Beyond Invariance: Test-Time Label-Shift Adaptation for Addressing "Spurious" Correlations: SUPPLEMENTARY MATERIAL

Qingyao Sun[*]
Cornell University
qs234@cornell.edu

Kevin Murphy
Google DeepMind
kpmurphy@google.com

Sayna Ebrahimi
Google Cloud AI Research
saynae@google.com

Alexander D'Amour
Google DeepMind
alexdamour@google.com

[*]Work done as a master's student at the University of Chicago.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

A Derivation of the EM algorithm

In this section we describe how to estimate the label prior on the target distribution, $p_t(y, z) = p_t(m) = \boldsymbol{\pi}_m$, using the unlabeled data $\mathcal{D}_t^x$. There are several approaches to this, including a moment matching method called black box shift learning [Lipton et al., 2018] and an MLE approach based on the EM algorithm [Saerens et al., 2002]. In [Alexandari et al., 2020], they show that the MLE approach is much better, provided the classifier is calibrated. (See also [Garg et al., 2020] for a unified analysis of these two approaches.)

Since our augmented label space is expanded to include both class labels y and meta-data z, the number of labels M can be large, which can result in problems when computing the MLE. We therefore expand the previous approach to compute the MAP estimate, using a Dirichlet prior of the form

$$\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{m=1}^{M} \boldsymbol{\pi}_m^{\alpha_m - 1} \tag{1}$$

where $B(\boldsymbol{\alpha})$ is the normalization constant. Note that the MLE solution can be recovered by setting $\boldsymbol{\alpha} = \mathbf{1}$, which represents a uniform prior.

The goal is to maximize the (unnormalized) log posterior of $\boldsymbol{\pi}$ given the unlabeled target data $\mathbf{X}$:

$$\begin{aligned}
\mathcal{L}(\mathbf{X}; \boldsymbol{\pi}) &= \log p_t(\boldsymbol{\pi}, \mathbf{X}) && (2)\\
&= \log p_t(\mathbf{X}|\boldsymbol{\pi}) + \log \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) && (3)\\
&= \sum_{n=1}^{N} \log p_t(\boldsymbol{x}_n|\boldsymbol{\pi}) + \log \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) && (4)\\
&= \sum_{n=1}^{N} \log \left[\sum_{m=1}^{M} \boldsymbol{\pi}_m p_t(\boldsymbol{x}_n|m) \right] + \log \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) && (5)
\end{aligned}$$

The first term can be rewritten as

$$\begin{aligned}
\sum_n \log \left[\sum_{m=1}^{M} \boldsymbol{\pi}_m p_s(\boldsymbol{x}_n|m) \right] &= \sum_n \log \left[\sum_{m=1}^{M} \boldsymbol{\pi}_m \frac{p_s(m|\boldsymbol{x}_n) p_s(\boldsymbol{x}_n)}{p_s(m)} \right] && (6)\\
&= \sum_n \log \sum_m \frac{p_s(m|\boldsymbol{x})}{p_s(m)} \boldsymbol{\pi}_m + \mathrm{const} && (7)
\end{aligned}$$

This objective is a sum of logs of a linear function of $\boldsymbol{\pi}$, as is the log prior. This needs to be maximized subject to the affine constraints $\boldsymbol{\pi}_m \geq 0$ and $\sum_{m=1}^{M} \boldsymbol{\pi}_m = 1$, so the problem is concave, with a unique global optimum [Alexandari et al., 2020].

One way to compute this optimum is to use EM. Let $\boldsymbol{\pi}^j$ be the estimate of $\boldsymbol{\pi}$ at iteration j; we initialize with $\boldsymbol{\pi}_m^0 = p_s(m)$. First note that

$$p_t(\boldsymbol{x}_n, m_n) = p_s(\boldsymbol{x}_n|m_n) p_t(m_n) = \prod_{m=1}^{M} \left[p_s(\boldsymbol{x}_n|m) \boldsymbol{\pi}(m) \right]^{\mathbb{I}(m_n = m)} \tag{8}$$

Hence the complete data log posterior is given by

$$\mathcal{L}(\mathbf{X}, \mathbf{M}; \boldsymbol{\pi}) = \sum_{n=1}^{N} \sum_{m=1}^{M} \mathbb{I}\left(m_n = m\right) \log[\boldsymbol{\pi}_m p_s(\boldsymbol{x}_n|m)] + \log \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \tag{9}$$

so the expected complete data log posterior is

$$Q\left(\boldsymbol{\pi}, \boldsymbol{\pi}^{(j)}\right) = E_{\mathbf{M}}[\mathcal{L}(\mathbf{X}, \mathbf{M}; \boldsymbol{\pi})|\mathbf{X}, \boldsymbol{\pi}^{(j)}] \tag{10}$$

$$= \sum_{n=1}^{N} \sum_{m=1}^{M} p(m_n = m|\mathbf{X}, \boldsymbol{\pi}^j) \log(\boldsymbol{\pi}_m p_s(\boldsymbol{x}_n|m)) + \log \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \tag{11}$$

$$= \sum_{m=1}^{M} N_m^j \log(\boldsymbol{\pi}_m p_s(\boldsymbol{x}_n|m)) + \sum_{m=1}^{M} (\alpha_m - 1) \log \boldsymbol{\pi}_m - \log B(\boldsymbol{\alpha}) \tag{12}$$

$$= \sum_{m=1}^{M} N_m^j \log \boldsymbol{\pi}_m + \underbrace{\sum_{m=1}^{M} N_m^j \log p_s(\boldsymbol{x}_n|m)}_{\text{const}} + \sum_{m=1}^{M} (\alpha_m - 1) \log \boldsymbol{\pi}_m + \text{const} \tag{13}$$

where we drop constants wrt $\boldsymbol{\pi}$, and where we defined the expected counts to be

$$N_m^j = \sum_{n=1}^{N} p(m_n = m|\boldsymbol{x}_n, \boldsymbol{\pi}^j) \tag{14}$$

Hence in the E step we just need to compute the posterior responsibilities for each label:

$$p(m_n = m|\boldsymbol{x}_n, \boldsymbol{\pi}^j) = \frac{\boldsymbol{\pi}^j(m) p_s(\boldsymbol{x}_n|m)}{\sum_{m'=1}^{M} \boldsymbol{\pi}^j(m') p_s(\boldsymbol{x}_n|m')} = \frac{\boldsymbol{\pi}^j(m) p_s(m|\boldsymbol{x}_n)/p_s(m)}{\sum_{m'=1}^{M} \boldsymbol{\pi}^j(m') p_s(m'|\boldsymbol{x}_n)/p_s(m')} \tag{15}$$

We plug this into Equation (14) and then maximize Equation (13), using a Lagrange multiplier to enforce the sum to one constraint. We then get the following (see e.g., Sec 4.2.4 of Murphy [2022] for the derivation):

$$\hat{\boldsymbol{\pi}}_m^{j+1} = \frac{\tilde{N}_m^j}{\sum_{m'=1}^{M} \tilde{N}_{m'}^j} \tag{16}$$

where $\tilde{N}_m^j$ are the prior pseudo counts plus the expected empirical counts:

$$\tilde{N}_m^j = N_m^j + \alpha_m - 1 \tag{17}$$

At convergence, we have

$$p_t(y, z) = \hat{\boldsymbol{\pi}}_{y,z}^J \tag{18}$$

If we assume that the class label prior is constant, and only the distribution of auxiliary labels has changed, then we can write

$$p_t(y, z) = p_s(y) p_t(z|y) \tag{19}$$

where

$$p_t(z|y) = \frac{p_t(y, z)}{\sum_{z'} p_t(y, z')} \tag{20}$$

However, we do not make this fixed label assumption in our experiments.

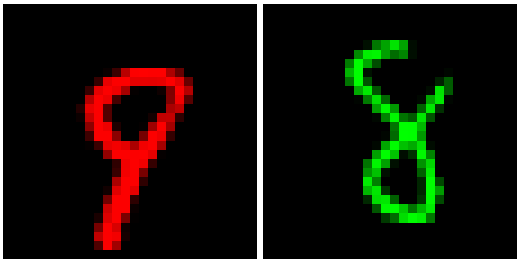

Figure 1: Samples from ColoredMNIST. (a): $y = 1$, $z = 0$. (b) $y = 1$, $z = 1$.

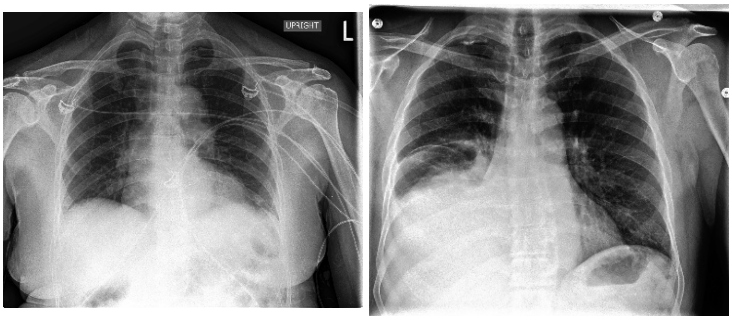

Figure 2: Samples from CheXpert. **Left**: Female patient without effusion. **Right**: Male patient with effusion.

B Datasets

In this section we discuss the datasets in more detail.

B.1 Colored MNIST

We show some sample images in Figure 1.

B.2 CheXpert

We show some sample images in Figure 2. We list all the target attributes in Table 1. To test the difficult of each task, we train a logistic regression model for each attribute on the embeddins. (We get similar results using an MLP.) The resulting AUC scores are shown in Table 1. This shows we can reliably predict all the attributes from the embeddings. The table also shows the marginal distribution of each attribute. Many labels are highly skewed, which means accuracy would be a poor measure of the predictive performance.

Interestingly, we see that we can predict sex with an AUC of 0.973, which is higher than the AUC for effusion (0.861). To understand why, note that we only use frontal scans; consequently breasts are often visible in female patients, and this is often easier to detect visually than detecting the disease itself (see Figure 2), providing a possible "shortcut" for models to exploit.

Attribute	**AUC**	**Prob.**
NO_FINDING	0.873	0.909
ENLARGED_CARDIOMEDIASTINUM	0.652	0.942
CARDIOMEGALY	0.843	0.867
AIRSPACE_OPACITY	0.711	0.480
LUNG_LESION	0.761	0.963
PULMONARY_EDEMA	0.848	0.696
CONSOLIDATION	0.683	0.911
PNEUMONIA	0.742	0.973
ATELECTASIS	0.694	0.815

Attribute	AUC	Prob.
PNEUMOTHORAX	0.883	0.875
EFFUSION	0.861	0.508
PLEURAL_OTHER	0.752	0.987
FRACTURE	0.784	0.962
SUPPORT_DEVICES	0.900	0.420
GENDER	0.973	0.586
AGE_AT_CXR	0.914	0.492
PRIMARY_RACE	0.731	0.459
ETHNICITY	0.681	0.728

Table 1: Metrics for all the attributes in the CheXpert dataset. (a) AUC using Logistic Regression on CXR embeddings. (b) Baseline prior probability for each attribute, illustrating the severe class imbalance for many attributes.

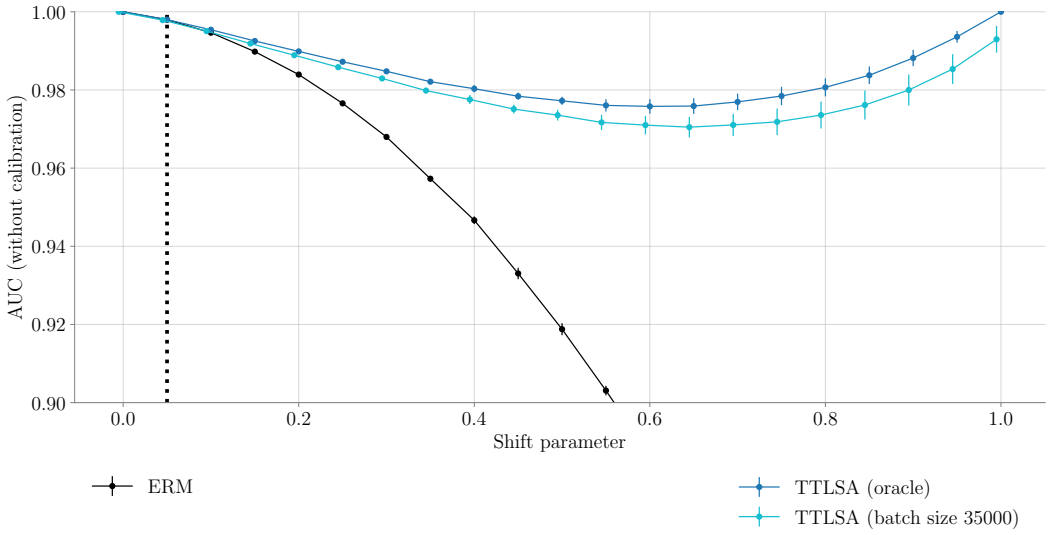

Figure 3: Performance on Colored MNIST using an uncalibrated tree classifier. TTSLA still improves the performance of the base model.

C Extra results

In this section, we include some extra experimental results.

C.1 Colored MNIST using gradient boosted tree classifier

In Figure 3, we show the results of various methods on the Colored MNIST dataset, where we use a Gradient Boosting Classification Tree as our base classifier, instead of a DNN. In particular, we use the `HistGradientBoostingClassifier` from scikit-learn [Pedregosa et al., 2011] with default parameters. The results are qualitatively similar to the DNN case.

C.2 The benefits of calibration

In Figure 4 we show the results on CheXpert if we remove the calibration step for our base classifier. Compared to **??**, we see that the overall AUC of all the methods is worse, and the variance is larger. [2] However, the rank ordering of the methods is the same. It is notable that a large gap opens up between the Oracle curve and the TTLSA implementations. This suggests that calibration primarily improves estimation of $p_t(y, z)$ estimation via EM, because the Oracle curve in this subfigure corresponds to using the correct weights with the uncalibrated $p_s(y, z|\boldsymbol{x})$ model.

C.3 CheXpert using CNN on raw pixels

In Figure 5 we show the result of various methods when applied to CheXpert images, as opposed to using embeddings. We use a ResNet-50 that was pretrained on Imagenet, which we then fine tune on CheXpert images by replicating the gray-scale image along all 3 RGB channels. The qualitative conclusions are the same as in the embedding case.

C.4 More results on the benchmark datasets

In Table 2 and Table 3 we report the per-group accuracy on the benchmark datasets.

[2] For a binary classification problem, calibration will not change the AUC, but since we derive the posterior over class labels by marginalizing a 4-way joint, $p(y|\boldsymbol{x}) = \sum_{z=0}^{1} p(y, z|\boldsymbol{x})$, calibration can help.

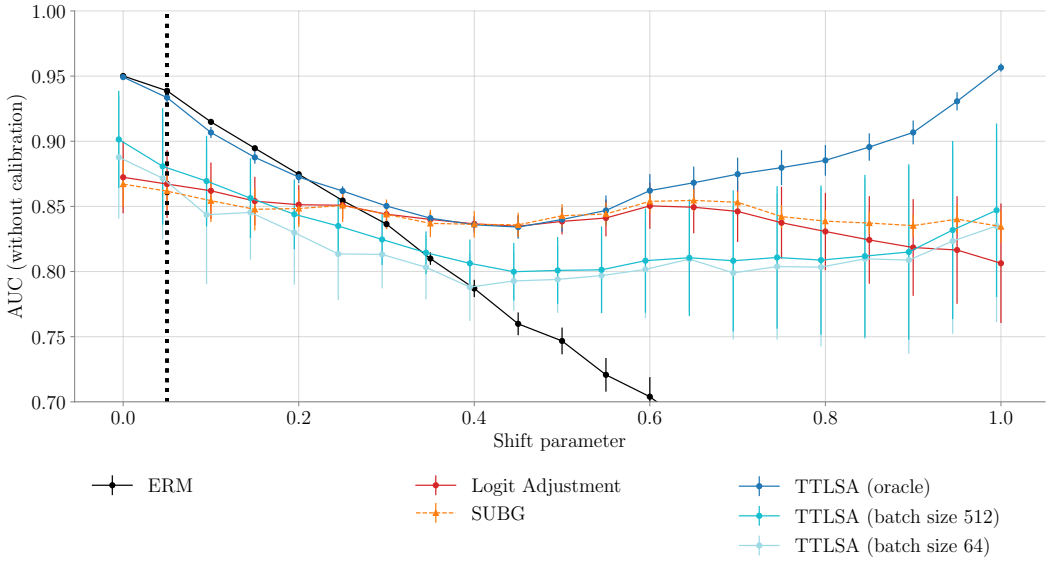

Figure 4: Performance across target domains on CheXpert embeddings, following the setup of Figure **??**. (a) Results using calibration. Performance mirrors those in Figure **??**. (b) Results without calibration. We see that calibration both improves performance and decreases variability between runs.

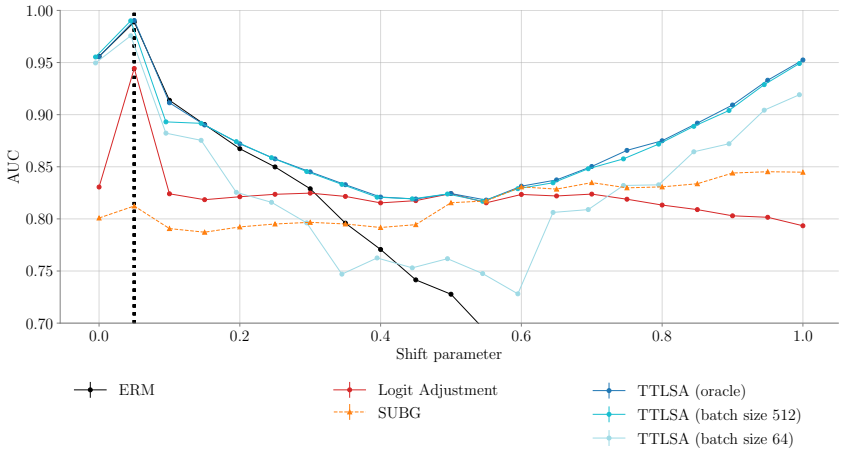

Figure 5: Performance on CheXpert using raw image (pixel) input instead of embeddings. These results are with calibration.

Data	Method	Group label (Y, Z) (0, 0)	(0, 1)	(1, 0)	(1, 1)	(2, 0)	(2, 1)
CelebA	ERM	86.71 (0.67)	92.65 (0.79)	96.80 (0.18)	80.83 (1.46)		
	gDRO	92.71 (0.05)	92.33 (0.09)	92.63 (0.35)	87.36 (0.47)		
	SUBG	91.76 (0.22)	91.91 (0.55)	90.96 (0.33)	87.22 (1.38)		
	LA	91.43 (0.11)	94.77 (0.08)	95.88 (0.12)	84.72 (0.58)		
	TTLSA	97.59 (0.18)	98.78 (0.03)	80.70 (1.22)	51.25 (0.27)		
Waterbirds	ERM	98.95 (0.13)	86.57 (0.48)	86.02 (0.13)	95.76 (0.16)		
	gDRO	93.46 (0.22)	88.00 (0.88)	90.15 (0.10)	92.06 (0.34)		
	SUBG	90.76 (0.69)	88.96 (0.19)	91.28 (0.35)	91.36 (0.24)		
	LA	94.43 (1.63)	88.38 (0.36)	91.32 (0.43)	93.15 (0.54)		
	TTLSA	94.59 (0.44)	93.68 (0.73)	95.72 (0.29)	97.12 (0.19)		
MultiNLI	ERM	80.75 (0.79)	94.94 (0.11)	83.18 (0.47)	78.05 (1.31)	81.98 (0.48)	68.60 (0.40)
	gDRO	80.36 (0.63)	85.27 (0.25)	82.48 (0.59)	81.21 (1.30)	79.39 (1.34)	76.87 (1.28)
	SUBG	69.63 (0.17)	82.85 (0.19)	74.39 (0.21)	79.68 (0.17)	69.84 (0.48)	68.40 (1.33)
	LA	81.63 (1.15)	87.79 (1.99)	84.36 (0.89)	80.95 (2.03)	78.77 (1.09)	76.33 (1.45)
	TTLSA	80.24 (0.87)	94.74 (0.58)	81.73 (2.45)	73.90 (1.72)	82.40 (1.49)	63.76 (2.15)
CivilComments	ERM	92.23 (0.42)	90.38 (0.46)	68.57 (1.07)	68.32 (0.97)		
	gDRO	83.94 (0.70)	79.92 (0.33)	80.97 (0.63)	81.09 (0.42)		
	SUBG	79.79 (0.56)	79.14 (0.34)	82.52 (0.46)	76.56 (0.25)		
	LA	84.45 (0.16)	79.27 (1.17)	83.00 (0.95)	84.20 (0.99)		
	TTLSA	85.53 (1.35)	74.94 (1.96)	84.34 (2.36)	84.61 (2.21)		

Table 2: Per-group accuracy on the benchmark datasets, where model selection is based on the worst (Y, Z) group accuracy on a validation set. Numbers in parentheses signify the standard error calculated based on 4 replication runs.

Data	Method	Group label (Y, Z) (0, 0)	(0, 1)	(1, 0)	(1, 1)	(2, 0)	(2, 1)
CelebA	ERM	96.54 (0.21)	99.58 (0.05)	86.47 (0.89)	40.28 (2.24)		
	gDRO	95.48 (0.14)	96.76 (0.14)	87.10 (0.34)	68.75 (1.14)		
	SUBG	95.45 (0.39)	95.93 (0.44)	79.81 (1.30)	67.64 (4.66)		
	LA	95.25 (0.55)	98.96 (0.49)	88.77 (1.81)	43.47 (12.05)		
	TTLSA	97.68 (0.13)	98.99 (0.07)	80.21 (1.04)	47.36 (1.81)		
Waterbirds	ERM	99.42 (0.11)	90.27 (1.24)	80.61 (2.51)	94.16 (0.84)		
	gDRO	97.04 (1.44)	92.84 (1.07)	83.84 (2.47)	89.10 (0.82)		
	SUBG	96.98 (0.29)	95.88 (0.42)	82.87 (1.51)	83.33 (1.86)		
	LA	98.23 (0.15)	92.42 (0.36)	85.98 (1.06)	92.91 (0.19)		
	TTLSA	99.06 (0.11)	93.61 (1.04)	87.66 (0.40)	95.02 (0.92)		
MultiNLI	ERM	82.43 (0.06)	95.47 (0.08)	83.62 (0.03)	77.14 (0.16)	80.45 (0.09)	67.36 (0.54)
	gDRO	80.37 (0.82)	86.32 (0.64)	81.06 (0.72)	78.22 (0.60)	81.22 (0.22)	78.83 (0.29)
	SUBG	68.30 (2.00)	83.95 (2.28)	75.72 (1.68)	79.40 (1.37)	69.91 (1.55)	66.44 (2.23)
	LA	82.74 (0.06)	92.92 (0.34)	83.97 (0.44)	79.88 (0.55)	79.77 (0.34)	71.49 (0.95)
	TTLSA	81.86 (0.19)	96.52 (0.11)	83.89 (0.37)	76.07 (0.71)	80.51 (0.17)	56.60 (1.36)
CivilComments	ERM	96.00 (0.38)	95.63 (0.53)	55.27 (1.88)	52.21 (2.43)		
	gDRO	89.59 (0.68)	86.60 (0.86)	71.56 (1.64)	71.94 (1.33)		
	SUBG	81.43 (1.09)	80.67 (1.28)	80.80 (1.34)	76.05 (0.44)		
	LA	84.45 (0.16)	79.27 (1.17)	83.00 (0.95)	84.20 (0.99)		
	TTLSA	91.53 (0.58)	87.18 (1.70)	70.14 (2.69)	71.32 (2.82)		

Table 3: Per-group accuracy on the benchmark datasets, where model selection is based on the average (Y, Z) group accuracy on a validation set. Numbers in parentheses signify the standard error calculated based on 4 replication runs.

C.5 Training with partial group labels

In this section, we evaluate an extension of our method where not all training samples have group labels z. In particular, we first train an ERM model to predict z on samples with group labels z, calculate $p(z|x)$ for training samples with missing z, and then fit a new $p(y, z|x)$ model on the augmented data. In particular, we represent each (y, z) target as a one-hot vector when z is known, and use a soft (predicted) encoding when z is unknown. We train with cross entropy loss. The use of soft labels may have the benefits of self-distillation Pham et al. [2022]. The validation set is always fully labeled for the purpose of hyperparameter tuning.

The results (on the 4 benchmark datasets) are shown in Table 4. The accuracy barely drops as missingness increases, which means our method is robust to the deficiency in group labels z.

	Missingness				
Data	0	0.5	0.75	0.875	0.9375
CelebA	84.72 / 95.55	78.33 / 95.68	77.78 / 95.37	79.44 / 94.44	77.22 / 95.20
Waterbirds	88.38 / 95.23	87.63 / 93.98	88.79 / 94.41	88.65 / 94.67	91.28 / 95.05
MultiNLI	76.33 / 82.60	74.87 / 79.55	74.72 / 82.49	76.05 / 82.61	78.75 / 81.72
CivilComments	79.27 / 85.03	76.26 / 85.87	73.87 / 83.55	73.41 / 84.57	66.64 / 80.36

Table 4: Accuracy of the worst / average (y, z) group on the benchmark datasets with partial training z labels, where model selection is based on average z accuracy. The *Missingness* columns stand for the proportion of training set with missing labels, e.g. 0.75 means only 25% of the training samples have z labels.

D Potential negative societal impacts

The proposed method in this work yields a model that can adapt to a new distribution and improves the performance at test time by exploiting spurious correlations to create a label shift correction technique that adapts to changes in the marginal distribution $p(y, z)$ using unlabeled samples from the target domain. In this way, there are potential societal benefits to our method, especially when z corresponds to a socially salient attribute, such as a protected class. However, use cases of this type require caution, especially given the limitations discussed in **??**. Further, as we discuss in a footnote in the main text, our method does not address concerns about cases where making decisions on the basis of z is discouraged or forbidden for *a priori* reasons. Given these limitations, there is a potential that the existence of adaptation methods of this type could be used to downplay the potential dangers of misusing sensitive information in machine learning systems. Here, we hope researchers and practitioners will instead acknowledge that, while beneficial use cases of z information exist, (1) there is a need to validate empirically that a particular use of z information is actually socially beneficial, and (2) there are valid reasons why one might want to avoid using z information altogether. Further, there is a potential risk that if the measurement quality of the labels y, z shift across distributions, such that they measure distinct concepts, or exhibit substantially different noise properties (i.e., become biased, or exhibit more outliers), our framework might absorb them during adaptation and eventually the outcomes of the system might be biased as well.

E Invariance Equivalences and Conditions

In this section, we review connections that have been established between risk invariance, ERM on balanced data, "separation" between a predictor $f(X)$ and the spurious factor Z, and worst-(y, z)-group performance. These results are useful for understanding why the application of logit adjustment at training time often yields a predictor that exhibits approximately invariant risk across the test sets that we study in our experiments.

E.1 Key Concepts

Risk invariance A predictor is risk-invariant with respect to a loss function ℓ and a family of test distributions $\mathcal{Q}$ iff it has the same risk $E_Q[\ell(f(X), Y)]$ for each $Q \in \mathcal{Q}$. The results we discuss apply to test distribution families that preserve both the generative distribution *and* the label distribution of the source distribution; that is, $\mathcal{Q}$ is the set of distributions such that $Q(Y) = P(Y)$ and $Q(X \mid Y, Z) = P(X \mid Y, Z)$ for each $Q \in \mathcal{Q}$. This formulation allows $Q(Z \mid Y)$ can change. This is is the family is considered in Makar et al. [2022] and Makar and D'Amour [2022], and is called a "causally compatible" family in Veitch et al. [2021], or a correlation shift in Yi et al..

Pure spuriousness The data generating process in Figure 1 is purely spurious if there exists some sufficient statistic $e(X)$ such that (1) $Y \perp X \mid e(X)$ and (2) $e(X) \perp Z \mid Y$. In words, if we know $e(X)$, there is no further dependence between Y and X, and further, $e(X)$ does not depend on the spurious factor Z except through Z's marginal dependence with Y. This is consistent with a causal model where the influence of Y on X is totally mediated by $e(X)$, and Z has no causal effect on $e(X)$.

Veitch et al. [2021] coined the term "purely spurious" in a context of a full counterfactual model of data generation, to refer to data generating processes where the portions of X that are causally related to Y and Z can be separated in a specific sense. Makar et al. [2022] consider the special case of pure spuriousness in the context of the anti-causal model in Figure 1. (They do not use the term "purely spurious" as the work in Veitch et al. [2021] was concurrent; Makar and D'Amour [2022] makes the connection explicit.) Here, we use formalism from Makar et al. [2022] to present the idea to minimize conceptual overhead.

Note that when the data X is rich, such as images are long passes of text, pure spuriousness is more plausible (or a better approximation to reality) because there is less possibility of descructive interfecence between Y and Z in the generation of X. Specifically, the simplest examples where pure spuriousness fails are ones where X is very low-content: e.g., Y and Z are binary, and $X := Y$ `OR` Z.

Separation Separation is a concept popularized in the literature on ML fairness [Barocas et al., 2019, Chapter 3], which stipulates that the predictor $f(X)$ should satisfy the the conditional independence $f(X)Z \perp |Y$. When Z is a sensitive attribute, this condition stipulates that the predictor $f(X)$ should contain no more information about Z than one could glean from knowing Y alone.

Data balancing Idrissi et al. [2022] study predictors trained on data subsampled so that the (Y, Z) distribution is uniform; they call this data-balancing. Makar et al. [2022] and Makar and D'Amour [2022] study a similar predictors optimized on a similar "ideal" distribution, where $Q(Y, Z) = P(Y)P(Z)$ for some source distribution P. This distribution does not "balance" the marginals of Y and Z, but it eliminates the marginal correlation between Y and Z.

Worst group performance Sagawa et al. [2020] define groups in terms of (z, y) values. The group conditional risk is $R_{z,y} = E_Q[\ell(f(X), Y) \mid Z = z, Y = y]$. Note that for all families of test sets that we consider, the group-conditional risks are equal for all Q. Worst group risk minimization attempts to minimize the group conditional risk of the worst subgroup. Saerens et al. [2002] propose a distributionally robust optimization algorithm for performing this minimization.

E.2 Connections

In the purely spurious setting, there are several connections and near-equivalences between risk invariance, separation, optimality on balanced data, and worst group risk minimization.

Yi et al. establish that for label distribution preserving target families, a predictor $f(X)$ that satisfies separation $f(X) \perp Z \mid Y$ will have invariant risk across the family $\mathcal{Q}$ defined above. Notably, this result does *not* require pure spuriousness.

Under pure spuriousness, the separation condition achieves a certain optimality. Veitch et al. [2021], Theorem 4.3 establishes that in the purely spurious case, the minimax optimal across the family $\mathcal{Q}$ satisfies separation $f(X) \perp Z \mid Y$. Similarly, under pure spuriousness, Makar and D'Amour [2022], Proposition 2, establishes that the optimal risk-invariant predictor satisfies separation.

Interestingly, this result establishes a connection between optimality under balanced data, separation, and optimal risk invariance. Specifically, Makar et al. [2022], Proposition 1 establishes that the optimal model for the "ideal" uncorrelated distribution for which $Q(Y, Z) = P(Y)P(Z)$ achieves risk invariance across the family $\mathcal{Q}$. Thus, minimizing risk under a separation constraint targets a similar predictor to the predictor that one would target simply optimizing on balanced data. Makar and D'Amour [2022] shows that the near-equivalence holds up empirically, such that learning algorithms targeted at efficiently learning the optimal predictor on balanced data can satisfy both risk invariance and separation criteria.

Idrissi et al. [2022] establish that, empirically, models trained to minimize risk on balanced data also yield favorable worst-group performance, showing that subsamping can be particularly effective. Sagawa et al. [2020] explore similar ideas, focusing on reweighting strategies, which both they and Idrissi et al. [2022] find to work relatively poorly with neural models in the data regimes they study. Sagawa et al. [2020] further establish that under certian convexity conditions, there does exist a reweighting of the data that optimizes worst-group performance, but provide a counterexample showing that this is not always the case with non-convex losses.

Based on the above results, in the purely spurious case, one can establish the following, for $\mathcal{Q}$ with a uniform distribution on Y:

1. There exists a predictor $f^*(X)$ that is optimal on the ideal balanced data, is the optimal risk-invariant predictor, and satisfies separation $f(X) \perp Z \mid Y$.
2. For all $Q \in \mathcal{Q}$, the group-specific risks are equal within labels, i.e., $E_Q[\ell(f^*(X), Y) \mid Y = y, Z = z] = E_Q[\ell(f^*(X), Y) \mid Y = y, Z = z']$ for all y.

The latter fact does not imply that $f^*(X)$ also optimizes worst-group risk, but it does imply that the worst group cannot be the worst due to a spurious correlation between Y and Z. This is because, for a fixed label value y, the risks of (y, z) subgroups are the same.

References

A. Alexandari, A. Kundaje, and A. Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 222–232. PMLR, July 2020. URL `https://proceedings.mlr.press/v119/alexandari20a.html`.

S. Barocas, M. Hardt, and A. Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

S. Garg, Y. Wu, S. Balakrishnan, and Z. Lipton. A unified view of label shift estimation. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3290–3300. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/219e052492f4008818b8adb6366c7ed6-Paper.pdf`.

B. Y. Idrissi, M. Arjovsky, M. Pezeshki, and D. Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pages 336–351. PMLR, 2022.

Z. Lipton, Y.-X. Wang, and A. Smola. Detecting and correcting for label shift with black box predictors. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on*

Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 3122–3130. PMLR, July 2018. URL https://proceedings.mlr.press/v80/lipton18a.html.

M. Makar and A. D'Amour. Fairness and robustness in anti-causal prediction. Sept. 2022. URL http://arxiv.org/abs/2209.09423.

M. Makar, B. Packer, D. Moldovan, D. Blalock, Y. Halpern, and A. D'Amour. Causally motivated shortcut removal using auxiliary labels. In *AISTATS*, volume 151, pages 739–766, 2022. URL https://proceedings.mlr.press/v151/makar22a/makar22a.pdf.

K. P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

M. Pham, M. Cho, A. Joshi, and C. Hegde. Revisiting self-distillation, 2022.

M. Saerens, P. Latinne, and C. Decaestecker. Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure. *Neural Computation*, 14(1):21–41, 01 2002. ISSN 0899-7667. doi: 10.1162/089976602753284446. URL https://doi.org/10.1162/089976602753284446.

S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for Worst-Case generalization. In *ICLR*, 2020. URL http://arxiv.org/abs/1911.08731.

V. Veitch, A. D'Amour, S. Yadlowsky, and J. Eisenstein. Counterfactual invariance to spurious correlations in text classification. In *NIPS*, Nov. 2021. URL https://openreview.net/forum?id=BdKxQp0iBi8.

M. Yi, R. Wang, J. Sun, Z. Li, and Z.-M. Ma. Breaking correlation shift via conditional invariant regularizer. In *The Eleventh International Conference on Learning Representations*.