalization error in ρ may be bounded using the divergence between ρ and π as seen in the following classical result due to McAllester.

Theorem 1 (Adapted from Thm. 2 in McAllester (2013))**.** *For a prior π and posterior ρ on $\mathcal{H}$, a bounded loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ and any fixed $\gamma, \delta \in (0, 1)$, we have w.p. at least $1 - \delta$ over the draw of m*

samples from $\mathcal{D}$, with $D_{\mathrm{KL}}(p\|q)$ the Kullback-Liebler (KL) divergence between p and q,

$$\mathop{\mathbb{E}}_{h\sim\rho} R_{\mathcal{D}}(h) \leq \frac{1}{\gamma} \mathop{\mathbb{E}}_{h\sim\rho} \hat{R}_{\mathcal{D}}(h) + \frac{D_{\mathrm{KL}}(\rho\|\pi) + \ln(\frac{1}{\delta})}{2\gamma(1-\gamma)m} \ .$$

The bound in Theorem 1 grows loose when prior π and posterior ρ diverge—when the posterior is sensitive to the training data. When learning with neural networks, π and ρ are typically taken to be distributions on the weights of the network before and after training. However, the weights of a trained deep neural network will be far away from any uninformed prior after only a few epochs. For this reason, Dziugaite et al. (2021) developed a methodology, based on work by Ambroladze et al. (2007) and Parrado-Hernández et al. (2012), for learning *data-dependent* neural network priors by a clever use of sample splitting. To ensure that the bound remains valid, any data which is used to fit the prior must be independent of the data used to evaluate the bound. In this work, we learn π and ρ following Dziugaite et al. (2021), as described below.

1. A fraction $\alpha \in [0, 1)$ is chosen and the available training data, S, is split randomly into two parts, S_α and $S \setminus S_\alpha$ of size αm and $(1-\alpha)m$, respectively.
2. A neural network is randomly initialized and trained using stochastic gradient descent on S_α for one epoch. From this we get the weights, w_α.
3. The same network is trained, starting from w_α, on all of S until a stopping condition is satisfied. In this work, we terminate training after 5 epochs. We save the weights, w_ρ.
4. From w_α and w_ρ we create our prior and posterior from Normal distributions centered on the learned weights, $\pi = \mathcal{N}(w_\alpha, \sigma I)$ and $\rho = \mathcal{N}(w_\rho, \sigma I)$ respectively. σ is a hyperparameter governing the specificity (variance) of the prior which may be chosen when evaluating.
5. Finally, we use the learned prior and posterior to evaluate the bounds on $S \setminus S_\alpha$.

2.3 PAC-Bayes bounds from the domain adaptation literature

Both the additive and multiplicative PAC-Bayes UDA bounds described in Section 2.1 are defined in Germain et al. (2020) and make use of a decomposition of the zero-one risk into the *expected joint error*

$$e_{\mathcal{D}}(\rho) = \mathop{\mathbb{E}}_{h,h'\sim\rho\times\rho} \mathop{\mathbb{E}}_{x,y\sim\mathcal{D}} \ell(h(x), y)\ell(h'(x), y),$$

which measures how often two classifiers drawn from ρ make the same errors, and the *expected disagreement*

$$d_{\mathcal{D}_x}(\rho) = \mathop{\mathbb{E}}_{h,h'\sim\rho\times\rho} \mathop{\mathbb{E}}_{x\sim\mathcal{D}_x} \ell(h(x), h'(x)),$$

which measures how often two classifier disagree on the labeling of the same point. Empirical variants $\hat{e}_{\mathcal{D}}(\rho)$ and $\hat{d}_{\mathcal{D}_x}(\rho)$ replace expectations with sample averages analogous to equation 2.

In the bound of Germain et al. (2016), the sample generalization component (KL-divergence between prior and posterior) is multiplied with a domain shift component (supremum density ratio).

Theorem 2 (Multiplicative bound, Germain et al. (2016))**.** *For any real numbers $a, b > 0$ and $\delta \in (0, 1)$, it holds, under Assumption 1, with probability at least $1-\delta$ over labeled source samples $S \sim (\mathcal{S})^m$ and unlabeled target samples $T_x \sim (\mathcal{T}_x)^n$, with constants $a' = \frac{a}{1-e^{-a}}$, $b' = \frac{b}{1-e^{-b}}$, for every posterior ρ on $\mathcal{H}$ that*

$$\mathop{\mathbb{E}}_{h\sim\rho} R_{\mathcal{T}}(h) \leq a'\frac{1}{2}\hat{d}_{\mathcal{T}_x} + b'\beta_\infty(\mathcal{T}\|\mathcal{S})\hat{e}_{\mathcal{S}} + \Big(\frac{a'}{na} + \frac{b'\beta_\infty(\mathcal{T}\|\mathcal{S})}{mb}\Big)\Big(2D_{\mathrm{KL}}(\rho\|\pi) + \ln\frac{2}{\delta}\Big) + \eta_{\mathcal{T}\setminus\mathcal{S}} \ ,$$

with $\beta_\infty(\mathcal{T}\|\mathcal{S}) = \sup_{x\in\mathrm{supp}(\mathcal{S}_x)} \mathcal{T}_x(x)/\mathcal{S}_x(x)$,[2] and $\eta_{\mathcal{T}\setminus\mathcal{S}} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{T}}[\mathbb{1}[(x, y) \notin \mathrm{supp}(\mathcal{S})]] \sup_{h\in\mathcal{H}} R_{\mathcal{T}\setminus\mathcal{S}}(h)$.

[2]Terms β_q based on the q:th moment of the density ratio for $q < \infty$ are considered in (Germain et al., 2020) but not here.

The bound is simplified slightly in our setting due to Assumption 1 as $\eta_{\mathcal{T}\setminus\mathcal{S}}$ will be 0. By Bayes rule, β_∞ can be computed as the maximum ratio between conditional probabilities of an input being sampled from $\mathcal{T}$ or $\mathcal{S}$. The second result due to Germain et al. is additive in its interaction between domain and sample generalization terms.

Theorem 3 (Additive bound, Germain et al. (2013))**.** *For any real numbers $\omega, \gamma > 0$ and $\delta \in (0,1)$, with probability at least $1-\delta$ over labeled source samples $S \sim (\mathcal{S})^m$ and unlabeled target samples $T_x \sim (\mathcal{T}_x)^m$; for every posterior ρ on $\mathcal{H}$, it holds with constants $\omega' = \frac{\omega}{1-e^{-\omega}}$ and $\gamma' = \frac{2\gamma}{1-e^{-2\gamma}}$ that*

$$\mathop{\mathbb{E}}_{h\sim\rho} R_{\mathcal{T}}(h) \leq \mathop{\mathbb{E}}_{h\sim\rho} \omega' \hat{R}_{\mathcal{S}}(h) + \gamma' \frac{1}{2} \hat{Dis}_\rho(S, T_x) + \Big(\frac{\omega'}{\omega} + \frac{\gamma'}{\gamma}\Big) \frac{D_{\mathrm{KL}}(\rho\|\pi) + \log\frac{3}{\delta}}{m} + \lambda_\rho + \frac{1}{2}(\gamma' - 1),$$

where $\hat{Dis}_\rho(S, T_x) = |\hat{d}_{\mathcal{T}_x} - \hat{d}_{\mathcal{S}_x}|$ is the empirical domain disagreement, $\lambda_\rho = |e_{\mathcal{T}}(\rho) - e_{\mathcal{S}}(\rho)|$.

Next we will combine the main techniques used to account for domain shift in Cortes et al. (2010) and Zhang et al. (2012) with the PAC-Bayes analysis of Theorem 1, producing two corollaries for the UDA setting.

2.4 Adapting the classical PAC-Bayes bound to unsupervised domain adaptation

First, we adapt the bound in Theorem 1 to UDA by incorporating importance weighting (Shimodaira, 2000; Cortes et al., 2010).We define a weighted loss $\ell^w(h(x), y) = w(x)\ell(h(x), y)$ where $w(x) = \frac{T(x)}{S(x)}$ and ℓ is the zero-one loss. The risk of a hypothesis using this loss is denoted by R^w.

Corollary 1. *(IW bound) Consider the conditions in Theorem 1 and let $\beta_\infty = \sup_{x\sim\mathcal{X}} w(x)$. We have, for any choice of $\gamma, \delta \in (0,1)$ and any pick of prior π and posterior ρ on $\mathcal{H}$,*

$$\mathop{\mathbb{E}}_{h\sim\rho} R_{\mathcal{T}}(h) \leq \frac{1}{\gamma} \mathop{\mathbb{E}}_{h\sim\rho} \hat{R}^w_{\mathcal{S}}(h) + \beta_\infty \frac{D_{\mathrm{KL}}(\rho\|\pi) + \ln(\frac{1}{\delta})}{2\gamma(1-\gamma)m} \ .$$

Proof. Since Theorem 1 holds for loss functions mapping onto $[0,1]$, we divide the weighted loss ℓ^w by the maximum weight, β_∞. The argument then follows naturally when we apply Theorem 1 with the loss function $\frac{\ell^w}{w_{max}}$.

$$\mathop{\mathbb{E}}_{h\sim\rho,(x,y)\sim\mathcal{T}}\Big[\frac{\ell(h(x),y)}{\beta_\infty}\Big] = \mathop{\mathbb{E}}_{h\sim\rho,(x,y)\sim\mathcal{S}}\Big[\frac{\ell^w(h(x),y)}{\beta_\infty}\Big] \leq \frac{1}{\gamma\beta_\infty}\hat{R}^w_{\mathcal{S}} + \frac{D_{\mathrm{KL}}(\rho\|\pi) + \ln(\frac{1}{\delta})}{2\gamma(1-\gamma)m}$$

The first equality holds due to Assumption 1 and the definitions of w and ℓ^w. □

Now we continue with applying an argument based on integral probability metrics (IPM) drawn from Zhang et al. (2012) and similar works. IPMs, such as the kernel maximum mean discrepancy (MMD) (Gretton et al., 2012) and the Wasserstein distance have been used to give tractable and even differentiable bounds on target error in UDA to guide algorithm development (Courty et al., 2017a; Long et al., 2015). The kernel MMD is a IPM, defined as follows, in terms of its square, given a reproducing kernel $k(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

$$\mathbf{MMD}_k(P, Q)^2 = \mathop{\mathbb{E}}_{X\sim P, X'\sim P}[k(X, X')] - 2\mathop{\mathbb{E}}_{X\sim p, Y\sim Q}[k(X, Y)] + \mathop{\mathbb{E}}_{Y\sim Q, Y'\sim Q}[k(Y, Y')].$$

Here, X and Y are random variables, and X' is an independent copy of X with the same distribution and Y' is an independent copy of Y. This measures a notion of discrepancy between the distributions P and Q based on their samples.

We combine the MMD with the bound from McAllester (2013) to arrive at what we will call the **MMD** bound.

Corollary 2. *(MMD bound) Let $\overline{\ell}_h(x) = \mathbb{E}[\ell(h(x), Y) \mid X = x]$ be the expected pointwise loss at $x \in \mathcal{X}$ and assume that, for any $h \in \mathcal{H}$, $\overline{\ell}_h$ can be uniformly bounded by a function in reproducing-kernel Hilbert $\mathcal{L}$ space with kernel k such that $\forall x, x' \in \mathcal{X} : 0 \leq k(x, x') \leq K$. Then, under Assumption 1, with $\gamma, \delta \in (0,1)$ and probability $\geq 1-\delta$ over labeled source samples $S \sim (\mathcal{S})^m$ and unlabeled target samples $S'_x \sim \mathcal{T}_x^m$,*

$$\mathop{\mathbb{E}}_{h\sim\rho} R_{\mathcal{T}}(h) \leq \frac{1}{\gamma} \mathop{\mathbb{E}}_{h\sim\rho} \hat{R}_{\mathcal{S}}(h) + \frac{D_{\mathrm{KL}}(\rho\|\pi) + \log\frac{2}{\delta}}{2\gamma(1-\gamma)m} + \hat{\mathrm{MMD}}_k(\mathcal{S}, \mathcal{T}) + 2\sqrt{\frac{K}{m}}\Big(2 + \sqrt{\log\frac{4}{\delta}}\Big) \ , \tag{4}$$

where $\hat{\mathrm{MMD}}_k(\mathcal{S}, \mathcal{T})$ *is the biased empirical estimate of the maximum mean discrepancy between* $\mathcal{S}$ *and* $\mathcal{T}$ *computed from* S_x *and* S'_x*, see eq. (2) in Gretton et al. (2012).*

Proof. By assumption, for any hypothesis $h \in \mathcal{H}$,

$$R_{\mathcal{T}}(h) = R_{\mathcal{S}}(h) + \mathbb{E}_{\mathcal{T}}[\ell(h(X), Y)] - \mathbb{E}_{\mathcal{S}}[\ell(h(X), Y)] \leq R_{\mathcal{S}}(h) + \sup_{l \in \mathcal{L}} |\mathbb{E}_{\mathcal{T}_x}[l_h(X)] - \mathbb{E}_{\mathcal{S}_x}[l_h(X)]| \ .$$

The inequality holds because $\mathbb{E}_{\mathcal{S}}[\ell(h(x), Y) \mid X = x] = \mathbb{E}_{\mathcal{T}}[\ell(h(x), Y) \mid X = x]$ due to Assumption 1 (covariate shift) and the assumption that ℓ_h is uniformly bounded by a function in $\mathcal{L}$. The RHS of the inequality is precisely $\mathrm{MMD}_{\mathcal{L}}$ and the full result follows by linearity of expectation (over ρ), since the MMD term is independent of h, and application of Theorem 1. The right-most term in equation 4 follows from a finite-sample bound on MMD, Theorem 7 in Gretton et al. (2012), and a union bound w.r.t. δ. □

Representation learning. When no function family $\mathcal{L}$ satisfying the conditions of Corollary 2 is known, an additional unobservable error term must be added to the bound, to account for excess error. Bounds based on the MMD and other IPMs have been used heuristically in representation learning to find representations which minimize the induced distance between domains and achieve better domain adaptation (Long et al., 2015). However, even if covariate shift holds in the input space, these are not guaranteed to hold in the learned representation. For this reason, we do not explore such approaches even though they might hold some promise. An example of this is recent work by Wu et al. (2019) which provided some interesting ideas about assumptions which constrains the structure of the source and target distributions under a specific representation mapping. Exploring such ideas further is an interesting direction for future work.

3 Experimental setup

We describe the experimental setup briefly, leaving more details in Appendix A. We examine the dynamics of the chosen bounds (**Add**, **Mult**, **MMD**, **IW**), which parts of the bounds dominate their value, the effect of varying the amount of data used to inform the prior and if the bounds have utility for early stopping indication or model selection. In addition to these we also want to answer if these bounds practically useful as guarantees, and if not, what is lacking and what future directions should be explored to reach our desiderata? Since the term λ_ρ in **Add** (Theorem 3) depends on target labels, we give access to these for purposes of analysis and illustration, keeping in mind that the bound is not fully estimable in practice.

We perform experiments on two image classification tasks, described further below, using three different neural network architectures. The architectures used are: a modified version of LeNet-5 due to Zhou et al. (2019), a fully connected network and a version of ResNet50 (He et al., 2016). These specific architectures were picked as examples due to their varying parameter size and complexity. The experiments are repeated for 5 distinct random seeds, with the exception of the varying of image size where we only conduct the experiment for one seed.

We learn prior and posterior network weights as described in Section 2.2. When both sets of weights have been trained, we use these as the means for prior and posterior distributions π and ρ, chosen to be isotropic Gaussians with equal variance, σ. Each bound is calculated for each pair of prior and posterior. To estimate the expectation over the posterior we sample 5 pairs of classifiers from the posterior and average their bound components (e.g., source risk) to get an estimate of the bound parts. When this has been calculated, we perform a small optimization step to choose the free parameters of the different bound through a simple grid search over a small number of combinations. We use the combination that produces the lowest minimum bound and account for testing all k combinations by applying a union bound argument, modifying the certainty parameter δ to δ/k. When calculating the MMD, we use the linear statistic for the MMD as detailed in Gretton et al. (2012) with the kernel $k(x, y) = \exp(\frac{-\|x-y\|^2}{2\kappa^2})$. This calculation is averaged over 10 random shuffles of the data for a chosen bandwidth, $\kappa > 0$. The process is repeated for different choices of bandwidth (see Appendix for details) and the maximum of the results is taken as the value of the MMD. Note that we calculate the MMD in the input space and have not adjusted it for sample variance. When calculating the importance weights we assume here that the maximum weight, β_∞, is uniformly bounded as

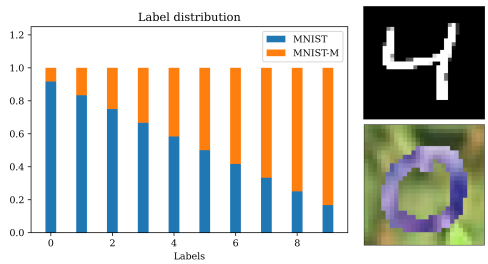


Figure 2: Example of source label densities in the task with the mix of MNIST and MNIST-M samples. An example from each of the two data sets can be seen on the right, the upper one being from MNIST and lower one from MNIST-M. The target domain is the complement of source samples.

the bound would potentially be vacuous otherwise. In addition, we will consider the importance weights to be perfectly computed for simplicity.

We construct two tasks from standard data sets which both fulfill Assumption 1 by design, one based on digit classification and one real-world task involving classification of X-ray images. These are meant to represent realistic UDA tasks where neural networks are the model class of choice.

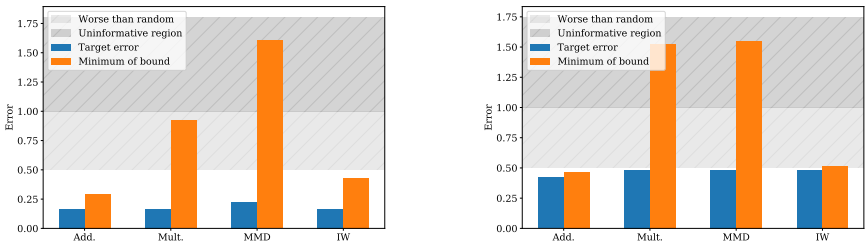
3.1 Task 1: MNIST mixture

MNIST (Lecun, 1998) is a digit classification data set containing 70000 images widely used as a benchmark for image classifiers. MNIST-M was introduced by Ganin et al. (2016) to study domain adaptation and is a variation of MNIST where the digits have been blended with patches taken from images from the BSDS500 data set (Arbeláez et al., 2011). We use MNIST and MNIST-M to construct source and target domains, both of which contain samples from each data set, but with different label density for images from MNIST and MNIST-M. To create the source data set, we start with images labeled "0" by adding 1/12th of the samples from MNIST-M and 11/12th of the samples from MNIST, we increase the proportion from MNIST-M by 1/12 for each subsequent label, "1", "2", and so on. The complement of the source samples is then used as the target data, see Figure 2 for an illustration. We make this into a binary classification problem by relabeling digits 0-4 to "0" and the rest to "1". The supremum density ratio $\beta_\infty \approx 11$ (see Theorem 2) is known and the mixture guarantees overlap in the support of the domains (Assumption 1).

3.2 Task 2: X-ray mixture

ChestX-ray14 (Wang et al., 2017) and CheXpert (Irvin et al., 2019) are data sets of chest X-rays and labeled according to the presence of common thorax diseases. The data sets contain 112,120 and 224,316 labeled images, respectively. Since the two data sets do not have full overlap in labels, we use the subset of labels for which there is overlap. The labels which occur in both data sets are: No finding, Cardiomegaly, Edema, Consolidation, Atelectasis, and Pleural Effusion. In addition, there is an uncertainty parameter present in the CheXpert data set which indicates how certain a label is. As we consider binary classification, we set all labels that are uncertain to positive. Therefore, a single image in the CheXpert data set might have multiple associated labels. For most experiments we resize all the images in the data sets to 32x32 to be able to use the same architectures for both tasks. However, we also conduct a small experiment with ResNet50 where larger image sizes are considered.

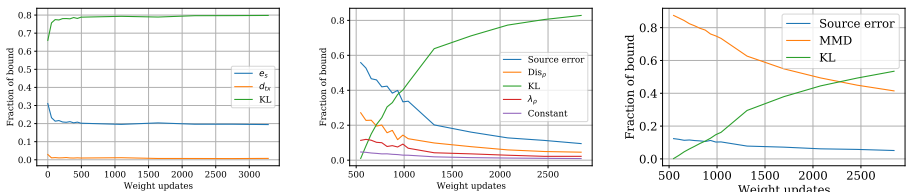
We take 20% of chestX-ray14 and add it to CheXpert to create our source data set. The target is the remaining part of ChestX-ray14 which is then 89,696 images compared to the 246,072 of the source. With this, we know from Appendix B that $\beta_\infty \approx 11$ and we have overlap. We turn this into a binary classification problem in a one-vs-rest fashion by picking one specific label to identify, relabeling images with "1" if it is



(a) MNIST/MNIST-M task.

(b) CheXpert+ChestX-ray14 task

Figure 3: The tightest bounds achieved on the LeNet-5 architecture. This illustrates the tightening effect of using data-dependent priors. Non-vacuous bounds are obtained when using data to inform the prior. The shaded area between 0.5 and 1 is where a random classifier would perform on average, and the shaded area above 1 signifies vacuity.



(a) **Mult** bound, $\alpha = 0$

(b) **Add** bound, $\alpha = 0.3$

(c) **MMD** bound, $\alpha = 0.3$

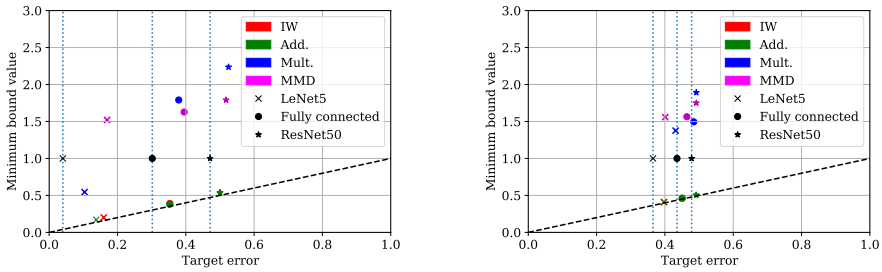
Figure 4: An illustration of constituent parts for three of the bounds with the fully connected architecture on the MNIST mixture task. $\sigma = 0.03$

present, and setting the labels of images with any other finding to "0". In this work, we consider only the task of classifying "No Finding".

4 Results

As expected, when we compute bounds with data-dependent priors, we achieve bounds which are substantially tighter than without them, as seen clearly by comparing Figure 1 to Figure 3. We also observe that the additive bound (**Add**) due to Germain et al. (2013) is the tightest overall for both tasks, followed closely by the **IW** bound. The latter is not so surprising as when we apply data-dependent priors, there is effectively a point in training where the D_{KL}-divergence between prior and posterior networks is very small. Moreover, due to overlap, the weighted source error is equal to the target error in expectation. Thus the only sources of looseness left is the error in the approximation of the expectation over the posterior and the $\log \frac{1}{\delta}$ term which is very small here. We can also see in Figure 5a and 5b that the minimum of the **IW** bound is often very close to the minimum of the additive bound. However, unlike **IW**, the **Add** bound relies on access to target labels in to compute the term λ_ρ (see further discussion below).

The evolution of the different bounds during training is shown for both tasks in Figure 6. Of course, all bounds will increase at some point as training progresses and the prior and posterior diverges further from each other and D_{KL} increases. While **Add** is consistently very tight, we note that the λ_ρ term which we cannot observe might be a significant part of the bound when the D_{KL}-term is low as we can see in Figure 4b. This is an issue for the additive bound since if we have sufficiently small variance of the posterior then the



(a) MNIST/MNIST-M (b) CheXpert+ChestX-ray14

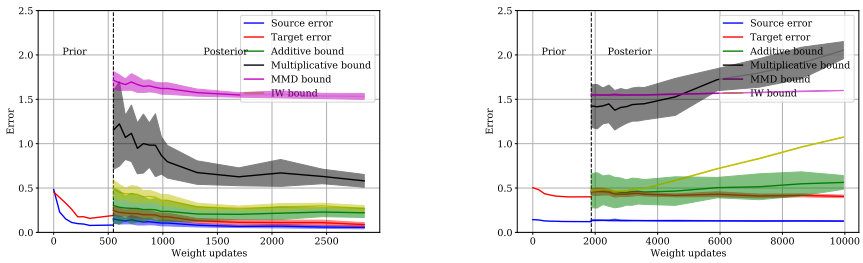
Figure 5: An illustration of the minimum bound value achieved by each of the three architectures on both tasks. The lowest target error achieved is indicated by a black marker with a vertical dotted line through it.

disagreement will be low, using informed priors will make the D_{KL} small while using neural networks often lead to having a low source error. This will leave only the constant term, $\log \frac{1}{\delta}$ and the unobservable λ_ρ terms and in those situations the bound might even be dominated by the unobservable term.

The multiplicative bound of (Germain et al., 2016) (**Mult**) suffers from the amplification of the source error e_S and D_{KL} term by the factor β_∞, and is generally larger than the **Add** and **IW** bounds. Conceptually, the **Mult** and **IW** bounds are similar, but in the former, the loss is multiplied uniformly by the largest weight. For tasks where certain inputs with high loss are more uncommon in the target domain than in the source domain, this is especially detrimental. The **MMD** bound is initially dominated by the MMD distance between inputs from the source and target domains, as shown in 4c, which is large and independent of the learned hypothesis. As such, this term cannot be reduced by optimization, without, for example, computing it in representation space (Long et al., 2015). With this approach, unobservable errors due to non-invertible representations must be accounted for (Johansson et al., 2019).

Experiments on using bounds for early stopping and model selection with different architectures yield the results seen in Figure 5. We can see that the errors achieved by terminating training at the smallest bound value (colored markers) do not coincide with the best-achieved target performance during training (denoted by the vertical dotted lines). Clearly, the bounds are not tight enough to do early stopping. This is a result of the sample generalization term D_{KL} increasing during training. For other analyses, this need not be the case. For larger architectures, the early-stopped models are closer to the best target models. If we instead look at the same figure again, but this time focus on utility for model selection we find something interesting. It seems that the bounds might be useful in this regard as they consistently have lower values for architectures/models which perform well. However, to be able to say this conclusively a more thorough study with different learning setups must be done. Both of the two previous observations should be contextualised with the fact that the domain shift terms are not dependent on the model as such, but amplify looseness in the case of the **Mult** and **IW** bounds. For the **MMD** and **Add** bounds, increased looseness during training is an artifact only of sample generalization.

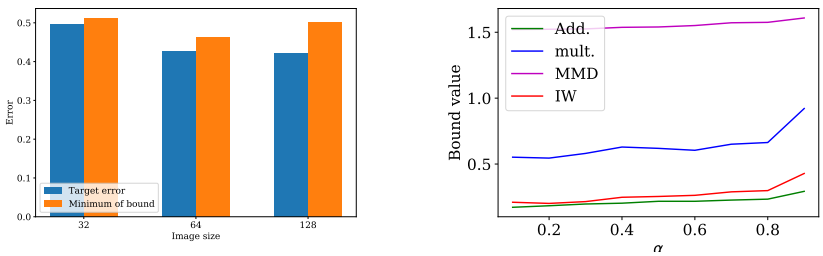
As we can see in Figure 7a, when we vary the size of the images we give to the ResNet50 architecture we observe that the error seems to decrease for the larger image sizes. Although, the minimum bound value achieved does not seem to follow the same trend consistently. This is likely the result of the amount of epochs trained for both prior and posterior. In Figure 7b, we see that the choice of prior sample proportion α makes some change to the smallest bound achieved. We also see the minimum bound values grow for large values of α, indicating that the remaining data is better spent calculating the bound than informing the prior in this case. We can also infer from Figure 1 that using no data to inform the prior is worse than using some. The overall shape is consistent with the results reported in Dziugaite et al. (2021, Figure 1).



(a) MNIST/MNIST-M task

(b) CheXpert+ChestX-ray14 task

Figure 6: Bounds evaluated at different points during training of the LeNet-5 architecture. $\alpha = 0.3$, $\sigma = 0.03$. The shaded areas represent one standard deviation.



(a) Minimum bound value and target error on the X-ray task with different sizes of input images. Architecture used is ResNet50, $\alpha = 0.3$, $\sigma = 0.003$

(b) The minimum value of the bounds on the MNIST/MNIST-M task for different values of α. The architecture used is LeNet-5.

Figure 7: a) show results of varying image sizes when training the ResNet50 network on the X-ray task. b) shows how the minimum value of the bound varies with α.

5 Discussion

From our survey of the literature, it is clear that only a small handful of analyses of UDA generalization can be informative as practical bounds on target domain performance. The main obstacle for computing existing bounds is that they are vacuous or intractable to compute for the kinds of models which perform the best on common UDA benchmarks—deep neural networks. A potential remedy is the use of PAC-Bayes bounds, which perform well once they are applied with data-dependent priors; without this they are vacuous. In our experiments, the **Add** bound with the unobservable term is the tightest which is unsurprising given its dependence on target labels. Furthermore, we note that the application of importance weights also performs very well as the setting is sufficiently benign. As such we can say that in this setting we can achieve the desiderata of a tractably computable, tight bound using the **IW** bound. However, recall that the guarantee we get is on a distribution over classifiers and not on one specific classifier. It should be noted, however, that the **IW** bound can become vacuous in certain situations where the worst-case density ratio, β_∞, is large and either the D_{KL} term or errors on underrepresented classes is large enough.

We found that the lowest value of the bounds achieved during training does not in general correspond to the best performing model on target. This tells us that these bounds are not useful metrics for early stopping. Further, the findings for using bound values for model selection are inconclusive, more experiments have to be conducted to answer this question satisfactorily. During training, we see that the dynamics are dominated by

the KL-divergence term, inherent to PAC-Bayes analysis, as training progresses. This reinforces our view that these bounds might be useful in getting performance estimates of methods at one particular point and not over several points during training if we do not have access to a large sample. This issue might be ameliorated by regularizing towards the prior during training, although this introduces yet another optimization as we now have to find the optimal regularization strength. In addition, it is not certain whether this will have any adverse effects on the final performance of the learned classifier.

A limitation of this work is that the bounds are cumbersome to compute and it is possible to do several optimizations in the process of producing the bounds. We have tried to do as few as possible in the name of practicality. We list some of the possible further optimizations in Appendix A for the reader's consideration. The impracticality of computing PAC-Bayes bounds is a known issue that has had some work done by Viallard et al. (2021) where they introduce an approach which would remove the computation of expectation over the posterior. In addition, in this work the computation of test errors have dominated the computation time. To produce the results one has to compute the predictions at least 50 times(5 pairs of sampled models from the posterior and 5 random seeds) for each datapoint in the bound for a single choice of prior. This will naturally consume increasing amounts of time with larger data sets.

Furthermore, the overlap assumption will not hold for all real-world applications. In fact, many of the benchmarks for algorithm development, such as the SVHN→MNIST task (Ganin et al., 2016) blatantly violate overlap, since images of house numbers and handwritten digits differ vastly in pixel space. Examples where overlap holds by definition include when the target domain represents a subpopulation of a larger population given by the source domain, e.g., women (target) among all patients (source) with a medical condition. Although an easy learning problem on its face, the optimal model in the full population may not be optimal for the subpopulation. Even when overlap is violated, many share the intuition that overlap may hold in a transformed space (Wu et al., 2019), representative of the core aspects of the problem—a digit is a digit, whether on a house or a postcard.

The strictness of the overlap assumption has been studied by D'Amour et al. (2021) where it was found that even for Gaussian distributions with insubstantial differences in mean parameters, overlap vanishes in high dimensions. Motivated by this fact we might wish to adopt relaxed versions of our assumptions or completely novel ones which still guarantee consistent estimation. A first step could be to require overlap only in a transformed space, not in the input space, like in Wu et al. (2019) or only requiring overlap in specific regions and leveraging assumptions on "closeness" in the other regions, as in Johansson et al. (2019). Further, task-specific assumptions are likely needed for a more complete description of out-of-distribution generalization. We mean task-specific in the sense that the assumptions will depend on the structure on the problem and the data-generating process (Hansen, 2008) or other approaches. Overcoming this gap is a important direction of future study.

Another limitation of this work is that the hypotheses do not optimize for adaptation to the target domain, which might be achieved through representation learning as in Ganin et al. (2016) or minimization of a weighted loss (Shimodaira, 2000). Our setting is representative for tasks where the target domain is unknown during training, but known when computing the bounds. Further, in this work we have assumed that we are able to estimate the importance weights exactly which may not be feasible in high-dimensional settings. In addition, there is no guarantee that the estimation error of the weights is small and thus even a small misestimation may have quite large implications for the resulting bound.

Future work regarding generalization bounds should preferably comment upon usefulness of their bound as a practical guarantee for performance, which is something that is often lacking. Ideally this would extend to explicit calculation if the bound is possible to compute. New bounds are often used as inspiration towards new algorithms which are hoped to result in more generalizable models. However, this is seldom guaranteed by theory and verified only in limited settings empirically.

Our results offer indications for how to obtain tractable and tight bounds for neural networks used in UDA tasks with available tools. If overlap can be assumed to hold, then use the **IW** bound, estimate importance weights using density estimation (Sugiyama et al., 2012) or probabilistic classifiers and apply data-dependent priors. The amount of data to use and how long to train your prior etc. are all task dependent and thus some engineering is necessary to pick optimal values. If this cannot be assumed, the most promising approach

to get bounds which fulfill our desiderata in this case would be to use the **MMD** bound as this does not technically rely on overlap and is tractable to compute for neural networks. This relies on the added assumptions of the pointwise loss being bounded by a function in the associated reproducing-kernel Hilbert space, which may or may not hold. The nature of this assumption makes it less useful since no test for this is available absent overlap and is similar in nature to assuming that joint optimal error is small. However, if the function under estimation is believed to be smooth, the assumption is more plausible. In conclusion, it is clear that the general case demands new research, and alternative, task-specific assumptions, to allow tight performance guarantees for realistic problems. In either setting, we conjecture that the tightest bounds will be coupled to the training procedure.

Acknowledgements

This work was supported in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

The computations and data handling were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at Chalmers Centre for Computational Science and Engineering (C3SE) partially funded by the Swedish Research Council through grant agreement no. 2018-05973. Mikael Öhman at C3SE is acknowledged for his assistance concerning technical and implementation aspects in making the code run on the C3SE resources.

References

David Acuna, Guojun Zhang, Marc T. Law, and Sanja Fidler. f-domain adversarial learning: Theory and algorithms. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 66–75. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/acuna21a.html`.

Ehab A. AlBadawy, Ashirbani Saha, and Maciej A. Mazurowski. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Medical Physics*, 45(3):1150–1158, 2018.

Amiran Ambroladze, Emilio Parrado-hernández, and John Shawe-taylor. Tighter pac-bayes bounds. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension and Pseudodimension Bounds for Piecewise Linear Neural Networks. *Journal of Machine Learning Research*, pp. 1–17, 2019.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, May 2010.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning Bounds for Domain Adaptation. *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pp. 8, 2008.

Daniel C. Castro, Ian Walker, and Ben Glocker. Causality matters in medical imaging. *Nature Communications*, 11(1):3673, July 2020. ISSN 2041-1723.

Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, January 2014. ISSN 0304-3975.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning Bounds for Importance Weighting. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 23, pp. 442–450. Curran Associates, Inc., 2010.

Corinna Cortes, Mehryar Mohri, and Andrés Muñoz Medina. Adaptation Algorithm and Theory Based on Generalized Discrepancy. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 169–178, Sydney NSW Australia, August 2015. ACM. ISBN 978-1-4503-3664-2.

Corinna Cortes, Mehryar Mohri, and Andrés Muñoz Medina. Adaptation based on generalized discrepancy. *Journal of Machine Learning Research*, 20(1):1–30, 2019. URL `http://jmlr.org/papers/v20/15-192.html`.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3733–3742, 2017a.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint Distribution Optimal Transportation for Domain Adaptation. *arXiv:1705.08848 [cs, stat]*, October 2017b. arXiv: 1705.08848.

Sofien Dhouib, Ievgen Redko, and Carole Lartizien. Margin-aware adversarial domain adaptation with optimal transport. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2514–2524. PMLR, 13–18 Jul 2020.

Sofien Dhouib and Ievgen Redko. Revisiting (ϵ , γ, τ)-similarity learning for domain adaptation. *NeurIPS*, pp. 7408–7417, 2018.

Gintare Karolina Dziugaite and Daniel M. Roy. Data-dependent PAC-Bayes priors via differential privacy. *arXiv:1802.09583 [cs, stat]*, April 2019. arXiv: 1802.09583.

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel M. Roy. On the role of data in PAC-Bayes bounds. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, October 2021.

Alexander D'Amour, Peng Ding, Avi Feller, Lihua Lei, and Jasjeet Sekhon. Overlap in observational studies with high-dimensional covariates. *Journal of Econometrics*, 221(2):644–654, 2021. ISSN 0304-4076. doi: https://doi.org/10.1016/j.jeconom.2019.10.014. URL `https://www.sciencedirect.com/science/article/pii/S0304407620302694`.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-Adversarial Training of Neural Networks. *arXiv:1505.07818 [cs, stat]*, May 2016. arXiv: 1505.07818.

Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. A PAC-Bayesian Approach for Domain Adaptation with Specialization to Linear Classifiers. In *International Conference on Machine Learning*, pp. 738–746. PMLR, May 2013. ISSN: 1938-7228.

Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. A new pac-bayesian perspective on domain adaptation. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 859–868, New York, New York, USA, 20–22 Jun 2016. PMLR.

Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. PAC-Bayes and Domain Adaptation. *Neurocomputing*, 379:379–397, February 2020. ISSN 09252312. arXiv: 1707.05712.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Ben B. Hansen. The prognostic analogue of the propensity score. *Biometrika*, 95(2):481–488, 06 2008. ISSN 0006-3444. doi: 10.1093/biomet/asn004. URL `https://doi.org/10.1093/biomet/asn004`.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Jeremy A. Irvin, Pranav Rajpurkar, M. Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, H. Marklund, Behzad Haghgoo, Robyn L. Ball, K. Shpanskaya, J. Seekins, D. Mong, S. Halabi, J. Sandberg, Ricky H Jones, D. Larson, C. Langlotz, B. Patel, M. Lungren, and A. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *AAAI*, 2019.

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

Seiichi Kuroki, Nontawat Charoenphakdee, Han Bao, Junya Honda, Issei Sato, and Masashi Sugiyama. Unsupervised Domain Adaptation Based on Source-Guided Discrepancy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4122–4129, July 2019.

Yann Lecun. Gradient-Based Learning Applied to Document Recognition. *proceedings of the IEEE*, 86(11): 47, 1998.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning Transferable Features with Deep Adaptation Networks. *arXiv:1502.02791 [cs]*, May 2015. arXiv: 1502.02791.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain Adaptation: Learning Bounds and Algorithms. In *Proceedings of the Conference on Learning Theory*, February 2009.

David A. McAllester. Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, COLT' 98, pp. 230–234, New York, NY, USA, July 1998. Association for Computing Machinery. ISBN 978-1-58113-057-7.

David A. McAllester. Pac-bayesian model averaging. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, COLT '99, pp. 164–170, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581131674.

David A. McAllester. A PAC-Bayesian Tutorial with A Dropout Bound. *arXiv e-prints*, 1307: arXiv:1307.2118, July 2013.

Emilie Morvant, Amaury Habrard, and Stéphane Ayache. Parsimonious unsupervised and semi-supervised domain adaptation with good similarity functions. *Knowledge and Information Systems*, 33(2):309–349, November 2012. ISSN 0219-1377, 0219-3116.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. Pac-bayes bounds with data dependent priors. *J. Mach. Learn. Res.*, 13(1):3507–3531, dec 2012. ISSN 1532-4435.

Christian S Perone, Pedro Ballester, Rodrigo C Barros, and Julien Cohen-Adad. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. *NeuroImage*, 194:1–11, July 2019. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2019.03.026.

Ievgen Redko. *Nonnegative matrix factorization for transfer learning*. PhD thesis, Paris North University, 2015.

Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical Analysis of Domain Adaptation with Optimal Transport. *arXiv:1610.04420 [cs, stat]*, July 2017. arXiv: 1610.04420.

Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia. Optimal transport for multi-source domain adaptation under target shift. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 849–858. PMLR, 16–18 Apr 2019.

Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv:2004.11829 [cs, stat]*, August 2020. arXiv: 2004.11829.

Omar Rivasplata, Ilja Kuzborskij, Csaba Szepesvari, and John Shawe-Taylor. PAC-Bayes Analysis Beyond the Usual Bounds. *arXiv:2006.13057 [cs, stat]*, December 2020. arXiv: 2006.13057.

John Shawe-Taylor and Robert C. Williamson. A pac analysis of a bayesian estimator. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, COLT '97, pp. 2–9, New York, NY, USA, 1997. Association for Computing Machinery. ISBN 0897918916.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein Distance Guided Representation Learning for Domain Adaptation. *arXiv:1707.01217 [cs, stat]*, March 2018. arXiv: 1707.01217.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, October 2000. ISSN 0378-3758.

M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge books online. Cambridge University Press, 2012. ISBN 978-0-521-19017-6.

L. G. Valiant. A theory of the learnable. *Communications of the ACM*, pp. 1134–1142, 1984.

Guillermo Valle-Pérez and Ard A Louis. Generalization bounds for deep learning. *arXiv preprint arXiv:2012.04115*, 2020.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 9 1998. ISBN 0471030031.

Paul Viallard, Pascal Germain, Amaury Habrard, and Emilie Morvant. A General Framework for the Disintegration of PAC-Bayesian Bounds. *arXiv:2102.08649 [cs, stat]*, October 2021. arXiv: 2102.08649.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. doi: 10.1017/9781108627771.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 3462–3471, 2017.

Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6872–6881. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/wu19f.html`.

Chao Zhang, Lei Zhang, and Jieping Ye. Generalization Bounds for Domain Adaptation. *Advances in Neural Information Processing Systems*, 25:3320–3328, 2012.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I. Jordan. Bridging Theory and Algorithm for Domain Adaptation. *arXiv:1904.05801 [cs, stat]*, April 2019. arXiv: 1904.05801 version: 1.

Yuchen Zhang, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. On Localized Discrepancy for Domain Adaptation. *arXiv:2008.06242 [cs, stat]*, August 2020. arXiv: 2008.06242.

Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J. Gordon. On Learning Invariant Representation for Domain Adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, January 2019.

Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P. Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: A pac-bayesian compression approach. In *International Conference on Learning Representations*, 2019.

A Experimental details

The experiments were carried out with a modified version of LeNet-5 due to Zhou et al. (2019), a 1024-600-600-2 fully connected network similar to the one used in Rivasplata et al. (2020) and a ResNet50 architecture (He et al., 2016) as mentioned before. Since the 0-1 loss is not differentiable we substitute it with the binary cross entropy loss which is and provides a tight upper bound on the 0-1 loss. We train with SGD with momentum 0.95 as the optimizer with a batch size of 128. The learning rate was chosen to be 3×10^{-3} for the LeNet-5 and fully connected architectures while for ResNet50 3×10^{-4} was used. The images from MNIST and MNIST-M were padded with zeros to 32×32 images. This was done to be able to use them with the ResNet50V2 implementation in Tensorflow which does not support smaller image sizes.

The training procedure went as follows. We load the data and construct our source and target. The source data is then split into an α-fraction S_α which is used to train the prior network on for $|S_\alpha|/b$ iterations, where b is the batch size, to get an informed prior. This is then used as the starting point when training the posterior which is done on all the available data. During training of the posterior network we save 10 network weights during the first epoch and then at the end of every subsequent epoch until termination. We terminate the training of the posterior network when we have trained 5 epochs.

When training the posterior is terminated we save the weights and proceed with the computation of the bound. In contrast with Dziugaite et al. (2021) we not only consider the bound at the point of termination but also at previous points during training. This is done with the goal of gaining an understanding of the bounds' behaviour during training.

We assume that our prior can be modeled by an isotropic gaussian akin to earlier work, this is done to get an easily computable closed form expression of the KL divergence. To pick a good value of the σ parameter one could sweep over some range of values and then use a union bound argument to be able to pick the best result with only a small penalty to the bound. We do not do such optimization and simply pick a value.

We perform a small optimization step when determining the free parameters of the bound. For the bounds from Germain et al. (2020), i.e. a, b, γ and ω, we iterate over values in the range $\{1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}, \ldots, 5 \times 10^{4}, 1 \times 10^{5}\}$ for both free parameters and pick the combination which yields the lowest bound. For the **MMD** and **IW** bounds we pick γ from the range $\{1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}, 5 \times 10^{-2}, 1 \times 10^{-1}, 5 \times 10^{-1}, 9.9 \times 10^{-1}\}$, choosing the one which yields the lowest bound.

A.1 Possible optimization when producing the bounds

When computing these bounds there are a lot of different parameter and hyperparameter choices to make, many of which can be optimized. We first have to train at least one model(depending on how many values of α to consider) with all parameter choices that entails. Then we sample models according to whatever the posterior distribution is; the amount depending on how well we want to estimate the expectation over the posterior. All PAC-Bayes bounds contain some sort of parameter which is free to choose and we must do a at least a rudimentary parameter search to arrive at a good bound. In addition to all these choices of parameters we can of course optimise these bounds even further. Some that we did not perform for this work are: Optimise the representation for smaller MMD, L2 regularisation towards the prior for each specific parameter set (also entails finding the optimal regularization strength) and perform even finer grid searches for the optimal bound parameters to name just a few.

B Importance weights and how to derive them

So assume that we do a mixing of two data sets (let's call them 0 and 1) to form two domains. We want to derive the way we should calculate and subsequently use the importance weights for this situation. We will do this first for the CXR task. In this task the underlying label set is multi-label and as such we need to make it into categorical variables before calculating and applying weights. We achieve this by making the problem into a binary classification problem where we try to predict if there is a finding or not. From this point we may calculate the importance weights as follows:

$$w = \frac{T(x,y)}{S(x,y)} = \frac{T(x|y)T(y)}{S(x|y)S(y)} = \frac{T(x|y)T(y)}{(S(x|y,D=1)S(D=1|y) + S(x|y,D=0)S(D=0|y))S(y)}$$

If we now assume that we have only no examples from data set 0 in the target as in the CXR task then we have the following

$$w = \frac{T(x|y)T(y)}{(S(x|y,D=1)S(D=1|y) + \underbrace{S(x|y,D=0)S(D=0|y)}_{=0,\text{ as T(y)=0 when this is non-zero}})S(y)} = \frac{T(x|y)T(y)}{(S(x|y,D=1)S(D=1|y)S(y)}$$

Now we note that $T(x|y)$ and $S(x|y,D=1)$ cancel as the conditional distribution of these should be the same as we mixed uniformly over the initial label and T(x|y,D=1)=T(x|y). We are thus left with

$$w(y) = \frac{T(y)}{S(y)S(D=1|y)} = \frac{\#\text{examples with label y in T}}{\#T} \Big/ \frac{\#\text{examples with label y in S which come from data set 1}}{\#S}$$

Through this argument we can see that the final importance weight is in the case where we use 20% of the images from data set 1, which will become the target, to mix with data set 0 to become the source. Assume that the initial amount from data set 1 is m_1.

$$w = \frac{\#S}{\#T} \cdot \frac{\#\text{examples w/ label y in T}}{\#\text{examples w/ label y in S which come from data set 1}} = \frac{\#S}{\#T}\frac{0.8m_1}{0.2m_1} = 4\frac{\#S}{\#T}$$

We can do the same type of argument for the MNIST/MNIST-M mix. There we have a more balanced data set where the classes are evenly distributed in amount across source and target.

$$w = \frac{T(x,y)}{S(x,y)} = \frac{T(x|y,D=1)T(D=1|y) + T(x|y,D=0)T(D=0|y))T(y)}{(S(x|y,D=1)S(D=1|y) + S(x|y,D=0)S(D=0|y))S(y)}$$

Since the labels are balanced between the data sets $\frac{T(y)}{S(y)} = 1$. Since we have mixed the datapoints for each label in a uniform fashion we know what $\frac{T(x|y,D=0)}{S(x|y,D=0)} = 1$ for every label. As such we can calculate the weight as

$$w(x|y,D=0) = \frac{\#\text{examples w/ label y in T which come from data set 0}}{\#\text{examples w/ label y in S which come from data set 0}}$$

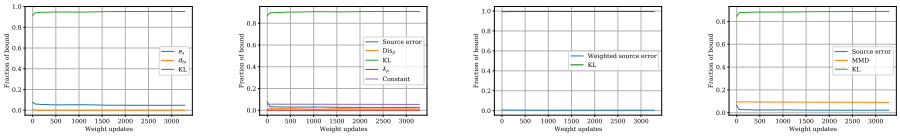
and similar for datapoints from the other data set.

C Additional results

C.1 Constituent parts of bounds

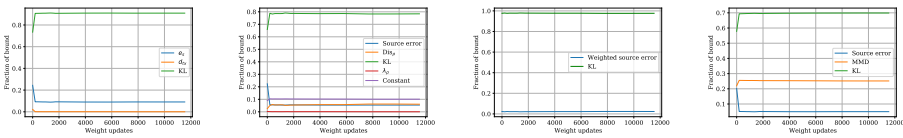
C.2 Best bounds achieved for different prior sample proportions

C.3 Bound during training



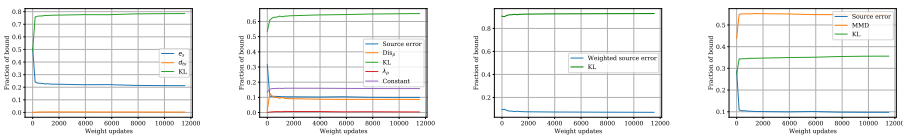
(a) Multiplicative bound (b) Additive bound (c) IW bound (d) MMD bound

Figure 8: An illustration of constituent parts of each of the four bounds with the fully connected architecture on the MNIST mixture task. $\alpha = 0$



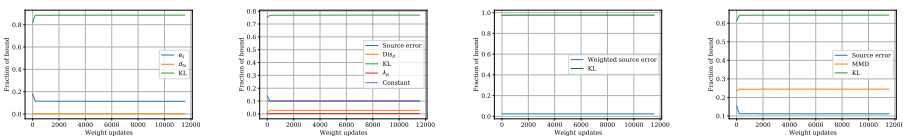
(a) Multiplicative bound (b) Additive bound (c) IW bound (d) MMD bound

Figure 9: An illustration of constituent parts of each of the four bounds with the fully connected architecture on the X-ray task. $\alpha = 0$



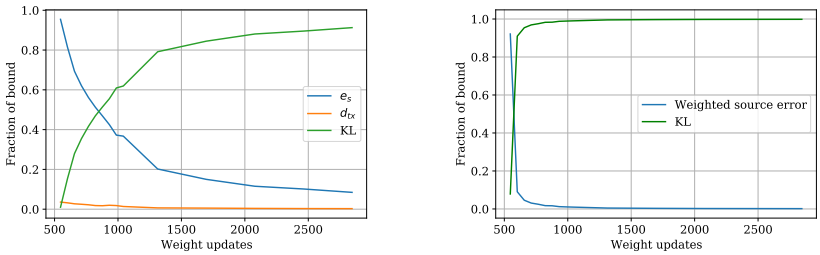
(a) Multiplicative bound (b) Additive bound (c) IW bound (d) MMD bound

Figure 10: An illustration of constituent parts of each of the four bounds with the LeNet-5 architecture on the X-ray task. $\alpha = 0$



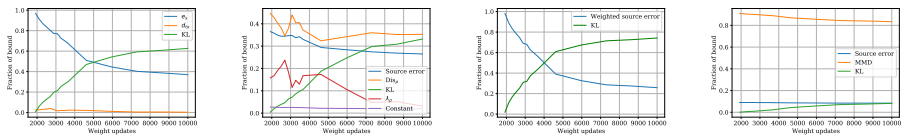
(a) Multiplicative bound (b) Additive bound (c) IW bound (d) MMD bound

Figure 11: An illustration of constituent parts of each of the four bounds with the ResNet50 architecture on the X-ray task. $\alpha = 0$



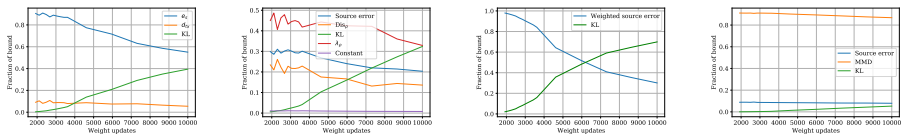
(a) Multiplicative bound, $\sigma = 0.03$ (b) IW bound, $\sigma = 0.03$

Figure 12: An illustration of constituent parts of each of the four bounds with the fully connected architecture on the MNIST mixture task. $\alpha = 0.3$



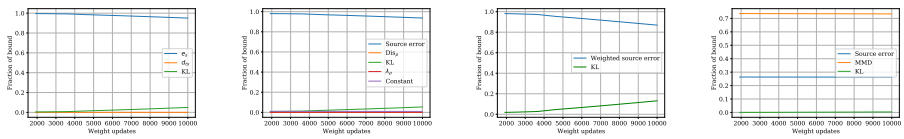
(a) Multiplicative bound (b) Additive bound (c) IW bound (d) MMD bound

Figure 13: An illustration of constituent parts of each of the four bounds with the fully connected architecture on the X-ray task. $\alpha = 0.3$, $\sigma = 0.03$



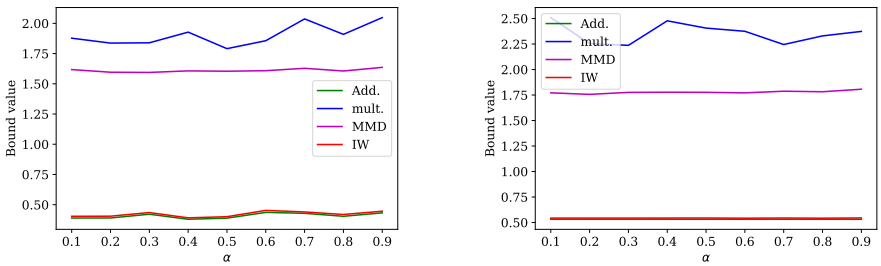
(a) Multiplicative bound (b) Additive bound (c) IW bound (d) MMD bound

Figure 14: An illustration of constituent parts of each of the four bounds with the LeNet-5 architecture on the X-ray task. $\alpha = 0.3$, $\sigma = 0.03$



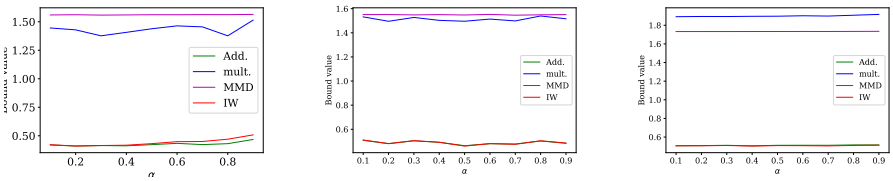
(a) Multiplicative bound (b) Additive bound (c) IW bound (d) MMD bound

Figure 15: An illustration of constituent parts of each of the four bounds with the ResNet50 architecture on the X-ray task. $\alpha = 0.3$, $\sigma = 0.03$



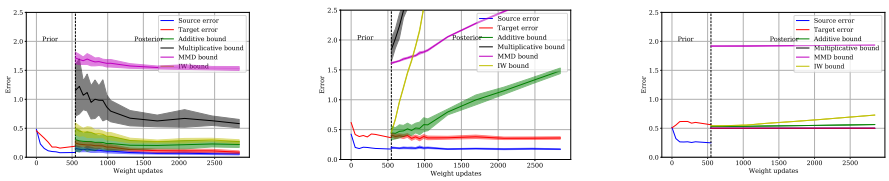
(a) Fully connected (b) ResNet50

Figure 16: The minimum value of the bound on the MNIST mixture task for different values of α.



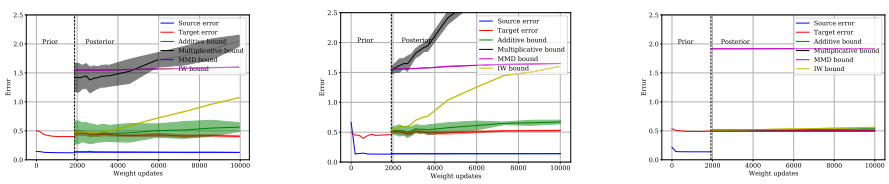
(a) LeNet-5 (b) Fully connected (c) ResNet50

Figure 17: The minimum value of the bounds on the X-ray task for different values of α.



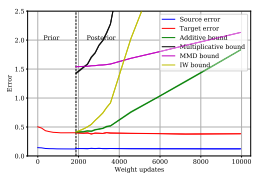
(a) LeNet-5, $\alpha = 0.3$, $\sigma = 0.03$ (b) Fully connected, $\alpha = 0.3$, $\sigma = 0.03$ (c) ResNet50, $\alpha = 0.3$, $\sigma = 0.03$

Figure 18: The evolution of the bounds during training on the MNIST mixture task when we use 30% of our sample to inform the prior.

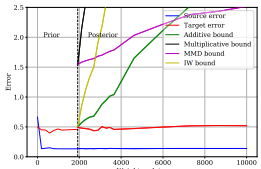


(a) LeNet-5, $\alpha = 0.3$, $\sigma = 0.03$ (b) Fully connected, $\alpha = 0.3$, $\sigma = 0.03$ (c) ResNet50, $\alpha = 0.3$, $\sigma = 0.03$

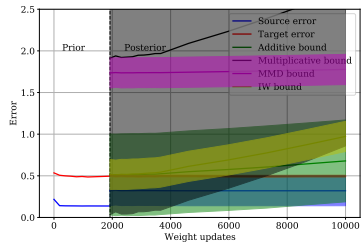
Figure 19: The evolution of the bounds during training on the X-ray task when we use 30% of our sample to inform the prior.



(a) LeNet-5, $\alpha = 0.3$, $\sigma = 0.003$



(b) Fully connected, $\alpha = 0.3$, $\sigma = 0.003$



(c) ResNet50, $\alpha = 0.3$, $\sigma = 0.003$

Figure 20: The evolution of the bounds during training on the X-ray task when we use 30% of our sample to inform the prior.

Paper II

Unsupervised Domain Adaptation by Learning using Privileged Information

A. Breitholtz,A. Matsson, F. D. Johansson

Transactions of Machine Learning Research (September 2024)

doi goes here

Unsupervised Domain Adaptation by Learning Using Privileged Information

Adam Breitholtz* *adambre@chalmers.se*
Department of Computer Science
Chalmers University of Technology

Anton Matsson* *antmats@chalmers.se*
Department of Computer Science
Chalmers University of Technology

Fredrik D. Johansson *fredrik.johansson@chalmers.se*
Department of Computer Science
Chalmers University of Technology

Reviewed on OpenReview: *https://openreview.net/forum?id=saV3MPHOkw*

Abstract

Successful unsupervised domain adaptation is guaranteed only under strong assumptions such as covariate shift and overlap between input domains. The latter is often violated in high-dimensional applications like image classification which, despite this limitation, continues to serve as inspiration and benchmark for algorithm development. In this work, we show that training-time access to side information in the form of auxiliary variables can help relax restrictions on input variables and increase the sample efficiency of learning at the cost of collecting a richer variable set. As this information is assumed available only during training, not in deployment, we call this problem unsupervised domain adaptation by learning using privileged information (DALUPI). To solve this problem, we propose a simple two-stage learning algorithm, inspired by our analysis of the expected error in the target domain, and a practical end-to-end variant for image classification. We propose three evaluation tasks based on classification of entities in photos and anomalies in medical images with different types of available privileged information (binary attributes and single or multiple regions of interest). We demonstrate across these tasks that using privileged information in learning can reduce errors in domain transfer compared to baselines, be robust to spurious correlations in the source domain, and increase sample efficiency.

1 Introduction

Deployment of machine learning (ML) systems relies on generalization from training samples to new instances in a target domain. When these new instances differ in distribution from the source of training data, performance tends to degrade and guarantees are often weak. For example, a supervised ML model trained to identify medical conditions in X-ray images from one hospital may work poorly in another hospital if the two sites have different equipment or examination protocols (Zech et al., 2018). In the *unsupervised domain adaptation* (UDA) problem (Ben-David et al., 2006), *no* labeled examples are available from the target domain and strong assumptions are needed for success. In this work, we ask: How can access to *auxiliary variables* during training help solve the UDA problem and weaken the assumptions necessary to guarantee domain transfer?

In standard UDA, a common assumption is that the object of the learning task is identical in source and target domains but that input distributions differ (Shimodaira, 2000). This "covariate shift" assumption is

*Equal contribution.

plausible in our X-ray example above: Doctors are likely to give the same diagnosis based on X-rays of the same patient from similar but different equipment. However, guarantees for consistent domain adaptation also require either distributional overlap between inputs from source and target domains or known parametric forms of the labeling function (Ben-David & Urner, 2012; Wu et al., 2019; Johansson et al., 2019). Without these, adaptation cannot be verified or guaranteed by statistical means.

Input domain overlap is implausible for the high-dimensional tasks that have become standard benchmarks in the UDA community, including image classification (Long et al., 2013; Ganin et al., 2016) and sentence labeling (Orihashi et al., 2020). If hospitals have different X-ray equipment, the probability of observing (near-)identical images from source and target domains is zero (Zech et al., 2018). Even when covariate shift and overlap are satisfied, large domain differences can have a dramatic effect on sample complexity (Breitholtz & Johansson, 2022). Despite promising developments (Shen et al., 2022), realistic guarantees for practical domain transfer remain elusive.

In supervised ML without domain shift, incorporating auxiliary variables in the training of models has been proposed to improve out-of-sample generalization. For example, learning using *privileged information* (Vapnik & Vashist, 2009; Lopez-Paz et al., 2016), variables available during training but unavailable in deployment, has been proven to require fewer examples compared to learning without these variables (Karlsson et al., 2021). In X-ray classification, privileged information (PI) can come from graphical annotations or clinical notes made by radiologists that are unavailable when the system is used. While PI has begun to see use in domain adaptation, see e.g., Sarafianos et al. (2017) or Vu et al. (2019), and a theoretical analysis exists for linear classifiers (Xie et al., 2020), the literature has yet to fully characterize the benefits of this practice.

We introduce *unsupervised domain adaptation by learning using privileged information* (DALUPI), in which auxiliary variables, related to the outcome of interest, are leveraged during training to improve test-time adaptation when the variables are unavailable. We summarize our contributions below:

- We formalize the DALUPI problem and give conditions under which it is possible to solve it consistently, i.e., to learn a model using privileged information that predicts optimally in the target domain. Importantly, these conditions do not rely on distributional overlap between source and target domains in the input variable (Section 2.1), making consistent learning without privileged information (PI) generally infeasible.

- We propose practical learning algorithms for image classification in the DALUPI setting (Section 3), designed to handle problems with three different types of PI, see Figure 1 for examples. As common UDA benchmarks lack auxiliary variables related to the learning problem, we propose three new evaluation tasks spanning the three types of PI using data sets with real-world images and auxiliary variables.

- On these tasks, we compare our methods to supervised learning baselines and well-known methods for unsupervised domain adaptation (Section 4). We find that our proposed models perform favorably to the alternatives for all types of PI, particularly when input overlap is violated and when training sets are small.

2 Privileged Information in Domain Adaptation

In unsupervised domain adaptation (UDA), the goal is to learn a hypothesis h to predict outcomes (or labels) $Y \in \mathcal{Y}$ for problem instances represented by input covariates $X \in \mathcal{X}$, drawn from a target domain with density $\mathcal{T}(X, Y)$. During training, we have access to labeled samples (x, y) only from a source domain $\mathcal{S}(X, Y)$ and unlabeled samples $\tilde{x}$ from $\mathcal{T}(X)$. As a running example, we think of $\mathcal{S}$ and $\mathcal{T}$ as radiology departments at two different hospitals, of X as the X-ray image of a patient, and of Y as the diagnosis made by a radiologist after analyzing the image.

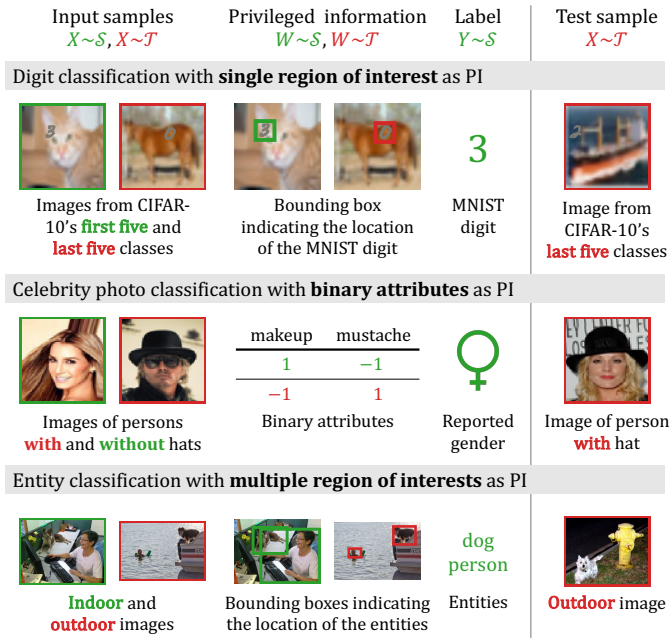


Figure 1: Examples of domain adaptation tasks with different types of privileged information (PI). During training, input samples X and PI W are drawn from both source and target domains. Labels Y are only available from the source domain. At test time, a target sample X is observed. We consider three types of PI: binary attribute vectors, a single region of interest, and multiple regions of interest.

We aim to learn a hypothesis $h \in \mathcal{H}$ from a hypothesis set $\mathcal{H}$ that minimizes the expected target-domain prediction error (risk) $R_{\mathcal{T}}$, with respect to a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, i.e., to solve

$$\min_{h \in \mathcal{H}} R_{\mathcal{T}}(h), \quad R_{\mathcal{T}}(h) \coloneqq \mathbb{E}_{\mathcal{T}(X,Y)}[L(h(X), Y)] \ , \tag{1}$$

where we use the subscript convention $\mathbb{E}_{p(X)}[f(X)] = \int_{x \in \mathcal{X}} p(x) f(x) dx$ to denote an expectation of some function f over a density p on the domain $\mathcal{X}$. A consistent solution to the UDA problem returns a minimizer of Equation 1 without ever observing labeled samples from $\mathcal{T}$. However, if $\mathcal{S}$ and $\mathcal{T}$ are allowed to differ arbitrarily, finding such a solution cannot be guaranteed (Ben-David & Urner, 2012). To make the problem feasible, we assume that *covariate shift* (Shimodaira, 2000) holds—that the labeling function is the same in both domains, but the covariate distributions differ.

Assumption 1 (Covariate shift)**.** *For domains $\mathcal{S}, \mathcal{T}$ on $\mathcal{X} \times \mathcal{Y}$,* covariate shift *holds with respect to X if*

$$\exists x \in \mathcal{X} : \mathcal{T}(X = x) \neq \mathcal{S}(X = x) \ \text{ and } \ \forall x \in \mathcal{X} : \mathcal{T}(Y \mid x) = \mathcal{S}(Y \mid x) \ .$$

In our example, covariate shift means that radiologists at either hospital would diagnose two patients with the same X-ray in the same way, but that the radiologists may encounter different distributions of patients and images. To guarantee consistent learning without further assumptions, these distributions cannot be *too* different—the source input domain $\mathcal{S}(x)$ must sufficiently *overlap* the target input domain $\mathcal{T}(x)$.

Assumption 2 (Domain overlap)**.** *A domain $\mathcal{S}$ overlaps another domain $\mathcal{T}$ with respect to X on $\mathcal{X}$ if*

$$\forall x \in \mathcal{X} : \mathcal{T}(X = x) > 0 \implies \mathcal{S}(X = x) > 0 \ .$$

Table 1: A summary of the different settings we consider in this work, what data is assumed to be available during training and if guarantees for identification are known for the setting under the assumptions of Proposition 1. The parentheses around source samples for DALUPI indicate that we need not necessarily observe these for the setting. Note that at test time only x from $\mathcal{T}$ is observed. *Under the more generous assumption of overlapping support in the input space $\mathcal{X}$, guarantees exist for all these settings.

Setting	Observed $\mathcal{S}$			Observed $\mathcal{T}$			Guarantee
	x	w	y	$\tilde{x}$	$\tilde{w}$	$\tilde{y}$	for $R_{\mathcal{T}}$
SL-T				✓		✓	✓
SL-S	✓		✓				*
UDA	✓		✓	✓			*
LUPI	✓	✓	✓				*
DALUPI	(✓)	✓	✓	✓	✓		✓

Covariate shift and domain overlap with respect to X guarantee that the target risk $R_{\mathcal{T}}$ can be identified by the sampling distribution described above, and thus, that a solution to Equation 1 may be found. Hence, they have become standard assumptions, used by most informative guarantees (Zhao et al., 2019).

Overlap is often violated in high-dimensional problems such as image classification, partly due to irrelevant information that has a spurious association with the label Y (Beery et al., 2018; D'Amour et al., 2021). In X-ray classification, it may be possible to perfectly distinguish hospitals (domains) based on protocol or equipment differences manifesting in the images (Zech et al., 2018). There are no guarantees for optimal UDA in this case. Some guarantees based on distributional distances do not rely on overlap (Ben-David et al., 2006; Long et al., 2013), but do not guarantee optimal learning either (Johansson et al., 2019).

Still, an image X may *contain* information W which is both *sufficient for prediction* and *supported in both domains*. For X-rays, this could be a region of pixels indicating a medical condition, ignoring parts that merely indicate differences in protocol (Zech et al., 2018). The learner does not know how to find this information a priori, but it can be supplied during training as added supervision. A radiologist could indicate regions of interest W using bounding boxes during training (Irvin et al., 2019), but would not be available to annotate images at test time. As such, W is *privileged information* (Vapnik & Vashist, 2009).

2.1 Unsupervised Domain Adaptation With Privileged Information

Learning using privileged information, variables that are available only during training but not at test time, has been shown to improve sample efficiency in diverse settings (Vapnik & Izmailov, 2015; Pechyony & Vapnik, 2010; Jung & Johansson, 2022). Next, we show that privileged information can also improve UDA by providing *identifiability* of the target risk—allowing it to be computed from the sampling distribution—even when overlap is not satisfied in X.

We define domain adaptation by learning using privileged information (DALUPI) as follows. During training, learners observe samples of covariates X, labels Y and privileged information $W \in \mathcal{W}$ from $\mathcal{S}$ in a dataset $D_{\mathcal{S}} = \{(x_i, w_i, y_i)\}_{i=1}^{m}$, as well as samples of covariates and privileged information from $\mathcal{T}$, $D_{\mathcal{T}} = \{(\tilde{x}_i, \tilde{w}_i)\}_{i=1}^{n}$. *At test time, trained models only observe covariates $\tilde{x} \sim \mathcal{T}(X)$ and our learning goal remains to minimize the target risk, see Equation 1.* We justify access to privileged information from $\mathcal{T}$, but not labels, by pointing out that it is often easier to annotate observations with privileged information W than with labels Y. For example, a non-expert may be able to reliably recognize the outline of an animal in an image, indicating the pixels W corresponding to it, but not identify its species (Y); see Figure 2, where it would likely be easier to identify the location of the cat in the image than to identify its breed.

To identify $R_{\mathcal{T}}$ (Equation 1) without overlap in X, we make the assumption that W is sufficient to predict Y in the following sense.

Assumption 3 (Sufficiency of privileged information)**.** *Privileged information W is sufficient for the outcome Y given covariates X if $Y \perp X \mid W$ in both $\mathcal{S}$ and $\mathcal{T}$.*

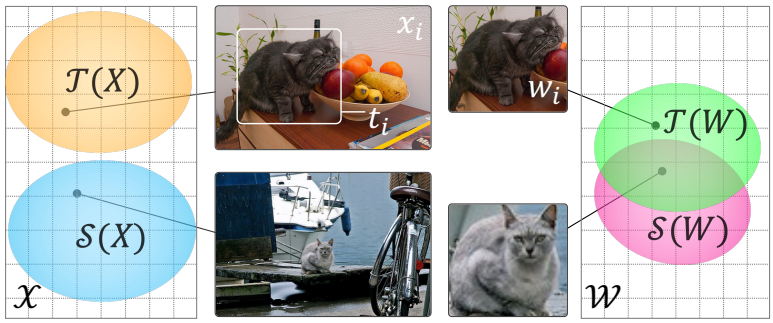


Figure 2: An illustration of domain overlap being more plausible when we consider appropriate forms of privileged information W, such as a region of interest of an image. Source and target domains $\mathcal{S}, \mathcal{T}$ are here indoor and outdoor images X and the task is to identify the animal Y in the image.

Assumption 3 is satisfied when X provides no more information about Y in the presence of W. If we consider W to be a subset of image pixels corresponding to an area of interest, the other pixels in X may be unnecessary to predict Y. This is illustrated in Figure 2 where the privileged information w_i is the region of interest indicated by the bounding box t_i. Here, overlap is more probable in $\mathcal{W}$ than in $\mathcal{X}$, as the extracted pixels mostly show cats. Moreover, when W retains more information, sufficiency becomes more plausible but domain overlap in W is reduced. The sufficiency assumption is used to replace $\mathcal{T}(y \mid x)$ with $\mathcal{T}(y \mid w)$ in Proposition 1. If sufficiency is violated but it is plausible that the degree of insufficiency is comparable across domains, we can still obtain a bound on the target risk which may be estimated from observed quantities. We give such a result in Appendix F.

We expect that some PI can be selected to be sufficient for a given task. However, if this sufficiency cannot be ensured, the overall performance may decrease, assuming covariate shift with respect to W is not violated. Even so, we still anticipate the generalization error to remain of a comparable magnitude. If covariate shift is violated in W, further performance declines are expected, as the problem becomes more complex and we are not guaranteed to identify the optimal hypothesis (Johansson et al., 2019).

Assumptions 1–2 holding with respect to privileged information W instead of X, along with Assumption 3, allow us to identify the target risk even for models $h \in \mathcal{H}$ that do not use W as input:

Proposition 1. *Let Assumptions 1 and 2 be satisfied with respect to W (not necessarily with respect to X) and let Assumption 3 hold as stated. Then, the target risk $R_{\mathcal{T}}$ is identified for hypotheses $h : \mathcal{X} \to \mathcal{Y}$,*

$$\begin{aligned} R_{\mathcal{T}}(h) &= \mathbb{E}_{\mathcal{T}(X)}\left[\mathbb{E}_{\mathcal{T}(W|X)}\left[\mathbb{E}_{\mathcal{S}(Y|W)}[L(h(X), Y) \mid X, W] \mid X\right]\right] \\ &= \int_x \mathcal{T}(x) \int_w \mathcal{T}(w \mid x) \int_y \mathcal{S}(y \mid w) L(h(x), y) \mathrm{d}y \mathrm{d}w \mathrm{d}x \ , \end{aligned}$$

and for L the squared loss, a minimizer of $R_{\mathcal{T}}$ is the function

$$h^*_{\mathcal{T}}(x) = \mathbb{E}_{\mathcal{T}(W|x)}[\mathbb{E}_{\mathcal{S}(Y|W)}[Y \mid W] \mid x] = \int_w \mathcal{T}(w \mid x) \int_y \mathcal{S}(y \mid w) y \ \mathrm{d}y \mathrm{d}w \ .$$

Proof sketch. $R_{\mathcal{T}}(h) = \int_{x,y} \mathcal{T}(x, y) L(h(x), y) \mathrm{d}x \mathrm{d}y$. We can then marginalize over W to get $\mathcal{T}(x, y) = \mathcal{T}(x)\mathbb{E}_{\mathcal{T}(W|x)}[\mathcal{T}(y \mid W) \mid x] = \mathcal{T}(x) \int_{w:\mathcal{S}(w)>0} \mathcal{T}(w \mid x)\mathcal{S}(y \mid w) \mathrm{d}w$, where the first equality follows by sufficiency and the second by covariate shift and overlap in W. $\mathcal{T}(x), \mathcal{T}(w \mid x)$ and $\mathcal{S}(y \mid w)$ are observable through training samples. That $h^*_{\mathcal{T}}$ is a minimizer follows from the first-order condition. See Appendix C. □

Proposition 1 shows that there are conditions where privileged information allows for identification of target-optimal hypotheses where identification is not possible without it, i.e., when overlap is violated in X. W

guides the learner toward the information in X that is relevant for the label Y. When W is deterministic in X, overlap in X implies overlap in W, but not vice versa. In the same case, under Assumption 3, if covariate shift holds for X, it holds also for W. Hence, if sufficiency can be justified, the requirements on X are weaker than in standard UDA, at the cost of collecting W. Surprisingly, Proposition 1 does not require that X is observed in the source domain as the result does not depend on $\mathcal{S}(x)$.

Figure 1 gives examples of problems with the DALUPI structure which we consider in this work. For comparison, we list related learning paradigms in Table 1. Supervised learning (SL-S) refers to learning from labeled samples from $\mathcal{S}$ without privileged information. SL-T refers to supervised learning with (infeasible) access to labeled samples from $\mathcal{T}$. UDA refers to the setting at the start of Section 2 and LUPI to learning using privileged information without data from $\mathcal{T}$ (Vapnik & Vashist, 2009). We compare DALUPI to these alternative settings in our experiments in Section 4.

2.2 A Two-stage Algorithm and Its Risk

In light of Proposition 1, a natural learning strategy is to model privileged information as a function of the input, $\mathcal{T}(W \mid x)$, and the outcome as a function of privileged information, $\hat{g}(w) \approx \mathbb{E}_{\mathcal{S}}[Y \mid w]$, and combining these. In the case where W is a deterministic function of X, $\mathcal{T}(W \mid x)$ is a map $f : \mathcal{X} \to \mathcal{W}$, which may be estimated as a regression $\hat{f}$ and combined with the outcome regression to form $\hat{h} = \hat{g}(\hat{f}(X))$. We may find such functions $\hat{f}$, $\hat{g}$ by separately minimizing the empirical risks

$$\hat{R}^W_{\mathcal{T}}(f) = \frac{1}{n}\sum_{i=1}^{n} L_W(f(\tilde{x}_i), \tilde{w}_i) \quad \text{and} \quad \hat{R}^Y_{\mathcal{S}}(g) \quad = \frac{1}{m}\sum_{i=1}^{m} L_Y(g(w_i), y_i) \ . \tag{2}$$

Hypothesis classes $\mathcal{F}, \mathcal{G}$ may be chosen so that $\mathcal{H} = \{h = g \circ f; (f, g) \in \mathcal{F} \times \mathcal{G}\}$ has a desired form. Note that L_W and L_Y may in general be different loss functions.

We can bound the generalization error of estimators $\hat{h} = \hat{g} \circ \hat{f}$ when $W \in \mathbb{R}^{d_W}$ and the loss is the squared loss. We do this by placing an assumption of Lipschitz smoothness on the space of prediction functions: $\forall g \in \mathcal{G}, w, w' \in \mathcal{W} : \|g(w) - g(w')\|_2 \leq M\|w - w'\|_2$. To arrive at a bound, we first define the ρ-weighted empirical risk of the outcome model g in the source domain, $\hat{R}^{Y,\rho}_{\mathcal{S}}(g) = \frac{1}{m}\sum_{i=1}^{m} \rho(w_i) L\_W(g(w_i), y_i)$ where ρ is the density ratio of $\mathcal{T}$ and $\mathcal{S}$, $\rho(w) = \frac{\mathcal{T}(w)}{\mathcal{S}(w)}$. When the density ratio ρ is unknown, we may use density estimation (Sugiyama et al., 2012) or probabilistic classifiers to estimate it. We arrive at the following result, proven for univariate Y but generalizable to multivariate outcomes.

Proposition 2. *Suppose that W and Y are deterministic in X and W, respectively, and that Assumptions 1–3 hold with respect to W. Let $\mathcal{G}$ comprise M-Lipschitz mappings $g : \mathcal{W} \to \mathcal{Y}$ with $\mathcal{W} \subseteq \mathbb{R}^{d_W}$, and let the loss be the squared Euclidean distance, assumed to be uniformly bounded over $\mathcal{W}$. Let $\rho(w) = \mathcal{T}(w)/\mathcal{S}(w)$ and d and d' be the pseudo-dimensions of $\mathcal{G}$ and $\mathcal{F}$, respectively. Assume that there are m labeled samples from $\mathcal{S}$ and n unlabeled samples from $\mathcal{T}$. Then, for any $h = g \circ f \in \mathcal{G} \times \mathcal{F}$, with probability at least $1 - \delta$,*

$$\frac{R_{\mathcal{T}}(h)}{2} \leq \hat{R}^{Y,\rho}_{\mathcal{S}}(g) + M^2 \hat{R}^W_{\mathcal{T}}(f) + \mathcal{O}\left(\sqrt[3/8]{\frac{d \log \frac{m}{d} + \log \frac{4}{\delta}}{m}} + \sqrt{\frac{d' \log \frac{n}{d'} + \log \frac{d_W}{\delta}}{n}} \right) \ .$$

Proof sketch. Decomposing the risk of $h \circ \phi$, we get

$$\begin{aligned}
R_{\mathcal{T}}(h) &= \mathbb{E}_{\mathcal{T}}[(g(f(X)) - Y)^2] \\
&\leq 2\mathbb{E}_{\mathcal{T}}[(g(W) - Y)^2 + (g(f(X)) - g(W))^2] \\
&\leq 2\mathbb{E}_{\mathcal{T}}[(g(W) - Y)^2 + M^2\|f(X)) - g(W)\|^2] \\
&\leq 2\mathbb{E}_{\mathcal{T}}[(g(W) - Y)^2] + 2M^2\mathbb{E}_{\mathcal{T}}[\|(f(X) - W)\|^2] \\
&= 2R^Y_{\mathcal{T}}(g) + 2M^2 R^W_{\mathcal{T}}(f) = 2R^{Y,\rho}_{\mathcal{S}}(g) + 2M^2 R^W_{\mathcal{T}}(f) \ .
\end{aligned}$$

The first inequality follows the relaxed triangle inequality, the second from the Lipschitz property, and the third equality from Overlap and Covariate shift. Treating each component of $\hat{w}$ as independent, using

standard PAC learning results, and application of Theorem 3 from Cortes et al. (2010) allows us to reweight the risk with the density ratio ρ by also adding an additional term which contains the Rényi divergence. Then with a union bound argument, we get the stated result. See Appendix D for a more detailed proof. □

When $\mathcal{F}$ and $\mathcal{G}$ contain the ground-truth mappings between X and W and between W and Y, in the infinite-sample limit, minimizers of Equation 2 minimize $R_{\mathcal{T}}$ as well. Our approach is not limited to classical PAC analysis but could, under suitable assumptions, be carried out under another framework, e.g. using PAC-Bayes analysis to obtain a bound that contains different sample complexity terms. However, such a bound would then hold in expectation over a posterior distribution on $\mathcal{H}$ instead of uniformly over $\mathcal{H}$. We sketch a proof of such a bound in Appendix E.

Furthermore, if sufficiency is violated but it is plausible that the degree of insufficiency is comparable across domains, we can still obtain a bound on the target risk which may be estimated from observed quantities. We give such a result in Appendix F.

3 Image Classification With Privileged Information

We use image classification, where X is an image and Y is a discrete label, as proof of concept for DALUPI. To show the versatility of our approach, we consider three different instantiations of privileged information W: a binary attribute vector, a single region of interest, or multiple regions of interest. The two-stage estimator, see Figure 3a, is used in the first two cases. With multiple regions of interest as privileged information, we use an end-to-end model based on Faster-R-CNN (Ren et al., 2016), see Figure 3b. We detail each setting below and illustrate them in Figure 1.

3.1 Binary Attributes as PI

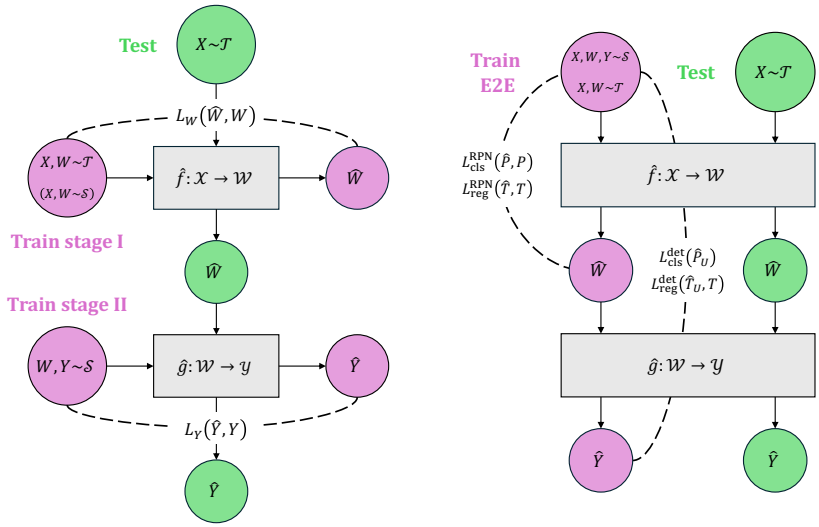
First, we consider the case where each image x_i is accompanied by privileged information in the form of a binary vector $w_i \in \{0,1\}^d$ indicating the presence of d attributes in the image. In this setting, we can directly apply our two-stage estimator (Equation 2). For the first estimator $\hat{f}$, we use a convolutional neural network (CNN) trained on observations from $\mathcal{T}$ (and possibly $\mathcal{S}$) to output a vector of attributes $\hat{w}_i$ from the input x_i. For the second estimator $\hat{g}$, we use a multi-layer perceptron classifier, trained on source domain observations, that predicts the image label $\hat{y}_i$ given the vector of attributes w_i. We use the categorical cross-entropy loss to train both $\hat{f}$ and $\hat{g}$. The resulting classifier, $\hat{h}(x) = \hat{g}(\hat{f}(x))$, is subsequently evaluated on target domain images.

3.2 Single Region of Interest as PI

Next, we consider privileged information as a subset of pixels w_i, taken from the image x_i and associated with an object or feature that determines the label $y_i \in \{1, \ldots, K\}$. In our experiments, this PI is provided as a *single* bounding box with coordinates $t_i \in \mathbb{R}^4$ enclosing the region of interest w_i. Here, we use two CNNs, $\hat{d}$ and $\hat{g}$, and a deterministic function ϕ to approximate the two-stage estimator (Equation 2). The network $\hat{d}$ is trained to output bounding box coordinates $\hat{t}_i$ as a function of the input x_i, and the pixels $\hat{w}_i$ within the bounding box are extracted from x_i and resized to pre-specified dimensions through ϕ. The composition of these two functions, $\hat{f}(x_i) = \phi(x_i, \hat{d}(x_i))$, returns $\hat{w}_i$. The second network $\hat{g}$ is trained to predict y_i given the pixels w_i contained in a bounding box t_i based on observations from $\mathcal{S}$. We use the mean squared error loss for $\hat{d}$ and the categorical cross-entropy loss for $\hat{g}$. Finally, $\hat{h}(x) = \hat{g}(\hat{f}(x))$ is evaluated on target domain images where the output of $\hat{f}$ is used for prediction with $\hat{g}$. See Appendix A.1 for further details.

3.3 Multiple Regions of Interest as PI

Finally, we consider a setting where privileged information indicates *multiple* regions of interest in an image. We use this PI in multi-label classification problems where the image x_i is associated with one or more categories k from a set $\{1, \ldots, K\}$, encoded in a multi-category label $y_i \in \{0,1\}^K$ (e.g., indicating findings



(a) Two-stage estimator.

(b) End-to-end estimator.

Figure 3: A schematic representation of the train and test flow for DALUPI using (a) the two-stage estimator presented in Section 2.2 and (b) an end-to-end architecture based on Faster R-CNN (Ren et al., 2016). In the two-stage procedure, the networks $\hat{f}$ and $\hat{g}$ are learned through empirical risk minimization of L_W and L_Y, respectively. At test time, $\hat{f}$ and $\hat{g}$ are combined into $\hat{h} = \hat{g}(\hat{f}(X))$. The end-to-end estimator uses a region proposal network (RPN) to produce regions of interest in the input image X. The RPN, which serves as the network $\hat{f}$, is followed by a detection network $\hat{g}$ that predicts the class of any object within a region proposal. Training is guided by regression losses $L_{\text{reg}}^{\text{PRN}}(\hat{T}, T)$ and $L_{\text{reg}}^{\text{det}}(\hat{T}_U, T)$, as well as by classification losses $L_{\text{cls}}^{\text{PRN}}(\hat{P}, P)$ and $L_{\text{cls}}^{\text{det}}(\hat{P}_U)$. Here, T and $\hat{T}$ denote ground-truth and predicted bounding box coordinates, respectively, and $\hat{T}_U$ are the predicted coordinates for a region proposal with ground-truth label U. Further, $\hat{P}$ is the RPN's predicted probability that a region proposal contains an object, P is a binary label assigned to the proposal based on its overlap with ground-truth bounding boxes, and $\hat{P}_u$ is the probability of the ground-truth class U within the proposal, as predicted by the detection network.

of one or more diseases). The partial label $y_i(k) = 1$ indicates the presence of features or objects in the image from category k. In our entity classification experiment, an object j of class $k \in [K]$ in the image, say "Bird", will be annotated by a bounding box $t_{ij} \in \mathbb{R}^4$ surrounding the pixels of the bird, and an object label $u_{ij} = k$. In X-ray classification, t_{ij} can indicate an abnormality j in the X-ray image, and $u_{ij} \in \{1, \ldots, K\}$ the label of the finding (e.g., "Pneumonia").

To make full use of privileged information, we train a deep neural network $\hat{h}(x) = \hat{g}(\hat{f}(x))$, where $\hat{f}$ produces *a set* of bounding box coordinates $\hat{t}_{ij}$ and extracts the pixels $\hat{w}_{ij}$ associated with each $\hat{t}_{ij}$, and where $\hat{g}$ predicts a label $\hat{u}_{ij}$ for each $\hat{w}_{ij}$. To this end, we adapt the Faster R-CNN architecture (Ren et al., 2016) which uses a region proposal network (RPN) to generate regions that are fed to a detection network for classification and refined bounding box regression. A CNN backbone in combination with the RPN region of interest pooling serves as the subnetwork $\hat{f}$, producing estimates $\hat{w}_i$ of the privileged information for an image x_i. For the detection network, which corresponds to the subnetwork $\hat{g}$, we use Fast-RCNN (Girshick, 2015).

Privileged information adds supervision through regression losses $L_{\text{reg}}^{\text{RPN}}(\hat{t}, t)$ and $L_{\text{reg}}^{\text{det}}(\hat{t}_u, t)$ for region proposals $\hat{t}$ and class-specific bounding box coordinates $\hat{t}_u$. We use the smooth L1 loss defined by Girshick (2015) for

both $L_{\text{reg}}^{\text{RPN}}$ and $L_{\text{reg}}^{\text{det}}$. Training is further guided by classification losses $L_{\text{cls}}^{\text{RPN}}(\hat{p}, p) = -(p \log \hat{p} + (1-p) \log \hat{p})$ and $L_{\text{cls}}^{\text{det}}(\hat{p}_u) = -\log \hat{p}_u$, where $\hat{p}$ is the RPN's predicted probability that a region proposal contains an object, p is a binary label assigned to the proposal based on its overlap with ground-truth bounding boxes, and $\hat{p}_u$ is the probability of the ground-truth class u within the proposal, as predicted by the detection network.

In Appendix A.2, we provide details of the learning objective and architecture and describe small modifications to the training procedure of Faster R-CNN to accommodate the DALUPI setting. Unlike the two-stage estimator, we train Faster R-CNN (both $\hat{f}$ and $\hat{g}$) end-to-end, minimizing both losses at once. In entity classification experiments (see Table 3 and Figure 5), we also train this model in a LUPI setting, where *no* information from the target domain is used, but privileged information from the source domain is used.

4 Experiments

We evaluate the empirical benefits of learning using privileged information, compared to the other data availability settings in Table 1, across four UDA image classification tasks where PI is available in the forms described in Section 3. Widely used datasets for UDA evaluation like OfficeHome (Venkateswara et al., 2017) and large-scale benchmark suites like DomainBed (Gulrajani & Lopez-Paz, 2021), VisDA (Peng et al., 2017) and WILDS (Koh et al., 2021) *do not* include privileged information and cannot be used for evaluation here. Thus, we first compare our method to baselines on the recent CelebA task (Xie et al., 2020) which includes PI in the form of binary attributes (Section 4.1). Additionally, we propose three new tasks based on well-known image classification data sets with regions of interest as PI (Section 4.2–4.4). In Section 4.1 and 4.2, we use the two-stage estimator with the subnetwork $\hat{f}$ based on the ResNet-18 architecture (He et al., 2016a). In Section 4.3 and 4.4, we use our variant of Faster R-CNN with a ResNet-50 backbone.

Our goal is to collect evidence that DALUPI improves adaptation bias and sample efficiency compared to methods that do not make use of PI. We choose baselines to illustrate these two disparate settings. First, we compare DALUPI to supervised learning baselines, SL-S and SL-T, trained on labeled examples from the source and target domain, respectively. SL-S is a simple but strong baseline: On benchmark suites like DomainBed and WILDS, there is still no UDA method that *consistently* outperforms SL-S (ERM) without transfer learning (Gulrajani & Lopez-Paz, 2021; Koh et al., 2021). SL-T serves as an oracle comparison since it uses labels from the target domain which are normally unavailable in UDA. Second, we compare DALUPI to two UDA methods—domain adversarial neural networks (DANN) (Ganin et al., 2016) and the margin disparity discrepancy (MDD) (Zhang et al., 2019)—which have theoretical guarantees but do not make use of PI. These baselines are all based on the ResNet architecture. In Section 4.1, we compare DALUPI also to In-N-Out (Xie et al., 2020), which was designed to make use of auxiliary (privileged) attributes for training domain adaptation models. We do not include this model in other experiments as it was not designed to use regions of interest as privileged information. The exact architectures of all models and baselines are described in Appendix A, along with details on experimental setup and hyperparameters.

For each task and task-specific setting (label skew, amount of privileged information, etc.), we train 10 models from each relevant class using hyperparameters randomly selected from given ranges (see Appendix A). For DANN and MDD, the trade-off parameter, which regularizes domain discrepancy in representation space, increases from 0 to 0.1 during training; for MDD, the margin parameter is set to 3. All models are evaluated on a held-out validation set from the source domain and the best-performing model in each class is then evaluated on held-out test sets from both domains. For SL-T, we use a held-out validation set from the target domain. We repeat this procedure over 5 or 10 seeds, controlling the data splits and the random number generation. We report accuracy and area under the ROC curve (AUC) with 95 % confidence intervals computed by bootstrapping over the seeds.

4.1 Celebrity Photo Classification With Binary Attributes as PI

In the case where privileged information is available as binary attributes, we follow Xie et al. (2020) who introduced a binary classification task based on the CelebA dataset (Liu et al., 2015), where the goal is to predict whether the person in an image has been identified as male or female (Y) in one of the binary

Table 2: Celebrity photo classification. DALUPI performs comparably to the In-N-Out models in Xie et al. (2020). Note: In-N-Out results are reported as the average of 5 trials with 90 % confidence intervals.

	Target accuracy
SL-T	86.6 (86.3, 86.9)
SL-S	78.4 (77.1, 80.0)
DANN	78.2 (76.2, 80.3)
MDD	78.3 (77.5, 79.1)
In-N-Out (w/o pretraining)	78.5 (77.2, 79.9)
In-N-Out (w. pretraining)	79.4 (78.7, 80.1)
In-N-Out (rep. self-training)	80.4 (79.7, 81.1)
DALUPI ($W \sim \mathcal{T}$)	76.4 (73.8, 78.6)
DALUPI ($W \sim \mathcal{S}, \mathcal{T}$)	80.3 (77.9, 82.7)

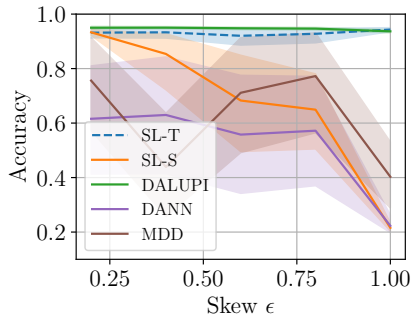


Figure 4: Digit classification. Target domain accuracy as a function of association ϵ between background and label in $\mathcal{S}$. As the skew increases, the target-domain performance of the non-privileged models deteriorates.

attributes that accompanies the data set's photos of celebrities (X). Like Xie et al. (2020), we use 7 of the 40 other attributes (`Bald`, `Bangs`, `Mustache`, `Smiling`, `5_o_Clock Shadow`, `Oval_Face`, and `Heavy_Makeup`) as a vector of privileged information $W \in \{0,1\}^7$. The target and source domains are defined by people wearing ($\mathcal{T}$) and not wearing ($\mathcal{S}$) a hat. The respective datasets contain 3,000 and 2,000 images. An extra 30,000 unlabeled source samples are available to train estimators (DALUPI and In-N-Out) that can utilize privileged information from both source and target. More details can be found in (Xie et al., 2020) and in Appendix A.3.

Table 2 shows the target accuracy for each model. We observe that when DALUPI is provided with PI from both source and target, it performs comparably to the best-performing In-N-Out model proposed by Xie et al. (2020), while outperforming other feasible baselines on average. Confidence intervals overlap for all feasible models. Notably, the best-performing In-N-Out models require four or more rounds of training to achieve their results (baseline, auxiliary input, auxiliary output pre-training, tuning and self-training) (Xie et al., 2020). Both DALUPI and In-N-Out benefit from access to privileged information from both the source and target domain (pre/self-training for In-N-Out).

Finally, it is worth noting that neither covariate shift, nor sufficiency are likely to hold with respect to W in this task. Specifically, photos with none of the 7 attributes active, $w = \mathbf{0}$, have different label rates and majority label in $\mathcal{S}$ and $\mathcal{T}$ (the rates of labels are $\bar{Y}_{\mathcal{S}} = 0.64$ and $\bar{Y}_{\mathcal{T}} = 0.46$, respectively) and therefore $P(Y|W)$ is not constant, i.e. covariate shift is violated. In addition, the best model we have found trained on W alone achieves only 65 % accuracy, compared to the results in Table 2—sufficiency is unlikely to hold. Thus, DALUPI is robust to violations of these assumptions.

4.2 Digit Classification With Single Region of Interest as PI

We construct a synthetic image dataset, based on the assumptions of Proposition 1, to verify that there are problems where DALUPI is guaranteed successful transfer but standard UDA is not. Starting from CIFAR-10 (Krizhevsky, 2009) images upscaled to 128×128, we insert a random 28×28 digit image from the MNIST dataset (Lecun, 1998), with a label in the range 0–4, into a random location of each CIFAR-10 image, forming the input image X (see Figure 1 (top) for examples). The label $Y \in \{0, \ldots, 4\}$ is determined by the MNIST digit. We store the bounding box around the inserted digit image and use the pixels contained within it as privileged information W during training. The domains are constructed using CIFAR-10's first five and last five classes as source and target backgrounds, respectively. Both source and target datasets contain 15,298 images each. To increase the difficulty of the task, we make the digit be the mean color of the

Table 3: Entity classification. UDA models have access to all unlabeled target samples, LUPI to all PI (source), and DALUPI to all PI (source and target).

	Source AUC	Target AUC
SL-T	60.1 (58.7, 61.5)	69.0 (68.1, 69.9)
SL-S	69.5 (68.6, 70.4)	63.0 (61.6, 64.2)
DANN	68.1 (67.5, 68.7)	62.5 (61.9, 63.1)
MDD	62.4 (61.1, 63.9)	57.7 (56.3, 59.2)
LUPI	69.3 (68.5, 70.1)	65.9 (65.0, 66.8)
DALUPI	71.4 (70.3, 72.4)	68.2 (66.3, 70.1)

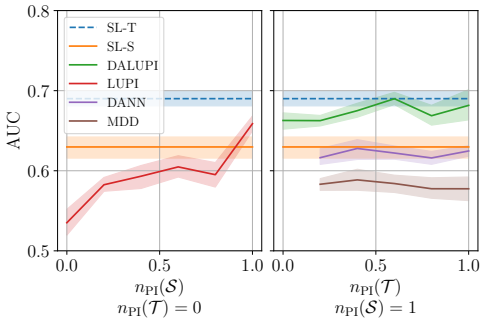


Figure 5: Entity classification. Target domain AUC. The performance of SL-S and SL-T is extended across the x-axes for visual purposes. DANN and MDD use an increasing fraction of target samples $\tilde{x}$ but no PI.

dataset and make the digit background transparent so that the border of the image is less distinct. This may slightly violate Assumption 2 with respect to the region of interest W since the backgrounds differ between domains.

To understand how successful transfer depends on domain overlap and access to sufficient privileged information, we include a *skew parameter* $\epsilon \in [\frac{1}{c}, 1]$, where $c = 5$ is the number of digit classes, which determines the correlation between digits and backgrounds. For a source image i with digit label $Y_i \in \{0, \ldots, 4\}$, we select a random CIFAR-10 image with class $B_i \in \{0, \ldots, 4\}$ with probability $P(B_i = b \mid Y_i = y) = \{\epsilon, \text{ if } b = y; \ (1-\epsilon)/(c-1), \text{otherwise}\}$. For target images, digits and backgrounds are matched uniformly at random. The choice $\epsilon = \frac{1}{c}$ yields a uniform distribution and $\epsilon = 1$ is equivalent to the background carrying as much signal as the privileged information. We hypothesize that $\epsilon = 1$ is the worst possible case where confusion of the model is likely, which would lead to poor adaptation under domain shift.

In Figure 4, we observe the conjectured behavior. As the skew ϵ and the association between background and label increases, the performance of SL-S decreases rapidly on the target domain. At $\epsilon = 1$, it performs no better than random guessing, likely because the model has learned to associate spurious features in the background with the label of the digit. We also observe that DANN and MDD deteriorate in performance with increased correlation between the label and the background. In contrast, DALUPI is unaffected by the skew as the subset of pixels extracted by $\hat{f}$ only carries some of the background with it, while containing sufficient information to make good predictions. Interestingly, DALUPI also seems to be as good or slightly better than the oracle SL-T in this setting. This may be due to improved sample efficiency from using PI.

4.3 Entitity Classification With Multiple Regions of Interest as PI

Next, we consider multi-label classification of the presence of four types of entities (persons, cats, dogs, and birds) indicated by a binary vector $Y \in \{0, 1\}^4$ for images X from the MS-COCO dataset (Lin et al., 2014). PI is used to localize regions of interest W related to the entities, provided as bounding box annotations. We define source and target domains $\mathcal{S}$ and $\mathcal{T}$ as indoor and outdoor images, respectively. Indoor images are extracted by filtering out images from the MS-COCO super categories "indoor" and "appliance" that also contain at least one of the four main label classes. Outdoor images are extracted using the super categories "vehicle" and "outdoor". In total, there are 5,231 images in the source and 5,719 images in the target domain; the distribution of labels is provided in Appendix A.5.

Sufficiency is likely to hold in this task because the pixels contained in a bounding box should be sufficient for an annotator to classify the entity according to the four categories above, irrespective of the pixels outside

Table 4: X-ray task. Test AUC for the three pathologies in the target domain for all considered models. Boldface indicates the best-performing feasible model; SL-T uses target labels.

	ATL	CM	PE
SL-T	57 (56, 58)	59 (55, 63)	79 (78, 80)
SL-S	**55 (55, 56)**	61 (58, 64)	73 (70, 75)
DANN	53 (51, 55)	55 (53, 58)	55 (51, 61)
MDD	49 (48, 50)	51 (51, 52)	51 (48, 54)
DALUPI	**55 (55, 56)**	**72 (71, 73)**	**74 (72, 76)**

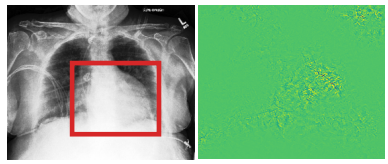


Figure 6: Left: Example from the X-ray target test set with label CM. The red rectangle indicates the bounding box predicted by DALUPI. Right: saliency map for CM for SL-S.

of the box. Similarly, covariate shift is likely to hold since the label attributed to the pixels in a bounding box should be the same, whether the entity is indoor or outdoor.

We study the effect of adding privileged information by first training the end-to-end model in a LUPI setting, using all (x, y) samples from the source domain and increasing the fraction of inputs for which PI is available, $n_{\text{PI}}(\mathcal{S})$, from 0 to 1. We then train the model in a DALUPI setting, increasing the fraction of $(\tilde{x}, \tilde{w})$ samples from the target domain, $n_{\text{PI}}(\mathcal{T})$, from 0 to 1, while keeping $n_{\text{PI}}(\mathcal{S}) = 1$. We train SL-S and SL-T using all available data and increase the fraction of unlabeled target samples used by DANN and MDD from 0.2 to 1 while using all data from the source domain.

Table 3 shows the models' source and target domain AUC, averaged over the four entity classes, when the UDA models have access to all unlabeled target samples, LUPI to all PI from the source domain, and DALUPI to all PI from both domains. Clearly, DALUPI yields a substantial gain in adaptation. As we see in Figure 5, the performance of LUPI increases as $n_{\text{PI}}(\mathcal{S})$ increases. When additional $(\tilde{x}, \tilde{w})$ samples from the target domain are added, DALUPI outperforms SL-S and approaches the performance of SL-T. We note that DANN and MDD do not benefit as much from added unlabeled target samples as DALUPI does. Their weak performance could be explained by difficulties in adversarial training. The gap between LUPI and SL-S for $n_{\text{PI}}(\mathcal{S}) = 0$ is anticipated; we do not expect the detection network to work well without bounding box supervision.

4.4 X-ray Classification With Multiple Regions of Interest as PI

As a real-world application, we study detection of pathologies in chest X-ray images. We use the ChestX-ray8 dataset (Wang et al., 2017) as source domain and the CheXpert dataset (Irvin et al., 2019) as target domain.[1] As PI, we use the regions of pixels associated with each found pathology, as annotated by domain experts using bounding boxes. For the CheXpert dataset, only pixel-level segmentations are available, and we create bounding boxes that tightly enclose the segmentations. It is not obvious that the pixels within such a bounding box are sufficient for classifying the pathology. For this reason, we suspect that some of the assumptions of Proposition 1 may be violated. However, as we find below, DALUPI improves empirical performance compared to baselines for small training sets, thereby demonstrating increased sample efficiency.

We consider the three pathologies that exist in both datasets and for which there are annotated findings: atelectasis (ATL: collapsed lung), cardiomegaly (CM: enlarged heart), and pleural effusion (PE: water around the lung). There are 457 and 118 annotated images in the source and target domain, respectively. We train DALUPI, DANN and MDD using all these images. SL-S is trained with the 457 source images and SL-T with the 118 target images as well as 339 labeled but non-annotated target images. Neither SL-S, SL-T, DANN, nor MDD support using privileged information. The distributions of labels and bounding box annotations are given in Appendix A.6.

[1]This study was granted IRB approval.

In Table 4, we present the per-class AUCs in the target domain. DALUPI outperforms all baseline models, including the target oracle, in detecting CM. For ATL and PE, it performs similarly to or better than the other feasible models. That SL-T is better at predicting PE is not surprising because this pathology is most prevalent in the target domain. In Figure 6, we show a single-finding image from the target test set with ground-truth label CM. The predicted bounding box of DALUPI with the highest probability is added to the image. DALUPI identifies the region of interest (the heart) and makes a correct classification. The rightmost panel shows the saliency map for the ground truth class for SL-S. We see that the gradients are mostly constant, indicating that the model is uncertain. In Appendix B, we show AUC for CM for the models trained with additional examples *without* bounding box annotations. We find that SL-S reaches the performance of DALUPI when a large amount of labeled examples are provided. This indicates that identifiability is not the main obstacle for adaptation and that PI improves sample efficiency.

5 Related Work

Learning using privileged information was first introduced by Vapnik & Vashist (2009) for support vector machines (SVMs), and was later extended to empirical risk minimization (Pechyony & Vapnik, 2010). Methods using PI, which is sometimes called hidden information or side information, has since been applied in many diverse settings such as healthcare (Shaikh et al., 2020), finance (Silva et al., 2010), clustering (Feyereisl & Aickelin, 2012) and image recognition (Vu et al., 2019; Hoffman et al., 2016). Related concepts include knowledge distillation (Hinton et al., 2015; Lopez-Paz et al., 2016), where a teacher model trained on additional variables adds supervision to a student model, and weak supervision (Robinson et al., 2020) where so-called weak labels are used to learn embeddings, subsequently used for the task of interest. Furthermore, in the realm of NLP, there is the related concept of learning using feature feedback, where additional annotations that are related to the associated task label are provided (Katakkar et al., 2022; Kaushik et al., 2021). These works are mostly of an empirical nature, and theoretical work on the subject either considers linear models/SVMs (Poulis & Dasgupta, 2017) or a teacher/student-type setup where additional supervision is given when the model predicts incorrectly (Dasgupta et al., 2018). The use of PI for deep image classification has been investigated by Chen et al. (2017) and Han et al. (2023) but these works only cover regular supervised learning where source and target domains coincide. Further, Sharmanska et al. (2014) used regions of interest in images as privileged information to improve the accuracy of image classifiers, but did not consider domain shift either.

Domain adaptation using PI has been considered before with SVMs (Li et al., 2022; Sarafianos et al., 2017), but not with more complex classifiers such as neural networks. Vu et al. (2019) used scene depth as PI in semantic segmentation using deep neural networks. However, they only used PI from the source domain and they did not provide any theoretical analysis. Xie et al. (2020) provide some theoretical results for a similar setup to ours. However, these are specifically for linear classifiers while our approach holds for any type of classifier. Motiian (2019) investigated PI and domain adaptation using the information bottleneck method for visual recognition. However, their setting differs from ours in that each observation comprises source-domain and target-domain features, a label and PI. Another related approach is that of subsidiary tasks (Kundu et al., 2022; Ye et al., 2022). However, in these settings the additional tasks performed are used to build a representation that helps with the main task through domain alignment. Our approach instead seeks to use information which directly relates to the main task.

6 Discussion

We have presented DALUPI: unsupervised domain adaptation by learning using privileged information (PI). The framework provides provable guarantees for adaptation under relaxed assumptions on the input features, at the cost of collecting a larger variable set, such as attribute or bounding box annotations, during training. Our analysis inspired practical algorithms for image classification which we evaluated using three kinds of privileged information. In our experiments, we demonstrated tasks where our approach is successful while existing adaptation methods fail. We observed empirically also that methods using privileged information are more sample-efficient than comparable non-privileged learners, in line with the literature. In fact, DALUPI

models occasionally even outperform oracle models trained using target labels due to their sample efficiency. Thus, we recommend considering these methods in small-sample settings.

The main contribution of the paper is the proposed learning paradigm for domain adaptation with privileged information. Since common benchmark datasets in UDA lack privileged information related to the learning problem, we created three new tasks for evaluating our framework, see Section 4.2–4.4, which itself is a notable contribution. We hope that this work inspires the community to develop additional datasets for UDA using privileged information.

To avoid assuming that domain overlap is satisfied with respect to input covariates, we require that the label is conditionally independent of the input features given the PI—that the PI is "sufficient". This is a limitation whenever sufficiency is difficult to verify. However, in our motivating example of image classification, a domain expert could *choose* PI so that sufficiency is reasonably justified. Moreover, in experiments on CelebA, we see empirical gains from our approach even when sufficiency is known to be violated. Another limitation is that we still rely on overlap in the privileged information, W, which may also be violated in some circumstances. It is more likely that overlap holds for W when, for example, it is a subset of X, as argued in Figure 2. Designing experiments to test how sensitive DALUPI is to violations of these assumptions is an interesting direction for future work.

The use of regions of interest as privileged information brings up an interesting point concerning the relationship between the label and the privileged information. In object detection tasks, it is natural to treat the bounding box coordinates as label information. In this work, however, the learning tasks were multi-class and multi-label image classification, not object detection. Producing a perfect box W was not the goal of the learning task, and the bounding boxes were therefore neither critical for the task nor for the labels. Instead, the bounding boxes were privileged information and our experiments in Section 4.2–4.4 sought to quantify the value of this added information, compared to not having it. Therefore, we compared our method to image classification baselines. It is not obvious a priori that learning from object locations improves the adaptation of image classifiers.

If there is a lack of PI available to the models one might mitigate this by either 1) using the limited amount of PI that is available to learn $\hat{g}$ and assume that it is good enough to achieve reasonable overall performance; or 2) using the learned $\hat{f}$ to create "weak" PI labels for the inputs that are missing PI, similar to the work of e.g. Robinson et al. (2020). However, one should note that the latter approach might bias the model in unintended ways and, as such, should be undertaken with some caution.

In future work, our framework could be applied to a more diverse set of tasks, with different modalities of inputs and privileged information to investigate if the findings here can be replicated and extended. Moreover, such work could consider different types and degrees of shifts to further corroborate the stability and resistance to noise which we observe here. More broadly, using PI may be viewed as "building in" domain knowledge in the structure of the adaptation problem and we see this as a promising direction for further research.

Acknowledgments

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation

The computations and data handling were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

References

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018.

Shai Ben-David and Ruth Urner. On the hardness of domain adaptation and the utility of unlabeled target samples. In *International Conference on Algorithmic Learning Theory*, pp. 139–153. Springer, 2012.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

Adam Breitholtz and Fredrik Daniel Johansson. Practicality of generalization guarantees for unsupervised domain adaptation with neural networks. *Transactions on Machine Learning Research*, 2022.

Yunpeng Chen, Xiaojie Jin, Jiashi Feng, and Shuicheng Yan. Training group orthogonal neural networks with privileged information. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 1532–1538, 2017.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

Sanjoy Dasgupta, Akansha Dey, Nicholas Roberts, and Sivan Sabato. Learning from discriminative feature feedback. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, 2018.

Antoine de Mathelin, François Deheeger, Guillaume Richard, Mathilde Mougeot, and Nicolas Vayatis. ADAPT: Awesome Domain Adaptation Python Toolbox. *arXiv preprint arXiv:2107.03049*, 2021.

Alexander D'Amour, Peng Ding, Avi Feller, Lihua Lei, and Jasjeet Sekhon. Overlap in observational studies with high-dimensional covariates. *Journal of Econometrics*, 221(2):644–654, 2021.

Jan Feyereisl and Uwe Aickelin. Privileged information for data clustering. *Information Sciences*, 194:4–23, 2012.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-Adversarial Training of Neural Networks. *arXiv:1505.07818 [cs, stat]*, May 2016. arXiv: 1505.07818.

Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. PAC-Bayes and Domain Adaptation. *Neurocomputing*, 379:379–397, February 2020. arXiv: 1707.05712.

Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

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

Dongyoon Han, Junsuk Choe, Seonghyeok Chun, John Joon Young Chung, Minsuk Chang, Sangdoo Yun, Jean Y. Song, and Seong Joon Oh. Neglected free lunch - learning image classifiers using annotation byproducts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20200–20212, October 2023.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016b.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with Side Information through Modality Hallucination. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 826–834. IEEE, 2016.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 590–597, 2019.

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

Bastian Jung and Fredrik Daniel Johansson. Efficient learning of nonlinear prediction models with time-series privileged information. In *Advances in Neural Information Processing Systems*, 2022.

Rickard Karlsson, Martin Willbo, Zeshan Hussain, Rahul G. Krishnan, David A. Sontag, and Fredrik D. Johansson. Using time-series privileged information for provably efficient learning of prediction models. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics 2022*, 2021.

Anurag Katakkar, Clay H. Yoo, Weiqin Wang, Zachary Lipton, and Divyansh Kaushik. Practical benefits of feature feedback under distribution shift. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 346–355, 2022.

D. Kaushik, A. Setlur, E. H. Hovy, and Z. C. Lipton. Explaining the efficacy of counterfactually augmented data. In *International Conference on Learning Representations*, 2021.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Jogendra Nath Kundu, Suvaansh Bhambri, Akshay Kulkarni, Hiran Sarkar, Varun Jampani, and R. Venkatesh Babu. Concurrent subsidiary supervision for unsupervised source-free domain adaptation. In *Proceedings of the 17th European Conference on Computer Vision (ECCV)*, pp. 177–194, 2022.

Yann Lecun. Gradient-Based Learning Applied to Document Recognition. *proceedings of the IEEE*, 86(11): 47, 1998.

Yanmeng Li, Huaijiang Sun, and Wenzhu Yan. Domain adaptive twin support vector machine learning using privileged information. *Neurocomputing*, 469:13–27, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*. European Conference on Computer Vision, September 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pp. 2200–2207, USA, 2013. IEEE Computer Society.

David Lopez-Paz, Leon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR 2016)*, 2016.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, second edition, 2018.

Saeid Motiian. *Domain Adaptation and Privileged Information for Visual Recognition*. PhD thesis, West Virginia University, 2019. Graduate Theses, Dissertations, and Problem Reports. 6271.

Shota Orihashi, Mana Ihori, Tomohiro Tanaka, and Ryo Masumura. Unsupervised Domain Adaptation for Dialogue Sequence Labeling Based on Hierarchical Adversarial Training. In *Interspeech 2020*, pp. 1575–1579. ISCA, October 2020.

Dmitry Pechyony and Vladimir Vapnik. On the theory of learnining with privileged information. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

Stefanos Poulis and Sanjoy Dasgupta. Learning with feature feedback: from theory to practice. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

S Ren, K He, R Girshick, and J Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2016.

Joshua Robinson, Stefanie Jegelka, and Suvrit Sra. Strength from Weakness: Fast Learning Using Weak Supervision. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 8127–8136. PMLR, November 2020.

Nikolaos Sarafianos, Michalis Vrigkas, and Ioannis A. Kakadiaris. Adaptive SVM+: Learning with privileged information for domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

Tawseef Ayoub Shaikh, Rashid Ali, and M. M. Sufyan Beg. Transfer learning privileged information fuels CAD diagnosis of breast cancer. *Machine Vision and Applications*, 31(1):9, February 2020.

Viktoriia Sharmanska, Novi Quadrianto, and Christoph H. Lampert. Learning to Transfer Privileged Information. *arXiv:1410.0389 [cs, stat]*, October 2014. arXiv: 1410.0389.

Kendrick Shen, Robbie M Jones, Ananya Kumar, Sang Michael Xie, Jeff Z. Haochen, Tengyu Ma, and Percy Liang. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19847–19878. PMLR, 17–23 Jul 2022.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Catarina Silva, Armando Vieira, Antonio Gaspar-Cunha, and Joao Carvalho das Neves. Financial distress model prediction using SVM+. In *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–7, July 2010.

M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge books online. Cambridge University Press, 2012.

Marian Tietz, Thomas J. Fan, Daniel Nouri, Benjamin Bossan, and skorch Developers. *skorch: A scikit-learn compatible neural network library that wraps PyTorch*, July 2017.

Vladimir Vapnik and Rauf Izmailov. Learning Using Privileged Information: Similarity Control and Knowledge Transfer. *Journal of Machine Learning Research*, 16:2023–2049, 2015.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, July 2009.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027, 2017.

Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7363–7372, 2019.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.

Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International conference on machine learning*, pp. 6872–6881. PMLR, 2019.

Sang Michael Xie, Ananya Kumar, Robbie Jones, Fereshte Khani, Tengyu Ma, and Percy Liang. In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. *arXiv preprint arXiv:2012.04550*, 2020.

Yalan Ye, Ziqi Liu, Yangwuyong Zhang, Jingjing Li, and Hengtao Shen. Alleviating style sensitivity then adapting: Source-free domain adaptation for medical image segmentation. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pp. 1935–1944, New York, NY, USA, 2022. Association for Computing Machinery.

John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683, 2018.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International conference on machine learning*, pp. 7404–7413. PMLR, 2019.

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

A Experimental Details

In this section, we give further details of the experiments. All code is written in Python and we mainly use PyTorch in combination with skorch (Tietz et al., 2017) for our implementations of the networks. For Faster R-CNN, we adapt the implementation provided by torchvision through the function `fasterrcnn_resnet50_fpn`. For DANN and MDD, we use the ADAPT TensorFlow implementation (de Mathelin et al., 2021) with a ResNet-50-based encoder. We initially set the trade-off parameter λ, which controls the amount of domain adaption regularization, to 0 and then increase it to 0.1 in 10,000 gradient steps according to the formula $\lambda = \beta(2/(1 + e^{-p}) - 1)/C$, where p increases linearly from 0 to 1, β is a parameter specified for each experiment, and $C = 2/(1 + e^{-1}) - 1$. For MDD, we fix the margin parameter γ to 3. The source and target baselines are based on the ResNet-50 architecture when PI is provided as multiple regions of interest; otherwise, the ResNet-18 architecture is used. The architecture of DALUPI in each experiment is specified in the respective subsection below.

We use the Adam optimizer in all experiments. Learning rate decay is treated as a hyperparameter. For ADAPT models (DANN and MDD), the learning rate is either constant or decayed according to $\mu_0/(1 + \alpha p)^{3/4}$, where μ_0 is the initial learning rate, p increases linearly from 0 to 1, and α is a parameter specified in each experiment (see below). For non-ADAPT models, the learning rate is either constant or decayed by a factor 0.1 every nth epoch, where n is another hyperparameter.

For all models except LUPI and DALUPI, the classifier network following the encoder is a simple MLP with two possible settings: Either it is a single linear layer from inputs to outputs or a three-layer network with ReLU activations between the layers. This choice is treated as a hyperparameter in our experiments. The nonlinear case has the following structure where n is the number of input features:

- fully connected layer with n neurons
- ReLU activation layer
- fully connected layer with $n//2$ neurons
- ReLU activation layer
- fully connected layer with $n//4$ neurons.

All models were trained using NVIDIA Tesla A40 GPUs and the development and evaluation of this study required approximately 30,000 hours of GPU training. The code is available on GitHub: `https://github.com/Healthy-AI/dalupi`.

A.1 DALUPI With Two-stage Classifier

Here, we describe in more detail how we construct our two-stage classifier for image classification when privileged information is provided as a single region of interest as in the digit classification task (Section 4.2). When privileged information is provided as binary attributes, we can directly learn the two-stage estimator according to Equation 2. In this task, it was found that using the cross entropy loss and using continuous outputs from f provided superior performance compared to other losses. In the digit classification task, each image x_i has a single label $y_i \in \{0, \ldots, 4\}$ determined by the MNIST digit. Privileged information is given by a single bounding box with coordinates $t_i \in \mathbb{R}^4$ enclosing a subset of pixels w_i corresponding to the digit. The training procedure is summarized in Algorithm 1 and further described below.

We first learn $\hat{d}$ which is a function that takes target image data, $\tilde{x}_i$, and bounding box coordinates, t_i, and learns to output bounding box coordinates, $\hat{t}_i$, which should contain the privileged information w_i. Note that we do not exactly follow the setup in Equation 2 since we do not need to actually predict the pixel values within the bounding box. If we find a good enough estimator of t_i we should minimize the loss of f in Equation 2. To obtain the privileged information we apply a deterministic function ϕ which crops and scales an image using the associated bounding box, t_i. We can now write the composition of these two functions as $\hat{f}(x_i) = \phi(x_i, \hat{d}(x_i))$ which outputs the privileged information. The function ϕ is hard-coded and therefore not learned.

In the second step, we learn $\hat{g}$ to predict the label from the privileged information w_i, which is a cropped version of x_i where the cropping is defined by the bounding box t_i around the digit. This cropping and resizing is performed by ϕ. When we evaluate the performance of this classifier we combine the two models into one, $\hat{h}(x) = \hat{g}(\phi(x, \hat{d}(x)))$. We use the mean squared error loss for learning $\hat{d}$ and categorical cross-entropy (CCE) loss for $\hat{g}$.

Algorithm 1 Training of the two-stage model.

procedure TWO_STAGE $(\tilde{x}_i, w_i, t_i, y_i)$
  Empirically minimize $\frac{1}{m}\sum_{i=1}^{m} \|d(\tilde{x}_i) - t_i\|^2$ and obtain $\hat{d}$.
  Empirically minimize $\frac{1}{n}\sum_{i=1}^{n} CCE(g(w_i), y_i)$ and obtain $\hat{g}$.
  Compose $\hat{d}$, $\hat{g}$ and ϕ into $\hat{h}(x) = \hat{g}(\phi(x, \hat{d}(x)))$.
end procedure

A.2 DALUPI With Faster R-CNN

For multi-label classification, we adapt Faster R-CNN (Ren et al., 2016) outlined in Figure 7 and described below. Faster R-CNN uses a region proposal network (RPN) to generate region proposals which are fed to a detection network for classification and bounding box regression. This way of solving the task in subsequent steps has similarities with our two-stage algorithm although Faster R-CNN can be trained end-to-end. We make small modifications to the training procedure of the original model in the end of this section.

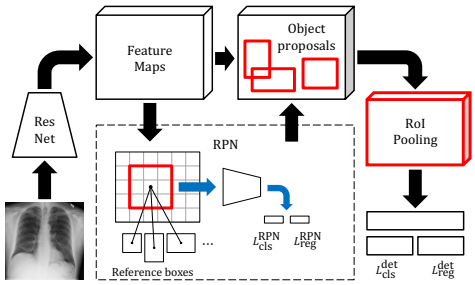


Figure 7: Faster R-CNN (Ren et al., 2016) architecture. The RoI pooling layer and the classification and regression layers are part of the Fast R-CNN detection network (Girshick, 2015).

The RPN generates region proposals relative to a fixed number of reference boxes—anchors—centered at the locations of a sliding window moving over convolutional feature maps. Each anchor is assigned a binary label $p \in \{0, 1\}$ based on its overlap with ground-truth bounding boxes; positive anchors are also associated with a ground-truth box with location t. The RPN loss for a single anchor is

$$L^{\text{RPN}}(\hat{p}, p, \hat{t}, t) := L^{\text{RPN}}_{\text{cls}}(\hat{p}, p) + pL^{\text{RPN}}_{\text{reg}}(\hat{t}, t), \tag{3}$$

where $\hat{t}$ represents the refined location of the anchor and $\hat{p}$ is the estimated probability that the anchor contains an object. The binary cross-entropy loss and a smooth L_1 loss are used for the classification loss $L^{\text{RPN}}_{\text{cls}}$ and the regression loss $L^{\text{RPN}}_{\text{reg}}$, respectively. For a mini-batch of images, the total RPN loss is computed based on a subset of all anchors, sampled to have a ratio of up to 1:1 between positive and negative ditto.

A filtered set of region proposals are projected onto the convolutional feature maps. For each proposal, the detection network—Fast R-CNN (Girshick, 2015)—outputs a probability $\hat{p}(k)$ and a predicted bounding box location $\hat{t}(k)$ for each class k. Let $\hat{p} = (\hat{p}(0), \ldots, \hat{p}(K))$, where $\sum_k \hat{p}(k) = 1$, K is the number of classes and 0 represents a catch-all background class. For a single proposal with ground-truth coordinates t and multi-class label $u \in \{0, \ldots, K\}$, the detection loss is

$$L^{\text{det}}(\hat{p}, u, \hat{t}_u, t) = L^{\text{det}}_{\text{cls}}(\hat{p}, u) + \mathbf{I}_{u \geq 1} L^{\text{det}}_{\text{reg}}(\hat{t}_u, t), \tag{4}$$

where $L^{\text{det}}_{\text{cls}}(\hat{p}, u) = -\log \hat{p}(u)$ and $L^{\text{det}}_{\text{reg}}$ is a smooth L_1 loss. To obtain a probability vector for the entire image, we maximize, for each class k, over the probabilities of all proposals.

During training, Faster R-CNN requires that all input images x come with at least one ground-truth annotation (bounding box) w and its corresponding label u. To increase sample-efficiency, we enable training the model using non-annotated but labeled samples (x, y) from the source domain and annotated but unlabeled samples $(\tilde{x}, \tilde{w})$ from the target domain. In the RPN, no labels are needed, and we simply ignore anchors from non-annotated images when sampling anchors for the loss computation. For the computation of Equation 4, we handle the two cases separately. We assign the label $u = -1$ to all ground-truth annotations from the target domain and multiply $L^{\text{det}}_{\text{cls}}$ by the indicator $\mathbf{I}_{u \geq 0}$. For non-annotated samples (x, y) from the source domain, there are no box-specific coordinates t or labels u but only the labels y for the entire image. In this case, 4 is undefined and we instead compute the binary cross-entropy loss between the per-image label and the probability vector for the entire image.

We train the RPN and the detection network jointly as described in Ren et al. (2016). To extract feature maps, we use a Feature Pyramid Network (Lin et al., 2017) on top of a ResNet-50 architecture He et al. (2016b). We use the modified model in the experiments in Section 4.3 and 4.4. In Section 4.3, we also train this model in a LUPI setting, where no information from the target domain is used.

A.3 Celebrity Photo Classification With Binary Attribute Vector

In our experiment based on CelebA (Liu et al., 2015), the input x is an RGB image which has been resized to 64×64 pixels, the target y is a binary label for gender of the subject of the image, and the privileged information w are 7 binary-valued attributes. The attributes used in this experiment are: `Bald`, `Bangs`, `Mustache`, `Smiling`, `5_o_Clock_Shadow`, `Oval_Face` and `Heavy_Makeup`. We use a subset of the CelebA dataset with 2,000 labeled source examples and 3,000 unlabeled target examples. We use 1,000 samples each for the source validation set, source test set, and target test set, respectively. The target oracle, SL-T, is trained using labels provided for the 3,000 target examples, with 20 % of these examples set aside for validation. The same unlabeled validation set is used to validate the first DALUPI network, $\hat{f}$. When using privileged information from the source domain to train $\hat{f}$, we use 30,000 extra samples (x, w) with PI.

For DALUPI, we use the two-stage estimator with the network $\hat{f}$ based on ResNet-18 followed by a non-linear MLP. The network $\hat{g}$ is an MLP with two hidden layers of with 256 neurons each. We train the models for 100 epochs. If the validation accuracy (or validation AUC for $\hat{f}$) does not improve for 10 subsequent epochs, we stop the training earlier. For DALUPI, the early stopping patience is 15 for each network. We treat the problem as multi-class classification with two classes and use the categorical cross entropy loss for SL-S, SL-T, DANN, and MDD.

A.3.1 Hyperparameters

We randomly choose hyperparameters from the following predefined sets of values:

- SL-S and SL-T:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - step size n for learning rate decay: (15, 30, 100)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - dropout (encoder): (0, 0.1, 0.2, 0.5)
  - nonlinear classifier: (`True`, `False`).
- DALUPI:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-5}, 1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - step size n for learning rate decay: (15, 30, 100)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$.
- DANN:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - parameter α for learning rate decay: (0, 1.0)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - dropout (encoder): $(0, 0.1, 0.2, 0.5)$
  - width of discriminator network: $(64, 128, 256)$
  - depth of discriminator network: $(2, 3)$
  - nonlinear classifier: (`True`, `False`)
  - parameter β for adaption regularization decay: (0.1, 1.0, 10.0).
- MDD:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$

- parameter α for learning rate decay: (0, 1.0)
- weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
- dropout (encoder): $(0, 0.1, 0.2, 0.5)$
- nonlinear classifier: (`True`, `False`)
- maximum norm value for classifier weights: (0.5, 1.0, 2.0)
- parameter β for adaption regularization decay: (0.1, 1.0, 10.0).

A.4 Digit Classification With Single Bounding Box as PI

In the digit classification task, we separate 20 % of the available source and target data into a test set. We likewise use 20 % of the training data for validation purposes. For DALUPI we use ResNet-18 for the function $\hat{f}$. We replace the default fully connected layer with a fully connected layer with 4 neurons to predict the coordinates of the bounding box. The predicted bounding box is resized to a 28×28 square no matter the initial size. We use a simple convolutional neural network for the function $\hat{g}$ with the following structure:

- convolutional layer with 16 output channels, kernel size of 5, stride of 1, and padding of 2
- max pooling layer with kernel size 2, followed by a ReLU activation
- convolutional layer with 32 output channels, kernel size of 5, stride of 1, and padding of 2
- max pooling layer with kernel size 2, followed by a ReLU activation
- dropout layer with $p = 0.4$
- fully connected layer with 50 out features, followed by ReLU activation
- dropout layer with $p = 0.2$
- fully connected layer with 5 out features.

The model training is stopped when the best validation accuracy (or validation loss for $\hat{f}$) does not improve over 10 epochs or when the model has been trained for 100 epochs, whichever occurs first. All models are trained from scratch, without pretrained weights. We use the categorical cross entropy loss for SL-S, SL-T, DANN, and MDD.

A.4.1 Hyperparameters

We randomly choose hyperparameters from the following predefined sets of values:

- SL-S and SL-T:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - step size n for learning rate decay: (15, 30, 100)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - dropout (encoder): (0, 0.1, 0.2, 0.5)
  - nonlinear classifier: (`True`, `False`).
- DALUPI:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - step size n for learning rate decay: (15, 30, 100)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$.

Table 5: Marginal label distribution in source and target domains for the entity classification task based on the MS-COCO dataset. The background class contains images where none of the four entities are present.

Domain	Person	Dog	Cat	Bird	Background
Source	2,963	569	1,008	213	1,000
Target	3,631	1,121	423	712	1,000

- DANN:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - parameter α for learning rate decay: (0, 1.0)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - dropout (encoder): $(0, 0.1, 0.2, 0.5)$
  - width of discriminator network: $(64, 128, 256)$
  - depth of discriminator network: $(2, 3)$
  - nonlinear classifier: (`True`, `False`)
  - parameter β for adaption regularization decay: (0.1, 1.0, 10.0).
- MDD:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - parameter α for learning rate decay: (0, 1.0)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - dropout (encoder): $(0, 0.1, 0.2, 0.5)$
  - nonlinear classifier: (`True`, `False`)
  - maximum norm value for classifier weights: (0.5, 1.0, 2.0)
  - parameter β for adaption regularization decay: (0.1, 1.0, 10.0).

A.5 Entitity Classification With Multiple Regions of Interest as PI

In the entity classification experiment, we train all models for at most 50 epochs. If the validation AUC does not improve for 10 subsequent epochs, we stop the training earlier. No pretrained weights are used in this experiment since we find that the task is too easy to solve with pretrained weights. For DALUPI and LUPI, we use the end-to-end solution based on Faster R-CNN (see Section A.2). We use the default anchor sizes for each of the feature maps (32, 64, 128, 256, 512), and for each anchor size we use the default aspect ratios (0.5, 1.0, 2.0). We use the binary cross entropy loss for SL-S, SL-T, DANN, and MDD.

We use the 2017 version of the MS-COCO dataset (Lin et al., 2014). As decribed in Section 4.3, we extract indoor images by sorting out images from the super categories "indoor" and "appliance" that also contain at least one of the entity classes. Outdoor images are extracted in the same way using the super categories "vehicle" and "outdoor". Images that match both domains (for example an indoor image with a toy car) are removed, as are any gray-scale images. We also include 1,000 negative examples, i.e., images with none of the entities present, in both domains. In total, there are 5,231 images in the source domain and 5,719 images in the target domain. From these pools, we randomly sample 3,000, 1,000, and 1,000 images for training, validation, and testing, respectively. In Table 5 we describe the label distribution in both domains. All images are resized to 320×320.

A.5.1 Hyperparameters

We randomly choose hyperparameters from the following predefined sets of values. For information about the specific parameters in LUPI and DALUPI, we refer to the paper by Ren et al. (2016). Here, RoI and NMS refer to region of interest and non-maximum suppression, respectively.

- SL-S and SL-T:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - step size n for learning rate decay: (15, 30, 100)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - dropout (encoder): (0, 0.1, 0.2, 0.5)
  - nonlinear classifier: (`True`, `False`).
- DANN:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - parameter α for learning rate decay: (0, 1.0)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - dropout (encoder): $(0, 0.1, 0.2, 0.5)$
  - width of discriminator network: $(64, 128, 256)$
  - depth of discriminator network: $(2, 3)$
  - nonlinear classifier: (`True`, `False`)
  - parameter β for adaption regularization decay: (0.1, 1.0, 10.0).
- MDD:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - parameter α for learning rate decay: (0, 1.0)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - dropout (encoder): $(0, 0.1, 0.2, 0.5)$
  - nonlinear classifier: (`True`, `False`)
  - maximum norm value for classifier weights: (0.5, 1.0, 2.0)
  - parameter β for adaption regularization decay: (0.1, 1.0, 10.0).
- LUPI and DALUPI:
  - batch size: $(16, 32, 64)$
  - learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - step size n for learning rate decay: (15, 30, 100)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - IoU foreground threshold (RPN): (0.6, 0.7, 0.8, 0.9)
  - IoU background threshold (RPN): (0.2, 0.3, 0.4)
  - batchsize per image (RPN): (32, 64, 128, 256)
  - fraction of positive samples (RPN): (0.4, 0.5, 0.6, 0.7)
  - NMS threshold (RPN): (0.6, 0.7, 0.8)
  - RoI pooling output size (Fast R-CNN): (5, 7, 9)
  - IoU foreground threshold (Fast R-CNN): (0.5, 0.6)
  - IoU background threshold (Fast R-CNN): (0.4, 0.5)
  - batchsize per image (Fast R-CNN): (16, 32, 64, 128)
  - fraction of positive samples (Fast R-CNN): (0.2, 0.25, 0.3)
  - NMS threshold (Fast R-CNN): (0.4, 0.5, 0.6)
  - detections per image (Fast R-CNN): (25, 50, 75, 100).

Table 6: Marginal distribution of labels of images and bounding boxes in the source and target domain, respectively, for the chest X-ray classification experiment. ATL=Atelectasis; CM=Cardiomegaly; PE=Effusion; NF=No Finding.

Data	ATL	CM	PE	NF
$x \sim \mathcal{S}$	11,559	2,776	13,317	60,361
$w \sim \mathcal{S}$	180	146	153	-
$\tilde{x} \sim \mathcal{T}$	14,278	20,466	74,195	16,996
$\tilde{w} \sim \mathcal{T}$	75	66	64	-

A.6 X-ray Classification With Multiple Regions of Interest as PI

In the X-ray classification experiment, we train all models for at most 50 epochs, using pre-trained weights in the ResNet architecture of each model. If the validation AUC does not improve for 10 subsequent epochs, we stop the training earlier. We then fine-tune all models, except DANN and MDD, for up to 20 additional epochs. The number of encoder layers that are fine-tuned is a hyperparameter for which we consider different values. We start the training with weights pretrained on ImageNet. For DALUPI, we use the end-to-end solution based on Faster R-CNN (see Section A.2). We use the default anchor sizes for each of the feature maps (32, 64, 128, 256, 512), and for each anchor size we use the default aspect ratios (0.5, 1.0, 2.0). We use the binary cross entropy loss for SL-S, SL-T, DANN, and MDD.

In total, there are 83,519 (457) and 120,435 (118) images (annotated images) in the source and target domain, respectively. The distributions of labels and bounding box annotations are provided in Table 6. Here, "NF" refers to images with no confirmed findings. In the annotated images, there are 180/146/153 and 75/66/64 examples of ATL/CM/PE in each domain respectively. Validation and test sets are sampled from non-annotated images and contain 10,000 samples each. All annotated images are reserved for training. We merge the default training and validation datasets before splitting the data and resize all images to 320×320. For the source dataset (ChestX-ray8), the bounding boxes can be found together with the dataset. The target segmentations can be found here: `https://stanfordaimi.azurewebsites.net/datasets/23c56a0d-15de-405b-87c8-99c30138950c`.

A.6.1 Hyperparameters

We choose hyperparameters randomly from the following predefined sets of values. For information about the specific parameters in DALUPI, we refer to the paper by Ren et al. (2016). RoI and NMS refer to region of interest and non-maximum suppression, respectively.

- SL-S and SL-T:
  - batch size: (16, 32, 64)
  - learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - dropout (encoder): (0, 0.1, 0.2, 0.5)
  - nonlinear classifier: (`True`, `False`)
  - number of layers to fine-tune: (3, 4, 5)
  - learning rate (fine-tuning): $(1.0 \times 10^{-5}, 1.0 \times 10^{-4})$.
- DANN:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - parameter α for learning rate decay: (0, 1.0)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$

  - number of trainable layers (encoder): (1, 2, 3, 4, 5)
  - dropout (encoder): $(0, 0.1, 0.2, 0.5)$
  - width of discriminator network: $(64, 128, 256)$
  - depth of discriminator network: $(2, 3)$
  - nonlinear classifier: (`True`, `False`)
  - parameter β for adaption regularization decay: (0.1, 1.0, 10.0).
- MDD:
  - batch size: $(16, 32, 64)$
  - initial learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - parameter α for learning rate decay: (0, 1.0)
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - number of trainable layers (encoder): (1, 2, 3, 4, 5)
  - dropout (encoder): $(0, 0.1, 0.2, 0.5)$
  - nonlinear classifier: (`True`, `False`)
  - maximum norm value for classifier weights: (0.5, 1.0, 2.0)
  - parameter β for adaption regularization decay: (0.1, 1.0, 10.0).
- DALUPI:
  - batch size: $(16, 32, 64)$
  - learning rate: (1.0×10^{-4})
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - IoU foreground threshold (RPN): (0.6, 0.7, 0.8, 0.9)
  - IoU background threshold (RPN): (0.2, 0.3, 0.4)
  - batchsize per image (RPN): (32, 64, 128, 256)
  - fraction of positive samples (RPN): (0.4, 0.5, 0.6, 0.7)
  - NMS threshold (RPN): (0.6, 0.7, 0.8)
  - RoI pooling output size (Fast R-CNN): (5, 7, 9)
  - IoU foreground threshold (Fast R-CNN): (0.5, 0.6)
  - IoU background threshold (Fast R-CNN): (0.4, 0.5)
  - batchsize per image (Fast R-CNN): (16, 32, 64, 128)
  - fraction of positive samples (Fast R-CNN): (0.2, 0.25, 0.3)
  - NMS threshold (Fast R-CNN): (0.4, 0.5, 0.6)
  - detections per image (Fast R-CNN): (25, 50, 75, 100)
  - learning rate (fine-tuning): $(1.0 \times 10^{-5}, 1.0 \times 10^{-4})$
  - number of layers to fine-tune: (3, 4, 5).

B Additional Results

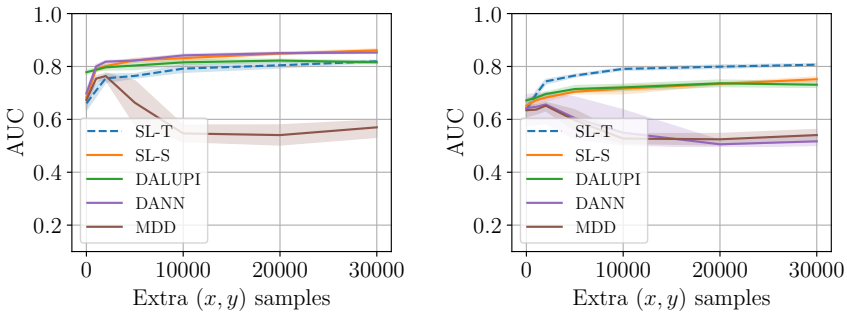
In Figure 8a and 8b, we show some example images from the digit classification task with associated saliency maps from the source-only model for different values of the skew parameter ϵ. We can see that for a lower value of epsilon the SL-S model activations seem concentrated on the area with the digit, while when the correlation with the background is large the model activations are more spread out.

In Figure 9, we show the *average* AUC when additional training data of up to 30,000 samples are added in the chest X-ray experiment. We see that, once given access to a much larger amount of labeled samples, SL-S and DALUPI perform comparably in the target domain.

In Figure 10, we show AUC for the pathology CM when additional training data *without* bounding box annotations are added. We see that SL-S catches up to the performance of DALUPI when a large amount of labeled examples are provided. These results indicate that identifiability is not the primary obstacle for adaptation, and that PI improves sample efficiency.

(a) SL-S, $\epsilon = 0.2$ (b) SL-S, $\epsilon = 1.0$

Figure 8: Example images (top) and saliency maps (bottom) from SL-S when trained with source skew $\epsilon = 0.2$ (a) and $\epsilon = 1$ (b).



(a) Source test AUC. (b) Target test AUC.

Figure 9: Classification of chest X-ray images. Model performance on source (a) and target (b) domains. The AUC is averaged over the three pathologies: ATL, CM and PE. The 95 % confidence intervals are computed using bootstrapping the results over five seeds.

C Proof of Proposition 1

Proposition. Let Assumptions 1 and 2 be satisfied w.r.t. W (not necessarily w.r.t. X) and let Assumption 3 hold as stated. Then, the target risk $R_{\mathcal{T}}$ is identified for hypotheses $h : \mathcal{X} \to \mathcal{Y}$,

$$R_{\mathcal{T}}(h) = \int_x \mathcal{T}(x) \int_w \mathcal{T}(w \mid x) \int_y \mathcal{S}(y \mid w) L(h(x), y) \mathrm{d}y \mathrm{d}w \mathrm{d}x \ .$$

and, for L the squared loss, a minimizer of $R_{\mathcal{T}}$ is $h^*_{\mathcal{T}}(x) = \int_w \mathcal{T}(w \mid x) \mathbb{E}_{\mathcal{S}}[Y \mid w] \mathrm{d}w$.

Proof. By definition, $R_{\mathcal{T}}(h) = \int_{x,y} \mathcal{T}(x, y) L(h(x), y) \mathrm{d}y \mathrm{d}x$. We marginalize over W to get

$$\begin{aligned}
\mathcal{T}(x, y) &= \mathcal{T}(x) \mathbb{E}_{\mathcal{T}(W|x)} \left[\mathcal{T}(y \mid W, x) \mid x \right] \\
&= \mathcal{T}(x) \mathbb{E}_{\mathcal{T}(W|x)} [\mathcal{T}(y \mid W) \mid x] \\
&= \mathcal{T}(x) \int_{w: \mathcal{T}(w) > 0} \mathcal{T}(w \mid x) \mathcal{S}(y \mid w) \mathrm{d}w \\
&= \mathcal{T}(x) \int_{w: \mathcal{S}(w) > 0} \mathcal{T}(w \mid x) \mathcal{S}(y \mid w) \mathrm{d}w \ .
\end{aligned}$$

where the second equality follows by sufficiency and the third by covariate shift and overlap in W. $\mathcal{T}(x), \mathcal{T}(w \mid x)$ and $\mathcal{S}(y \mid w)$ are observable through training samples. That $h^*_{\mathcal{T}}$ is a minimizer follows from the first-order

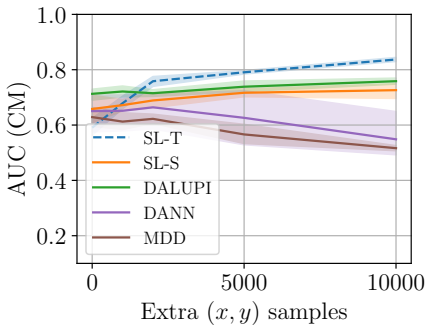


Figure 10: Test AUC for CM in $\mathcal{T}$. DALUPI outperforms the other models when no extra (x, y) samples are provided. Adding examples without bounding box annotations improves the performance of SL-S and SL-T, eventually causing the latter to surpass DALUPI.

condition of setting the derivative of the risk with respect to h to 0. This strategy yields the well-known result that

$$h^*_{\mathcal{T}} = \arg\min_h \mathbb{E}_{\mathcal{T}}[(h(X) - Y^2)] = \mathbb{E}_{\mathcal{T}}[Y \mid X] \ .$$

By definition and the previous result, we have that

$$\begin{aligned}
\mathbb{E}_{\mathcal{T}}[Y \mid X = x] &= \int_y y \frac{\mathcal{T}(x, y)}{\mathcal{T}(x)} \mathrm{d}y \\
&= \int_y \int_{w:\mathcal{S}(w)>0} \mathcal{T}(w \mid x)\mathcal{S}(y \mid w) y \mathrm{d}w \mathrm{d}y \\
&= \int_w \mathcal{T}(w \mid x)\mathbb{E}_{\mathcal{S}}[Y \mid x] \mathrm{d}w
\end{aligned}$$

and we have the result. □

D Proof of Proposition 2

Proposition 2. *Assume that $\mathcal{G}$ comprises M-Lipschitz mappings from the privileged information space $\mathcal{W} \subseteq \mathbb{R}^{d_W}$ to $\mathcal{Y}$. Further, assume that both the ground truth privileged information W and label Y are deterministic in X and W respectively. Let ρ be the domain density ratio of W and let Assumptions 1–3 (Covariate shift, Overlap and Sufficiency) hold w.r.t. W. Further, let the loss L be uniformly bounded by some constant B and let d and d' be the pseudo-dimensions of $\mathcal{G}$ and $\mathcal{F}$ respectively. Assume that there are n observations from the source (labeled) domain and m from the target (unlabeled) domain. Then, with L the squared Euclidean distance, for any $h = h \circ f \in \mathcal{G} \times \mathcal{F}$, w.p. at least $1 - \delta$,*

$$\begin{aligned}
\frac{R_{\mathcal{T}}(h)}{2} \leq\ & \hat{R}^{Y,\rho}_{\mathcal{S}}(g) + M^2 \hat{R}^{W}_{\mathcal{T}}(f) \\
& + 2^{5/4} \sqrt{d_2(\mathcal{T} \| \mathcal{S})} \sqrt[4]{\frac{d \log \frac{2me}{d} + \log \frac{4}{\delta}}{m}} \\
& + d_{\mathcal{W}} B M^2 \left(\sqrt{\frac{2d' \log \frac{en}{d'}}{n}} + \sqrt{\frac{\log \frac{d_{\mathcal{W}}}{\delta}}{2n}} \right) .
\end{aligned}$$

Proof. Decomposing the risk of $h \circ \phi$, we get

$$\begin{aligned}
&R_{\mathcal{T}}(h) = \mathbb{E}_{\mathcal{T}}[(g(f(X)) - Y)^2] \\
&\leq 2\mathbb{E}_{\mathcal{T}}[(g(W) - Y)^2 + (g(f(X)) - g(W))^2] \\
&\leq 2\mathbb{E}_{\mathcal{T}}[(g(W) - Y)^2 + M^2\|f(X)) - g(W)\|^2] \\
&\leq 2\mathbb{E}_{\mathcal{T}}[(g(W) - Y)^2] + 2M^2\mathbb{E}_{\mathcal{T}}[\|(f(X) - W)\|^2] \\
&= 2R_{\mathcal{T}}^{Y}(g) + 2M^2 R_{\mathcal{T}}^{W}(f) = 2\underbrace{R_{S}^{Y,\rho}(g)}_{(I)} + 2M^2\underbrace{R_{\mathcal{T}}^{W}(f)}_{(II)}.
\end{aligned}$$

The first inequality follows from the relaxed triangle inequality, the second inequality from the Lipschitz property and the third equality from Overlap and Covariate shift. We will bound these quantities separately starting with (I).

We assume that the pseudo-dimension of $\mathcal{G}$, d is bounded. Further, we assume that the second moment of the density ratios, equal to the Rényi divergence $d_2(\mathcal{T}\|\mathcal{S}) = \Sigma_{w \in cG}\mathcal{T}(w)\frac{\mathcal{T}(w)}{\mathcal{S}(w)}$ are bounded and that the density ratios are non-zero for all $w \in \mathcal{G}$. Let $D_1 = \{w_i, y_i\}_{i=0}^{m}$ be a dataset drawn i.i.d from the source domain. Then by application of Theorem 3 from Cortes et al. (2010) we obtain with probability $1 - \delta$ over the choice of D_1,

$$(I) = R_{S}^{Y,\rho}(g) \leq \hat{R}_{S}^{Y,\rho}(g) + 2^{5/4}\sqrt{d_2(\mathcal{T}\|\mathcal{S})}\sqrt[3/8]{\frac{d\log\frac{2me}{d} + \log\frac{4}{\delta}}{m}}$$

Now for (II) we treat each component of $w \in \mathcal{W}$ as a regression problem independent from all the others. So we can therefore write the risk as the sum of the individual component risks

$$R_{\mathcal{T}}^{W}(f) = \Sigma_{i=1}^{d_{\mathcal{W}}} R_{\mathcal{T},i}^{W}(f)$$

Let the pseudo-dimension of $\mathcal{F}$ be denoted d, $D_2 = \{x_i, w_i\}_{i=0}^{n}$ be a dataset drawn i.i.d from the target domain. Then, using theorem 11.8 from Mohri et al. (2018) we have that for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of D_2, the following inequality holds for all hypotheses $f \in \mathcal{F}$ for each component risk

$$R_{\mathcal{T},i}^{W}(f) \leq \hat{R}_{\mathcal{T},i}^{W}(f) + B\left(\sqrt{\frac{2d'\log\frac{en}{d'}}{n}} + \sqrt{\frac{\log\frac{1}{\delta}}{2n}}\right)$$

We then simply make all the bounds hold simultaneously by applying the union bound and having it so that each bound must hold with probability $1 - \frac{\delta}{d_{\mathcal{W}}}$ which results in

$$\begin{aligned}
R_{\mathcal{T}}^{W}(f) &= \Sigma_{i=1}^{d_{\mathcal{W}}} R_{\mathcal{T},i}^{W}(f) \leq \Sigma_{i=1}^{d_{\mathcal{W}}}\hat{R}_{\mathcal{T},i}^{W}(f) + \Sigma_{i=1}^{d_{\mathcal{W}}} B\left(\sqrt{\frac{2d'\log\frac{en}{d'}}{n}} + \sqrt{\frac{\log\frac{d_{\mathcal{W}}}{\delta}}{2n}}\right) \\
&= \hat{R}_{\mathcal{T}}^{W}(f) + d_{\mathcal{W}}B\left(\sqrt{\frac{2d'\log\frac{en}{d'}}{n}} + \sqrt{\frac{\log\frac{d_{\mathcal{W}}}{\delta}}{2n}}\right)
\end{aligned}$$

Combination of these two results then yield the proposition statement.

Consistency follows as Y is a deterministic function of W and W is a deterministic fundtion of X and both $\mathcal{H}$ and $\mathcal{F}$ are well-specified. Thus both empirical risks and sample complexity terms will converge to 0 in the limit of infinite samples. □

The parts of the bound shown above can be described as falling into three main categories: Empirical risk(s), domain shift and sample complexity components. A central term that figures both in the weighted empirical

risk and the Rényi divergence is the density ratio $\frac{\mathcal{T}(w)}{\mathcal{S}(w)}$. Therefore, the size of the bound is governed at least in part based on the proximity in $\mathcal{W}$-space the source and target domains are. This is similar to other importance weighting bounds, however, since the experiment designer may choose the form of PI this can be more well-behaved than the density ratio in the input space.

E Proof Sketch for PAC-Bayes Bound

We will here detail a proof sketch for a PAC-Bayes version of the bound we propose in the main text. For the purposes of this bound we will consider the quantity $\mathbb{E}_{h\sim\psi}R_{\mathcal{T}}(h)$, where ψ is a posterior distribution over classifiers $h \sim \psi$. As we are basing the bound on the two-step methodology where we train two different classifiers on separate datasets we assume that we can obtain the posteriors over the component functions separately and independently i.e. $h = f \circ g \sim \psi = \psi_f \times \psi_g$, where $f \sim \psi_f$ and $g \sim \psi_g$. Let the assumptions from proposition 2 hold here. Similar to the previous section we decompose the risk into two parts

$$\begin{aligned}&\mathbb{E}_{h\sim\psi}R_{\mathcal{T}}(h) = \mathbb{E}_{h\sim\psi}\mathbb{E}_{\mathcal{T}}[(g(f(X)) - Y)^2] \\ &= \mathbb{E}_{h\sim\psi}[2R^Y_{\mathcal{T}}(g) + 2M^2R^W_{\mathcal{T}}(f)] = 2\mathbb{E}_{h\sim\psi}\underbrace{R^Y_{\mathcal{T}}(g)}_{(I)} + 2M^2\mathbb{E}_{h\sim\psi}\underbrace{R^W_{\mathcal{T}}(f)}_{(II)}.\end{aligned}$$

We note that since we now have expectations over the composite function h on expressions which depend on only one of the components we can, for example, write the following:

$$\mathbb{E}_{h\sim\psi}R^Y_{\mathcal{T}}(g) = \mathbb{E}_{g\sim\psi_g}R^Y_{\mathcal{T}}(g)$$

This holds as we assume that f and g are not dependent on each other. Therefore, we can just marginalize out the part which is not in use. From this point we can use some of the available bounds from the literature to estimate the resulting part e.g. Corollary 1 from Breitholtz & Johansson (2022). Application of this result yields the following bound on the first term

$$\mathop{\mathbb{E}}_{g\sim\psi_g} R^Y_{\mathcal{T}}(g) \leq \frac{1}{\gamma}\mathop{\mathbb{E}}_{g\sim\psi_g} R^{\hat{Y},\rho}_{\mathcal{S}}(g) + \beta_\infty\frac{\mathrm{KL}(\psi_g\|\pi_g) + \ln(\frac{1}{\delta})}{2\gamma(1-\gamma)m}\ .$$

Thereafter we can use another bound from the literature to estimate the second term, e.g. Theorem 6 from Germain et al. (2020). Using this we obtain the following:

$$\mathop{\mathbb{E}}_{f\sim\psi_f} R^W_{\mathcal{T}}(f) \leq \frac{\alpha}{1-e^{-\alpha}}\left(\mathop{\mathbb{E}}_{f\sim\psi_f}\hat{R}^W_{\mathcal{T}}(f) + \frac{\mathrm{KL}(\psi_f\|\pi_f) + \ln(\frac{1}{\delta})}{n\alpha}\right)\ .$$

Then a bound can be constructed by combining these two results using a union bound argument.

$$\begin{aligned}\mathbb{E}_{h\sim\psi}R_{\mathcal{T}}(h) \leq \frac{2}{\gamma}\mathop{\mathbb{E}}_{g\sim\psi_g} R^{\hat{Y},\rho}_{\mathcal{S}}(g) + \beta_\infty\frac{\mathrm{KL}(\psi_g\|\pi_g) + \ln(\frac{2}{\delta})}{2\gamma(1-\gamma)m} \\ + \frac{2M^2\alpha}{1-e^{-\alpha}}\left(\mathop{\mathbb{E}}_{f\sim\psi_f}\hat{R}^W_{\mathcal{T}}(f) + \frac{\mathrm{KL}(\psi_f\|\pi_f) + \ln(\frac{2}{\delta})}{n\alpha}\right)\end{aligned}$$

F A Bound on the Target Risk Without Suffiency

The sufficiency assumption is used to replace $\mathcal{T}(y \mid x)$ with $\mathcal{T}(y \mid w)$ in the proof of Proposition 1. If sufficiency is violated but it is plausible that the degree of insufficiency is comparable across domains, we can still obtain a bound on the target risk which may be estimated from observed quantities. One way to formalize such an assumption is that there is some $\gamma \geq 1$, for which

$$\sup_{x\in\mathcal{T}(x|w)} \mathcal{T}(y \mid w, x)/\mathcal{T}(y \mid w) \leq \gamma \sup_{x\in\mathcal{S}(x|w)} \mathcal{S}(y \mid w, x)/\mathcal{S}(y \mid w) \tag{5}$$

This may be viewed as a relaxation of suffiency. If Assumption 3 holds, both left-hand and right-hand sides of the inequality are 1. Under Equation 5, with $\Delta_\gamma(w, y)$ equal to the right-hand side the inequality,

$$R_{\mathcal{T}}(h) \leq \int_x \mathcal{T}(x) \int_w \mathcal{T}(w \mid x) \int_y \Delta_\gamma(w, y) \mathcal{S}(y \mid w) L(h(x), y) \mathrm{d}y \mathrm{d}w \mathrm{d}x \ .$$

However, the added assumption is not verifiable statistically.

Paper III

Overcoming label shift in target-aware federated learning

E. Listo Zec, **A. Breitholtz**, F. D. Johansson

Submitted, under review

Overcoming label shift with target-aware federated learning

Edvin Listo Zec*
RISE Research Institutes of Sweden
KTH Royal Institute of Technology
edvin.listo.zec@ri.se

Adam Breitholtz *
Chalmers University of Technology
& University of Gothenburg
adambre@chalmers.se

Fredrik D. Johansson†
Chalmers University of Technology
& University of Gothenburg
fredrik.johansson@chalmers.se

August 27, 2025

Abstract

Federated learning enables multiple actors to collaboratively train models without sharing private data. Existing algorithms are successful and well-justified in this task when the intended *target domain*, where the trained model will be used, shares data distribution with the aggregate of clients, but this is often violated in practice. A common reason is label shift—that the label distributions differ between clients and the target domain. We demonstrate empirically that this can significantly degrade performance. To address this problem, we propose FedPALS, a principled and practical model aggregation scheme that adapts to label shifts *to improve performance in the target domain* by leveraging knowledge of label distributions at the central server. Our approach ensures unbiased updates under federated stochastic gradient descent which yields robust generalization across clients with diverse, label-shifted data. Extensive experiments on image classification tasks demonstrate that FedPALS consistently outperforms baselines by aligning model aggregation with the target domain. Our findings reveal that conventional federated learning methods suffer severely in cases of extreme label sparsity on clients, highlighting the critical need for target-aware aggregation as offered by FedPALS.

1 Introduction

Federated learning (FL) has emerged as a powerful paradigm for training machine learning models collaboratively across multiple clients without sharing data [McMahan et al., 2017, Kairouz et al., 2021]. This is attractive in problems where privacy is paramount, such as healthcare [Sheller et al., 2020], finance [Byrd and Polychroniadou, 2020], and natural language processing [Hilmkil et al., 2021]. While effective when data from different clients are identically distributed, the performance of federated learning can degrade significantly when clients exhibit systematic data heterogeneity, such as label shift [Zhao et al., 2018, Woodworth et al., 2020].

Most federated learning research, even that addressing data heterogeneity, focuses on what we term *standard* federated learning, where the test distribution matches the combined distribution of training clients. However, many real-world applications require generalization to a target client or domain with a distinct, unknown data distribution. Consider a retail scenario: multiple stores (clients) collaboratively train a sales prediction model using their local purchase histories. While each store's data reflects its unique customer base, the goal is to deploy the model in a *new* store (target client) with different customer preferences, and no historical records. This is *target-aware* federated learning, a more challenging paradigm than standard federated learning due to the inherent distributional shift between training and test data.

The problem of generalizing under distributional shifts has been extensively studied in centralized settings, often under the umbrella of domain adaptation [Blanchard et al., 2011, Ganin et al., 2016]. However, traditional domain adaptation techniques, such as sample re-weighting [Lipton et al., 2018] or domain-invariant representation learning [Arjovsky et al., 2020], typically require access to data from both source and target domains. This requirement is incompatible with the decentralized nature of federated learning, where neither the server nor the clients share data between them. While several

*Equal contribution. Order decided by coin toss.
†https://www.healthyai.se/

techniques have been proposed to address client heterogeneity in standard FL, such as regularization [Li et al., 2020a, 2021], clustering [Ghosh et al., 2020, Vardhan et al., 2024], and meta-learning [Chen et al., 2018, Jiang et al., 2019], they do not address the challenge of generalizing to a target client with differing data distribution.

Contributions We introduce the problem of *target-aware* federated learning under label shift, where the goal is to train a model that generalizes well to a target client (domain) whose label distribution differs from those of the training clients (see Section 2). To address this problem, we propose a novel aggregation scheme called FedPALS that optimizes a convex combination of client models to ensure that the aggregated model is better suited for the label distribution of the target domain (Section 3). We prove that the resulting stochastic gradient update behaves, in expectation, as centralized learning in the target domain (Proposition 1), and examine its relation to standard federated averaging (Proposition 3.1). We demonstrate the effectiveness of FedPALS through an extensive empirical evaluation (Section 5), showing that it outperforms traditional approaches in scenarios where distributional shifts pose significant challenges, at the small cost of sharing client label marginals with the central server. Moreover, we observe that traditional methods struggle particularly in scenarios where training clients have sparse label distributions, highlighting the need for target-aware aggregation strategies.

2 Target-aware federated learning with label shift

In federated learning, a global model h_θ is produced by a central server by aggregating updates to model parameters θ from multiple clients [McMahan et al., 2017]. We focus on classification tasks in which the goal is for h_θ to predict the most probable label $Y \in \{1, ..., K\}$ for a given d-dimensional input $X \in \mathcal{X} \subset \mathbb{R}^d$. Each client $i = 1, ..., M$ holds a data set $D_i = \{(x_{i,1}, y_{i,1}), ..., (x_{i,n_i}, y_{i,n_i})\}$ of n_i labeled examples, assumed to be drawn i.i.d. from a *local* client-specific distribution $S_i(X, Y)$. Due to constraints on privacy or communication, these data sets cannot be shared directly with other clients or with the central server.

Learning proceeds over rounds $t = 1, ..., t_{max}$, each comprising three steps: (1) The central server broadcasts the current global model parameters θ_t to all clients; (2) Each client i computes updated parameters $\theta_{i,t}$ based on their local data set D_i, and sends these updates back to the server; (3) The server aggregates the clients' updates, for example, using federated averaging (FedAvg) [McMahan et al., 2017] or related techniques, to produce the new global model θ_{t+1}.

A common implicit assumption in federated learning is that the learned model will be applied in a target domain $T(X, Y)$ that coincides with the aggregate distribution of clients,

$$\bar{S}(X, Y) = \sum_{i=1}^{M} \frac{n_i}{N} S_i(X, Y), \tag{1}$$

where $N = \sum_{i=1}^{M} n_i$. To this end, trained models are evaluated in terms of their average performance over clients. However, in applications, the intended target domain may be different entirely [Bai et al., 2024]. Here, we assume that the target domain is distinct from all client distributions: $\forall i : T(X, Y) \neq S_i(X, Y)$ and from the client aggregate, $T(X, Y) \neq \bar{S}(X, Y)$. We refer to this setting as *target-aware federated learning*. While distributional shift between clients is a well-recognized problem in federated learning, the target domain is still typically the client aggregate $\bar{S}$ [Karimireddy et al., 2020, Li et al., 2020b]. Our setting differs also from federated domain generalization which lacks a specific target domain [Bai et al., 2024].

In target-aware federated learning, the goal is to train a model to predict well in a target domain $T(X, Y)$ *without access to samples from* T. Formally, our objective is to minimize the expected target risk, R_T of a classifier $h_\theta : \mathcal{X} \to \mathcal{Y}$, with respect to a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$,

$$\underset{\theta}{\text{minimize}}\ R_T(h_\theta) \coloneqq \mathop{\mathbb{E}}_{(X,Y)\sim T} [\ell(h_\theta(X), Y)] \ . \tag{2}$$

To make solving (2) possible, we assume that target and client label marginal distributions $T(Y), \{S_i(Y)\}$ are known to the central server. This is much less restrictive than it sounds: (i) Estimating each client label distribution $S_i(Y)$ merely involves computing the proportion of each label in the client sample D_i, (ii) The target client (domain) may have collected label statistics without logging context features X. In our retail example, the label distribution corresponds to the proportion of sales $T(Y = y)$ of each product category y, and many companies store this information without logging customer features X. The central server is given access to all label distributions to facilitate the learning process, but these are *not available to the clients*. Retailers may be hesitant to share their exact sales statistics $T(Y)$ with competitors but

could share this information with a neutral third party (central server) responsible for coordinating the federated learning process. There is a privacy-accuracy trade-off in all FL settings. In our experiments, *we show that substantial performance improvements can be gained at the small privacy cost of sharing the label marginals with the central server.*

As in standard FL, clients $i \neq j$ do not communicate directly with each other directly but interact with the central server through model parameters. While it is technically possible for the server to *infer* each client's label distribution $S_i(Y)$ based on their parameter updates [Ramakrishna and Dán, 2022], doing so would likely be considered a breach of trust in practical applications and sharing would be preferred.

We assume that the distributional shifts between clients and the target are restricted to *label shift*—while the label distributions vary across clients and the target, the class-conditional input distributions are identical.

Assumption 1 (Label shift)**.** *For the client distributions $S_1, ..., S_M$ and the target distribution T,*

$$\forall i, j \in [M] : S_i(X \mid Y) = S_j(X \mid Y) = T(X \mid Y) \, . \tag{3}$$

This setting has been well studied in non-federated learning, see e.g., [Lipton et al., 2018]. In the retail example, label shift means that the proportion of sales across product categories ($S_i(Y)$ and $T(Y)$) varies between different retailers and the target, but that the pattern of customers who purchase items in each category ($S_i(X \mid Y)$ and $T(X \mid Y)$) remain consistent. In other words, although retailers may sell different quantities of products across categories, the characteristics of customers buying a particular product (conditional on the product category) are assumed to be the same. Note that both label shift and *covariate shift* may hold, that is, there are cases where $\forall i : S_i(X \mid Y) = T(X \mid Y)$ *and* $S_i(Y \mid X) = T(Y \mid X)$, but $S_i(X), T(X)$ differ, such as when the labeling function is deterministic.

2.1 Limitations of classical aggregation

When either all clients $\{S_i\}$ or their aggregate $\bar{S}$, see (1), are identical in distribution to the target domain, the empirical risk on aggregated client data is identical in distribution ($\stackrel{d}{=}$) to the empirical risk of a hypothetical data set $D_T = \{(x_{T,j}, y_{T,j})\}_{j=1}^{n_T}$ drawn from the target domain,

$$\hat{R} \coloneqq \sum_{i=1}^{M} \sum_{j=1}^{n_i} \frac{\ell\left(h(x_{i,j}), y_{i,j}\right)}{N} \stackrel{d}{=} \sum_{j=1}^{n_T} \frac{\ell\left(h(x_{T,j}), y_{T,j}\right)}{n_T} =: \hat{R}_T$$

Thus, if each client performs a single gradient descent update, the mean of these, weighted by the client sample sizes, is equal in distribution to a centralized batch update for the target domain, given the previous parameter value. This property justifies the federated stochastic gradient (FedSGD) and federated averaging principles [McMahan et al., 2017], both of which aggregate parameter updates in this way,

$$\theta_{t+1}^{FA} = \sum_{i=1}^{M} \alpha_i^{FA} \theta_{i,t} \quad \text{where} \quad \alpha_i^{FA} = \frac{n_i}{\sum_{j=1}^{M} n_j} \, . \tag{4}$$

Unfortunately, when the target domain T is not the aggregate of clients $\bar{S}$, the aggregate risk gradient $\nabla \hat{R}$ and, therefore, the FedSGD update are no longer unbiased gradients and updates for the risk in the target domain. As we see in Table 1, this can have large effects on model quality.

Our central question is: How can we *aggregate* the parameter updates $\theta_{i,t}$ of the M clients, whose data sets are drawn from distributions $S_1, ..., S_M$, such that the resulting federated learning algorithm minimizes the target risk, R_T?

3 FedPALS: Target-aware adjustment for label shift

Next, we develop a model aggregation strategy for target-aware federated learning. Under Assumption 1 (label shift), the target risk is a weighted sum of class-conditional client risks,

$$\forall i : R_T(h) = \sum_{y=1}^{K} T(y) \mathbb{E}_{S_i}[\ell(h(X), y) \mid Y = y] \, .$$

In centralized learning, this insight is often used to re-weight the training objective in a source domain $S(y)$ by the importance ratio $T(y)/S(y)$ [Lipton et al., 2018, Japkowicz and Stephen, 2002]. *That is not an option here since $T(Y)$ is not revealed to the clients.* For now, assume instead that the target label distribution is *covered* by the convex hull of the set of client label distributions $S = \{S_i(Y)\}_{i=1}^{M}$.

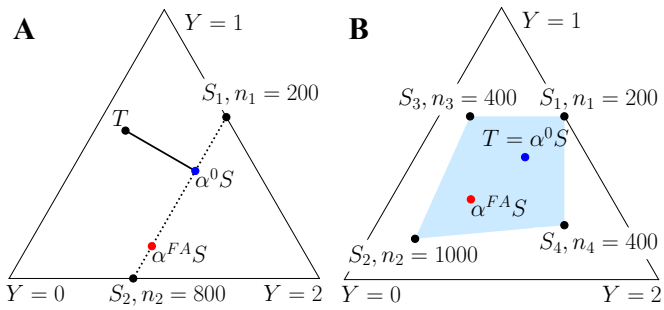


Figure 1: Illustration of the target label marginal T and client marginals $S_1, ..., S_4$ in a ternary classification task, $Y \in \{0, 1, 2\}$. A: there are fewer clients than labels, $M < K$, and $T \notin \mathrm{Conv}(S)$; $\alpha^0 S$ is a projection of T onto $\mathrm{Conv}(S)$. B: $T \in \mathrm{Conv}(S)$ and coincides with $\alpha^0 S$. In both cases, the label marginal $\alpha^{FA} S$ implied by FedAvg is further from the target distribution.

Assumption 2 (Target coverage)**.** *The target label distribution $T(Y)$ is covered by the convex hull of client label distributions $S_1(Y), ..., S_M(Y)$, that is $T \in \mathrm{Conv}(S)$, or*

$$\exists \alpha^c \in \Delta^{M-1} : T(y) = \sum_{i=1}^{M} \alpha_i^c S_i(y) \quad \forall y \in [K] \,. \tag{5}$$

Under label shift, Assumption 2 implies that $T(X, Y) = \sum_{i=1}^{M} \alpha_i^c S_i(X, Y)$, as well. Thus, under Assumptions 1–2, we have for any α^c satisfying (5),

$$R_T(h) = \sum_{y=1}^{K} \left(\sum_{i=1}^{M} \alpha_i^c S_i(y) \right) \mathbb{E}[\ell(h(X), y) \mid Y = y] \tag{6}$$

$$= \sum_{i=1}^{M} \alpha_i^c R_{S_i}(h) \,. \tag{7}$$

Consequently, aggregating client updates with weights α^c will be an unbiased estimate of the update.

Proposition 1 (Unbiased SGD update)**.** *Consider a single round t of federated learning in the batch stochastic gradient setting with learning rate η. Each client $i \in [M]$ is given parameters θ_t by the server, computes their local gradient, and returns the update $\theta_{i,t} = \theta_t - \eta \nabla_\theta \hat{R}_i(h_{\theta_t})$. Let Assumptions 1–2 hold and α^c satisfy* (5)*. Then, the aggregate update $\theta_{t+1} = \sum_{i=1}^{M} \alpha_i^c \theta_{i,t}$ satisfies*

$$\mathbb{E}[\theta_{t+1} \mid \theta_t] = \mathbb{E}[\theta_{t+1}^T \mid \theta_t] \,,$$

where $\theta_{t+1}^T = \theta_t - \eta \nabla_\theta \hat{R}_T(h_{\theta_t})$ is the batch stochastic gradient descent (SGD) update for $\hat{R}_T$ that would be obtained with a sample from the target domain. A proof is in Appendix C.

By Proposition 1, we may compute unbiased parameter updates for the target domain by replacing the aggregation step of FedSGD with aggregation weighted according to α^c. In practice, many federated learning systems, including FedAvg, allow clients several steps of local optimization (e.g., an epoch) before aggregating the parameter updates at the server. Strictly speaking, this is not justified by Proposition 1, but we find in all experiments that aggregating client updates computed over an epoch performs very well, see Section 5.

In applications, the target may not be covered by clients (Assumption 2 may not hold), and α^c may not exist. For example, if the target label marginal $T(y)$ is sparse, only clients with *exactly the same sparsity pattern* as T can be used in a convex combination $\alpha^c S = T$. That is, if we aim to classify images of animals and T contains no tigers, then no clients contributing to the combination can have data containing tigers. Since $\{S_i(Y)\}_{i=1}^{M}, T(Y)$ are known to the server, it is straightforward to verify Assumption 2.

A pragmatic choice when Assumption 2 is violated is to look for the convex combination α^0 that most closely aligns with the target label distribution, and use that for aggregation,

$$\alpha^0 = \underset{\alpha \in \Delta^{M-1}}{\arg\min} \left\| \sum_{i=1}^{M} \alpha_i S_i(Y) - T(Y) \right\|_2^2 \tag{8}$$

We illustrate the label distributions implied by weighting with α^0 and α^{FA} (FedAvg) in Figure 1.

Effective sample size of aggregates. A limitation of aggregating using α^0 as defined in (8) is that, unlike FedAvg, it does not give higher weight to clients with larger sample sizes, which can lead to a higher variance in the model estimate. The variance of importance-weighted estimators can be quantified through the concept of *effective sample size* (ESS) [Kong, 1992], which measures the number of samples needed from the target domain to achieve the same variance as a weighted estimate computed from source-domain samples. ESS is often approximated as $1/(\sum_{i=1}^{m} w_j^2)$ where w are normalized sample weights such that $w_j \geq 0$ and $\sum_{j=1}^{n} w_j = 1$. In federated learning, we can interpret the aggregation step as assigning a total weight α_i to each client i, which has n_i samples. Consequently, each sample $(x_j, y_j) \in D_i$ has the same weight $\tilde{w}_j = \alpha_i / n_i$. The ESS for the aggregate is then given by $1/(\sum_{i=1}^{m}(\sum_{j \in S_i} \tilde{w}_j^2)) = 1/(\sum_{i=1}^{m} n_i \alpha_i^2 / n_i^2) = 1/(\sum_{i=1}^{m} \alpha_i^2 / n_i)$.

In light of the above, we propose a client aggregation step such that the weighted sum of clients' label distributions will a) closely align with the target label distribution, and b) minimize the variance due to weighting using the inverse of the ESS. For a given regularization parameter $\lambda \in [0, \infty)$, we define weights α^λ as the solution to the following problem

$$\alpha^\lambda = \underset{\alpha \in \Delta^{M-1}}{\arg\min} \|T(Y) - \sum_{i=1}^{M} \alpha_i S_i(Y)\|_2^2 + \lambda \sum_i \frac{\alpha_i^2}{n_i} , \tag{9}$$

with aggregate client parameters as $\theta_{t+1}^\lambda = \sum_{i=1}^{M} \alpha_i^\lambda \theta_{i,t}$. We refer to this strategy as Federated learning with Parameter Aggregation for Label Shift (FedPALS).

3.1 FedPALS in the limits

In the FedPALS aggregation scheme ((9)), there exists a trade-off between closely matching the target label distribution and minimizing the variance of the model parameters. This trade-off gives rise to two notable limit cases: $T \in \mathrm{Conv}(S), \lambda \to 0$, and $\lambda \to \infty$. If all source distributions $\{S_i\}_{i=1}^{M}$ are identical and match the target distribution, this corresponds to the classical i.i.d. setting.

Case 1: $\lambda \to \infty \Rightarrow$ Federated averaging In the limit $\lambda \to \infty$, as the regularization parameter λ grows large, FedPALS aggregation approaches FedAvg aggregation.

Proposition 2. *The limit solution α^λ to* (9)*, as $\lambda \to \infty$, is*

$$\lim_{\lambda \to \infty} \alpha_i^\lambda = \frac{n_i}{\sum_{j=1}^{M} n_j} = \alpha_i^{FA} \quad \textit{for} \quad i = 1, \ldots, M \, . \tag{10}$$

The result is proven in Appendix C. By Proposition 2, the FedAvg weights α^{FA} minimize the ESS and coincide with FedPALS weights α^λ in the limit $\lambda \to \infty$. As a rare special case, whenever $T(Y) = \tilde{S} = \sum_{i=1}^{M} \frac{n_i}{N} S_i(Y)$, FedAvg weights $\alpha^{FA} = \alpha^\lambda$ for any value of λ, since both terms attain their mimima at this point. However, this violates the assumption that $T(Y) \neq \tilde{S}(Y)$.

Case 2: Covered target, $T \in \mathrm{Conv}(S)$ Now, consider when the target label distribution is in the convex hull of the source label distributions, $\mathrm{Conv}(S)$. Then, we can find a convex combination α^c of source distributions $S_i(Y)$ that recreate $T(Y)$, that is, $T(Y) = \sum_{i=1}^{M} \alpha_i^c S_i(Y)$. However, when there are more clients than labels, $M > K$, such a *satisfying combination* α^c need not be unique and different combinations may have different effective sample size. Let $A^c = \{\alpha^c \in \Delta^{M-1} : T(Y) = (\alpha^c)^\top S(Y)\}$ denote all satisfying combinations where $S(Y) \in \mathbb{R}^{M \times K}$ is the matrix

of all client label marginals. For a sufficiently small regularization penalty λ, the solution to (9) will be the satisfying combination with largest effective sample size.

$$\lim_{\lambda \to 0} \alpha^\lambda = \arg\min_{\alpha \in A^c} \sum_{i=1}^{M} \frac{\alpha_i^2}{n_i} \, .$$

If there are fewer clients than labels, $M < K$, the set of target distributions for which a satisfying combination exists has measure zero, see Figure 1 (left). Nevertheless, the two cases above allow us to interpolate between being as faithful as possible to the target label distribution ($\lambda \to 0$) and retaining the largest effective sample size ($\lambda \to \infty$), the latter coinciding with FedAvg. Finally, when $T \in \mathrm{Conv}(S)$ and $\lambda \to 0$, Proposition 1 applies also to FedPALS; the aggregation strategy results in an unbiased estimate of the target risk gradient in the SGD setting. However, like the unregularized weights, Proposition 1 does not apply for multiple local client updates.

Case 3: $T \notin \mathrm{Conv}(S)$ If the target distribution does not lie in $\mathrm{Conv}(S)$, see Figure 1 (left), FedPALS projects the target to the "closest point" in $\mathrm{Conv}(S)$ if $\lambda = 0$, and to a tradeoff between this projection and the FedAvg aggregation if $\lambda > 0$. We have a discussion on how to choose λ in the second and third cases in Appendix A.1.

Sparse clients and targets In problems with a large number of labels, $K \gg 1$, it is common that any individual domain (clients or target) supports only a subset of the labels. For example, in the IWildCam benchmark, not every wildlife camera captures images of all animal species. When the target $T(Y)$ is *sparse*, meaning $T(y) = 0$ for certain labels y, it becomes easier to find a good match $(\alpha^\lambda)^\top S(Y) \approx T(Y)$ if the client label distributions are also sparse. Achieving a perfect match, i.e., $T \in \mathrm{Conv}(S)$, requires that (i) the clients collectively cover all labels in the target, and (ii) each client contains only labels that are present in the target. If this is also beneficial for learning, it would suggest that the client-presence of labels that are not present in the target would *harm* the aggregated model. We study the implications of sparsity of label distributions empirically in Section 5.

4 Related work

Efforts to mitigate the effects of distributional shifts in federated learning can be broadly categorized into client-side and server-side approaches. Client-side methods use techniques such as clustering clients with similar data distributions and training separate models for each cluster [Ghosh et al., 2020, Sattler et al., 2020, Vardhan et al., 2024], and meta-learning to enable models to quickly adapt to new data distributions with minimal updates [Chen et al., 2018, Jiang et al., 2019, Fallah et al., 2020]. Other notable strategies include regularization techniques that penalize large deviations in client updates to ensure stable convergence [Li et al., 2020b, 2021] and recent work on optimizing for flatter minima to enhance model robustness [Qu et al., 2022, Caldarola et al., 2022]. Server-side methods focus on improving model aggregation or applying post-aggregation adjustments. These include optimizing aggregation weights [Reddi et al., 2021], learning adaptive weights [Li et al., 2023], iterative moving averages to refine the global model [Zhou et al., 2023], and promoting gradient diversity during updates [Zeng et al., 2023]. Both categories of work overlook shifts in the target distribution, leaving this area unexplored.

Another related area is personalized federated learning, which focuses on fine-tuning models to optimize performance on each client's specific local data [Collins et al., 2022, Boroujeni et al., 2024]. This setting differs fundamentally from our work, which focuses on improving generalization to new target clients without any training data available for fine-tuning. Label distribution shifts have also been explored with methods such as logit calibration [Zhang et al., 2022, Xu et al., 2023], novel loss functions [Wang et al., 2021], feature augmentation [Xia et al., 2023], gradient reweighting [Xiao et al., 2023], and contrastive learning [Wu et al., 2023]. However, like methods aimed at mitigating the effects of general shifts, these do not address the challenge of aligning models with an unseen target distribution, as required in our setting.

Generalization under domain shift in federated learning remains underdeveloped [Bai et al., 2024]. The work most similar to ours is that of agnostic federated learning (AFL) [Mohri et al., 2019], which aims to learn a model that performs robustly across all possible target distributions within the convex hull of client distributions. One notable approach is tailored for medical image segmentation, where clients share data in the frequency domain to achieve better generalization across domains [Liu et al., 2021]. However, this technique requires data sharing, making it unsuitable for privacy-sensitive applications like ours. A different line of work focuses on addressing covariate shift in federated learning through importance weighting [Ramezani-Kebrya et al., 2023]. Although effective, this method requires sending samples from the test distribution to the server, which violates our privacy constraints.

Table 1: Comparison of mean accuracy and standard deviation ($\pm$) across different algorithms. The reported values are over 8 independent random seeds for the CIFAR-10 and Fashion-MNIST tasks, and 3 for PACS. C indicates the number of labels per client and β the Dirichlet concentration parameter. M is the number of clients. The *Oracle* method refers to a FedAvg model trained on clients whose distributions are identical to the target.

Data set	Label split	M	FedPALS	FedAvg	FedProx	SCAFFOLD	AFL	FedRS	Oracle
Fashion-MNIST	$C = 3$	10	$\mathbf{92.4 \pm 2.1}$	67.1 ± 22.0	66.9 ± 20.8	69.5 ± 19.3	78.9 ± 14.7	85.3 ± 13.5	97.6 ± 2.1
	$C = 2$		$\mathbf{80.6 \pm 23.7}$	53.9 ± 36.2	52.9 ± 35.7	54.9 ± 36.8	78.6 ± 20.0	63.14 ± 20.2	97.5 ± 4.0
CIFAR-10	$C = 3$		$\mathbf{65.6 \pm 10.1}$	44.0 ± 8.4	43.5 ± 7.2	43.3 ± 7.4	53.2 ± 0.9	44.0 ± 8.0	85.5 ± 5.0
	$C = 2$	10	$\mathbf{72.8 \pm 17.4}$	46.7 ± 15.8	47.7 ± 15.6	46.7 ± 14.9	54.7 ± 0.1	49.4 ± 9.5	89.2 ± 3.9
	$\beta = 0.1$		$\mathbf{62.6 \pm 17.9}$	40.8 ± 9.2	41.9 ± 9.7	43.5 ± 10.5	53.4 ± 11.5	57.1 ± 11.2	79.2 ± 3.7
PACS	$C = 6$	3	$\mathbf{86.0 \pm 2.9}$	73.4 ± 1.6	75.3 ± 1.3	73.9 ± 0.3	74.5 ± 0.9	76.1 ± 1.6	90.5 ± 0.3

5 Experiments

We perform a series of experiments on benchmark data sets to evaluate FedPALS in comparison with baseline federated learning algorithms. The experiments aim to demonstrate the value of the central server knowing the label distributions of the client and target domains when these differ substantially. Additionally, we seek to understand how the parameter λ, controlling the trade-off between bias and variance in the FedPALS aggregation scheme, impacts the results. Finally, we investigate how the benefits of FedPALS are affected by the sparsity of label distributions and by the distance $d(T, S) \coloneqq \min_{\alpha \in \Delta^{M-1}} \|T(Y) - \alpha^\top S(Y)\|_2^2$ from the target to the convex hull of clients.

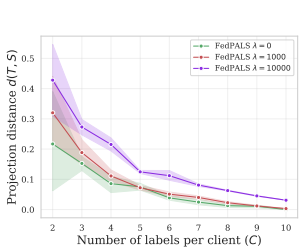
Experimental setup While numerous benchmarks exist for federated learning [Caldas et al., 2018, Chen et al., 2022] and domain generalization [Gulrajani and Lopez-Paz, 2020, Koh et al., 2021], respectively, until recently none have addressed tasks that combine both settings. To fill this gap, Bai et al. [2024] introduced a benchmark specifically designed for federated domain generalization (DG), evaluating methods across diverse datasets with varying levels of client heterogeneity. In our experiments, we use the PACS Li et al. [2017] and iWildCAM data sets from the Bai et al. [2024] benchmark to model realistic label shifts between the client and target distributions. We modify the PACS dataset to consist of three clients, each missing a label that is present in the other two. Additionally, one client is reduced to one-tenth the size of the others, and the target distribution is made sparse in the same label as that of the smaller client. For further details see Appendix A.

Furthermore, we construct two additional tasks by introducing label shift to standard image classification data sets, Fashion-MNIST [Xiao et al., 2017] and CIFAR-10 [Krizhevsky, 2009]. We apply two label shift sampling strategies: sparsity sampling and Dirichlet sampling. Sparsity sampling involves randomly removing a subset of labels from clients and the target domain, following the data set partitioning technique first introduced in McMahan et al. [2017]. Each client is assigned C random labels, with an equal number of samples for each label and no overlap among clients.

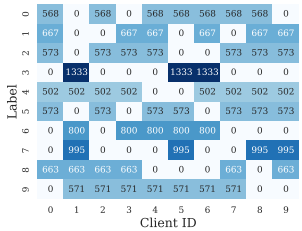
Dirichlet sampling simulates realistic non-i.i.d. label distributions by, for each client i, drawing a sample $p_i \sim \text{Dirichlet}_K(\beta)$, where $p_i(k)$ represents the proportion of samples in client i that have label $k \in [K]$. We use a symmetric concentration parameter $\beta > 0$ which controls the sparsity of the client distributions. See Appendix A.2.

While prior works have focused on inter-client distribution shifts assuming that client and target domains are equally distributed, *we apply these sampling strategies also to the target set*, thereby introducing label shift between the client and target data. Figures 2b & 5b (latter in appendix) illustrate an example with $C = 6$ for sparsity sampling and Dirichlet sampling with $\beta = 0.1$, where the last client (Client 9) is chosen as the target. In addition, we investigate the effect of $T(Y) \notin \text{Conv}(S)$ in a task described in B.4.

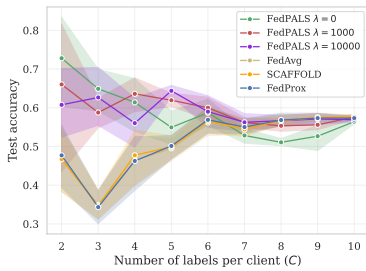
Baseline algorithms and model architectures Alongside FedAvg, we use SCAFFOLD, FedProx, AFL and FedRS [Karimireddy et al., 2020, Li et al., 2020b, Mohri et al., 2019, Li and Zhan, 2021] as baselines. The first two chosen due to their prominence in the literature for handling non-iid data, and AFL which is similar in concept to FedPALS and aims to optimize for an unseen domain. We also include FedRS, designed specifically to address label distribution skew. For the synthetic experiment in Section B.4, we use a logistic regression model. For CIFAR-10 and Fashion-MNIST, we use small, two-layer convolutional networks, while for PACS and iWildCAM, we use a ResNet-50 pre-trained on ImageNet. Early stopping, model hyperparameters, and λ in FedPALS are tuned using a validation set that reflects the target distribution in the synthetic experiment, CIFAR-10, Fashion-MNIST, and PACS. This tuning process consistently resulted in setting the number of local epochs to $E = 1$ across all experiments. For iWildCAM, we adopt the hyperparameters reported by Bai et al. [2024] and select λ using the same validation set used in their work. We report the mean test accuracy and



(a) Projection distance between target and client convex hull, varying C.

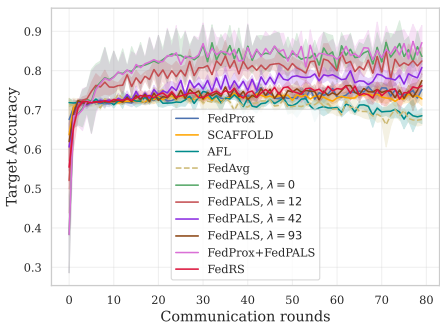


(b) Marginal label distributions of clients and target with $C = 6$ labels per client.

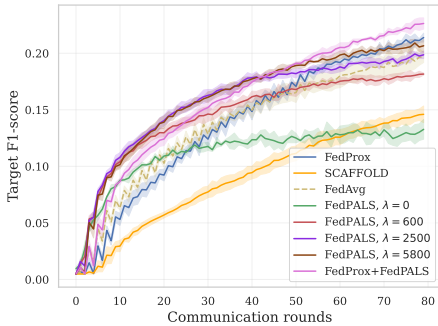


(c) Accuracy vs labels per client, C.

Figure 2: Results on CIFAR-10 with sparsity sampling, varying the number of labels per clients C across 10 clients. Clients with IDs 0–8 are used in training, and Client 9 is the target client. The task is more difficult for small C, when fewer clients share labels, and the projection distance is larger.



(a) PACS, $M = 3$.



(b) iWildCam, $M = 100$.

Figure 3: Target accuracy/F1-score during training of FedPALS compared to baselines on PACS (a) and iWildCam (b), averaged over 3 random seeds. M is the number of training clients. Non-zero λ-values chosen to correspond to an ESS of 25%, 50% and 75%.

standard deviation for each method over 3 independent random seeds for PACS and iWildCam and 8 seeds for the smaller Fashion-MNIST and CIFAR-10, to ensure robust evaluation.

5.1 Experimental results on benchmark tasks

We present results for three tasks with selected skews in Table 1 and explore detailed results below. Across these tasks, FedPALS consistently outperforms or matches the best-performing baseline. For PACS, Fashion-MNIST and CIFAR-10, we include results for an *Oracle* FedAvg model, which is trained on clients whose distributions are identical to the target distribution, eliminating any client-target distribution shift (see Appendix A for details). A FedPALS-*Oracle* would be equivalent since there is no label shift. The *Oracle*, which has perfect alignment between client and target distributions, achieves superior performance, underscoring the challenge posed by distribution shifts in real-world scenarios where such alignment is absent.

CIFAR-10/Fashion-MNIST. Figure 2c shows the results for the CIFAR-10 data set, where we vary the label sparsity across clients. In the standard i.i.d. setting, where all labels are present in both the training and target clients ($C = 10$), all methods perform comparably. However, as label sparsity increases and fewer labels are available in client data sets (i.e., as C decreases), we observe a performance degradation in standard baselines. In contrast, our proposed method, FedPALS, leverages optimized aggregation to achieve a lower target risk, resulting in improved test accuracy under these challenging conditions. Similar trends are observed for Fashion-MNIST, as shown in Figure 6 in Appendix B. Furthermore, the results in the highly non-i.i.d. cases ($C = 2, 3$ and $\beta = 0.1$) are summarized in Table 1. Additional experiments in Appendix B

examine how the algorithms perform with varying numbers of local epochs (up to 40) and clients (up to 100).

PACS. As shown in Figure 3a, being faithful to the target distribution is crucial for improved performance. Lower values of λ generally correspond to better performance. Notably, FedAvg struggles in this setting because it systematically underweights the client with the distribution most similar to the target, leading to suboptimal model performance. In fact, this even causes performance to degrade over time. Interestingly, the baselines also face challenges on this task: both FedProx, FedRS and SCAFFOLD perform similarly to FedPALS when $\lambda = 93$. However, FedPALS demonstrates significant improvements over these methods, highlighting the effectiveness of our aggregation scheme in enhancing performance. We also see that FedPALS + FedProx performs comparably to just using FedPALS in this case, although it does have higher variance. Additionally, in Table 1, we present the models selected based on the source validation set, where FedPALS outperforms all other methods. For comprehensive results, including all FedPALS models and baseline comparisons, refer to Table 4 in Appendix B.

iWildCam. The test performance across communication rounds is shown in Figure 3b. Initially, FedPALS widens the performance gap compared to FedAvg, but as training progresses, this gain diminishes. While FedPALS quickly reaches a strong performing model, it eventually plateaus. The rate of convergence and level of performance reached appears to be influenced by the choice of λ, with lower values of λ leading to faster plateaus at lower levels compared to larger ones. This suggests that more uniform client weights and a larger effective sample size are preferable in this task. Given the iWildCam dataset's significant class imbalance – with many classes having few samples – de-emphasizing certain clients can degrade performance. We also note that our assumption of label shift need not hold in this experiment, as the cameras are in different locations, potentially leading to variations in the conditional distribution $p(X \mid Y)$. The performance of the models selected using the source validation set is shown in Table 3 in Appendix B. There we see that FedPALS performs comparably to FedAvg and FedProx while outperforming SCAFFOLD. Unlike in other tasks, where FedProx performs comparably or worse than FedPALS, we see FedProx achieve the highest F1-score on this task. Therefore, we conduct an additional experiment where we use both FedProx and FedPALS together, as they are not mutually exclusive. This results in the best performing model, see Figure 3b. Due to memory issues with the implementation FedRS was not able to run for this experiment and is omitted. AFL fails to learn in this task and is thus also omitted, although results are shown in Table 3 in Appendix B. Finally, as an illustration of the impact of increasing λ, we provide the weights of the clients in this experiment alongside the FedAvg weights in 4 in Appendix B. We note that as λ increases, the weights increasingly align with those of FedAvg while retaining weight on the clients whose label distributions most resemble that of the target.

6 Discussion

We have explored *target-aware federated learning under label shift*, a scenario where client data distributions differ from a target domain with a known label distribution, but no target samples are available. We demonstrated that traditional approaches, such as federated averaging (FedAvg), which assume identical distributions between the client aggregate and the target, fail to adapt effectively in this context due to biased aggregation of client updates. To address this, we proposed FedPALS, a novel aggregation strategy that optimally combines client updates to align with the target distribution, ensuring that the aggregated model minimizes target risk. Empirically, across diverse tasks, we showed that under label shift, FedPALS significantly outperforms standard methods like FedAvg, FedProx, FedRS and SCAFFOLD, as well as AFL. Specifically, when the target label distribution lies within the convex hull of the client distributions, FedPALS finds the solution with the largest effective sample size, leading to a model that is most faithful to the target distribution. More generally, FedPALS balances the trade-off between matching the target label distribution and minimizing variance in the model updates. Our experiments further highlight that FedPALS excels in challenging scenarios where label sparsity and client heterogeneity hinder the performance of conventional federated learning methods.

One of the limitations of FedPALS is label shift–the assumption that label-conditional input distributions are equal in all clients and the target. We observed empirically that selecting the trade-off parameter λ is crucial for optimal performance in tasks such as iWildCam, where this assumption may not fully hold. Future work should aim to detect and overcome violations of this assumption.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization. *arXiv:1907.02893*, 2020. arXiv: 1907.02893.

Ruqi Bai, Saurabh Bagchi, and David I Inouye. Benchmarking algorithms for federated domain generalization. In *The Twelfth International Conference on Learning Representations*, 2024.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24, 2011.

Mahrokh Ghoddousi Boroujeni, Andreas Krause, and Giancarlo Ferrari-Trecate. Personalized federated learning of probabilistic models: A pac-bayesian approach. *ArXiv*, abs/2401.08351, 2024.

David Byrd and Antigoni Polychroniadou. Differentially private secure multi-party computation for federated learning in financial applications. In *Proceedings of the First ACM International Conference on AI in Finance*, 2020.

Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *Computer Vision – ECCV 2022*, 2022. ISBN 978-3-031-20050-2.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Daoyuan Chen, Dawei Gao, Weirui Kuang, Yaliang Li, and Bolin Ding. pfl-bench: A comprehensive benchmark for personalized federated learning. *Advances in Neural Information Processing Systems*, 35:9344–9360, 2022.

Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv:1802.07876*, 2018.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Fedavg with fine tuning: Local updates lead to representation learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in neural information processing systems*, 33:3557–3568, 2020.

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

Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020.

Agrin Hilmkil, Sebastian Callh, Matteo Barbieri, Leon René Sütfeld, Edvin Listo Zec, and Olof Mogren. Scaling federated learning for fine-tuning of large language models. In *International Conference on Applications of Natural Language to Information Systems*, pages 15–23, 2021.

Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5): 429–449, 2002.

Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 2020.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021.

Augustine Kong. A note on importance sampling using standardized weights. *University of Chicago, Dept. of Statistics, Tech. Rep*, 348:14, 1992.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5543–5551, 2017. doi: 10.1109/ICCV.2017.591.

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 965–978. IEEE, 2022.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, 2020b.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pages 6357–6368. PMLR, 2021.

Xin-Chun Li and De-Chuan Zhan. Fedrs: Federated learning with restricted softmax for label distribution non-iid data. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021.

Zexi Li, Tao Lin, Xinyi Shang, and Chao Wu. Revisiting weighted aggregation in federated learning with neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pages 3122–3130. PMLR, 2018.

Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, 2017.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4615–4625. PMLR, 09–15 Jun 2019.

Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In *International conference on machine learning*, pages 18250–18280. PMLR, 2022.

Raksha Ramakrishna and György Dán. Inferring class-label distribution in federated learning. In *Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security*, pages 45–56, 2022.

Ali Ramezani-Kebrya, Fanghui Liu, Thomas Pethick, Grigorios Chrysos, and Volkan Cevher. Federated learning under covariate shifts with generalization guarantees. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Sashank Reddi, Zachary Burr Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Brendan McMahan, editors. *Adaptive Federated Optimization*, 2021.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32 (8):3710–3722, 2020.

Micah J Sheller, Brandon Edwards, G Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R Colen, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):12598, 2020.

Harsh Vardhan, Avishek Ghosh, and Arya Mazumdar. An improved federated clustering algorithm with model-based clustering. *Transactions on Machine Learning Research*, 2024.

Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. Addressing class imbalance in federated learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(11), 2021. doi: 10.1609/aaai.v35i11.17219.

Blake E Woodworth, Kumar Kshitij Patel, and Nati Srebro. Minibatch vs local sgd for heterogeneous distributed learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 6281–6292, 2020.

Nannan Wu, Li Yu, Xin Yang, Kwang-Ting Cheng, and Zengqiang Yan. Fediic: Towards robust federated learning for class-imbalanced medical image classification. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, pages 692–702, 2023. ISBN 978-3-031-43895-0.

Tong Xia, Abhirup Ghosh, and Cecilia Mascolo. Flea: Improving federated learning on scarce and label-skewed data via privacy-preserving feature augmentation. *ArXiv*, abs/2312.02327, 2023.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Zikai Xiao, Zihan Chen, Songshan Liu, Hualiang Wang, Yang Feng, Jinxiang Hao, Joey Tianyi Zhou, Jian Wu, Howard H. Yang, and Zuo-Qiang Liu. Fed-grab: Federated long-tailed learning with self-adjusting gradient balancer. *ArXiv*, abs/2310.07587, 2023.

Jian Xu, Mei Yang, Wenbo Ding, and Shao-Lun Huang. Stabilizing and improving federated learning with non-iid data and client dropout. *ArXiv*, abs/2303.06314, 2023.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR, 2019.

Dun Zeng, Zenglin Xu, Yu Pan, Qifan Wang, and Xiaoying Tang. Tackling hybrid heterogeneity on federated optimization via gradient diversity maximization. *arXiv preprint arXiv:2310.02702*, 2023.

Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. Federated learning with label distribution skew via logits calibration. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162. PMLR, 2022.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Tailin Zhou, Zehong Lin, Jinchao Zhang, and Danny H. K. Tsang. Understanding and improving model averaging in federated learning on heterogeneous data. *IEEE Transactions on Mobile Computing*, 2023.

A Experimental details

Here we provide additional details about the experimental setup for the different tasks. The code will be made available upon acceptance.

A.1 Choice of hyperparameter λ

A salient question in Cases 2 & 3, as discussed in Section 3.1, is how to choose the strength of the regularization, λ. A larger value will generally favor influence from more clients, provided that they have sufficiently many samples. When $T \notin \mathrm{Conv}(S)$, the convex combination closest to T could have weight on a single vertex. This will likely hurt the generalizability of the resulting classifier. In experiments, we compare values of λ that yield different effective sample sizes, such as 10%, 25%, 50% or 75% of the original sample size, N. We can find these using binary search by solving (9) and calculate the ESS. One could select λ heuristically based on the the ESS, or treat λ as a hyperparameter and select it using a validation set. Although this would entail training and evaluating several models which can be seen as a limitation. We elect to choose a small set of λ values based on the ESS heuristic and train models for these. Then we use a validation set to select the best performing model. This highlights the usefulness of the ESS as a heuristic. If it is unclear which values to pick, one could elect for a simple strategy of taking the ESS of $\lambda = 0$ and 100% and taking l equidistributed values in between the two extremes, for some small integer l.

A.2 Sampling strategies

Sparsity sampling entails randomly removing a subset of labels from clients and the target domain following the data set partitioning technique introduced in McMahan et al. [2017]. Each client is assigned C random labels, with an equal number of samples for each label and no overlap among clients. Sparsity sampling has been extensively used in subsequent studies [Geyer et al., 2017, Li et al., 2020a, 2022].

Dirichlet sampling simulates non-i.i.d. label distributions by, for each client i, drawing a sample $p_i \sim \mathrm{Dirichlet}_K(\beta)$, where $p_i(k)$ represents the proportion of samples in client i that have label $k \in [K]$. The concentration parameter $\beta > 0$ controls the sparsity of the client distributions. In dirichlet sampling, using a smaller β results in more heterogeneous client data sets, while a larger value approximates an i.i.d. setting. This widely-used method for sampling clients was first introduced by Yurochkin et al. [2019].

A.3 Oracle construction

The *Oracle* method serves as a benchmark to illustrate the performance upper bound when there is no distribution shift between the clients and the target. To construct this *Oracle*, we assume that the client label distributions are identical to the target label distribution, effectively eliminating the label shift that exists in real-world scenarios.

In practice, this means that for each dataset, the client data is drawn directly from the same distribution as the target. The aggregation process in the *Oracle* method uses FedAvg, as no adjustments for label shift are needed. Since the client and target distributions are aligned, FedPALS would behave equivalently to FedAvg under this setting, as there is no need for reweighting the client updates.

This method allows us to assess the maximum possible performance that could be achieved if the distributional differences between clients and the target did not exist. By comparing the *Oracle* results to those of our proposed method and other baselines, we can highlight the impact of label shift on model performance and validate the improvements brought by FedPALS.

A.4 Perturbation of target marginal $\mathcal{T}$

In an experiment we perturb the given target label marginals, $\mathcal{T}$, to evaluate the performance impact of noise in the estimate. We do this by generating gaussian noise, ϵ, and then we add the noise to the label marginal to create a new target $\mathcal{T}_p$. We modulate the size of the noise with a parameter δ and only add the positive noise values.

$$\mathcal{T}_p = \mathcal{T} + \delta \max(\epsilon, 0)$$

This is then normalised and used as the new target label marginal. This perturbation was done on the PACS experiment with $\delta \in [10^{-3}, 10^{-2}, 5 \times 10^{-1}]$ and repeated for three seeds. The results are given in Table 2 where we see that the performance decreases with increasing noise.

δ	Accuracy
10^{-3}	88.8
10^{-2}	85.2
5×10^{-1}	81.4

Table 2: Results of perturbing $\mathcal{T}$ with varying noise levels δ.

A.5 Synthetic task

We randomly sampled three means $\mu_1 = [6, 4.6], \mu_2 = [1.2, -1.6]$, and $\mu_3 = [4.6, -5.4]$ for each label cluster, respectively.

A.6 PACS

In this task we use the official source and target splits which are given in the work by Bai et al. [2024]. We construct the task such that the training data is randomly assigned among three clients, then we remove the samples of one label from each of the clients. This is chosen to be labels '0', '1' and '2'. Then the client that is missing the label '2' is reduced so that it is 10% the amount of the original size. For the target we modify the given one by removing the samples with label '2', thereby making it more similar to the smaller client. To more accurately reflect the target distribution we modify the source domain validation set to also lack the samples with label '2'. This is reasonable since we assume that we have access to the target label distribution.

We pick four values of λ, [0,12,42,93], which approximately correspond to an ESS of $15\%, 25\%, 50\%$ and 75% respectively. We use the same hyperparameters during training as Bai et al. [2024] report using in their paper. Furthermore, we use the cross entropy loss in this task.

A.7 iWildCam

We perform this experiment using the methodology described in Bai et al. [2024] with the heterogeneity set to the maximum setting, i.e., $\lambda = 0$ in their construction.[1] We use the same hyperparameters which is used for FedAvg in the same work to train FedPALS. We perform 80 rounds of training and, we then select the best performing model based on held out validation performance and report the mean and standard deviation over three random seeds. This can be seen in Table 3. We pick four values of λ, [0,600,2500,5800], which approximately correspond to an ESS of $8\%, 25\%, 50\%$ and 75% respectively. We use the cross entropy loss in this task.

Due to FedProx performing comparably to FedPALS on this task, in contrast with other experiments, we also perform an experiment where we do both FedProx and FedPALS. This is easily done as FedProx is a client side method while FedPALS is a weighting method applied at the server. This results in the best performing model.

We use the same hyperparameters during training as Bai et al. [2024] report using in their paper. However, we set the amount of communication rounds to 80.

B Additional empirical results

Figure 4 illustrates the aggregation weights of clients in the iWildCam experiment for λ corresponding to different effective sample sizes.

We report the performance of the models selected using the held out validation set in Table 3 and Table 4 for the iWildCam and PACS experiments respectively.

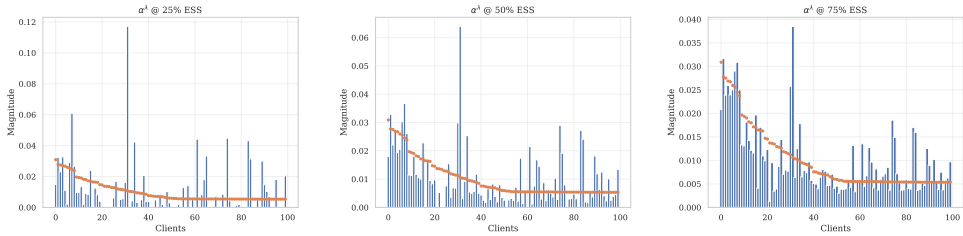
B.1 Results on CIFAR-10 with Dirichlet sampling

Figure 5 shows the results for the CIFAR-10 experiment with Dirichlet sampling of client and target label distributions.

B.2 Training dynamics for Fashion-MNIST

Figure 7 shows the training dynamics for Fashion-MNIST and CIFAR-10 with different label marginal mechanisms.

[1] Note that this is not the same λ used in the trade-off in FedPALS.



(a) (b) (c)

Figure 4: An illustration of the aggregation weights of clients in the iWildCam experiment using FedPALS for different ESS. The clients are sorted by amount of samples in descending order. The magnitude of the weights produced by federated averaging is shown as dots. Note that with increasing the ESS, the magnitudes more closely resemble that of federated averaging.

Table 3: Results on iWildCam with 100 clients, standard deviation reported over 3 random seeds.

Algorithm	F1 (macro)
FedPALS, $\lambda = 0$	0.13 ± 0.00
FedPALS, $\lambda = 600$	0.18 ± 0.00
FedPALS, $\lambda = 2500$	0.19 ± 0.00
FedPALS, $\lambda = 5800$	0.21 ± 0.00
FedProx+FedPALS, $\lambda = 5800$	0.23 ± 0.00
FedAvg	0.20 ± 0.01
FedProx	0.21 ± 0.00
SCAFFOLD	0.15 ± 0.01
AFL	0.005 ± 0.0

B.3 Local epochs and number of clients

In Figure 8c we show results for varying number of clients for each method. For the cases with number of clients 50 and 100, we use the standard sampling method of federated learning where a fraction of 0.1 clients are sampled in each communication round. In this case, we optimize α^λ for the participating clients in each communication round. Interestingly, we observe that while FedAvg performs significantly worse than FedPALS on a target client under label shift, it outperforms both FedProx and SCAFFOLD when the number of local epochs is high ($E = 40$), as shown in Figure 8b.

B.4 Synthetic experiment: effect of projection distance on test error

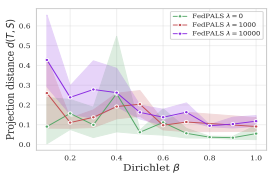
When the target distribution $T(Y)$ is not covered by the clients, FedPALS finds aggregation weights corresponding to a regularized projection of T onto $\mathrm{Conv}(S)$. To study the impact of this, we designed a controlled experiment where the distance of the projection is varied. We create a classification task with three classes, $\mathcal{Y} = \{0, 1, 2\}$, and define $p(X \mid Y = y)$ for each label $y \in \mathcal{Y}$ by a unit-variance Gaussian distribution $\mathcal{N}(\mu_y, I)$, with randomly sampled means $\mu_y \in \mathbb{R}^2$. We simulate two clients with label distributions $S_1(Y) = [0.5, 0.5, 0.0]^\top$ and $S_2(Y) = [0.5, 0.0, 0.5]^\top$, and $n_1 = 40, n_2 = 18$ samples, respectively. Thus, FedAvg gives larger weight to Client 1. We define a target label distribution $T(Y)$ parameterized by $\delta \in [0, 1]$ which controls the projection distance $d(T, S)$ between $T(Y)$ and $\mathrm{Conv}(S)$,

$$T_\delta(Y) \coloneqq (1 - \delta)T_{\text{proj}}(Y) + \delta T_{\text{ext}}(Y) \,,$$

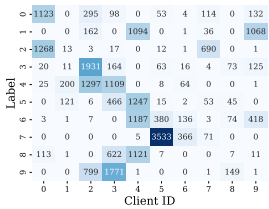
with $T_{\text{ext}}(Y) = [0, 0.5, 0.5]^\top \notin \mathrm{Conv}(S(Y))$ and $T_{\text{proj}}(Y) = [0.5, 0.25, 0.25]^\top \in \mathrm{Conv}(S(Y))$. By varying δ, we control the projection distance $d(T, S)$ between each T_δ and $\mathrm{Conv}(S)$ from solving (8), allowing us to study its effect on model performance.

Table 4: Results on PACS with 3 clients with mean and standard deviation reported over 3 random seeds.

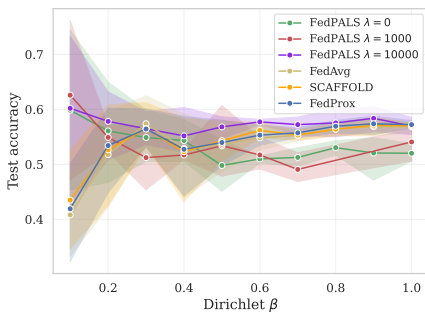
Algorithm	Accuracy
FedPALS, $\lambda = 0$	86.0 ± 2.9
FedPALS, $\lambda = 12$	84.3 ± 2.5
FedPALS, $\lambda = 42$	81.7 ± 1.2
FedPALS, $\lambda = 93$	77.3 ± 1.6
FedProx+FedPALS, $\lambda = 0$	87.2 ± 4.1
FedAvg	73.4 ± 1.6
FedProx	75.3 ± 1.3
SCAFFOLD	73.9 ± 0.3
AFL	74.5 ± 0.9



(a) Projection distance between target label marginal and client convex hull



(b) Marginal label distributions of clients and target with $\beta = 0.1$.



(c) Accuracy under Dirichlet sampling.

Figure 5: Results on CIFAR-10 with Dirichlet sampling across 10 clients, varying concentration parameter β. Clients with IDs 0–8 are clients present during training, and client with ID 9 is the target client.

We evaluate the global model on a test set with $n_{\text{test}} = 2000$ samples drawn from the target distribution $T(Y)$ for each value of δ and record the target accuracy for FedPALS and FedAvg. Figure 9 illustrates the relationship between the target accuracy and the projection distance $d(T, S)$ due to varying δ. When $d(S, T) = 0$ (i.e., $T(Y) \in \text{Conv}(S)$), the target accuracy is highest, indicating that our method successfully matches the target distribution. As $d(S, T)$ increases (i.e., T moves further away from $\text{Conv}(S)$), the task becomes harder and accuracy declines. For all values, FedPALS performs better than FedAvg. For more details on the synthetic experiment, see Appendix A.

C Proofs

C.1 FedPALS updates

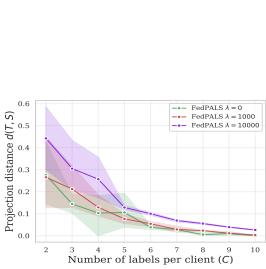
Proposition 1 (Repeated) (Unbiased SGD update)**.** Consider a single round t of federated learning in the batch stochastic gradient setting with learning rate η. Each client $i \in [M]$ is given parameters θ_t by the server, computes their local gradient, and returns the update $\theta_{i,t} = \theta_t - \eta \nabla_\theta \hat{R}_i(h_{\theta_t})$. Let weights α^c satisfy $T(X, Y) = \sum_{i=1}^M \alpha_i^c S_i(X, Y)$. Then, the aggregate update $\theta_{t+1} = \sum_{i=1}^M \alpha_i^c \theta_{i,t}$ satisfies

$$\mathbb{E}[\theta_{t+1} \mid \theta_t] = \mathbb{E}[\theta_{t+1}^T \mid \theta_t] \,,$$

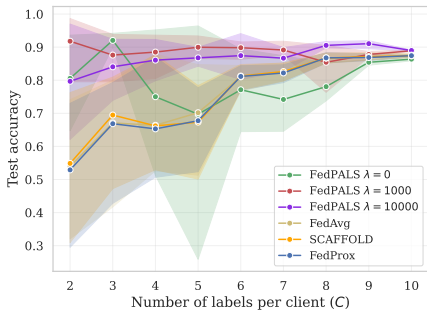
where θ_{t+1}^T is the batch stochastic gradient update for $\hat{R}_T$ that would be obtained with a sample from the target domain.

Proof.

$$\theta_{t+1} = \sum_{i=1}^M \alpha_i^c \theta_{i,t} = \sum_{i=1}^M \theta_i^c (\theta_t - \eta \nabla \hat{R}_i(h_{\theta_t})) = \theta_t - \eta \sum_{i=1}^M \alpha_i \nabla \hat{R}_i(h_{\theta_t}) \tag{11}$$

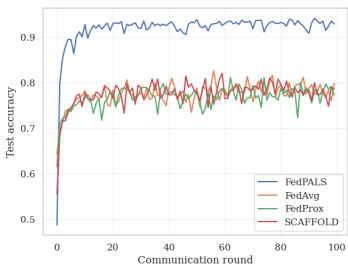


(a) Projection distance between target label marginal and client convex hull

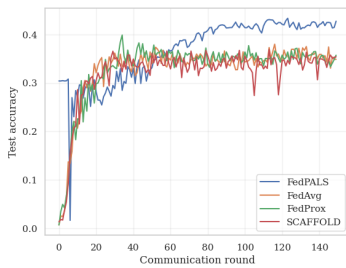


(b) Accuracy under label sampling.

Figure 6: Results on Fashion-MNIST with label sampling across 10 clients, varying parameter C. Clients with IDs 0–8 are clients present during training, and client with ID 9 is the target client.



(a) Fashion-MNIST with $C = 6$.



(b) CIFAR-10 with Dirichlet $\beta = 0.2$.

Figure 7: Test accuracy during training rounds.

$$\mathbb{E}[\theta_{t+1} \mid \theta_t] = \theta_t - \eta \cdot \mathbb{E}\left[\sum_{i=1}^{M} \alpha_i \nabla \hat{R}_i(h_{\theta_t}) \mid \theta_t\right] \tag{12}$$

$$= \theta_t - \eta \cdot \sum_{x,y} \mathbb{E}\left[\sum_{i=1}^{M} \hat{S}_i(x,y)\alpha_i\right] \nabla L(y, h_{\theta_t}(x)) \tag{13}$$

$$= \theta_t - \eta \cdot \sum_{x,y} T(x,y) \nabla L(y, h_{\theta_t}(x)) \tag{14}$$

$$= \theta_t - \eta \cdot \mathbb{E}\left[\sum_{x,y} \hat{T}(x,y)\right] \nabla L(y, h_{\theta_t}(x)) = \mathbb{E}[\theta_{t+1}^T \mid \theta_t] \,. \tag{15}$$

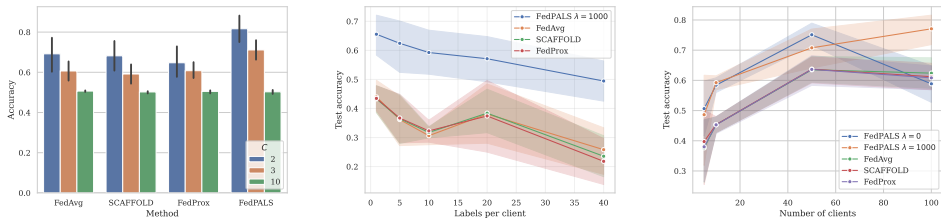
□

C.2 FedPALS in the limits

As $\lambda \to \infty$, because the first term in (9) is bounded, the problem shares solution with

$$\min_{\alpha_1,\ldots,\alpha_M} \sum_i \frac{\alpha_i^2}{n_i} \quad \text{s.t.} \quad \sum_i \alpha_i = 1, \quad \forall i : \alpha_i \geq 0 \,. \tag{16}$$

Moreover, we have the following result.



(a) (b) (c)

Figure 8: Comparison of CIFAR-10 results with different clients and settings. (a) 100 clients for $C = 2, 3, 10$, $\lambda = 1000$. (b) 10 clients and number of labels $C = 3$. We plot test accuracy as a function of number of local epochs E. The total number of communication rounds T are set such that $T = E/150$, where 150 is the number of rounds used for $E = 1$. (c) Test accuracy as a function of number of clients, with $C = 3$.

Proposition 3. *The optimization problem*

$$\min_{\alpha} \sum_i \frac{\alpha_i^2}{n_i} \quad s.t \quad \sum_i \alpha_i = 1 \quad \alpha_i \geq 0 \, \forall \, i \, ,$$

has the optimal solution $\alpha_i^* = \frac{n_i}{\sum_i n_i}$ *where* $i \in [1, m]$

Proof. From the constrained optimization problem we form a Lagrangian formulation

$$\mathcal{L}(\alpha, \mu, \tau) = \sum_i \frac{\alpha_i^2}{n_i} + \mu \underbrace{(1 - \sum_i \alpha_i)}_{h(\alpha)} + \tau \underbrace{-\alpha}_{g(\alpha)}$$

We then use the KKT-theorem to find the optimal solution to the problem.

$$\nabla_\alpha \mathcal{L}(\alpha^*) = 0 \implies \forall i : \; 2\frac{\alpha_i^*}{n_i} - \mu - \tau = 0 \, . \tag{17}$$

In other words, the following ratio is a constant,

$$\forall i \quad \frac{\alpha_i^*}{n_i} = c$$

for some constant c. We have the additional conditions of primal feasibility, i.e.

$$h(\alpha^*) = 0$$
$$g(\alpha^*) \leq 0$$

From the first constraint, we have $\sum_{i=1}^M \alpha_i^* = 1$, and thus,

$$\sum_{i=1}^M \alpha_i^* = c \sum_{i=1}^M n_i = 1$$

which implies that $c = 1/\sum_{i=1}^M n_i$ and thus

$$\forall i : \alpha_i^* = \frac{n_i}{\sum_{i=1}^M n_i} \, .$$

□

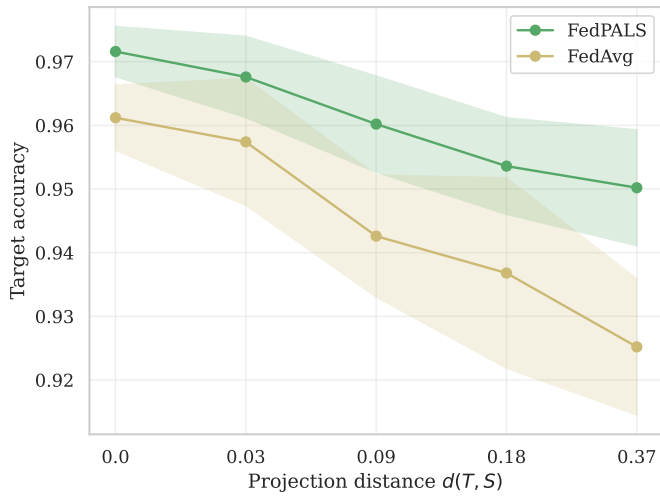


Figure 9: Synthetic experiment. Accuracy of the global model as a function of the projection distance $d(T, S)$ between the target distribution $T(Y)$ and client label distributions $\mathrm{Conv}(S(Y))$. Means and standard deviations reported over 5 independent runs.

Paper IV

Federated Learning with Heterogeneous and Private Label Sets

A. Breitholtz, E. Listo Zec, F. D. Johansson

Submitted, under review

Federated Learning with Heterogeneous and Private Label Sets

Adam Breitholtz*
Chalmers University of Technology
& University of Gothenburg
adambre@chalmers.se

Edvin Listo Zec*
RISE Research Institutes of Sweden
KTH Royal Institute of Technology
edvin.listo.zec@ri.se

Fredrik D. Johansson†
Chalmers University of Technology
& University of Gothenburg
fredrik.johansson@chalmers.se

August 27, 2025

Abstract

Although common in real-world applications, heterogeneous client label sets are rarely investigated in federated learning (FL). Furthermore, in the cases they are, clients are assumed to be willing to share their entire label sets with other clients. Federated learning with *private* label sets, shared only with the central server, adds further constraints on learning algorithms and is, in general, a more difficult problem to solve. In this work, we study the effects of label set heterogeneity on model performance, comparing the public and private label settings—when the union of label sets in the federation is known to clients and when it is not. We apply classical methods for the classifier combination problem to FL using centralized tuning, adapt common FL methods to the private label set setting, and discuss the justification of both approaches under practical assumptions. Our experiments show that reducing the number of labels available to each client harms the performance of all methods substantially. Centralized tuning of client models for representational alignment can help remedy this, but often at the cost of higher variance. Throughout, our proposed adaptations of standard FL methods perform well, showing similar performance in the private label setting as the standard methods achieve in the public setting. This shows that clients can enjoy increased privacy at little cost to model accuracy.

1 Introduction

Federated learning (FL) enables collaborative model training across distributed clients without centralizing their private data [16]. While promising, the effectiveness of FL is often challenged by statistical heterogeneity, where the data distributions vary significantly across clients. A particularly common and disruptive form of this is *label shift* [15], where the distribution of class labels differs from one client to another. Even more challenging is *label set heterogeneity* [7], where clients' local label sets are disjoint subsets of a global label set.

In applications where access to instances of particular classes holds a competitive advantage, clients may be unwilling to reveal the identities of the classes they observe. Consider, for instance, a consortium of competing pharmaceutical companies wanting to train a model to predict which drug compounds individual patients will have adverse reactions to [3, 12]. Each company has proprietary data on its own set of compounds and reactions, some of which are used by other clients in the federation, and some which are not. They are willing to collaborate to build a more powerful, generalizable model, but would never share the full list of compounds they classify with other clients, as this would reveal information about their R&D pipeline. Instead, clients must communicate model updates with the central server pertaining *only* to their *private label set*.

Learning with heterogenous client label distributions has been tackled by several methods, including model distillation [5], contrastive learning [7], and latent space alignment based on class names [23]. However, the

*Equal contribution.
†https://www.healthyai.se/

literature is more sparse when considering label sets which are not identical across clients, although there are some works which consider it [15]. Moreover, classical methods for federated learning, adapted for label shift or otherwise, can not be applied directly with private label sets. In this setting, the models learned on the clients will necessarily be incompatible for regular aggregation since they will make classifiers for different label sets. Therefore, there is a need to develop methods to deal with this complication to ensure that learning is successful.

In this work, we investigate the effects of client label sparsity and heterogeneity on federated learning performance when client label sets are shared by the whole federation (public) and when they are unknown to other clients (private). We define the private label set problem in Section 3 and adapt popular FL model aggregation strategies for it in Section 4.1. We show that such methods are not well justified when client representations are poorly aligned and propose an alternative method based on the literature on classifier combination in Section 4.2, tuning the central classifier for heterogeneous client representations using an unlabeled dataset at the server. We conduct experiments in image classification on two data sets and show empirically how the sparsity and privacy of label sets affect performance (Section 5). We find that both the private and public label settings are more challenging when clients hold smaller and more diverse subsets of the global label set. Finally, in the private label setting, our proposed adaptations achieve comparable performance to methods for public labels, implying that clients can retain more privacy at little to no cost in accuracy.

2 Related Work

The question of how to combine classifiers trained on disparate label sets have been studied in the binary setting previously in the context of centralized (non-federated) learning. See for example [20] for a comparison of several methods which focuses on combining binary classifiers based on the classifier probabilites.

A similar line of work is the Open-set literature, where the label sets in the clients may be incomplete and the sets may not match between clients. In [2] they learn distribution estimators in the clients to approximate the overall label distribution using uncertainty of the global model.

[23] deals with clients which do not share the same label set. They propose having the clients share the names of the labels and aligning the embedding of these names across the representations of clients. However, this requires sharing the names of the labels which may be undesirable, especially in a private label set setting. In a similar vein, [15] restricts the softmax to account for the missing labels in the clients. The algorithm proposed hinges on knowing which labels a client has to account for which precludes its use in the private label set setting.

Moreover, our setting is also related to shift in client label distributions. There are many works which aim to handle cases when there is a imbalance between client label distributions. This can be done by using regularization which penalizes large deviations in client updates [13, 14] or using control variates to steer the learning [10]. Other techniques include clustering clients with similar data distributions and training separate models for each cluster [4, 17, 19] and meta-learning to enable models to quickly adapt to new data distributions with minimal updates [1, 9]. However, these techniques are not adapted to the private label set setting.

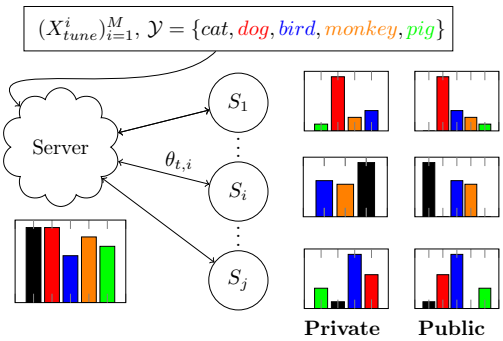
Another related field is that of semi supervised federated learning where some works make use of unlabeled data in FL settings [8]. In these works the unlabeled data is usually available on the client side, which differs from our setting.

3 Problem setup

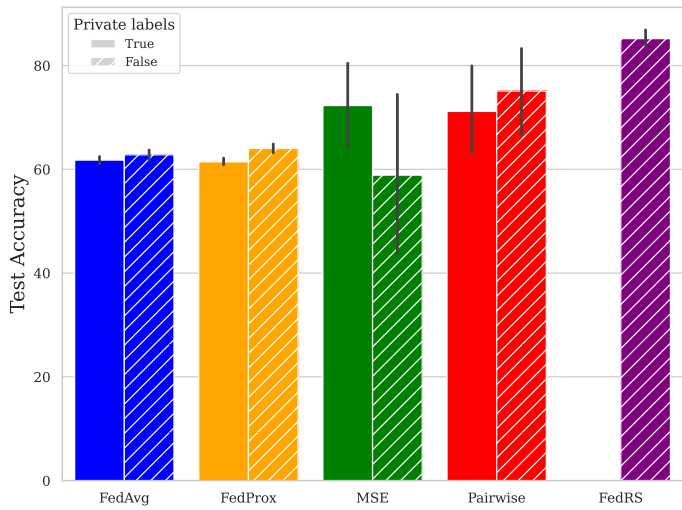
We consider the problem of federated learning (FL) of a single model h, trained to classify points $\boldsymbol{x} \in \mathcal{X}$ into classes $y \in \mathcal{Y} = \{1, ..., |\mathcal{Y}|\}$ given inputs $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$. A central server coordinates learning on m clients, indexed by $k = 1, ..., m$, each observing labeled data from an unknown distribution $p_k(\boldsymbol{X}, Y)$. Central to our setting is that clients *do not* have all labels in their data, i.e., clients k are exposed only to a subset $\mathcal{Y}_k \subset \mathcal{Y}$ of labels.

Our goal is to learn a probabilistic classifier $h : \mathcal{X} \to \Delta_{\mathcal{Y}}$, where $\Delta_{\mathcal{Y}}$ is the simplex over *all* classes, that minimizes the expected prediction risk with respect to a loss function $L : \Delta_{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}$,

$$\underset{h}{\text{minimize}}\ R(h) \quad R(h) \coloneqq \mathbb{E}[L(h(\boldsymbol{X}), Y)]\ . \tag{1}$$



(a)



(b)

Figure 1: **a):** A schematic view of the two settings which we consider in our work. The private setting where the clients are unaware of the full label set and the public setting where this is known. **b):** Results on CIFAR10 where each client has 5 labels available in their respective dataset. The tuning methods with MSE and Pairwise losses perform the best in the private setting. Errorbars represent a 95% confidence interval. Note, FedRS is not applicable with private labels as it needs knowledge of the full label set.

Here, the expectation $\mathbb{E}$ is defined over an unknown distribution $p(\boldsymbol{X}, Y)$, assumed to be a convex combination of clients $p(\boldsymbol{X}, Y) = \sum_{k=1}^{m} w_k p_k(\boldsymbol{X}, Y)$ where $w \in \Delta_m$ assigns a weight to each client. In the classical FL setting, it is assumed implicitly that the weight is proportional to the number of samples n_k held by the client, $w_k = n_k / (\sum_{k'=1}^{m} n_{k'})$, and the risk is computed over a distribution that matches the client aggregate, as exploited in the aggregation scheme of federated averaging (FedAvg) [16]. However, our methods can be adapted to targeted federated learning where p cannot be expressed by a convex combination of clients [22]. The model $h(x) = \psi(\phi(x))$ typically consists of feature extractor ϕ and a classifier ψ, typically parameterized by a neural network with parameters θ_ϕ and a linear-softmax classifier with parameters θ_ψ, respectively. That is, $h_\theta(x) = \sigma(\theta_\psi^\top \phi(x))$ where $\theta = (\theta_\phi, \theta_\psi)$.

Algorithm 1: FedAvg with private label sets

Data: Client label sets $\{\mathcal{Y}_k\}$ and reverse indices $\{I_k\}$
Result: Classifier $h(x) = \sigma(\theta_\psi^\top \phi(x))$
Initialize central parameters $\theta^0 = (\theta_\phi^0, \theta_\psi^0)$
for *each round* $t = 0, ..., T-1$ **do**
  for *each client* $k = 1, ..., m$ **do**
    Distribute $(\theta_\phi^t, \theta_\psi^t[\mathcal{Y}_k])$ to client k
    Receive client update $(\theta_{\phi,k}^t, \theta_{\psi,k}^t)$
  end
  $\theta_\phi^{t+1} = \sum_{k=1}^m \theta_{\phi,k}^t \frac{n_k}{n}$ where $n = \sum_{k=1}^m n_k$
  for *each label* $y \in \mathcal{Y}$ **do**
    $\theta_\psi^{t+1}(y) = \sum_{k:y\in\mathcal{Y}_k} \theta_{\psi,k}^t(I_k(y)) \frac{n_k}{n'_y}$ where $n'_y = \sum_{k:y\in\mathcal{Y}_k} n_k$
  end
end
Return classifier with parameters $\theta = (\theta_\phi^T, \theta_\psi^T)$

We will consider two settings for the label set (illustrated in Figure 1):

- **Public labels**: All clients know the full global label set $\mathcal{Y}$. This is the standard FL setting but with emphasis on label set heterogeneity.
- **Private labels**: Each client know only their local label set $\mathcal{Y}_k$ and all communication with the central server is restricted to this set. That is, classifier parameters $\theta_\psi(y)$ for labels $y \notin \mathcal{Y}_k$ are not shared with client k.

In both cases, the central server knows the full label set and the label sets of all clients to allow for tailoring communication to clients with heterogeneous and private label sets. We expand on methods for handle the private label case next.

4 Methods for heterogeneous label sets

When the label set is public, we can simply use existing FL methods to learn our classifiers. The issue of label set heterogeneity still remains, and aligning the models to combat any effects of misaligned representations may be warranted. However, in the case where the label set is private, some further modifications have to be made. We detail this and a method of tuning models for alignment in the following sections.

4.1 Model averaging with private label sets

The main challenge addressed in this work is *private label set heterogeneity*: each client k observes labels from a subset $\mathcal{Y}_k \subseteq \mathcal{Y}$ and are *unaware* of other labels $\mathcal{Y} \setminus \mathcal{Y}_k$. Without loss of generality, we assume that for all clients k, every label in $\mathcal{Y}_k$ is observed with positive probability, $\forall y \in \mathcal{Y}_k : p_k(Y = y) > 0$, and other labels are unobserved, $\forall y \notin \mathcal{Y}_k : p_k(Y = y) = 0$.

In the private setting, standard methods (e.g., FedAvg [16], FedProx [13], FedRS [15]) cannot be applied without modification as clients do not have access to the full set of parameters θ_ψ of the shared classifier ψ. Moreover, the server can only receive updates from client k to parameters concerning their subset of labels $\mathcal{Y}_k$. To overcome this obstacle, we propose a simple modification to common model averaging strategies that handles the lack of a full classifier by using the restricted classifiers and reweighting them.

Client-side modification In each round t, each client k is sent the full set of current encoder parameters θ_ϕ^t and the subset of current classifier parameters corresponding to their label set, $\theta_\psi^t[\mathcal{Y}_k] \coloneqq [\theta_\psi^t(y) : y \in \mathcal{Y}_k]^\top \in \mathbb{R}^{|\mathcal{Y}_k|}$. Clients then proceed with local updates as normal.

Server-side modification In each round, t, the server receives parameter updates $(\theta^t_{\phi,k}, \theta^t_{\psi,k})$ from each client k and averages the classifier parameters for each label y based on the clients which have the label in their label set, weighted according to their sample size (see Algorithm 1). Encoder parameter updates $\theta^t_{\phi,k}$ are averaged as normal.

Surprisingly, this simple method is well-justified under the softmax classifier model, provided that the classifier $h(x) = \sigma(\theta_\psi^\top \phi(x))$ is well-specified and clients' conditional label distributions (mechanisms) are what we call *subset consistent*.

Assumption 1 (Subset-consistent labeling mechanisms)**.** *The labeling mechanisms of clients $k = 1, .., n$, each with a distributions $p_k(\boldsymbol{X}, Y)$ on a label set $\mathcal{Y}_k$, are* subset-consistent *if the target label distribution $p(\boldsymbol{X}, Y)$ satisfies*

$$\forall k, \boldsymbol{x} : p_k(Y = y \mid Y \in \mathcal{Y}_k, \boldsymbol{X} = \boldsymbol{x}) = p(Y = y \mid Y \in \mathcal{Y}_k, \boldsymbol{X} = \boldsymbol{x}) \ .$$

Now, suppose that $h_\theta(x)$ is well-specified for the true labeling function $p(Y \mid X)$ given an optimal encoder ϕ, that is, there are parameters θ_ψ such that

$$p(Y = y \mid X = x) = \frac{e^{-\theta_\psi(y)^\top \phi(x)}}{\sum_{y'} e^{-\theta_\psi(y')^\top \phi(x)}} = \sigma(\theta_\psi^\top \phi(x))_y \ .$$

Then, the subset-conditional outcome can be parameterized as a softmax classifiers with parameters $\theta_\psi[\mathcal{Y}_k]$, the subset of θ_ψ restricted to $\mathcal{Y}_k$,

$$\begin{aligned} p(Y = y \mid X = x, Y \in \mathcal{Y}_k) &= \frac{p(Y = y \mid X = x)}{\sum_{y' \in \mathcal{Y}_k} p(Y = y' \mid X = x)} = \frac{e^{-\theta_y^\top \phi(x)}}{\sum_{y' \in \mathcal{Y}_k} e^{-\theta_{y'}^\top \phi(x)}} \\ &= \sigma(\theta_\psi[\mathcal{Y}_k]^\top \phi(x))_y, \end{aligned} \tag{2}$$

since the normalization terms over the full label set cancel. As a result, the optimal model in this circumstance has the same parameters $\theta(y)$ both centrally and in all clients k with $y \in \mathcal{Y}_k$. Consequently, given an optimal encoder $\phi(x)$ in the sense above, *any* convex combination of unbiased estimates $\hat{\theta}_{\psi,k}$ of client-optimal parameters is unbiased for the server-optimal parameters θ. Client weighting based on sample size (as in Algorithm 1) achieves the largest effective sample size (smallest variance) [22].

Remark. In the deterministic case, where $\forall \boldsymbol{x}, \exists y^* : p(Y = y^* \mid \boldsymbol{X} = \boldsymbol{x}) = 1$, Assumption 1 corresponds to the often-used *covariate shift* assumption [18] since the event $Y \in \mathcal{Y}_k$ does not alter the distribution of Y for a given on $\boldsymbol{x}$. In this case, aggregating *perfect* client models $h_k(y \mid x)$ is trivial for a given x, since all of them will return 1 for the correct label. In general, for stochastic labels $p_k(y \mid Y \in \mathcal{Y}_k, \boldsymbol{x}) \neq p_l(y \mid Y \in \mathcal{Y}_l, \boldsymbol{x})$ for $\mathcal{Y}_k \neq \mathcal{Y}_l$. In either case, Assumption 1 allows both marginal distributions $p_k(\boldsymbol{X})$ and $p_k(Y)$ to vary with k.

Based on the simple modifications above, we can also adapt the FedProx [13] algorithm and other centralized model-averaging strategies. For FedProx, we simply omit a comparison of the final layers in the regularization term on the client as their sizes do not match.

The approach detailed in this section is by itself a viable method and will produce a classifier for all classes. However, when representations are not optimal for all clients at once, or when there isn't a single classifier that is optimal in all clients, the justification from (2) fails, and simply averaging client parameters. may not be the best strategy. We explore an alternative strategy next.

4.2 Representation alignment by central tuning

The fundamental problem of federated learning is the aggregation of multiple client models into a single central model that is beneficial to the whole federation. The classical approach of parameter averaging, and its adaptation to private label sets above, is specific to a few model classes (e.g., neural networks) and poorly justified when the averaged representation is suboptimal for some clients. Stepping back, the aggregation problem may be viewed as a special case of classifier combination or couplings [20, 6]. Classifier combination methods were developed to combine several *binary* classifiers, e.g., support vector machines, on different pairs of labels into a single multi-class classifier. Today, this technique is rare as multi-class classifiers are trained routinely using neural networks with softmax outputs or (ensembles of) decision trees. However, in federated learning with

heterogenous and private label sets, we face the same problem again since no client nor the server has access to labeled data from all classes.

Traditionally, classifier combination operates on the classifier functions themselves not on their parameters. In our case, the classifiers are estimates of the conditional label probability $p_k(Y \mid \boldsymbol{X})$ specific to each client k and their label sets $\mathcal{Y}_k$. It is appropriate to ask whether there exists a perfect combination of perfect client classifiers, one that yields minimal error on the target distribution $p(\boldsymbol{X}, Y)$. To understand this, we draw inspiration from the binary-to-multi-class problem of classifier combination [6] and note an important distinction to our setting: usually, classifier combination applies to multiple classifiers trained on different subsets of the same data, or at least on data from the same distribution. This implies a structure between the probability distributions that the classifiers aim to fit. For example, with $\mathcal{Y} = \{0, 1, 2\}$, a binary classifier can be used to distinguish classes 0 and 1 by training on samples $(\boldsymbol{x}, y)$ labeled with $y \in \{0, 1\}$. By design, $p(Y = 1 \mid Y \in \{0, 1\}, \boldsymbol{X} = \boldsymbol{x}) = p(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})/p(Y \in \{0, 1\} \mid \boldsymbol{X} = \boldsymbol{x})$.

In general, the clients in federated learning may have completely unrelated label distributions. However, if we suppose again that Assumption 1 holds, a perfect combination of perfect classifiers may be found.

Proposition 1 (Perfect classifier combination)**.** *Let Assumption 1 hold for a set of clients $k = 1, ..., m$ such that clients jointly cover all labels, $\cup_{k=1}^{m} \mathcal{Y}_k = \mathcal{Y}$. Then, the perfect central classifier $p(Y = y \mid \boldsymbol{X} = \boldsymbol{x})$ can be aggregated from perfect client classifiers $\{p_k(Y = y \mid \boldsymbol{X} = \boldsymbol{x})\}_{k=1}^{m}$.*

Proof. By Assumption 1, for all clients k, inputs $\boldsymbol{x} \in \mathcal{X}$, and outputs $y \in \mathcal{Y}_k$,

$$p_k(y \mid Y \in \mathcal{Y}_k, \boldsymbol{x}) = \frac{p(y \mid \boldsymbol{x})}{p(Y \in \mathcal{Y}_k \mid \boldsymbol{x})} = \frac{p(y \mid \boldsymbol{x})}{\sum_{y' \in \mathcal{Y}_k} p(y' \mid \boldsymbol{x})} \ . \tag{3}$$

In other words, $\forall \boldsymbol{x}, y, \exists k : p(y \mid \boldsymbol{x}) = c(\boldsymbol{x}) p_k(y \mid Y \in \mathcal{Y}_k, \boldsymbol{x})$ with $c(\boldsymbol{x})$ a normalizing constant. □

The result for softmax classifiers in (2) is a special case of this result.

In practice, of course, we cannot expect to have perfect models of each client to combine—especially not *during* federated learning. However, Proposition 1 gives direction for what a good aggregated model should satisfy. Consider an estimated client model $h_k(y \mid \boldsymbol{x}) \approx p_k(y \mid Y \in \mathcal{Y}_k, \boldsymbol{x})$ and a good central model $h(y \mid \boldsymbol{x}) \approx p(y \mid \boldsymbol{x})$. By (3), it should hold that,

$$\forall k \in [m], y, y' \in \mathcal{Y}_k : h_k(y \mid \boldsymbol{x}) h(y' \mid \boldsymbol{x}) \approx h_k(y' \mid \boldsymbol{x}) h(y \mid \boldsymbol{x}) \ .$$

This is a generalization of the argument in [20] to multi-class subset classifiers. We may use this to construct an aggregation criterion for the central model h, given a set of client models $\{h_k(y \mid \mathcal{Y}_k, \boldsymbol{x})\}_{k=1}^{m}$, first for a fixed input $\boldsymbol{x}$,

$$\underset{h_{\boldsymbol{x}} \in \Delta_{\mathcal{Y}}}{\text{minimize}} \sum_{k=1}^{m} w_k \sum_{\substack{y, y' \in \mathcal{Y}_k \\ y \neq y'}} \left(h_k(y \mid \boldsymbol{x}) h_{\boldsymbol{x}}(y') - h_k(y' \mid \boldsymbol{x}) h_{\boldsymbol{x}}(y) \right)^2 \ . \tag{4}$$

If all client models are perfect, the minimizer of (4) is a perfect central model at $\boldsymbol{x}$ under the conditions of Proposition 1. For high-dimensional or continuous $\boldsymbol{x}$, it is not feasible to fit a separate central classifier to each possible input. Instead, we may use function approximation by fitting a classifier $h(y \mid \boldsymbol{X})$ from a class $\mathcal{H} \subset \{h : \mathcal{X} \to \Delta_{\mathcal{Y}}\}$ to minimize the expected error over $p(\boldsymbol{X})$.

$$\underset{h \in \mathcal{H}}{\text{minimize}} \sum_{k=1}^{m} w_k \sum_{\substack{y, y' \in \mathcal{Y}_k \\ y \neq y'}} \mathbb{E}_{\boldsymbol{X}} \left[\left(h_k(y \mid \boldsymbol{X}) h(y' \mid \boldsymbol{X}) - h_k(y' \mid \boldsymbol{X}) h(y \mid \boldsymbol{X}) \right)^2 \right] \tag{5}$$

We call this the *pairwise* tuning loss and will use this as one of our objectives when combining classifiers centrally. In practice, the marginal distribution $p(\boldsymbol{X})$ is unknown and the expectation is intractable to compute, so we must solve (5) with respect to the empirical expectation $\hat{\mathbb{E}}[\boldsymbol{X}]$ over a sample of data. Consequently, to use this method, we require that the central server has access to an *unlabeled* data set of points $\boldsymbol{x}_1, ..., \boldsymbol{x}_m$ drawn from $p(\boldsymbol{X})$. Since access to tuning data is not required by methods based on parameter averaging, we must bear that in mind when comparing the empirical performance of the two approaches.

For additional comparison, we also consider tuning-based classifier combination using the direct *MSE* loss used in [20],

$$\underset{h \in \mathcal{H}}{\text{minimize}} \sum_{k=1}^{m} w_k \sum_{y \in \mathcal{Y}_k} \mathbb{E}_{\boldsymbol{X}} \left[\left(h_k(y \mid \boldsymbol{X}) - h(y \mid \boldsymbol{X}) \right)^2 \right] . \tag{6}$$

In summary, we solve one of the two optimization problems above at each update to tune the classifier to be more aligned with the predictions of the client models. The tuned classifier is sent back to the clients and training proceeds as normal.

5 Experiments

We use the well-known datasets CIFAR-10 [11] and Fashion-MNIST [21] for constructing our experiments. We perform ablations where we vary the number of labels that a client has access to from 2-10. This entails choosing a random set of labels for each client which they then get distributed from the dataset equally. This means that, absent further intervention, the clients will not have an identical amount of labeled examples across the ablation points. To control for this, we perform a subsampling step where we subsample the client dataset randomly to consistently have 2000 samples in each client. When evaluating the impact of tuning on an unlabeled dataset centrally (Section 4.2), we use an unlabeled dataset with 5000 samples for CIFAR10 and 6000 for FashionMNIST. We use the standard test set splits for both datasets, both have 10000 samples.

In the public label set setting, we use FedAvg, FedProx [13] and FedRS [15] as baselines. In the private setting, we adapt FedAvg and FedProx to compare this approach to the central tuning (see Sections 4.1–4.2 for further details). As FedRS depends on knowledge of the full label set on the clients, we cannot use this method in the private setting.

When performing central tuning, we train the server classifier for 3 epochs using one of two loss functions (Pairwise or MSE) in equations (5) and (6), respectively. The model aggregation then follows that of FedAvg (or our adaptation of FedAvg in the private setting). We aggregate the results for different labels per client over 10 independent random seeds and the error bars denote a 95% bootstrapped confidence interval over these splits. For the ablation over epochs per client, we aggregate over 3 random seeds. Further details, including the choice of hyperparameters for each method, can be found in Appendix A.

5.1 Experimental Results

We present the detailed results of our experiments below. For each method, we show the test accuracy of the model snapshot that achieves the highest validation accuracy during a run and then aggregate this across several seeds. More results, including an ablation over client epochs, can be found in Appendix C.

CIFAR10: We present the results varying the amount of labels per client in Figure 2 with the specific case of 5 labels per client being shown in Figure 1b. We clearly see performance decreasing with decreasing number of labels. This effect is not due to decreasing sample size as that is fixed in the experiments. In the public label setting we see that FedRS performs the best while the tuning approach with the pairwise loss performs better than FedAvg and FedProx. Tuning with MSE loss seems to struggle here while, in the private setting, it performs the best. In the private setting, we can see that the tuning approach is superior to the adapted methods, although their variance is higher. Moreover, the MSE loss performs slightly better than the pairwise loss. Interestingly, the pairwise tuning seems to outperform FedAvg and FedProx in the public setting suggesting that the tuning of the representation can be of use in this setting also. This is likely because representational (mis)alignment can be an issue for federated learning whether labels are private or public. It is noteworthy that the adaptation of FedAvg and FedProx seem to exhibit a surprising robustness against the challenges of the private label sets, since they aggregate models from clients with disparate label sets. However, as the clients have an identical amount of labels, each of the feature extractors are trained to output the same label amounts which could help explain the robustness.

Fashion-MNIST: As we can see in Figure 3, the tuning methods do not outperform the adapted methods in the private setting on Fashion-MNIST. However, their large variance suggests that with more careful training they might perform at least equivalently. This may be due to the task being simple and an alignment of classifiers is unnecessary. We show results for 3 labels per client in Figure 4, where we see that the methods perform similarly in the private setting.

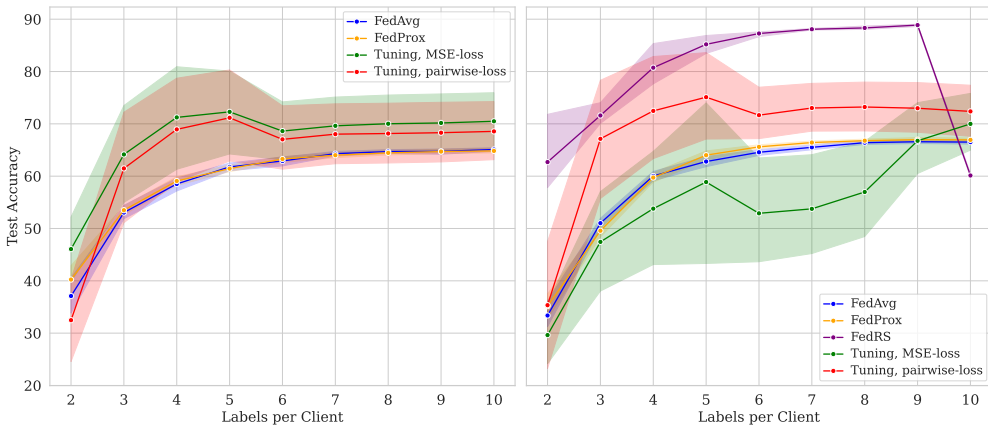


Figure 2: The performance of the methods in both the private (left) and public (right) settings on CIFAR10. Note that in the regular setting the pairwise loss performs better than FedAvg and the MSE loss while in the private setting the relationship is reversed.

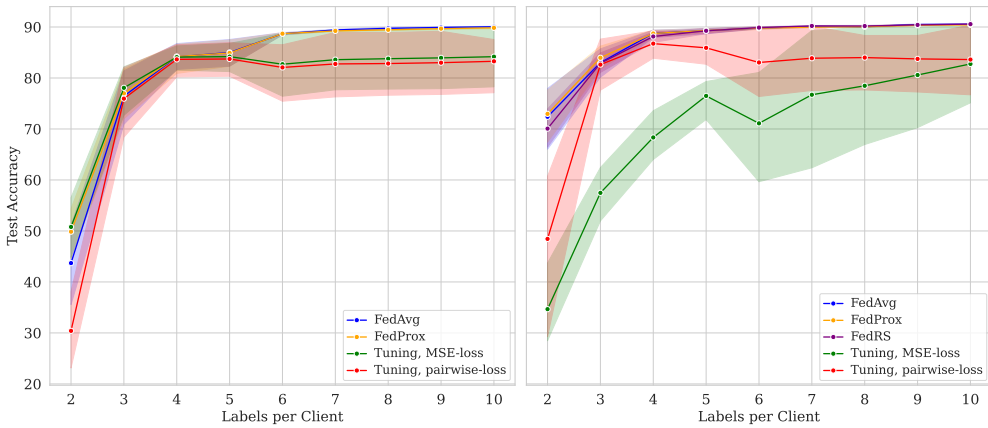


Figure 3: The performance of the methods in both the private and public settings on FashionMNIST. We note that the tuning approaches do not outperform the adapted methods in the private setting.

The relationship that the pairwise loss performs better than the MSE loss in the public setting and worse in the private setting holds true here also. This could be due to the pairwise loss having cases where the loss is large due to lack of labels in clients. See Appendix B for a discussion of this issue. We see a robustness of FedAvg and Fedprox to the private label sets here as well. This indicates that the adapted methods are a pragmatic choice that also works well in this restricted setting.

In the public setting, we see that the tuning methods underperform the other baselines, with FedAvg, FedProx, and FedRS all performing about equally well. This suggests that there is either limited misalignment between client representations or that it does not adversely affect performance. Instead, the central tuning seems to interfere with the successful learning during federation in the public setting, possibly due to increased variance in the central model. We can observe this variance in the figures over communication rounds in Appendix C.1

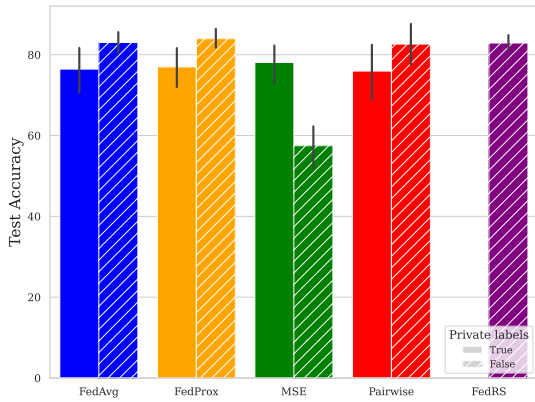


Figure 4: The results for the private and public settings for 3 labels per client on the FashionMNIST dataset.

6 Discussion

This work investigated the impact of label set heterogeneity on Federated Learning, with a specific focus on the challenging and practical *private label set* setting where clients are unaware of the global label space. Our experiments reveal several key insights into the behavior of both standard and adapted FL algorithms under these conditions.

A primary finding is the surprising robustness of our adaptation of FedAvg to private label sets. We hypothesize that even when individual clients train on a small subset of labels, the shared feature extractor learns a common, semantically rich representation space. The simple aggregation strategy for the classifier weights (Algorithm 1), which combines knowledge on a per-class basis, proves to be a powerful and efficient method for stitching together these partial views into a coherent whole. This establishes adapted FedAvg as a formidable baseline, suggesting that complex alignment mechanisms may not always be necessary if there is sufficient label overlap across the client population.

Further, the tuning approaches do seem to work well for the private setting in some cases while performing worse than adapted methods in others. We also observe that the tuning yields better performance when the sparsity is more extreme. This may be due to the alignment problem being harder with an increasing amount of labels. Notably, we observed a performance reversal between the two tuning losses. In the public setting, the pairwise loss was superior, whereas in the more challenging private setting, the MSE loss performed better. We attribute this phenomenon to a critical vulnerability in the pairwise loss, detailed in Appendix B. When a label is globally absent from all participating clients in a round (a scenario far more likely with fewer labels per client) the pairwise loss generates problematic gradients by comparing against a class for which no client has information. The MSE loss, by directly comparing the global model's predictions to each client's predictions on their known classes, appears more resilient to this issue. It provides a more stable, albeit less constrained, learning signal in the face of extreme sparsity.

A key limitation of the tuning methods are the fact that solving the optimization problem for each communication round could become difficult computationally as the amount of labels, and the number of clients, increases. Also, the existence of unlabeled data is an additional burden to bear that may be impractical in application. The adapted methods do not share these limitations and could be an alternative if tuning methods are computationally infeasible or if there does not exist an unlabeled dataset centrally.

In future work, more realistic datasets could be considered which naturally exhibit label set heterogeneity. In addition, some other FL methods could perhaps be adapted to the private label set setting. Moreover, there could be further consideration of and comparison with other tuning losses.

References

[1] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv:1802.07876*, 2018.

[2] Zhipeng Deng, Luyang Luo, and Hao Chen. Scale federated learning for label set mismatch in medical image classification. In Hayit Greenspan, Anant Madabhushi, Parvin Mousavi, Septimiu Salcudean, James Duncan, Tanveer Syeda-Mahmood, and Russell Taylor, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, pages 118–127, 2023.

[3] I Ralph Edwards and Jeffrey K Aronson. Adverse drug reactions: definitions, diagnosis, and management. *The lancet*, 356(9237):1255–1259, 2000.

[4] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

[5] Gautham Krishna Gudur and Satheesh K Perepu. Federated learning with heterogeneous labels and models for mobile activity monitoring. *arXiv preprint arXiv:2012.02539*, 2020.

[6] Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. *Advances in neural information processing systems*, 10, 1997.

[7] Chenghao Huang, Xiaolu Chen, Yanru Zhang, and Hao Wang. Fedcrl: Personalized federated learning with contrastive shared representations for label heterogeneity in non-iid data. *arXiv preprint arXiv:2404.17916*, 2024.

[8] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Federated semi-supervised learning with inter-client consistency & disjoint learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=ce6CFXBh30h`.

[9] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

[10] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 2020.

[11] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

[12] Amanda Hanora Lavan and Paul Gallagher. Predicting risk of adverse drug reactions in older adults. *Therapeutic advances in drug safety*, 7(1):11–22, 2016.

[13] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, 2020.

[14] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pages 6357–6368. PMLR, 2021.

[15] Xin-Chun Li and De-Chuan Zhan. Fedrs: Federated learning with restricted softmax for label distribution non-iid data. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021.

[16] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[17] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

[18] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[19] Harsh Vardhan, Avishek Ghosh, and Arya Mazumdar. An improved federated clustering algorithm with model-based clustering. *Transactions on Machine Learning Research*, 2024.

[20] Ting-Fan Wu, Chih-Jen Lin, and Ruby Weng. Probability estimates for multi-class classification by pairwise coupling. *Advances in Neural Information Processing Systems*, 16, 2003.

[21] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[22] Edvin Listo Zec, Adam Breitholtz, and Fredrik D Johansson. Overcoming label shift in targeted federated learning. *arXiv preprint arXiv:2411.03799*, 2024.

[23] Jiayun Zhang, Xiyuan Zhang, Xinyang Zhang, Dezhi Hong, Rajesh K Gupta, and Jingbo Shang. Navigating alignment for non-identical client class sets: A label name-anchored federated learning framework. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3297–3308, 2023.

A Experimental details

Here we detail the specifics of our experiments. In general we do a 80/20 train and validation split. For the tuning methods we train the classifier for three epochs over the unlabeled dataset each round. Both the central tuning and the client training uses the Adam optimizer with identical learning rates.

A.1 Model

We use a simple CNN model for both experiments.

Layer	# Filters
Convolution	3
Convolution	3
2×2 Max pooling	-
Convolution	3
Convolution	3
2×2 Max pooling	-
Flatten	-
Fully connected	-
Dropout (0.5)	

Table 1: Layers of the CNN used in the experiments.

A.2 Hyperparameters

Learning rate	1×10^{-3}
Batch size	64
Epochs per round	1
Number of communication rounds	100
FedProx μ	1×10^{-2}
FedRS α	0.5

Table 2: hyperparameters used during training.

B Effect of pairwise loss function with a missing label

Consider the loss function:

$$\text{loss\_tensor} = (h_{k,y_j} \cdot p_i - h_{k,y_i} \cdot p_j)^2$$

where:

- h_{k,y_i} is the probability output by client model k for class i (mapped to the client's local label space).
- h_{k,y_j} is the probability output by client model k for class j (mapped to the client's local label space).
- p_i is the probability output by the global model for class i.
- p_j is the probability output by the global model for class j.

Scenario: A single label, label 9, is not present in any client's dataset. All other labels (0-8) are present in at least one client.

Consequences:

1. **Zero client probabilities for label 9:** Because no client has seen label 9, their models will output near-zero (or zero) probabilities for this label:

$$h_{k,y_9} \approx 0 \quad \forall k$$

2. **Problematic loss calculation when label 9 is involved:** The loss calculation becomes problematic when either $i = 9$ or $j = 9$. Let's analyze both cases:

   - **Case 1:** $i = 9$

   $$\text{loss\_tensor} = (h_{k,y_j} \cdot p_9 - h_{k,y_9} \cdot p_j)^2 \approx (h_{k,y_j} \cdot p_9 - 0 \cdot p_j)^2 = (h_{k,y_j} \cdot p_9)^2$$

   The loss depends on the global model's probability for label 9 (p_9) and the client's probability for other labels j. The global model is penalized if p_9 is non-zero, even though no client provides information about label 9.

   - **Case 2:** $j = 9$

   $$\text{loss\_tensor} = (h_{k,y_9} \cdot p_i - h_{k,y_i} \cdot p_9)^2 \approx (0 \cdot p_i - h_{k,y_i} \cdot p_9)^2 = (h_{k,y_i} \cdot p_9)^2$$

   Similar to Case 1, the loss depends on p_9 and client probabilities for other labels. The global model is penalized, and gradients related to label 9 are based on "noise" from the clients.

3. **Global model degradation:** The global model's representation for label 9 is negatively impacted. The loss pushes p_9 towards zero because that's the only way to reduce the loss when paired with the near-zero client probabilities. This harms the global model's ability to generalize, even for labels present in client data.

4. **Unfair penalization:** The global model receives gradients that are based on a comparison against the absent label, creating unstable behaviour.

C Additional empirical results

Here we show some additional results.

C.1 Test accuracy over communication rounds

Here we present the test accuracy over time for the public and private settings. We compare FedAvg to the tuning methods we propose. We can see in Figures 5—8 that the convergence of the methods seem to occur at similar times in the training. To note is the increased variance in the tuning methods.

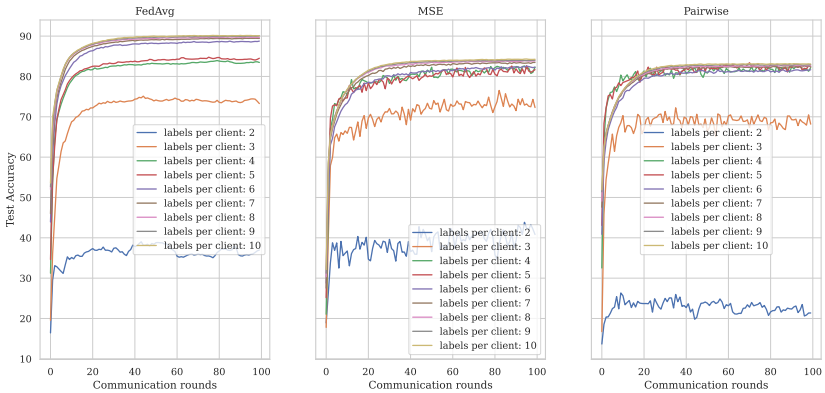


Figure 5: Test accuracy over rounds in the FashionMNIST task. The labels sets are private.

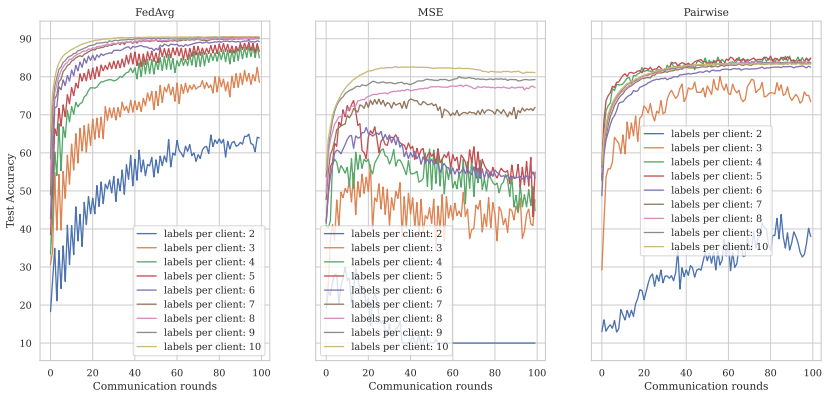


Figure 6: Test accuracy over rounds in the FashionMNIST task. The labels sets are public.

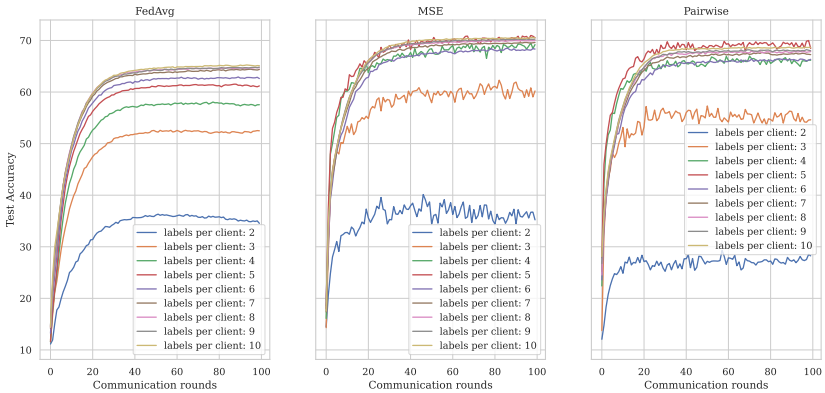


Figure 7: Test accuracy over rounds in the CIFAR10 task. The labels sets are private.

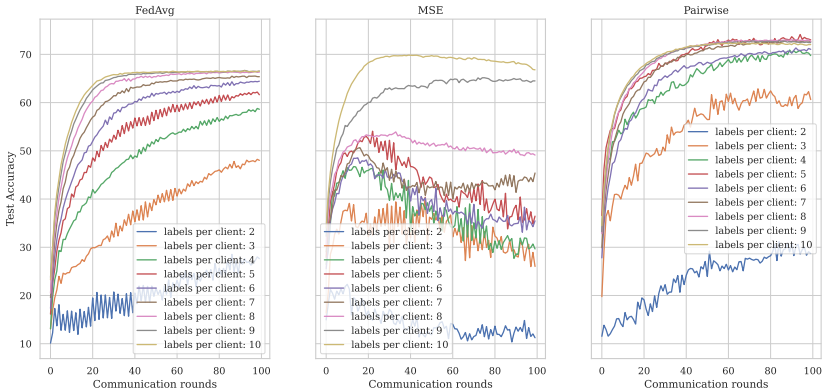


Figure 8: Test accuracy over rounds in the FashionMNIST task. The labels sets are Public.

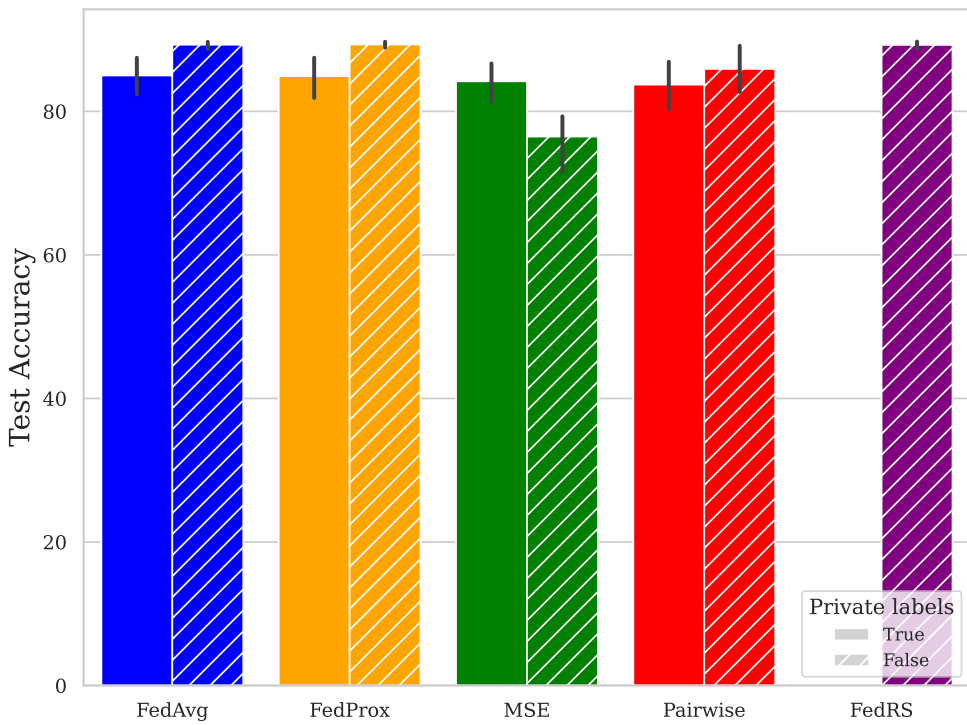


Figure 9: The results for the private and public settings for 5 labels per client on the FashionMNIST dataset.

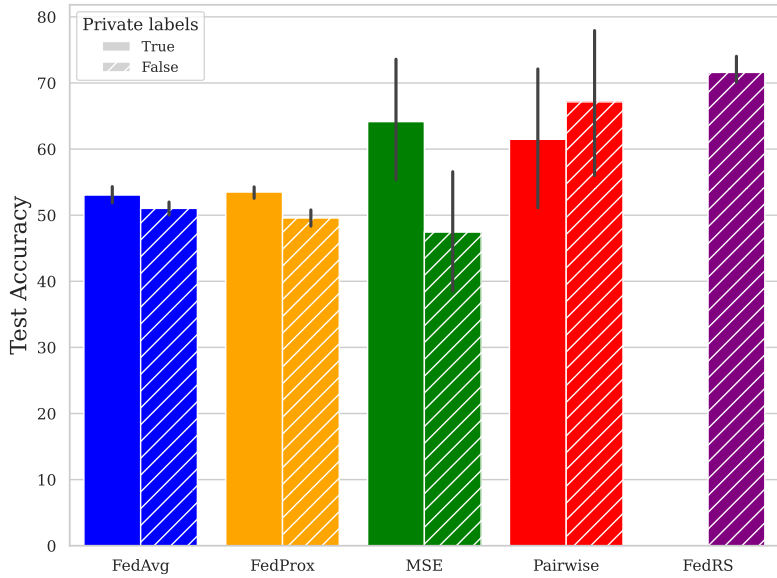


Figure 10: The results for the private and public settings with 3 labels per client on the CIFAR10 dataset.

C.2 Labels per client barplots

Here are additional barplots to compare performance at different labels per client.

C.3 Local epochs

Here we present the results of experiments on CIFAR10 with varying labels per client in figures 11–13. Note, that the central tuning step is done here with a SGD optimizer and not Adam as in other experiments. We see that our ablations across different values of epochs per round do not seem to meaningfully impact performance. Furthermore, the same pattern of the MSE loss performing better in the private setting than the pairwise is seen here.

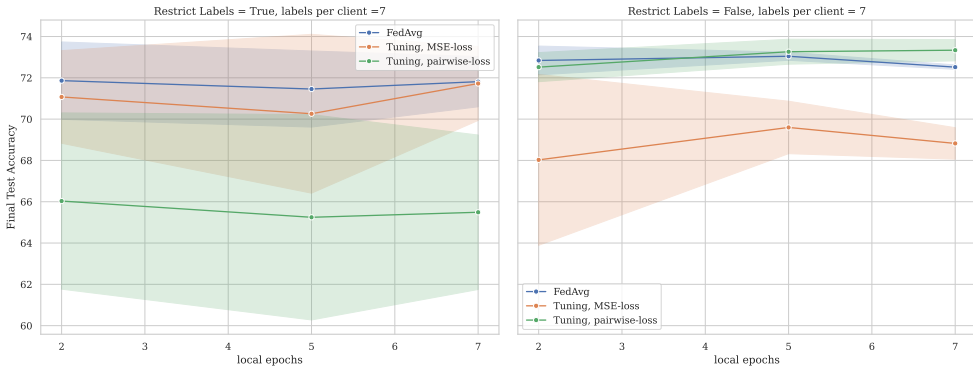


Figure 11: Ablation with differing amounts of local epochs on CIFAR10. We do not observe any particular drop in performance for either FedAvg or the tuning methods.

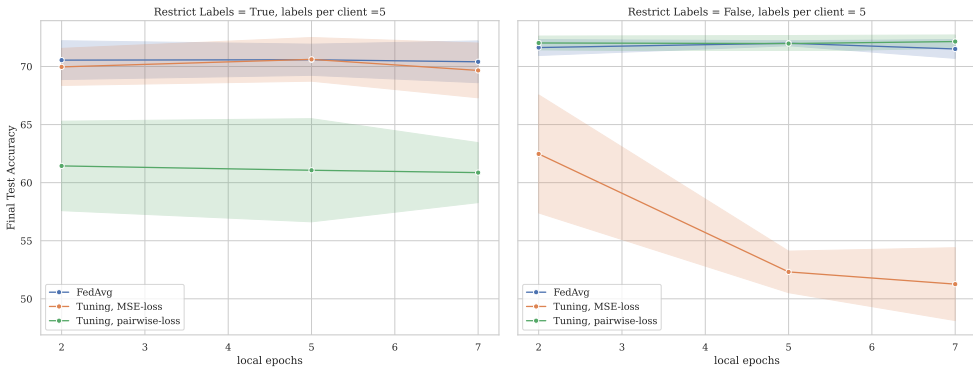


Figure 12: Comparison of different labels per client on CIFAR10 with E=5.

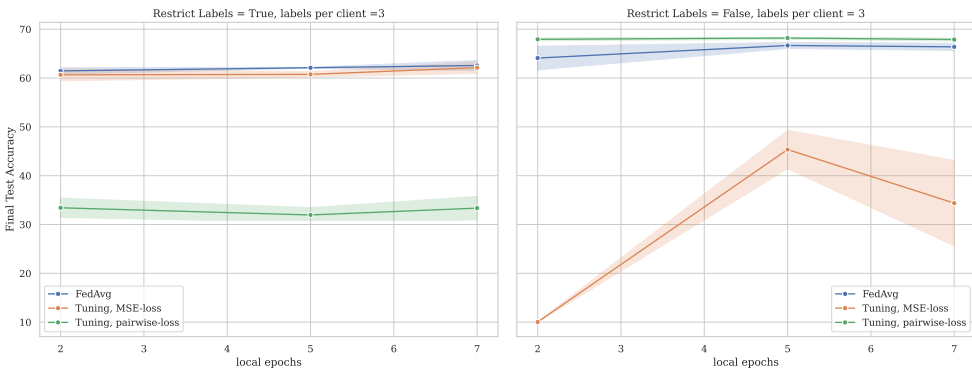


Figure 13: Comparison of different labels per client on CIFAR10 with E=3.