Uniformly Distributed Feature Representations for Fair and Robust Learning

Anonymous authors
Paper under double-blind review

Abstract

A fundamental challenge in machine learning is training models that generalize well to distributions different from the training distribution. Empirical Risk Minimization (ERM), which is the predominant learning principle, is known to under-perform in minority sub-populations and fail to generalize well in unseen test domains. In this work, we propose a novel learning principle called Uniform Risk Minimization (URM) to alleviate these issues. We first show theoretically that uniform training data distributions and feature representations support robustness to unknown distribution shifts. Motivated by this result, we propose an empirical method that trains deep neural networks to learn a uniformly distributed feature representation in their final activation layer for improved robustness. Our experiments on multiple datasets for sub-population shifts and domain generalization show that URM improves the generalization of deep neural networks without requiring knowledge of groups or domains during training. URM is competitive with the best existing methods designed for these tasks and can also be easily combined with them for improved performance. Our work sheds light on the importance of the distribution of learned feature representations for downstream robustness and fairness.

1 Introduction

ERM has been the predominant approach to train machine learning models. Consistently achieving high accuracy on test distributions identical to the training data distribution was a challenge until the resurgent success of deep learning. However, machine learning practitioners today realize that a major challenge is robustness in test distributions different from the training data distribution. Hence, the common assumption that training and test data distributions are identical no longer fully capture the challenges of modern machine learning.

In this work, we consider the challenging problem of training models that perform well in distributions different from the training distribution without assuming knowledge of the test distribution during the learning process. Specifically, we explore the question of what is the best data distribution to train classifiers on for improved downstream generalization. We consider two downstream generalization tasks in our work, i.e., sub-population shifts and domain generalization. In the sub-population or group shift problem setup, we have input $\mathbf{x} \in \mathcal{X}$ and labels $y \in \mathcal{Y}$, with the objective to learn a function $f : \mathcal{X} \to \mathcal{Y}$. Moreover, there exist attributes of the data $a_1, ..., a_i, ..., a_m$, $a_i \in \mathcal{A}_i$, which may not be available during training. Hence, distinct sub-populations or groups of the data can be defined using combinations of the attribute and label, for example $h : \mathcal{A} \times \mathcal{Y} \to \mathcal{G}$. The training distribution over $(\mathbf{x}, y)$ is a mixture of group-specific distributions p_g i.e. $p_{tr} = \sum_{g \in \mathcal{G}} \gamma_g p_g$, where $\gamma \in \Delta_{|\mathcal{G}|}$. The test distribution, which is not observed during training, is: $p_{te} = \sum_{g \in \mathcal{G}} \beta_g p_g$, where $\beta \in \Delta_{|\mathcal{G}|}$ and $\gamma \neq \beta$. The objective for f is to perform accurately in the unobserved p_{te}. Generally, models perform most poorly when the test distribution only comprises minority sub-populations of the training distribution (Hashimoto et al., 2018; Zhang et al., 2020). In sensitive applications of machine learning such as healthcare, this leads to *fairness* issues when models perform better for some groups of people than others. Multiple methods have been proposed to tackle this problem of robustness to sub-population shifts. A common approach so far has been identifying the samples where models performs poorly and up-weighting these samples during training. This *identify-and-emphasise*

paradigm has been proposed in multiple works (Liu et al., 2021; Zhang & Sabuncu, 2018; Yaghoobzadeh et al., 2021; Sanh et al., 2021; Utama et al., 2020). Many proposed methods also rely on the knowledge of minority sub-population attributes for training or validation to achieve good performance (Gowda et al., 2021; Izmailov et al., 2022; Menon et al., 2021; Nam et al., 2022; Yao et al., 2022). However, knowledge of relevant sub-populations may not always be known, especially in novel settings, so this assumption may be impractical.

In domain generalization (DG), models are provided a set of training domains, each containing examples about the same task, with the goal that they must perform well on unseen target domains (Blanchard et al., 2011; Muandet et al., 2013). While challenging, DG captures real-world scenarios where unforeseen distribution shifts between training and test distributions can be encountered by models. Both scenarios, sub-population shifts and domain generalization, require novel solutions that enable generalization while requiring limited privileged information such as access to minority group annotations or access to samples from the test domains.

In this work, we present a novel learning principle called Uniform Risk Minimization (URM) to alleviate these issues. We first propose a novel risk measure, *uniform risk*, to define the *expected* risk of the model when the test distribution is not equal to the training distribution. Specifically, uniform risk is equal to the expected risk of a model over all possible test distributions under a uniform prior. Our main theoretical result is as follows: *training classifiers on uniformly distributed training data or learned feature representations is an optimal choice to lower uniform risk and supports robustness and fairness.* Motivated by this result, we propose an empirical method to encourage deep neural networks to learn uniformly distributed feature representations using an adversarial training objective. We show using experiments that encouraging models to have uniformly distributed feature representations significantly improves their robustness to sub-population shifts and domain generalization, without requiring knowledge of groups or domains during training.

The major contributions of our work are as follows:

1. We propose Uniform Risk Minimization (URM), a novel learning principle for out of distribution (ood) robustness and fairness (Section 2.3).
2. We theoretically show that training classifiers on uniformly distributed data and feature representation spaces in deep neural networks are optimal for lowering *uniform risk* and support robustness (Propositions 2.2, 2.3).
3. Motivated by our findings, we propose a method to learn uniformly distributed feature representations in the final activation layer of deep neural networks using an adversarial objective (Section 2.6). We empirically demonstrate the efficacy of this method for sub-population shifts and domain generalization, without requiring knowledge of task-specific knowledge such as group or domain labels (Section 3).

2 Uniform Risk Minimization

In this section, we present a novel learning principle called Uniform Risk Minimization (URM) for robustness under distribution shifts. Motivated by real-world scenarios where we do not have knowledge of the environments models will be tested on, we propose a novel risk measure called *uniform risk* that is the *expected* test risk over *all* possible test distributions under a uniform prior (Section 2.3). We then show theoretically that uniformly distributed training data (Proposition 2.2) or feature distributions (Proposition 2.3) are optimal in lowering uniform risk. We then present a practical method for deep learning models that encourages their feature representations to be uniformly distributed using adversarial distribution matching to improve their robustness in various distribution shift scenarios (Section 2.6).

2.1 Notation and Problem Setup

We describe the general setting of distribution shifts between the training and test distributions for the classification problem. We denote the joint training distribution as $p_{tr}(x, y)$ and the test distribution as

$p_{te}(x, y)$, where x denotes the input and y denotes the label. We assume these two distributions have the same support sets $\mathcal{X}, \mathcal{Y}$. We denote a labeled training dataset of size N_{tr} sampled from the training distribution $(x_{tr}^{(i)}, y_{tr}^{(i)})_{i=1}^{N_{tr}}$, where $(x_{tr}^{(i)}, y_{tr}^{(i)}) \sim p_{tr}(x, y)$, and a dataset of size N_{te} from the test distribution $(x_{te}^{(i)}, y_{te}^{(i)})_{i=1}^{N_{te}}$, where $(x_{te}^{(i)}, y_{te}^{(i)}) \sim p_{te}(x)$.

2.2 A Generalization Bound for Distribution Shifts

We first define the test loss under distribution shift. The negative log-loss on the test distribution is:

$$l_{test} = \mathbb{E}_{p_{te}(x,y)}[-\log \hat{p}(y|x)] \tag{1}$$

Next, we propose a bound of the test loss using the training loss and the Kullback-Leibler (KL) divergence between train and test distributions by extending Proposition 1 in Nguyen et al. (2022).

Proposition 2.1. *If the loss $-\log \hat{p}(y|x)$ is bounded by M[1] and the labeling mechanism, $p(y|x)$, is unchanged between train and test distributions (covariate-shift assumption), we have:*

$$l_{test} \leq l_{train} + \frac{M}{\sqrt{2}}[\mathrm{KL}[p_{te}(x)|p_{tr}(x)] + \frac{M}{4\sqrt{2}} \tag{2}$$

Proof. Proof in Appendix A. □

An alternate proposition based on the *label* or *target*-shift assumption is discussed in Appendix E.1.

2.3 Uniform Risk

In real-world deployments of machine learning, we cannot know in advance which distributions a model will encounter. Hence, we propose a novel risk measure that incorporates the *expected* risk over all possible test distributions. As we do not assume any prior knowledge on the test distribution our model may encounter in the wild, the expected risk is defined using an uninformative *uniform prior* over all possible test distributions. A key fact enabling the definition of such a uniform prior is that all computer systems represent data using finite-precision floating points, making all data representations discrete in practice. This enables us to define the uniform prior over all test distributions using the Dirichlet distribution. Let N be the size of the support set of the data distribution i.e. the total number of possible inputs. For N-dimensional distributions, the uniform prior can be represented using the Dirichlet distribution, such that $p_{te}(x) \sim Dir(\boldsymbol{\alpha} = \mathbb{1})$. Note that $Dir(\boldsymbol{\alpha} = \mathbb{1})$ is the uniform prior with parameters $\alpha_i = 1, \forall i \in \{1, \ldots, N\}$, and $\alpha_0 = \sum_{i=1}^{N} \alpha_i = N$. In Eq. 3, we define *uniform risk*, $\mathcal{R}_U$, as the expectation of the test loss under a uniform prior over all possible test distributions. We assume that training and test distributions have the same support.

$$\mathcal{R}_U \coloneqq \mathbb{E}_{p_{te}(x) \sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[l_{test}] \tag{3}$$

We now upper bound *uniform risk* using Eq. 2:

$$\mathbb{E}_{p_{te}(x) \sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[l_{test}] \leq l_{train} + \frac{M}{\sqrt{2}}\mathbb{E}_{p_{te}(x) \sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[\mathrm{KL}[p_{te}(x)|p_{tr}(x)]] + \frac{M}{4\sqrt{2}} \tag{4}$$

[1] As suggested by (Nguyen et al., 2022), in a classification problem, this can be enforced easily by augmenting the output softmax of the classifier such that each class probability is always at least $\exp(-M)$. For example, if we choose $M = 3 \Rightarrow \exp(-M) \approx 0.05$, and if the output softmax is $(p_1, p_2, ..., p_C)$, we can augment it as $(p_1 \cdot K + 0.05, p_2 \cdot K + 0.05, ..., p_C \cdot K + 0.05)$, where $K = 1 - 0.05 \cdot C$ and C is the number of classes. This ensures the bound for the loss on a sample, while leaving the output prediction class unchanged.

Note that the bound depends on the training loss, the expected KL-divergence between training and test distributions, and a constant (Equation 4). Modern deep learning systems can minimize the training loss nearly to zero as long as the task is well-defined and realizable by modern architectures. Hence, the main component of Eq. 4 to minimize would be the *expected KL-divergence between training and test distributions.* This leads us to our first key result:

Proposition 2.2. *The expected KL-divergence between train and test distributions in Eq. 4 is minimized when the training data distribution $p_{tr}(x)$ is uniform:*

$$\arg\min_{p_{tr}(x)} \mathbb{E}_{p_{te}(x)\sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[\mathrm{KL}[p_{te}(x)|p_{tr}(x)]] = u^*_{tr}(x)$$

$$\textit{where } u^*_{tr}(x_i) = \frac{1}{N}, \forall i \in \{1, \ldots, N\}$$

Proof. Proof in Appendix B. □

Proposition 2.2 states that the training data distribution that minimises the upper bound on *uniform risk* is the uniform distribution, $u^*_{tr}(x)$. This means that training on uniform data distributions may improve robustness to distribution shifts. However, most practical training datasets may not be uniformly distributed. While uniformity could in principle be achieved by re-weighting the data distribution i.e. by down-weighting high-density regions and up-weighting low-density regions of the distribution, this would require knowledge of the data distribution or an accurate estimation of it, which is non-trivial in practice. Modern deep learning systems are usually trained on high-dimensional image or audio datasets whose distributions are not known and are difficult to estimate. Hence, we will consider a similar proposition in the learned feature space of deep neural networks, which can be trained to be uniformly distributed (Section 2.5).

2.4 Revisiting Group-balanced and Class-balanced Training via URM

Rather than considering the data distribution over the input space, we can also consider *sub-group* and *class* distribution shifts i.e., shifts in $p(g)$ and $p(y)$, respectively. Similar to Proposition 2.2, we can show that training with uniform sub-group or class distributions—equivalent to group-balanced and class-balanced training—reduces uniform risk as defined in these scenarios (discussion and proof in Appendix E). Additionally, in these scenarios, *uniform risk* can be shown to be equivalent to the balanced risk across sub-groups or classes, respectively (proof in Appendix E). Thus, minimizing uniform risk in these scenarios directly corresponds to minimizing the balanced risk across sub-groups or classes. In fact, prior work (Idrissi et al., 2022) has shown that group-balanced training is an effective approach for robustness to sub-population shifts when group-attributes of the training samples are known. Prior work (Buda et al., 2018) has also demonstrated the efficacy of class-balanced training. We provide further evidence below that class-balanced training is effective for label-shifts below in Section 3.3. Hence, URM provides a unified perspective on various algorithms that perform group or class balanced training.

2.5 Lowering Uniform Risk in Learned Feature Representation Spaces

We have above defined uniform risk using the input space (Eq. 3) and shown that uniform training data distributions lower uniform risk (Proposition 2.2). To implement this in practice would require training our models on uniformly distributed training data distributions. However, most practical training datasets are not uniformly distributed and it is difficult to modify their distribution to match a uniform distribution. Hence, we re-define the bound on *uniform risk* using the *learned* feature representation space of a deep model instead. As in Proposition 2.2, we will similarly show that uniformly distributed *feature representations* also lower uniform risk. This allows us to implement URM by training deep neural networks to *learn* a uniformly distributed feature representations rather than training on an input data distribution that is already uniformly distributed.

Let G be a model parameterized by a deep neural network that outputs a latent variable z i.e. $z = G(x)$, which has dimensionality Z and is used by a downstream task head such as linear classifier. In order to extend the propositions to the latent space, z, we make additional assumptions about the input data x, y

and the latent variable, z, following Nguyen et al. (2022). The key assumption is that $I_{tr}(z, y) = I_{tr}(x, y)$ where $I(\cdot, \cdot)$ is mutual information. This is a reasonable assumption that the latent variable z is *sufficient* to solve the task and contains the same information about the output variable y as does the input variable x. As the latent representation z is learned while optimizing the downstream task objective, it must learn to retain information about the label. The remaining assumptions are weak assumptions about the data distribution and the standard covariate-shift assumption (all assumptions are detailed in Appendix D). Using these assumptions, Proposition 1 of Nguyen et al. (2022) can be re-written as follows:

$$l_{test} \leq l_{train} + \frac{M}{\sqrt{2}}[\mathrm{KL}[p_{te}(z)|p_{tr}(z)] + \frac{M}{4\sqrt{2}} \tag{5}$$

We have thus replaced x (input space) with z (latent space). The rest of our claims in Propositions 2.1 and 2.2 can then straightforwardly applied to the latent space rather than the input space. Uniform risk can be bounded using the latent feature representation space as follows,

$$\mathbb{E}_{p_{te} \sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[l_{test}] \leq l_{train} + \frac{M}{\sqrt{2}}\mathbb{E}_{p_{te}(z) \sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[\mathrm{KL}[p_{te}(z)|p_{tr}(z)]] + \frac{M}{4\sqrt{2}} \tag{6}$$

As above, we aim to minimize the expected divergence between the training and test feature distributions in Eq. 6. This leads us to our second key result: *uniform feature distributions minimize the expected divergence between training and test feature distributions* (Proposition 2.3).

Proposition 2.3. *The expected KL-divergence between train and test distributions in Eq. 6 is minimized when the training feature distribution $p_{tr}(z)$ is uniform:*

$$\arg\min_{p_{tr}(z)} \mathbb{E}_{p_{te}(z) \sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[\mathrm{KL}[p_{te}(z)|p_{tr}(z)]] = u^*_{tr}(z)$$

$$\textit{where } u^*_{tr}(z_i) = \frac{1}{Z}, \forall i \in \{1, \ldots, Z\}$$

Proof. Follows proof of Proposition 2.2. □

Hence, we have shown that uniformly distributed *feature* representations also lower *uniform risk*. This motivates our proposed method below that encourages deep neural networks to learn a feature representation space that is uniformly distributed for improved robustness (Section 2.6).

2.6 Encouraging Feature Representation Uniformity using Adversarial Distribution Matching

Motivated by Proposition 2.3, we perform URM by learning deep models with uniformly distributed feature representations. In order to encourage the distribution of deep feature vectors to match a uniform distribution, we use an adversarial training approach. Adversarial training is a useful approach to match distributions as it does not require significant modifications to the output of the encoder. Alternate approaches such as minimizing KL-divergence would require limiting the expressivity of the model to output a parametric distribution to analytically compute its divergence from a prior. Therefore, we train an additional *domain classifier* (*discriminator*) that is trained to differentiate between deep feature vectors and samples from a uniform noise distribution. The feature extractor or encoder is trained to fool this *domain classifier* by learning to produce feature vectors that are uniformly distributed. We use adversarial training to encourage the feature representation of deep neural networks to

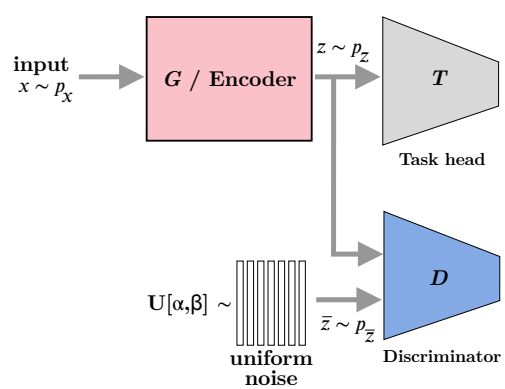

Figure 1: Overview of adversarial distribution matching to encourage the encoder to output uniformly distributed representations.

be uniformly distributed by additionally training a domain discriminator, D, to differentiate between encoded feature vectors and uniformly distributed random noise vectors. A deep encoder model, G, is trained to both fool D to classify encoded feature vectors, $z = G(x)$, as belonging to a uniform distribution while learning representations useful for the downstream task head T, typically a linear classifier (see Figure 1 for an overview). Our approach is similar to Adversarial Autoencoders (Makhzani et al., 2016) whose aim is to match the latent distribution of autoencoders to arbitrary prior distributions using adversarial training.

On the one hand, we want the discriminator D's predictions over *real* uniform noise to be accurate by maximizing $\mathbb{E}_{\bar{z}\sim p_u}[\log D(\bar{z})]$. On the other hand, given an encoded feature vector $G(x)$ the discriminator must output a probability, $D(G(x))$, close to zero by maximizing $\mathbb{E}_{x\sim p_x}[\log(1 - D(G(x)))]$. The encoder must be trained to fool D to produce a high probability for feature vectors by minimizing $\mathbb{E}_{x\sim p_x}[\log(1-D(G(x)))]$ or maximising $\mathbb{E}_{x\sim p_x}[\log D(G(x))]$. The combined objective function to update D and G represents a *minimax game* with the following loss function:

$$\min_G \max_D L(D, G) = \mathbb{E}_{\bar{z}\sim p_u}[\log D(\bar{z})] + \mathbb{E}_{z\sim p_z}[\log(1 - D(z)]$$

where $\mathbb{E}_{\bar{z}\sim p_u}[\log D(\bar{z})]$ is not used to update G. Generative adversarial networks (GAN) (Goodfellow et al., 2014) theory shows that the above objective minimizes the Jensen–Shannon Divergence between the encoded feature distribution and the uniform distribution when the discriminator is optimal. Let the θ be the parameters of the task model, which includes the encoder G as its backbone and a task head T such as a linear classifier i.e. $\theta = \{G, T\}$. $\mathcal{L}_T$ is the task-specific loss function. We can summarize the overall URM training objective for θ and D as follows:

$$\min_\theta \max_D \mathcal{L}(\theta, D) = \mathbb{E}_{(x,y)\sim p_{tr}}[\mathcal{L}_T(T(G(x)), y)] + \lambda \mathbb{E}_{x\sim p_{tr}}[\log(1 - D(G(x))] + \mathbb{E}_{\bar{z}\sim p_u}[\log D(\bar{z})]$$

The hyper-parameter λ determines the weight for training the encoder and discriminator to classify the feature outputs $z = G(x)$. A higher λ increases the strength of regularization at the expense of feature learning for the downstream task and lower λ weaken regularization but allow the encoder to pay more attention to the downstream task. The same λ is applied to both update the discriminator and encoder so that one does not overpower the other. The choice of activation function applied to the output of the encoder G can be changed for each task. We choose from either the hyperbolic Tangent (TanH) or ReLU activations. In case of TanH activations, the uniform noise distribution is $\bar{z} \sim \mathcal{U}(-1, 1)$ and in case of ReLU it is $\bar{z} \sim \mathcal{U}(0, 1)$. We use Leaky ReLUs in the discriminator to improve gradient flow to the generator. We train the encoder G and discriminator D alternately.

3 Experiments

3.1 Experimental Settings and Datasets

For group robustness, our experimental setup followed Yang et al. (2023). Specifically, we used the evaluation setting where group attributes are unknown in the training set. The *worst-class* accuracy is used to perform model selection, i.e., choice of the best checkpoint from each training run, as it was found to be an effective alternative in the absence of group-attributes by Yang et al. (2023). For all methods, group attributes in the validation set were used to select hyper-parameters as well as to fine-tune methods that require a group-balanced dataset e.g. DFR (Izmailov et al., 2022). We report *worst-group* and *balanced* (across classes) accuracies. All results were averaged across three random seeds. For domain generalization experiments, our experimental setup followed Gulrajani & Lopez-Paz (2021) using the *leave-one-domain-out* cross-validation scheme. We report the average accuracy across all test domains. For both tasks, we searched over sixteen random hyper-parameter combinations and applied adversarial distribution matching to the penultimate layer before the linear classification head. Hyper-parameter search ranges for URM are included in the Appendix (Table 4). For group robustness, we evaluate on Waterbirds (Wah et al., 2011), CelebA (Liu et al., 2015), CivilComments (Borkan et al., 2019) and MultiNLI (Williams et al., 2018). For Waterbirds and CelebA, we use the ResNet50 (He et al., 2016) architecture pretrained on ImageNet1k (Russakovsky

et al., 2015). For MultiNLI and CivilComments, we use BERT (Devlin et al., 2019) pretrained on Book Corpus and English Wikipedia data. For domain generalization (DG), we benchmark on ColoredMNIST (Arjovsky et al., 2020), RotatedMNIST (Ghifary et al., 2015), PACS (Li et al., 2017), VLCS (Fang et al., 2013), OfficeHome (Venkateswara et al., 2017) and TerraIncognita (Beery et al., 2018). For DG benchmarks, all methods used the ResNet50 (He et al., 2016) architecture pretrained on ImageNet1k (Russakovsky et al., 2015). Other training and dataset parameters are based on (Gulrajani & Lopez-Paz, 2021). We further describe all benchmark datasets in Appendix F. All experiments were run on a computer with 8 A100 GPUs.

3.2 Baselines

We consider baselines including ERM and various methods proposed for sub-population shifts and domain generalization. For sub-population shifts, we consider GroupDRO, LfF, JTT, LISA, DFR, Mixup, IRM, CORAL, MMD, DANN, C-DANN, ReSample, ReWeight, Focal Loss, CBLoss (Class-balanced loss), LDAM, Balanced Softmax (BSoftmax), CRT and ReWeightCRT. We follow Yang et al. (2023) for implementation of these baselines. For domain generalization, we consider Inter-domain Mixup (Mixup), MLDF, MTL, ARM, SagNet, RSC, and VREx. We follow Gulrajani & Lopez-Paz (2021) for implementation of these baselines. For a description of these baselines, please refer to our Appendix G. We note that URM is a generic robust representation learning method that does not require any knowledge of training domains or groups for both sub-population shifts and domain generalization.

3.3 URM Supports Class-balanced Training for Robustness under Label-shifts

As discussed in Section 2.4, when considering shifts in the class-distribution (*label-shifts*) alone, we demonstrate that URM, which corresponds to class-balancing, produces the best *balanced accuracy*. We consider the binary classification tasks in the Waterbirds and ColoredMNIST datasets. We trained models with different class-distributions and compute each model's *balanced accuracy*, which is the average of the class-specific accuracies on the test set. Balanced accuracy is a commonly used metric in imbalanced classification tasks. We train models with different values of p ($p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$), where p corresponds to the binary class distribution parameter (Table 1). Note that $p = 0.5$ corresponds to class-balanced training, which corresponds to URM for label shift scenarios. The URM model ($p = 0.5$) achieves the highest *balanced accuracy* on the test set (Table 1). *Balanced accuracy* deteriorates when the the training class distribution is highly imbalanced e.g. $p = 0.9$ or 0.1. As discussed in Section 2.4, *uniform risk* corresponds to balanced accuracy across classes in the label-shift scenario, hence URM (class-balancing) is a useful strategy in dealing with unexpected label shifts. Our results further support and provide a unified perspective on the common-practice of class-balanced training.

Table 1: *Balanced accuracies (%)* after training with different class distributions. Results are averaged across three random seeds.

Training	Waterbirds	ColoredMNIST
$p = 0.1$	85.7	55.3
$p = 0.2$	86.0	68.0
$p = 0.3$	85.3	76.5
$p = 0.4$	86.1	**76.9**
$p = 0.5$ (URM)	**86.8**	**76.9**
$p = 0.6$	85.7	76.5
$p = 0.7$	86.2	76.3
$p = 0.8$	86.4	75.7
$p = 0.9$	85.7	66.6

3.4 Uniformly Distributed Feature Representations Improve Sub-group Robustness

URM learns more group-robust representations compared to ERM on multiple datasets (Table 2). URM can also be easily combined with other methods as it is a generic representation learning method. Notably, URM, when combined with Deep Feature Reweighting (DFR), achieved the best worst-group accuracy on the Waterbirds and MultiNLI datasets (URM (DFR), Table 2). On MultiNLI, URM (DFR) achieved 4% higher worst-group accuracy than the existing best methods. URM also achieved the highest balanced accuracy on

Table 2: Worst-group and balanced accuracies on group robustness benchmarks. Results for other methods are from Yang et al. (2023).

Algorithm	Waterbirds		CelebA		CivilComments		MultiNLI	
	worst (%)	balanced (%)	worst (%)	balanced (%)	worst (%)	balanced (%)	worst (%)	balanced (%)
ERM	69.1 ±4.7	83.1 ±2.0	57.6 ±0.8	93.0 ±0.0	63.2 ±1.2	79.4 ±0.2	66.4 ±2.3	81.0 ±0.3
Mixup	77.5 ±0.7	88.9 ±0.2	57.8 ±0.8	93.1 ±0.1	65.8 ±1.5	79.7 ±0.0	66.8 ±0.3	81.7 ±0.1
GroupDRO	73.1 ±0.4	86.3 ±0.5	68.3 ±0.9	93.9 ±0.1	61.5 ±1.8	81.3 ±0.1	64.1 ±0.8	81.1 ±0.2
CVaRDRO	75.5 ±2.2	88.5 ±0.3	60.2 ±3.0	93.1 ±0.1	62.9 ±3.8	80.8 ±0.1	48.2 ±3.4	75.4 ±0.2
JTT	71.2 ±0.5	86.8 ±0.2	48.3 ±1.5	92.4 ±0.2	51.0 ±4.2	77.7 ±0.8	65.1 ±1.6	81.4 ±0.0
LfF	75.0 ±0.7	86.3 ±0.3	53.0 ±4.3	85.3 ±2.9	42.2 ±7.2	69.7 ±4.7	57.3 ±5.7	71.4 ±1.6
LISA	77.5 ±0.7	88.9 ±0.2	57.8 ±0.8	93.1 ±0.1	65.8 ±1.5	79.7 ±0.0	66.8 ±0.3	81.7 ±0.1
ReSample	70.0 ±1.0	85.0 ±0.2	**74.1 ±2.2**	93.8 ±0.1	61.0 ±0.6	80.7 ±0.1	66.8 ±0.5	81.5 ±0.0
ReWeight	71.9 ±0.6	86.2 ±0.1	69.6 ±0.2	94.0 ±0.1	59.3 ±1.1	81.3 ±0.0	64.2 ±1.9	79.4 ±0.2
SqrtReWeight	71.0 ±1.4	87.2 ±0.6	66.9 ±2.2	93.9 ±0.1	**68.6 ±1.1**	80.6 ±0.2	63.8 ±2.4	80.6 ±0.2
CBLoss	74.4 ±1.2	86.2 ±0.6	65.4 ±1.4	93.8 ±0.1	67.3 ±0.2	80.3 ±0.3	63.6 ±2.4	80.6 ±0.3
Focal	71.6 ±0.8	87.1 ±0.3	56.9 ±3.4	92.6 ±0.3	61.9 ±1.1	78.7 ±0.3	62.4 ±2.0	80.9 ±0.2
LDAM	70.9 ±1.7	86.0 ±0.2	57.0 ±4.1	93.2 ±0.2	28.4 ±7.7	69.5 ±3.2	65.5 ±0.8	80.9 ±0.1
BSoftmax	74.1 ±0.9	87.0 ±1.0	69.6 ±1.2	**94.2 ±0.1**	58.3 ±1.1	81.1 ±0.1	63.6 ±2.4	80.6 ±0.2
DFR	89.0 ±0.2	91.2 ±0.1	73.7 ±0.8	93.2 ±0.0	64.4 ±0.1	79.0 ±0.0	63.8 ±0.0	80.2 ±0.0
CRT	76.3 ±0.8	87.9 ±0.1	69.6 ±0.7	93.6 ±0.0	67.8 ±0.3	80.7 ±0.0	65.4 ±0.2	80.2 ±0.0
ReWeightCRT	76.3 ±0.2	88.0 ±0.2	70.7 ±0.6	93.6 ±0.0	64.7 ±0.2	80.7 ±0.0	65.2 ±0.2	80.2 ±0.0
URM	76.9 ±1.9	87.3 ±0.7	65.6 ±1.2	93.4 ±0.3	66.2 ±0.5	80.1 ±0.4	67.7 ±0.7	**81.8 ±0.1**
URM (DFR)	**89.7 ±0.2**	**91.7 ±0.1**	67.4 ±0.8	93.5 ±0.1	61.3 ±1.0	**81.5 ±0.0**	**70.7 ±0.9**	80.8 ±0.1

MultiNLI. The better worst-group accuracy achieved by URM (DFR) reflects the improved representation learned by URM in the absence of any group attribute information. We emphasise that URM is a generic robust representation learning method and is not specific to sub-group robustness like many of the other baseline methods.

3.5 Uniformly Distributed Feature Representations Improve Domain Generalization

URM improves the generalization performance of models in novel domains. URM outperformed competing methods in the RotatedMNIST and VLCS and PACS benchmarks (Table 3). Moreover, when combined with Inter-domain Mixup, URM achieved the best accuracy on the PACS dataset achieving 3.5% better accuracy than the best competing method, as well as the OfficeHome and Terraincognita benchmarks (Table 3). Notably, URM does not use any knowledge of the DG task and yet achieves competitive accuracy with methods that do. As URM is performed by learning a feature representation that is uniformly distributed over a bounded range e.g. $[-1, 1]^Z$, where Z is the dimensionality of the feature space, feature representations of test domain samples fall within the familiar feature space of the URM model, reducing the likelihood of unexpected outputs. The URM model sees the entire range of feature representations equally during training, helping generalization.

3.6 Visualizing Uniformity of Learned Feature Representations

Data distributions often have multiple peaks and troughs, and models trained on such distributions tend to be biased towards high-density regions (majority groups) since they contribute more to gradient updates during training. Consequently, the learned parameters are more optimized for these high-density regions, resulting in poorer performance on low-density regions (the troughs). URM addresses this by learning a feature space that is more uniformly distributed, effectively flattening the training feature distribution and reducing bias. This approach encourages the model to learn features that are more evenly distributed across both high- and low-density regions. To demonstrate this, we visualized the feature representations learned by URM compared to those learned by ERM using UMAP (McInnes et al., 2018) (Figure 2). We extracted feature vectors for test samples in the Waterbirds dataset using ERM and URM trained models. While UMAP does not provide a complete picture of the data due to significant dimensionality reduction, we observed that URM's features were significantly more uniformly distributed than ERM's. This suggests that

Table 3: Domain generalization accuracies using leave-one-domain-out validation. Results were averaged over three random seeds. Results for other methods are from Gulrajani & Lopez-Paz (2021).

Algorithm	ColoredMNIST	RotatedMNIST	VLCS	PACS	OfficeHome	TerraIncognita
ERM	36.7 ± 0.1	97.7 ± 0.0	77.2 ± 0.4	83.0 ± 0.7	65.7 ± 0.5	41.4 ± 1.4
IRM	40.3 ± 4.2	97.0 ± 0.2	76.3 ± 0.6	81.5 ± 0.8	64.3 ± 1.5	41.2 ± 3.6
GroupDRO	36.8 ± 0.1	97.6 ± 0.1	77.9 ± 0.5	83.5 ± 0.2	65.2 ± 0.2	44.9 ± 1.4
Mixup	33.4 ± 4.7	97.8 ± 0.0	77.7 ± 0.6	83.2 ± 0.4	67.0 ± 0.2	48.7 ± 0.4
MLDG	36.7 ± 0.2	97.6 ± 0.0	77.2 ± 0.9	82.9 ± 1.7	66.1 ± 0.5	46.2 ± 0.9
CORAL	39.7 ± 2.8	97.8 ± 0.1	78.7 ± 0.4	82.6 ± 0.5	68.5 ± 0.2	46.3 ± 1.7
MMD	36.8 ± 0.1	97.8 ± 0.1	77.3 ± 0.5	83.2 ± 0.2	60.2 ± 5.2	46.5 ± 1.5
DANN	**40.7 ± 2.3**	97.6 ± 0.2	76.9 ± 0.4	81.0 ± 1.1	64.9 ± 1.2	44.4 ± 1.1
CDANN	39.1 ± 4.4	97.5 ± 0.2	77.5 ± 0.2	78.8 ± 2.2	64.3 ± 1.7	39.9 ± 3.2
MTL	35.0 ± 1.7	97.8 ± 0.1	76.6 ± 0.5	83.7 ± 0.4	65.7 ± 0.5	44.9 ± 1.2
SagNet	36.5 ± 0.1	94.0 ± 3.0	77.5 ± 0.3	82.3 ± 0.1	67.6 ± 0.3	47.2 ± 0.9
ARM	36.8 ± 0.0	**98.1 ± 0.1**	76.6 ± 0.5	81.7 ± 0.2	64.4 ± 0.2	42.6 ± 2.7
VREx	36.9 ± 0.3	93.6 ± 3.4	76.7 ± 1.0	81.3 ± 0.9	64.9 ± 1.3	37.3 ± 3.0
RSC	36.5 ± 0.2	97.6 ± 0.1	77.5 ± 0.5	82.6 ± 0.7	65.8 ± 0.7	40.0 ± 0.8
URM	36.9 ± 0.2	**98.1 ± 0.2**	**84.3 ± 4.8**	**84.0 ± 0.2**	67.5 ± 0.5	48.3 ± 1.4
URM (Mixup)	32.4 ± 3.6	96.7 ± 0.4	77.1 ± 0.2	**87.2 ± 3.4**	**68.9 ± 0.6**	**49.3 ± 0.9**

URM achieves the training objective for the feature representations to be more uniformly distributed and contributes to the robustness of the model.

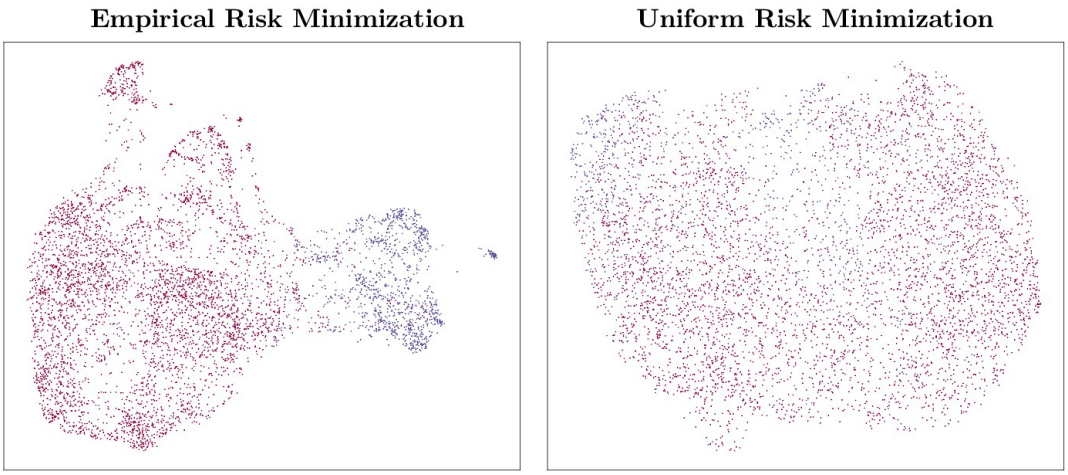

Figure 2: UMAP (McInnes et al., 2018) visualization of feature vectors generated by ERM and URM on Waterbirds dataset. URM generates a more uniformly distributed feature representation space compared to ERM.

4 Related Work

Group robustness: The topic of robustness to sub-population shifts has been increasingly studied in recent years as machine learning models are being deployed in real-world applications where we wish to avoid downstream performance biases. Many such methods involve training on a group balanced dataset with improved performance. As shown above, URM in fact supports this approach of training on uniform group distributions. For example, one of the existing best methods, DFR, retrains the linear classifier head on a group balanced dataset. However, group-attributes may not available if many practical scenarios. Hence,

a method such as URM that does not require group annotations would be useful. Moreover, as shown above, URM can also be combined with methods such as DFR for improved performance (Section 3.4).

Uniform priors: We note that our work is closely related to the ideas presented in uniform priors for data-efficient learning (Sinha et al., 2022). Compared to (Sinha et al., 2022), we proposed a novel learning principle and framework to motivate the use of uniform priors for robustness. In fact, our results may explain the empirical results of (Sinha et al., 2022). In other work, uniform priors have also been employed in self-supervised representation learning, both explicitly and implicitly (Assran et al., 2023).

Robust Representation Learning: URM is also related to the idea of state reification (Lamb et al., 2019), which aims to project out-of-distribution hidden states back on to the manifold of familiar hidden states from the training distribution. Rather than training additional networks to map out-of-distribution hidden states back on to the training data manifold, URM learns bounded feature spaces that are uniformly distributed. Hence, even out-of-distribution samples will fall in the familiar feature space of the model and is less likely to provide unexpected outputs. Other work has introduced input-space transformations that mitigate the effect of irrelevant input features to improve representation learning (Taghanaki et al., 2021).

5 Conclusion

We presented a novel learning principle called Uniform Risk Minimization (URM) and showed theoretically that uniform training data distributions are an optimal choice to lower *uniform risk*, a novel risk measure for ood robustness. Our results also provide a unified perspective on various existing algorithms that perform class-balanced or group-balanced training from the perspective of distribution shift. Finally, we proposed an empirical method to learn uniformly distributed feature representations in deep neural networks using adversarial distribution matching. We empirically demonstrated the effectiveness of our method to improve robustness to sub-population shifts as well as domain generalization, without knowledge of any privileged group or domain attribute information during training. Our work sheds light on the importance of the distribution of learned feature representations for downstream robustness and fairness. We hope our work encourages further research into uniform data distributions and priors for robust and fair representation learning.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization, March 2020. URL `http://arxiv.org/abs/1907.02893`. arXiv:1907.02893 [cs, stat].

Mido Assran, Randall Balestriero, Quentin Duval, Florian Bordes, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, and Nicolas Ballas. The hidden uniform cluster prior in self-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=04K3PMtMckp`.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in Terra Incognita. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, pp. 472–489, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01270-0.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, May 2010. ISSN 1573-0565. doi: 10.1007/s10994-009-5152-4. URL `https://doi.org/10.1007/s10994-009-5152-4`.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from Several Related Classification Tasks to a New Unlabeled Sample. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL `https://proceedings.neurips.cc/paper_files/paper/2011/file/b571ecea16a9824023ee1af16897a582-Paper.pdf`.

Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435. Publisher: JMLR.org.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, pp. 491–500, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 978-1-4503-6675-5. doi: 10.1145/3308560.3317593. URL https://doi.org/10.1145/3308560.3317593. event-place: San Francisco, USA.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, October 2018. ISSN 08936080. doi: 10.1016/j.neunet.2018.07.011. URL http://arxiv.org/abs/1710.05381. arXiv: 1710.05381.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/621461af90cadfdaf0e8d4cc25129f91-Paper.pdf.

Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie. Class-Balanced Loss Based on Effective Number of Samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9260–9269, Los Alamitos, CA, USA, June 2019. IEEE Computer Society. doi: 10.1109/CVPR.2019.00949. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00949.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Chen Fang, Ye Xu, and Daniel N. Rockmore. Unbiased Metric Learning: On the Utilization of Multiple Datasets and Web Images for Softening Bias. In *2013 IEEE International Conference on Computer Vision*, pp. 1657–1664, 2013. doi: 10.1109/ICCV.2013.208.

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. ISSN 1533-7928. URL http://jmlr.org/papers/v17/15-239.html.

M. Ghifary, W. Kleijn, M. Zhang, and D. Balduzzi. Domain Generalization for Object Recognition with Multi-task Autoencoders. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2551–2559, Los Alamitos, CA, USA, December 2015. IEEE Computer Society. doi: 10.1109/ICCV.2015.293. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.293. ISSN: 2380-7504.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

Sindhu C. M. Gowda, Shalmali Joshi, Haoran Zhang, and Marzyeh Ghassemi. Pulling Up by the Causal Bootstraps: Causal Data Augmentation for Pre-Training Debiasing. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pp. 606–616, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 978-1-4503-8446-9. doi: 10.1145/3459637.3482380. URL https://doi.org/10.1145/3459637.3482380. event-place: Virtual Event, Queensland, Australia.

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.

Ishaan Gulrajani and David Lopez-Paz. In Search of Lost Domain Generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=lQdXeXDoWtI.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness Without Demographics in Repeated Loss Minimization. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1929–1938. PMLR, July 2018. URL https://proceedings.mlr.press/v80/hashimoto18a.html. ISSN: 2640-3498.

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, June 2016. doi: 10.1109/CVPR.2016.90. ISSN: 1063-6919.

Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging Improves Cross-Domain Generalization. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, volume 12347, pp. 124–140. Springer International Publishing, Cham, 2020. ISBN 978-3-030-58535-8 978-3-030-58536-5. doi: 10.1007/978-3-030-58536-5_8. URL https://link.springer.com/10.1007/978-3-030-58536-5_8. Series Title: Lecture Notes in Computer Science.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Proceedings of the First Conference on Causal Learning and Reasoning*, pp. 336–351. PMLR, June 2022. URL https://proceedings.mlr.press/v177/idrissi22a.html. ISSN: 2640-3498.

Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew Gordon Wilson. On Feature Learning in the Presence of Spurious Correlations. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=wKhUPzqVap6.

Nathalie Japkowicz. The Class Imbalance Problem: Significance and Strategies. *Proceedings of the 2000 International Conference on Artificial Intelligence ICAI*, June 2000.

John C. Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, June 2021. doi: 10.1214/20-AOS2004. URL https://doi.org/10.1214/20-AOS2004.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling Representation and Classifier for Long-Tailed Recognition. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1gRTCVFvB.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-Distribution Generalization via Risk Extrapolation (REx). In *Proceedings of the 38th International Conference on Machine Learning*, pp. 5815–5826. PMLR, July 2021. URL https://proceedings.mlr.press/v139/krueger21a.html. ISSN: 2640-3498.

Alex Lamb, Jonathan Binas, Anirudh Goyal, Sandeep Subramanian, Ioannis Mitliagkas, Yoshua Bengio, and Michael Mozer. State-Reification Networks: Improving Generalization by Modeling the Distribution of Hidden Representations. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3622–3631. PMLR, May 2019. URL https://proceedings.mlr.press/v97/lamb19a.html. ISSN: 2640-3498.

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)*, pp. 5543–5551, Venice, October 2017. IEEE. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.591. URL http://ieeexplore.ieee.org/document/8237853/.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: meta-learning for domain generalization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018a. ISBN 978-1-57735-800-8. Place: New Orleans, Louisiana, USA.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain Generalization with Adversarial Feature Learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018b. doi: 10.1109/CVPR.2018.00566.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. doi: 10.1109/TPAMI.2018.2858826.

Evan Z. Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just Train Twice: Improving Group Robustness without Training Group Information. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 6781–6792. PMLR, July 2021. URL https://proceedings.mlr.press/v139/liu21f.html. ISSN: 2640-3498.

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. doi: 10.1109/ICCV.2015.425.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial Autoencoders. In *International Conference on Learning Representations*, 2016. URL http://arxiv.org/abs/1511.05644.

L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, February 2018. _eprint: 1802.03426.

Aditya Krishna Menon, Ankit Singh Rawat, and Sanjiv Kumar. Overparameterisation and worst-case generalisation: friend or foe? In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jphnJNOwe36.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain Generalization via Invariant Feature Representation. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 10–18, Atlanta, Georgia, USA, June 2013. PMLR. URL https://proceedings.mlr.press/v28/muandet13.html. Issue: 1.

H. Nam, H. Lee, J. Park, W. Yoon, and D. Yoo. Reducing Domain Gap by Reducing Style Bias. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8686–8695, Los Alamitos, CA, USA, June 2021. IEEE Computer Society. doi: 10.1109/CVPR46437.2021.00858. URL https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.00858.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from Failure: De-biasing Classifier from Biased Classifier. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20673–20684. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/eddc3427c5d77843c2253f1e799fe933-Abstract.html.

Junhyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. Spread Spurious Attribute: Improving Worst-group Accuracy with Spurious Attribute Estimation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_F9xpOrqyX9.

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.

Jiawei Ren, Cunjun Yu, shunan sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and hongsheng Li. Balanced Meta-Softmax for Long-Tailed Visual Recognition. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4175–4186. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/2ba61cc3a8f44143e1f2f13b2b729ab3-Abstract.html.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, December 2015. ISSN 1573-1405. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization. *arXiv:1911.08731 [cs, stat]*, April 2020. URL http://arxiv.org/abs/1911.08731. arXiv: 1911.08731.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M. Rush. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Hf3qXoiNkR.

Samarth Sinha, Karsten Roth, Anirudh Goyal, Marzyeh Ghassemi, Zeynep Akata, Hugo Larochelle, and Animesh Garg. Uniform Priors for Data-Efficient Learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4016–4027, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-66548-739-9. doi: 10.1109/CVPRW56347.2022.00447. URL https://ieeexplore.ieee.org/document/9857327/.

Baochen Sun and Kate Saenko. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In Gang Hua and Hervé Jégou (eds.), *Computer Vision – ECCV 2016 Workshops*, 2016.

Saeid A. Taghanaki, Kristy Choi, Amir Hosein Khasahmadi, and Anirudh Goyal. Robust Representation Learning via Perceptual Similarity Metrics. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 10043–10053. PMLR, July 2021. URL https://proceedings.mlr.press/v139/taghanaki21a.html. ISSN: 2640-3498.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Towards Debiasing NLU Models from Unknown Biases. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7597–7610, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.613. URL https://aclanthology.org/2020.emnlp-main.613.

Vladimir Vapnik. *Statistical Learning Theory*. Wiley, 1998. ISBN 978-0-471-03003-4.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep Hashing Network for Unsupervised Domain Adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5385–5394, 2017. doi: 10.1109/CVPR.2017.572.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-UCSD. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Yufei Wang, Haoliang Li, and Alex C. Kot. Heterogeneous Domain Generalization Via Domain Mixup. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3622–3626, 2020. doi: 10.1109/ICASSP40776.2020.9053273.

Adina Williams, Nikita Nangia, and Samuel Bowman. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

Yadollah Yaghoobzadeh, Soroush Mehri, Remi Tachet des Combes, T. J. Hazen, and Alessandro Sordoni. Increasing Robustness to Spurious Correlations using Forgettable Examples. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 3319–3332, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.291. URL https://aclanthology.org/2021.eacl-main.291.

Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change is Hard: A Closer Look at Subpopulation Shift. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 39584–39622. PMLR, July 2023. URL https://proceedings.mlr.press/v202/yang23s.html.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving Out-of-Distribution Robustness via Selective Augmentation. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 25407–25437. PMLR, June 2022. URL https://proceedings.mlr.press/v162/yao22b.html. ISSN: 2640-3498.

Haoran Zhang, Amy X. Lu, Mohamed Abdalla, Matthew McDermott, and Marzyeh Ghassemi. Hurtful Words: Quantifying Biases in Clinical Contextual Word Embeddings. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, CHIL '20, pp. 110–120, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 978-1-4503-7046-2. doi: 10.1145/3368555.3384448. URL https://doi.org/10.1145/3368555.3384448. event-place: Toronto, Ontario, Canada.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain Adaptation under Target and Conditional Shift. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. III–819–III–827. JMLR.org, 2013. Place: Atlanta, GA, USA.

Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive Risk Minimization: Learning to Adapt to Domain Shift. In *Advances in Neural Information Processing Systems*, volume 34, pp. 23664–23678. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/c705112d1ec18b97acac7e2d63973424-Abstract.html.

Zhilu Zhang and Mert Sabuncu. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/f2925f97bc13ad2852a7a551802feea0-Abstract.html.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6): 1452–1464, 2018. doi: 10.1109/TPAMI.2017.2723009.

A Proof of Proposition 2.1

Proof.

$$
\begin{aligned}
l_{test} &\leq l_{train} + \frac{M}{\sqrt{2}}\sqrt{\mathrm{KL}[p_{te}(x,y)|p_{tr}(x,y)]} && \text{(using Proposition 1 of (Nguyen et al., 2022))} \\
&\leq l_{train} + \frac{M}{\sqrt{2}}[\mathrm{KL}[p_{te}(x,y)|p_{tr}(x,y)] + \frac{1}{4}] && \text{(square the square root}^2\text{)} \\
&= l_{train} + \frac{M}{\sqrt{2}}\mathrm{KL}[p_{te}(x,y)|p_{tr}(x,y)] + \frac{M}{4\sqrt{2}} && (7) \\
&= l_{train} + \frac{M}{\sqrt{2}}\mathbb{E}_{p_{te}(x,y)}[\log p_{te}(x) + \log p_{te}(y|x) - \log p_{tr}(x) - \log p_{tr}(y|x)] + \frac{M}{4\sqrt{2}} && (8) \\
&= l_{train} + \frac{M}{\sqrt{2}}\mathbb{E}_{p_{te}(x,y)}[\log p_{te}(x) - \log p_{tr}(x) \\
&\qquad + \mathbb{E}_{p_{te}(x,y)}[\log p_{te}(y|x) - \log p_{tr}(y|x)]] + \frac{M}{4\sqrt{2}} && (9) \\
&= l_{train} + \frac{M}{\sqrt{2}}\mathbb{E}_{p_{te}(x)}[\log p_{te}(x) - \log p_{tr}(x)] \\
&\qquad + \frac{M}{\sqrt{2}}\mathbb{E}_{p_{te}(x)}\left[\mathbb{E}_{p_{te}(y|x)}[\log p_{te}(y|x) - \log p_{tr}(y|x)]\right] + \frac{M}{4\sqrt{2}} && (10) \\
&= l_{train} + \frac{M}{\sqrt{2}}[\mathrm{KL}[p_{te}(x)|p_{tr}(x)] + \mathbb{E}_{p_{te}(x)}\left[\mathrm{KL}[p_{te}(y|x)|p_{tr}(y|x)]\right]] + \frac{M}{4\sqrt{2}} && (11) \\
&= l_{train} + \frac{M}{\sqrt{2}}[\mathrm{KL}[p_{te}(x)|p_{tr}(x)] + \frac{M}{4\sqrt{2}} && \{\text{covariate-shift}\} \quad (12)
\end{aligned}
$$

□

Using the *covariate-shift* assumption, the second KL divergence term in Eq. 11 involving the conditional probabilities $p(y|x)$ equals 0 and can be dropped. This is a reasonable assumption as generalization to distributions different from the training distribution is challenging if the labeling mechanism changes (Ben-David et al., 2010). Hence, we can make the common assumption that $\mathrm{KL}[p_{te}(y|x)|p_{tr}(y|x)] = 0$. We can then focus on the divergence between the marginal distributions i.e. $\mathrm{KL}[p_{te}(x)|p_{tr}(x)]$.

B Proof of Proposition of 2.2

Proof.

$$
\begin{aligned}
&\arg\min_{p_{tr}(x)} \mathbb{E}_{Dir(\boldsymbol{\alpha}=\mathbb{1})}[\mathrm{KL}[p_{te}(x)|p_{tr}(x)]] && (13) \\
&= \arg\min_{p_{tr}(x)} \mathbb{E}_{Dir(\boldsymbol{\alpha}=\mathbb{1})}[\sum_{i=1}^{N} p_{te}(x_i) ln(\frac{p_{te}(x_i)}{p_{tr}(x_i)})] && (14) \\
&= \arg\min_{p_{tr}(x)} \mathbb{E}_{Dir(\boldsymbol{\alpha}=\mathbb{1})}[\sum_{i=1}^{N} p_{te}(x_i) ln(p_{te}(x_i)) - p_{te}(x_i) ln(p_{tr}(x_i))] && (15) \\
&= \arg\min_{p_{tr}(x)} \mathbb{E}_{Dir(\boldsymbol{\alpha}=\mathbb{1})}[\sum_{i=1}^{N} p_{te}(x_i) ln(p_{te}(x_i)) - \sum_{i=1}^{N} p_{te}(x_i) ln(p_{tr}(x_i))] && (16) \\
&= \arg\min_{p_{tr}(x)} \mathbb{E}_{Dir(\boldsymbol{\alpha}=\mathbb{1})}[\sum_{i=1}^{N} p_{te}(x_i) ln(p_{te}(x_i))] - \mathbb{E}_{Dir(\boldsymbol{\alpha}=\mathbb{1})}[\sum_{i=1}^{N} p_{te}(x_i) ln(p_{tr}(x_i))] && (17)
\end{aligned}
$$

[2]Note that KL-divergence is non-negative. $\frac{1}{4}$ is added as a buffer in case the KL-divergence is less than 1, as $\forall x >= 0, x + \frac{1}{4} \geq \sqrt{x}$.

$$= \arg\min_{p_{tr}(x)} \sum_{i=1}^{N} \mathbb{E}_{Dir(\boldsymbol{\alpha}=\mathbb{1})}[p_{te}(x_i) ln(p_{te}(x_i))] - \sum_{i=1}^{N} \mathbb{E}_{Dir(\boldsymbol{\alpha}=\mathbb{1})}[p_{te}(x_i) ln(p_{tr}(x_i))] \tag{18}$$

$$= \arg\min_{p_{tr}(x)} - \sum_{i=1}^{N} \mathbb{E}_{Dir(\boldsymbol{\alpha}=\mathbb{1})}[p_{te}(x_i) ln(p_{tr}(x_i))] \qquad \text{(remove terms independent of } p_{tr}(x_i))$$

$$= \arg\min_{p_{tr}(x)} - \sum_{i=1}^{N} ln(p_{tr}(x_i)) \mathbb{E}_{Dir(\boldsymbol{\alpha}=\mathbb{1})}[p_{te}(x_i)] \qquad \text{(expectation of Dirichlet dist. is } \tfrac{\alpha_i}{\alpha_0})$$

$$= \arg\min_{p_{tr}(x)} - \sum_{i=1}^{N} ln(p_{tr}(x_i)) [\frac{\alpha_i}{\alpha_0}] \qquad (\alpha_i = 1,\ \alpha_0 = \text{N})$$

$$= \arg\min_{p_{tr}(x)} - \frac{1}{N} \sum_{i=1}^{N} ln(p_{tr}(x_i)) \tag{19}$$

$$\therefore \arg\min_{p_{tr}(x)} \mathbb{E}_{Dir(\boldsymbol{\alpha}=\mathbb{1})}[\mathrm{KL}[p_{te}(x)|p_{tr}(x)]] = \arg\min_{p_{tr}(x)} - \frac{1}{N} \sum_{i=1}^{N} ln(p_{tr}(x_i)) \tag{20}$$

$$\text{subject to } \sum_{i=1}^{N} p_{tr}(x_i) = 1$$

The Lagrangian for this constrained optimization problem is:

$$\mathcal{L}[p_{tr}(x_1), \ldots, p_{tr}(x_N), \lambda] = -\frac{1}{N} \sum_{i=1}^{N} ln(p_{tr}(x_i)) + \lambda(\sum_{i=1}^{N} p_{tr}(x_i) - 1) \tag{21}$$

$$\text{solve } \nabla_{p_{tr}(x_1),\ldots,p_{tr}(x_N),\lambda} \mathcal{L}[p_{tr}(x_1), \ldots, p_{tr}(x_N), \lambda] = 0$$

$$\Rightarrow p_{tr}(x_i) = \frac{1}{N\lambda}, \forall i \in \{1, \ldots, N\}$$

$$\Rightarrow \lambda = 1, \because \sum_{i=1}^{N} p_{tr}(x_i) = 1 \text{ (constraint)} \tag{22}$$

$$\Rightarrow p_{tr}(x_i) = \frac{1}{N}, \forall i \in \{1, \ldots, N\} \text{ (i.e. uniform distribution)} \tag{23}$$

$$\therefore \arg\min_{p_{tr}(x)} \mathbb{E}_{p_{te} \sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[\mathrm{KL}[p_{te}(x)|p_{tr}(x)]] = u^*_{tr}(x) \tag{24}$$

$$\text{where } u^*_{tr}(x_i) = \frac{1}{N}, \forall i \in \{1, \ldots, N\}.$$

□

C Proof of Proposition of E.1

Proof.

$$l_{test} \leq l_{train} + \frac{M}{\sqrt{2}} \sqrt{\mathrm{KL}[p_{te}(x,y)|p_{tr}(x,y)]} \qquad \text{(using Proposition 1 of (Nguyen et al., 2022))}$$

$$\leq l_{train} + \frac{M}{\sqrt{2}} [\mathrm{KL}[p_{te}(x,y)|p_{tr}(x,y)] + \frac{1}{4}] \qquad \text{(square the square root}^3\text{)}$$

$$= l_{train} + \frac{M}{\sqrt{2}} \mathrm{KL}[p_{te}(x,y)|p_{tr}(x,y)] + \frac{M}{4\sqrt{2}} \tag{25}$$

$$= l_{train} + \frac{M}{\sqrt{2}} \mathbb{E}_{p_{te}(x,y)}[\log p_{te}(y) + \log p_{te}(x|y) - \log p_{tr}(y) - \log p_{tr}(x|y)] + \frac{M}{4\sqrt{2}} \tag{26}$$

$$
\begin{aligned}
&= l_{train} + \frac{M}{\sqrt{2}}\mathbb{E}_{p_{te}(x,y)}[\log p_{te}(y) - \log p_{tr}(y) \\
&\qquad + \mathbb{E}_{p_{te}(x,y)}[\log p_{te}(x|y) - \log p_{tr}(x|y)]] + \frac{M}{4\sqrt{2}} && (27)\\
&= l_{train} + \frac{M}{\sqrt{2}}\mathbb{E}_{p_{te}(y)}[\log p_{te}(y) - \log p_{tr}(y)] \\
&\qquad + \frac{M}{\sqrt{2}}\mathbb{E}_{p_{te}(y)}\left[\mathbb{E}_{p_{te}(x|y)}[\log p_{te}(x|y) - \log p_{tr}(x|y)]\right] + \frac{M}{4\sqrt{2}} && (28)\\
&= l_{train} + \frac{M}{\sqrt{2}}[\mathrm{KL}[p_{te}(y)|p_{tr}(y)] + \mathbb{E}_{p_{te}(y)}\left[\mathrm{KL}[p_{te}(x|y)|p_{tr}(x|y)]\right]] + \frac{M}{4\sqrt{2}} && (29)\\
&= l_{train} + \frac{M}{\sqrt{2}}[\mathrm{KL}[p_{te}(y)|p_{tr}(y)] + \frac{M}{4\sqrt{2}} \qquad \{\text{target-shift}\} && (30)
\end{aligned}
$$

□

In Eq. 29, we assume that the conditional distribution of $p(x|y)$ does not change and that only the label distribution $p(y)$ has shifted. This assumes that the generative process $p(x|y)$ has not changed at test time. This is referred to *target shift* in the domain adaptation literature (Zhang et al., 2013). Hence, we can drop the second term involving the conditional misalignment is zero and focus on the target shift in Eq. 30. Based on Eq. 30, we can similarly show that the expected test loss under target shift is minimized by the uniform training class-distribution.

D Extending Propositions 2.1 and 2.2 to Deep Feature Layers

Here we extend Propositions 2.1 and 2.2 to feature spaces such as those learned by a deep neural network. This will be useful below when we wish to apply the above results to a method that modifies the feature space to be uniformly distributed in order to improve robustness. Let G be a model parameterized by a deep neural network that outputs a latent variable z i.e. $z = G(x)$, which can then be used for a downstream task such as classification. To extend the propositions to the latent space, z, we make additional assumptions about the input data x, y and the latent variable, z:

Assumption D.1. $I_{tr}(z, y) = I_{tr}(x, y)$, where $I(\cdot, \cdot)$ is mutual information, defined as:

$$
I_{tr}(z,y) = \mathbb{E}_{p_{tr}(z,y)}\left[\log \frac{p_{tr}(z,y)}{p_{tr}(z)p_{tr}(y)}\right]; \quad I_{tr}(x,y) = \mathbb{E}_{p_{tr}(x,y)}\left[\log \frac{p_{tr}(x,y)}{p_{tr}(x)p_{tr}(y)}\right] \tag{31}
$$

This is a reasonable assumption that the latent variable z is *sufficient* to solve the task and contains the same information about the output variable y as does the input variable x. As the latent representation z is learned while optimizing the task objective, it must learn to retain information about the label.

Assumption D.2. $p_{tr}(y|x) = \mathbb{E}_{p(z|x)}[p_{tr}(y|z)], \quad \forall x, y \in \mathcal{X}, \mathcal{Y}$

This assumption simply allows us to re-write the true predictive distribution $p_{tr}(y|x)$ as a conditional expectation over the latent variable z. When this assumption holds, the predictive distribution, $\hat{p}(y|x) = \mathbb{E}_{p(z|x)}[\hat{p}(y|z)]$, will approximate $p_{tr}(y|x)$ when $\hat{p}(y|z) = p_{tr}(y|z)$, which is a goal of training.

Assumption D.3. $\frac{p_{te}(x,y)}{p_{tr}(x,y)} < \infty, \quad \forall x, y \in \mathcal{X}, \mathcal{Y}$

This assumption is satisfied when train and test distributions have the same support set $\mathcal{X}, \mathcal{Y}$. This is a reasonable assumption for sub-population shift scenarios when the test distribution is assumed to comprise the same groups as the training distribution but in different proportions.

Assumption D.4. $\mathrm{KL}[p_{te}(y|x)|p_{tr}(y|x)] = 0$

This is the covariate-shift assumption.

[3]Note that KL-divergence is non-negative. $\frac{1}{4}$ is added as a buffer in case the KL-divergence is less than 1, as $\forall x >= 0, x + \frac{1}{4} \geq \sqrt{x}$.

Using these four assumptions, Proposition 1 of (Nguyen et al., 2022) can be re-written as follows (see (Nguyen et al., 2022) for proofs):

$$l_{test} \leq l_{train} + \frac{M}{\sqrt{2}}[\mathrm{KL}[p_{te}(z)|p_{tr}(z)] + \frac{M}{4\sqrt{2}} \tag{32}$$

We have thus replaced x (input space) with z (latent space). The rest of our claims in Propositions 2.1 and 2.2 can then straightforwardly applied to the latent space rather than the input space.

E Uniform Risk for Label Shifts and Sub-population Shifts

In the context of sub-population and label shifts, *uniform risk* can be shown to be the balanced risk across groups or classes, respectively.

Let $\mathcal{R}_g$ be the group-conditional risk computed over each group-specific distribution $p_g \coloneqq p(x, y|g)$ i.e. $\mathcal{R}_g \coloneqq \mathbb{E}_{p_g}[l(x, y)]$. In the case of *label-shift*, the group can be treated as a class instead. Assume that the distribution shift is strictly over the group or class distribution.

$$\begin{aligned}
\mathcal{R}_U &\coloneqq \mathbb{E}_{p_{te} \sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[l_{\text{test}}] \\
&= \mathbb{E}_{p_{te} \sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[\mathbb{E}_{p_g}[\mathcal{R}_g]] \\
&= \mathbb{E}_{p_{te} \sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[\sum_{g \in \mathcal{G}} \beta_g \mathcal{R}_g], \text{where } \beta \in \Delta_{|\mathcal{G}|} \\
&= \sum_{g \in \mathcal{G}} \mathcal{R}_g \mathbb{E}_{p_{te} \sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[\beta_g] \\
&= \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \mathcal{R}_g && (\textit{balanced risk})
\end{aligned}$$

Thus, *uniform risk* is the balanced risk of a model over groups or classes (Eq. *balanced risk*) in the context of sub-population or label shifts. However, in the sub-population shift literature, authors generally focus on the worst-group performance. The limitations of over-emphasis on the *worst-group accuracy* metric has been reported in prior work (Section 5.5 of Yang et al. (2023)). In contrast, *uniform risk* provides a balanced risk measure that equally weighs all groups or classes.

E.1 Class Balancing Lowers Uniform Risk

Here we discuss the common practice of class-balancing datasets from the perspective of *uniform risk* and robustness to *label* shifts at test time. In Eq. 7 of the Proof of Proposition 2.1, we can alternatively decompose the KL-divergence between the joint training and test distributions $p(x, y)$ using the marginal misalignment in $p(y)$ and the conditional misalignment in $p(x|y)$. We can then similarly show that training on data with a uniform *class or label* distribution minimizes the upper bound on *uniform risk*, as pertaining to the *label-shift* problem.

Proposition E.1. *If the loss $-\log \hat{p}(y|x)$ is bounded by M and the generative process, $p(x|y)$, is the same between train and test distributions*[4]*, we have:*

$$l_{test} \leq l_{train} + \frac{M}{\sqrt{2}}\mathrm{KL}[p_{te}(y)|p_{tr}(y)] + \frac{M}{4\sqrt{2}} \tag{33}$$

Proof. Proof in Appendix C. □

[4]This is known as the *label* or *target shift* assumption in the distribution-shift literature.

The expectation of Eq. 33 with respect to a uniform prior distribution over test class-distributions is:

$$\mathbb{E}_{p_{te}\sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[l_{test}] \leq l_{train} + \frac{M}{\sqrt{2}}\mathbb{E}_{p_{te}(y)\sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[\mathrm{KL}[p_{te}(y)|p_{tr}(y)]] + \frac{M}{4\sqrt{2}} \tag{34}$$

We can now aim to minimize the upper bound on the expectation of test-loss under label-shift i.e. *uniform risk* for label-shifts, by minimizing the expected KL-divergence between train and test class distributions. Let C be the number of classes in the classification task. The proof follows the same steps as the proof of Proposition 2.2 but for the class-distribution p(y) instead of p(x).

Proposition E.2. *The expected KL-divergence between train and test class distributions in Eq. 4 is minimized when the training class distribution $p_{tr}(y)$ is uniform:*

$$\underset{p_{tr}(y)}{\arg\min}\, \mathbb{E}_{p_{te}(y)\sim Dir(\boldsymbol{\alpha}=\mathbb{1})}[\mathrm{KL}[p_{te}(y)|p_{tr}(y)]] = u^*_{tr}(y) \tag{35}$$

$$\textit{where } u^*_{tr}(y_i) = \frac{1}{C}, \forall i \in \{1, \ldots, C\}.$$

Proof. Same as proof of Proposition 2.2. □

Therefore our results support the common practice of class-balancing from the perspective of robustness under label shifts. Indeed, prior work has demonstrated the efficacy of balancing classes during training (Buda et al., 2018). We further experimentally demonstrate this below (Section 3.3).

F Datasets

We describe each dataset used in our group robustness benchmarks below.

- **Waterbirds (Wah et al., 2011).** Waterbirds is a binary classification image dataset containing spurious correlations, constructed by placing images from the Caltech-UCSD Birds-200-2011 (CUB) dataset (Wah et al., 2011) over backgrounds from the Places dataset (Zhou et al., 2018). The task is to classify landbirds from waterbirds and the spurious attribute is the background (water or land). As most images of waterbirds have a water background and most images of landbirds have a land background, models may latch on to the spurious correlation between the background and the type of bird.

- **CelebA (Liu et al., 2015).** CelebA is a binary classification image dataset. The task is to predict hair color from images of celebrity faces (blond *vs.* non-blond), where the spurious correlation is gender. As most blond haired people in this dataset are female and most non-blond haired people are male, this creates a spurious correlation between gender and hair color.

- **CivilComments (Borkan et al., 2019).** CivilComments is a binary classification text dataset where models must predict whether an internet comment contains toxic language. The spurious attribute is the presence of references to eight demographic identities (*male, female, LGBTQ, Christian, Muslim, other religions, Black,* and *White*).

- **MultiNLI (Williams et al., 2018).** MultiNLI is a text classification dataset with 3 classes and the target is the natural language inference relationship between the premise and the hypothesis (neutral, contradiction, or entailment). The spurious attribute is the presence of negation in the text, as negation is highly correlated with the contradiction label.

Domain Generalization Datasets: We benchmarked on multiple challenging DG datasets. ColoredMNIST (Arjovsky et al., 2020) is a variant of the MNIST digit recognition dataset where each domain contains a disjoint set of digits colored either red or blue. The binary label is a noisy function of digit and color, such that color is correlated with the label to varying degree in each domain and the digit bears correlation 0.75 with the label. RotatedMNIST (Ghifary et al., 2015) is also a variant of MNIST where each domain contains

digits rotated to different degrees (0, 15, 30, 45, 60, 75 and contains 10 classes. PACS (Li et al., 2017) contains four diverse domains (art, cartoons, photos, sketches) and 7 classes. VLCS (Fang et al., 2013) contains four photographic domains (Caltech101, LabelMe, SUN09, VOC2007) and 5 classes. OfficeHome (Venkateswara et al., 2017) also includes four domains (art, clipart, product, real) and 65 classes. TerraIncognita (Beery et al., 2018) comprises photographs of animals taken by camera traps at four different locations.

G Baselines

Vanilla Training: Empirical Risk Minimization (ERM, (Vapnik, 1998)), minimizes the errors across all training samples. **Subgroup Robust Methods:** Group distributionally robust optimization (GroupDRO) (Sagawa et al., 2020) performs ERM while emphasising sub-populations or domains with high losses during training. CVaRDRO (John C. Duchi & Hongseok Namkoong, 2021) is a variant of GroupDRO that up-weights training samples with the highest losses. LfF (Nam et al., 2020) trains two models where the first model is biased and the second one is debiased using a re-weighted objective. Just train twice (JTT) (Liu et al., 2021) first trains a standard ERM model to identify minority sub-populations in the dataset and then trains another ERM model while up-weighting minority samples. LISA (Yao et al., 2022) trains invariant predictors using data interpolation within and across attributes. Deep feature re-weighting (DFR) (Izmailov et al., 2022) first trains an ERM model and then retrains only the last layer of the model using a dataset balanced among different sub-populations. **Data Augmentation Method:** Mixup (Zhang et al., 2018) trains on linear interpolations of randomly sampled training data points and their labels. **Domain-Invariant Representation Learning Methods:** Invariant risk minimization (IRM) (Arjovsky et al., 2020) learns a feature space such that the optimal linear classifier is the same across domains. Deep correlation alignment (CORAL) (Sun & Saenko, 2016) matches the second order moments of the feature distributions. Maximum mean discrepancy (MMD) (Li et al., 2018b) matches the MMD (Gretton et al., 2012) of feature distributions across domains. Domain Adversarial Neural Networks (DANN, (Ganin et al., 2016)) employs adversarial training to match feature distributions across domains. Class-conditional DANN (C-DANN, (Li et al., 2018b)) is similar to DANN as it matches the class-conditional feature distribution across domains. **Imbalanced Learning Methods:** ReSample (Japkowicz, 2000) and ReWeight (Japkowicz, 2000) simply re-sample or re-weight the inputs according to the number of samples per class. Focal loss (Focal) (Lin et al., 2020) reduces the relative loss for well-classified samples and focuses on difficult samples. Class-balanced loss (CBLoss) (Cui et al., 2019) proposes re-weighting by the inverse effective number of samples. The LDAM loss (LDAM) (Cao et al., 2019) employs a modified marginal loss that favors minority samples more. Balanced-Softmax (BSoftmax) (Ren et al., 2020) extends Softmax to an unbiased estimation that considers the number of samples in each class. Classifier re-training (CRT) (Kang et al., 2020) decomposes the representation and classifier learning into two stages, where it fine-tunes the classifier using class-balanced sampling with representation fixed in the second stage. ReWeightCRT (Kang et al., 2020) is a re-weighting variant of CRT.

Domain Generalization Baselines: Inter-domain Mixup (Mixup, (Zhang et al., 2018; Wang et al., 2020)) trains on linear iterpolations between samples from different domains. Meta-Learning for Domain Generalization (MLDG, (Li et al., 2018a)) uses MAML to meta-learn generalizing across domains. Marginal Transfer Learning (MTL, (Blanchard et al., 2011; 2021)) estimates a mean embedding for each domain that is passed as a second argument to the model. Adaptive Risk Minimization (ARM, (Zhang et al., 2021)) extends MTL using a separate embedding model. Style-Agnostic Networks (SagNet, (Nam et al., 2021)) trains models by keeping image content and randomizing style. Representation Self Challenging (RSC, (Huang et al., 2020)) learns robust models by iteratively pruning the most activated features. Risk Extrapolation (VREx, (Krueger et al., 2021)) approximates IRM using a variance penalty.

H Training Hyper-parameters

Table 4: Search ranges for URM-specific hyper-parameters. Other training parameters ranges are based on (Yang et al., 2023).

Method	**Parameter**	**Random Distribution**
URM	lambda λ	$\text{Uniform}(0, 0.2)$
	generator output	$\text{RandomChoice}([TanH, ReLU])$
	number of layers in discriminator	$\text{RandomChoice}([1, 2, 3])$
	learning rate (for image or tabular datasets)	$10^{\text{Uniform}(-5,-3)}$
	optimizer (for image or tabular datasets)	$\text{RandomChoice}([SGD])$
	learning rate (text datasets)	$10^{\text{Uniform}(-6,-5)}$
	optimizer (text datasets)	$\text{RandomChoice}([AdamW])$