LEARNING REPRESENTATIONS THAT SUPPORT ROBUST TRANSFER OF PREDICTORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Ensuring generalization to unseen environments remains a challenge. Domain shift can lead to substantially degraded performance unless shifts are well-exercised within the available training environments. We introduce a simple robust estimation criterion – transfer risk – that is specifically geared towards optimizing transfer to new environments. Effectively, the criterion amounts to finding a representation that minimizes the risk of applying any optimal predictor trained on one environment to another. The transfer risk essentially decomposes into two terms, a direct transfer term and a weighted gradient-matching term arising from the optimality of per-environment predictors. Although inspired by IRM, we show that transfer risk serves as a better out-of-distribution generalization criterion, both theoretically and empirically. We further demonstrate the impact of optimizing such transfer risk on two controlled settings, each representing a different pattern of environment shift, as well as on two real-world datasets. Experimentally, the approach outperforms baselines across various out-of-distribution generalization tasks.

1 INTRODUCTION

Training and test examples are rarely sampled from the same distribution in real applications. Indeed, training and test scenarios often represent somewhat different domains. Such discrepancies can degrade generalization performance or even cause serious failures, unless specifically mitigated. For example, standard empirical risk minimization approach (ERM) that builds on the notion of matching training and test distributions rely on statistically informative but non-causal features such as textures (Geirhos et al., 2019), background scenes (Beery et al., 2018), or word co-occurrences in sentences (Chang et al., 2020).

Learning to generalize to domains that are unseen during training is a challenging problem. One approach to domain generalization or out-of-distribution generalization is based on reducing variation due to sets or environments one has access to during training. For example, one can align features of different environments (Muandet et al., 2013; Sun & Saenko, 2016) or use data-augmentation to help prevent overfitting to environment-specific features (Carlucci et al., 2019; Zhou et al., 2020). At one extreme, domain adaptation assumes access to unlabeled test examples whose distribution can be then matched in the feature space (*e.g.*, (Ganin et al., 2016)).

More recent approaches build on causal invariance as the foundation for out-of-distribution generalization. The key assumption is that the available training environments represent nuisance variation, realized by intervening on non-causal variables in the underlying Structural Causal Model (Pearl, 2000). Since causal relationships can be assumed to remain invariant across the training environments as well as any unseen environments, a number of recent approaches (Peters et al., 2015; Arjovsky et al., 2019; Krueger et al., 2020) tailor their objectives to remove spurious (non-causal) features specific to training environments.

In this paper, we propose a simple robust criterion termed Transfer Risk Minimization (TRM). The goal of TRM is to directly translate model's ability to generalize across environments into a learning objective. As in prior work, we decompose the model into a feature mapping and a predictor operating on the features. Our transfer risk in this setting measures the average risk of applying the optimal predictor learned in one environment to examples

from another adversarially chosen environment. The feature representation is then tailored to support such robust transfer. Although our work is greatly inspired by IRM (Arjovsky et al., 2019), we show that TRM serves as a better out-of-distribution criterion with both empirical and theoretical analysis in non-linear case. We further show that the TRM objective decomposes into two terms, direct transfer term and a weighted gradient-matching term with connections to meta-learning. We then propose an alternating updating algorithm for optimizing TRM.

To evaluate robustness we introduce two patterns of environment shifts based on 10C-CMNIST and SceneCOCO datasets. We construct these controlled settings so as to exercise different combinations of invariant and non-causal features, highlighting the impact of non-causal features in the training environments. In the absence of non-causal confounders, we show that all the methods achieve decent out-of-distribution generalization. When non-causal features are present, however, TRM offers greater robustness against biased training environments. We further demonstrate that our approach leads to good performance on the two real-world datasets, PACS and Office-Home.

2 Background and related works

Domain generalization Machine learning models trained with Empirical Risk Minimization may not perform well in unseen environments where examples are sampled from a distribution different from training. The problem is known as out-of-distribution generalization or domain generalization (Blanchard et al., 2011; Muandet et al., 2013). A number of recent approaches have been proposed in this context. We only touch some of them for brevity. A typical approach to out-of-distribution generalization involves (distributionally) aligning training environments (Muandet et al., 2013; Ganin et al., 2016; Sun & Saenko, 2016; Li et al., 2018b; Shi et al., 2021). Related approaches such as Nam et al. (2019) encourage the model to focus more on shapes via style adversarial learning, adopt data augmentations (Carlucci et al., 2019; Zhou et al., 2020) or meta-learning (Li et al., 2018a).

Causal invariance A recent line of work focuses on promoting invariance as a way to isolate causally meaningful features. Ideally, one would specify a structural equation model (Pearl, 2000), expressing direct and indirect causes, distinguishing them from spurious, environment specific influences that are unlikely to generalize (Peters et al., 2015; Rojas-Carulla et al., 2018; Müller et al., 2020). Invariance serves as a statistically more amenable proxy criterion towards identifying causally relevant features for predictors. Arjovsky et al. (2019) proposed *invariant risk minimization* over feature-predictor decompositions. The main idea is that the predictor operating on causal features can be assumed to be simultaneously optimal across training environments. A number of related approaches have been proposed. For example, Krueger et al. (2020) uses variance of losses as regularization, Jin et al. (2020) minimizes the regret loss induced by held-out environments and Parascandolo et al. (2020) aligns gradient signs across environments by and-mask.

Distributionally robust optimization (DRO) DRO specifies a minimax criterion for estimating predictors where an adversary gets to modify the training distribution. The allowed modifications are typically expressed in terms of divergence balls around the training distribution (Ben-Tal et al., 2013; Duchi et al., 2016; Esfahani & Kuhn, 2018). Closer to our work, Group DRO (Hu et al., 2018; Sagawa et al., 2019) defines uncertainty regions in terms of a simplex over (fixed) training groups. Both DRO and Group DRO minimize the worst-case loss of the predictor within the uncertain regions. Unlike these methods, we use a predictor-representation decomposition, and define a regularizer over the representation using a minimax criterion. Moreover, we explicitly measure the risk of a predictor trained in one environment but applied to another.

3 Transfer Risk Minimization

Consider a classification problem from input space $\mathcal{X}$ (e.g. images) to output space $\mathcal{Y}$ (e.g. labels). We are given E training environments $\Omega = \{P_1(\mathcal{X} \times \mathcal{Y}), \ldots, P_E(\mathcal{X} \times \mathcal{Y})\}$, where P_i is the empirical distribution for environment i. We decompose our model into two parts: feature extractor $\Phi : \mathcal{X} \to \mathcal{Z}$, which maps the input to a feature representation, and predictor $w : \mathcal{Z} \to \mathcal{Y}$ that operates on the features to realize the final output. We call their concatenation

$w \circ \Phi$ as a classifier. We use $\ell(w \circ \Phi(x); y)$ to denote the cross-entropy loss on a training point $(x, y) \in \mathcal{X} \times \mathcal{Y}$. As a shorthand, the expected loss with respect to a distribution P is given as $\mathbb{E}_P[\ell(w \circ \Phi)]$. The broader goal is to learn a pair of feature extractor and predictor that minimize the risk on some unseen environment $\hat{P}$:

$$\mathcal{R}(\Phi, w) = \mathbb{E}_{\hat{P}}[\ell(w \circ \Phi)]$$

As a step towards this goal, we learn a predictively robust model (Φ, w) across the available training environments (defined later). While the high level aim here resembles invariant risk minimization (Arjovsky et al., 2019), our proposed estimation criterion is based on robustness rather than invariance.

3.1 Estimation criterion

We define group (environment) robustness based on exchangeability of predictors. Specifically, we require that environment-specific predictors w generalize also to other training environments. Note that this doesn't imply that a single predictor is per-environment optimal or invariant as in IRM. Instead, our representation Φ aims to minimize *transfer risk* across a set of training environments Ω

$$\mathcal{R}(\Phi; \Omega) = \sum_{Q \in \Omega} \left(\sup_{P \in \text{Conv}(\Omega \backslash Q)} \mathbb{E}_P[\ell(w(Q; \Phi) \circ \Phi)] \right) \tag{1}$$

where $w(Q; \Phi) = \arg\min_w \mathbb{E}_Q[\ell(w \circ \Phi)]$ refers to the optimal predictor w with respect to distribution Q. $\text{Conv}(\Omega \backslash Q) = \left\{ \sum_{P_i \in \Omega \backslash Q} \alpha_i(Q) P_i \mid \alpha_i(Q) \geq 0, \parallel \alpha(Q) \parallel_1 = 1 \right\}$ is the convex hull of environment specific distributions, excluding Q. Unlike methods in the DRO family (Ben-Tal et al., 2013; Sagawa et al., 2019) that do not decompose the predictors, the robust estimation criterion here is specifically tailored to measure the goodness of features in terms of their ability to permit generalization across the environments. We will show in later sections that transfer risk (Eq. (1)) indeed ensures better out-of-distribution generalization.

Remark We introduced transfer risk in Eq. (1) as a "sum-sup" criterion with respect to outer and inner terms. Other possible versions with similar estimation consequences include sum-sum, *i.e.*, $\mathcal{R}(\Phi; \Omega) = \sum_{Q \in \Omega} \left(\sum_{P \in \text{Conv}(\Omega \backslash Q)} \mathbb{E}_P[\ell(w(Q; \Phi) \circ \Phi)] \right)$. Note that the criterion still measures whether the feature representation allows a predictor trained in one environment to generalize to another. We expect this version to behave similarly when training environments have comparable noise levels, complexities. However, sum-sum version can be more resistant to environmental outliers.

3.2 Comparison with IRM

IRM (Arjovsky et al., 2019) is a popular objective for learning features that are invariant across training environments. Specifically, IRM finds a feature extractor such that the associated predictor is simultaneously optimal for every training environment. In our notation

$$\text{(IRM)} \quad \min_{\Phi, w} \sum_{P \in \Omega} \mathbb{E}_P[\ell(w \circ \Phi)] \qquad \text{subject to } w \in \arg\min_w \mathbb{E}_P[\ell(w \circ \Phi)], \forall P \in \Omega$$

IRM specifies a more restrictive set of admissible feature extractors Φ than transfer risk. Specifically, per-environment optimal predictors in IRM must agree (contain a common predictor) whereas transfer risk uses the per-environment optimal predictor to guide the representation learning. Due to the difficulty of solving the IRM bi-leveled optimization problem, Arjovsky et al. (2019) introduced a relaxed objective called IRMv1 where the constraints are replaced by gradient penalties:

$$\text{(IRMv1)} \quad \min_{\Phi, w} \quad \sum_{P \in \Omega} \mathbb{E}_P[\ell(w \circ \Phi)] + \lambda \parallel \nabla_w \mathbb{E}_P[\ell(w \circ \Phi)] \parallel^2 \tag{2}$$

To compare with IRM, we use the theoretical framework in Rosenfeld et al. (2020). For each environment, the data are defined by the following process: the binary label y is sampled

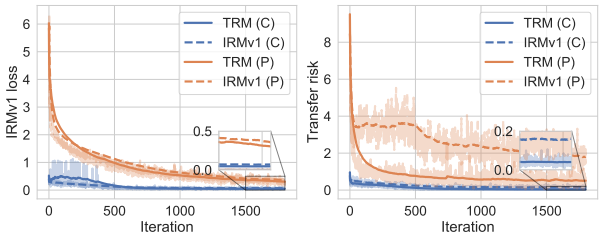


Figure 1: IRMv1 loss (Left) and transfer risk (Right) versus training iterations for models trained by IRMv1 and TRM on 10C-CMNIST (C) and PACS (P).

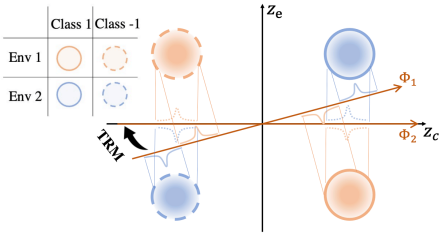


Figure 2: 2-d scenario of the linear case. TRM drives the non-invariant Φ_1 toward the invariant Φ_2.

uniformly from $\{\pm 1\}$ and the environmental features $[z_c, z_e]$ are sampled subsequently from label-conditioned Gaussians:

$$z_c \sim \mathcal{N}(y * \mu_c, \sigma_c^2 I); z_e \sim \mathcal{N}(y * \mu_i, \sigma_e^2 I), i \in \{1, \ldots, E\}$$

with $\mu_c \in \mathbb{R}^{d_c}, \mu_e \in \mathbb{R}^{d_e}$. The invariant feature mean μ_c remains the same for all environments while non-causal means μ_is vary across environments. The observation x is generated as a function of the latent features: $x = f(z_c, z_e)$, where f is a injective function that maps low dimensional features to high dimensional observations x.

Theorem 3.3 in Rosenfeld et al. (2020) shows that for non-linear f, there exists a non-linear classifier (Φ, w) that has nearly optimal IRMv1 loss. In addition, it is equivalent to ERM solution on nearly all test points when the non-causal mean in the test environment is sufficiently different from those in training. Below we show that TRM can avoid the failure mode of IRM.

Theorem 1 (Informal)**.** *Under some mild assumptions, there exists a classifier that achieves near-optimal IRMv1 loss (Eq. (2)) and has high transfer risk (Eq. (1)). In addition, for any test environment with a non-causal mean far from those in training, this classifier behaves like an ERM-trained classifier on most fractions of the test distribution.*

We defer the formal statement and proof to Appendix A.1. We prove the above theorem by constructing a classifier *only* using invariant features for prediction on the high-density region but behaving like ERM solution on the tails, which can still have near-optimal IRMv1 loss. However, the per-environment optimal predictors are distinct when using ERM-solution on the tails. The discrepancy in the per-environment optimal predictors leads to large transfer risk.

In addition to the theoretical analysis, we provide further empirical analysis to characterize the difference between TRM and IRM. Fig. 1 reports the IRMv1 loss and transfer risk on the 10C-CMNIST (C) and PACS (P) datasets discussed later (section 5). Although IRMv1 solutions achieve small IRMv1 losses, it has significantly higher transfer risks than TRM solutions. Conversely, TRM solutions have slightly lower IRMv1 loss than IRMv1 solutions. Besides, the out-of-distribution test accuracies on 10C-CMNIST / PACS are: $57\%/73\%$ (IRMv1), $57\%/74\%$ (ERM) and $78\%/81\%$ (TRM). IRMv1 solutions have close performance to ERM solutions, while TRM outperforms others by a large margin. The empirical results support the statement in Theorem 1 that models with near-optimal IRMv1 loss can have large transfer risks and behave like ERM solutions on test environments.

Together, these results suggest that transfer risk is a better criterion than IRMv1 for assessing model's out-of-distribution generalization. In Fig. 2, we demonstrate the effect of TRM on a a toy 2-d example ($d_c = d_e = 1$) with linear f, Φ. TRM drives the non-invariant feature to the invariant one. We defer details of the analysis in the 2-d case to Appendix A.2.

4 Method

In this section, we discuss how to optimize the TRM objective (Eq. (1)). We first introduce an exponential gradient ascent algorithm for optimizing the inner supremum, and then

discuss how to optimize the feature extractor with per-environment optimal predictor. An alternating updating algorithm incorporating these steps is summarized in Algorithm 1.

4.1 TRANSFER RISK OPTIMIZATION

Solving the inner sup Assume E different environments with associated densities $\Omega = \{P_1, P_2, \ldots, P_E\}$. Given Q, we find the corresponding worst-case environment P in the inner max of Eq. (1). The search space for P is the convex hull of all environment distributions with the exception of Q: $P(Q) \in \text{Conv}(\Omega \backslash Q) = \left\{ \sum_{P_i \in \Omega \backslash Q} \alpha_i(Q) P_i \mid \alpha_i(Q) \geq 0, \| \alpha(Q) \|_1 = 1 \right\}$. Since the optimization is over a simplex, the solution can be found exactly by just selecting the worst environment in Ω: $P = \arg\max_{P \in \Omega} \mathbb{E}_P[\ell(w(Q) \circ \Phi)]$. Empirically, we find that updating α by gradient ascent instead of selecting the worst environment leads to a more stable training process. This has been observed in related contexts (Sagawa et al., 2019).

The gradient for α_i is $\mathbb{E}_{P_i}[\ell(w(Q) \circ \Phi)]$, indicating that the inner supremum simply up-weights the environments with larger losses relative to the predictor $w(Q)$. We adopt an exponential gradient ascent algorithm (EG) for the updates:

$$\alpha_i(Q) = \text{EG}(\alpha(Q), \eta_\alpha, \mathcal{L}(Q))_i = \alpha_i(Q) \exp(\eta_\alpha \frac{\partial \mathcal{L}(Q)}{\partial \alpha_i(Q)}) / \sum_{P_i \in \Omega \backslash Q} \alpha_i(Q) \exp(\eta_\alpha \frac{\partial \mathcal{L}(Q)}{\partial \alpha_i(Q)}) \quad (3)$$

where η_α is the learning rate, and the subscript i denotes the ith component of the vector.

Updating the feature extractor Φ Given Q and the corresponding worst-case environment $P(Q)$, we consider here how to update the feature extractor Φ in Eq. (1). Denote the risk of using predictor $w(Q; \Phi)$ with data distribution $P(Q)$ as $\mathcal{L}_P(Q) = \mathbb{E}_{P(Q)}[\ell(w(Q; \Phi) \circ \Phi)]$. Recall that the optimal predictor $w(Q; \Phi)$ is an implicit function of the feature extractor Φ, *i.e.*, $w(Q; \Phi) = \arg\min_w \mathbb{E}_Q[\ell(w \circ \Phi)]$. In the remainder, we use a shorthand $w(Q)$ to refer to the predictor $\texttt{sg}(w(Q; \Phi))$, where sg stands for the `stop_gradient` operator. In other words, the value of $w(Q)$ follows Φ but it's partial derivatives w.r.t Φ are set to zero. It is helpful to distinguish $w(Q)$ from $w(Q; \Phi)$ to clarify what is meant by the different expressions below. Now, the full gradient of the transfer risk w.r.t Φ comprises two terms since $w(Q; \Phi)$ also depends on Φ

$$\frac{\mathrm{d}\mathcal{L}_P(Q)}{\mathrm{d}\Phi} = \underbrace{\frac{\partial \mathcal{L}_P(Q)}{\partial \Phi}}_{\text{direct gradient}} + \underbrace{(\frac{\partial \mathcal{L}_P(Q)}{\partial w(Q; \Phi)})^T \frac{\mathrm{d}w(Q; \Phi)}{\mathrm{d}\Phi}}_{\text{implicit gradient}} \quad (4)$$

We show in Proposition 1 that the implicit gradient can be further simplified as a weighted gradient-matching term.

Proposition 1. *Denote the Hessian as $H_{w(Q)} = \frac{\partial^2 \mathbb{E}_Q[\ell(w(Q) \circ \Phi)]}{\partial w(Q)^2}$. Suppose the loss $\ell(w \circ \Phi)$ is continuously differentiable and $H_{w(Q)}$ is non-singular, then we have:*

$$(\frac{\partial \mathcal{L}_P(Q)}{\partial w(Q; \Phi)})^T \frac{\mathrm{d}w(Q; \Phi)}{\mathrm{d}\Phi} = -\frac{\partial (\texttt{sg}(v_Q)^T \frac{\partial \mathbb{E}_Q[\ell(w(Q) \circ \Phi)]}{\partial w(Q)})}{\partial \Phi}$$

where $v_Q = (\frac{\partial \mathcal{L}_P(Q)}{\partial w(Q)})^T H_{w(Q)}^{-1}$, and is treated as a constant vector in the above equation (note the use of stop-gradient version $w(Q)$).

We can interpret the numerator of RHS as a gradient-matching objective: it measures the similarity of gradients depending on whether the distribution over which the loss is measured is Q or $P(Q)$, weighted by the Hessian inverse $H_{w(Q)}^{-1}$. It shows that TRM naturally aims to find a representation Φ where the gradients are matched when moving from Q to P (cf. (Shi et al., 2021)). By integrating, we can write down an objective function whose gradient with respect to Φ matches Eq. 4:

$$\int \frac{\mathrm{d}\mathcal{L}_P(Q)}{\mathrm{d}\Phi} \mathrm{d}\Phi = \underbrace{\mathbb{E}_P[\ell(w(Q) \circ \Phi)]}_{\text{direct transfer term}} - \underbrace{\texttt{sg}(v_Q)^T \frac{\partial \mathbb{E}_Q[\ell(w(Q) \circ \Phi)]}{\partial w(Q)}}_{\text{weighted gradient-matching term}} + \underbrace{C}_{\text{constant term}} \quad (5)$$

Note the use of stop-gradient versions $w(Q)$ in these expressions. Effectively, the TRM objective for Φ decomposes into two terms: (i) the direct transfer term, which encourages predictor $w(Q)$ to do well even if the distribution were $P(Q)$, and (ii) the weighted gradient-matching term. The second term attempts to match the gradient of w in the original environment Q and worst-case environment $P(Q)$ by updating the features. The weighted gradient-matching term plays a role analogous to meta-learning (Li et al., 2018a; Shi et al., 2021) encouraging simultaneous loss descent. Note that the weighted gradient-matching term actually evaluates to zero at the current value of Φ since $w(Q)$ is set to the per-environment optimal value, but the gradient of this term with respect to Φ is not zero.

4.2 Approximation of Inverse Hessian Vector Product

We denote the number of parameters in w as P and the gradient $\frac{\partial \mathcal{L}_P(Q)}{\partial w(Q)}$ as v. Dropping the $w(Q)$ subscript for clarity, the weight gradient matching term in Eq. (5) involves the computation of inverse Hessian vector product $v_Q = H^{-1}v$. For minibatch data $\mathcal{B}(x)$, computing the inverse Hessian H^{-1} requires $O(|\mathcal{B}(x)|P^2 + P^3)$ operations. To avoid heavy computation, we use the similar approach in Agarwal et al. (2017) to get good approximations by Taylor expansion and efficient Hessian-vector product (HVP) (Pearlmutter, 1994). Let $H_j^{-1} = \sum_{i=0}^{j}(I - H)^i$ be the first j terms in the Taylor expansion of H^{-1}. Note that $\lim_{j\to\infty} H_j^{-1} = H^{-1}$. We can solve the corresponding matrix vector product $H_j^{-1}v = \sum_{i=0}^{j}(I - H)^i v$ in linear time by recursively computing $(I - H)^i v$ with fast HVP. These computation are easy to implement in auto-grad systems like PyTorch (Paszke et al., 2019).

4.3 Algorithm

In addition to the TRM objective in Eq. (1), we include standard ERM term $\mathbb{E}_Q[\ell(w_{all} \circ \Phi)]$ for updating the predictor w_{all} on the top of features. Overall, given the distributions $Q, P(Q)$, the per-environment objective for updating the (Φ, w_{all}) pair consists of three terms:

$$\mathcal{R}(\Phi, w_{all}; Q) = \mathbb{E}_Q[\ell(w_{all} \circ \Phi)] + \mathbb{E}_P[\ell(w(Q) \circ \Phi)] - \lambda \mathtt{sg}(v_Q)^T \frac{\partial \mathbb{E}_Q[\ell(w(Q) \circ \Phi)]}{\partial w(Q)} \tag{6}$$

where λ is a hyper-parameter for adjusting the gradient-matching term to have the same gradient magnitude as the other terms. We interleave gradient updates on the model parameters (Φ, w_{all}) and the environmental weights $\{\alpha(Q) \mid Q \in \Omega\}$, as shown in Algorithm 1.

Algorithm 1 TRM algorithm

Input: Inital model parameters Φ, w_{all}, learning rates $\eta_\phi, \eta_w, \eta_\alpha$. and environment set Ω
for $t = 1$ **to** T **do**
  Randomly pick a environment $Q \in \Omega$
  Get the optimal $w(Q)$ on Q
  Update the model parameters:
    $\Phi^t \leftarrow \Phi^{t-1} - \eta_\phi \nabla \mathcal{R}(\Phi, w_{all}; Q)$, $w_{all}^t \leftarrow w_{all}^{t-1} - \eta_w \nabla \mathcal{R}(\Phi, w_{all}; Q)$
  Update the environmental weights by Eq. (3):
    $\alpha(Q) \leftarrow \mathrm{EG}(\alpha(Q), \eta_\alpha, \mathcal{L}(Q))$
end for

Algorithm 1 updates $\alpha(Q)$ in an online manner. With some convexity, boundness, and smoothness assumptions, we can prove that the on-line updating has a convergence rate of $\mathcal{O}(1/\sqrt{T})$ by using the techniques in Nemirovski et al. (2009). We defer more discussions to Appendix B.3.1.

5 Experiments

In our experiments, we focus on the out-of-distribution generalization tasks. We first evaluate our method on two synthesized datasets (10C-CMNIST, SceneCOCO). We simulate three

(a) Label-correlated shift (b) Combined shift

Figure 3: Visualization of the training environment 1 (**top**) and the test environment (**bottom**) of 10C-CMNIST and SceneCOCO on (a) Label-correlated shift and (b) Combined shift.

kinds of domain shifts by controlled experiments. Next, we evaluate all the methods on two real-world datasets (PACS, Office-Home). We compare **TRM** with standard empirical risk minimization (**ERM**), and recent methods developed for out-of-distribution generalization: **IRM** (Arjovsky et al., 2019), **REx** (Krueger et al., 2020), **GroupDRO** (Sagawa et al., 2019), **MLDG** (Li et al., 2018a) and **Fish** (Shi et al., 2021). We also use ERM trained with data sampled from the test domain to serve as an upper bound (**Oracle**).

Experiments in the main body use training-domain validation sets for hyper-parameter selection, which are arguably more practical for out-of-distribution generalization task (Ahmed et al., 2021; Gulrajani & Lopez-Paz, 2020; Krueger et al., 2020). We defer the results of the test-domain validation set to Appendix C. We also show the efficacy of TRM on group distributional robustness in Appendix C.4.

5.1 Experiments on 10C-CMNIST and SceneCOCO

5.1.1 Datasets

Evaluating the out-of-distribution generalization performance in an unambiguous manner necessitates controlled experiments. We synthesize the data by three latent features: (i) invariant (causal) feature, (ii) non-causal feature, which is spuriously correlated with labels and (iii) the dummy feature, which is not predictive of the labels. We conduct the controlled experiments on two synthetic datasets:

10C(lasses)-C(olored)MNIST is a more general 10 classes version of the 2-classes ColoredMNSIT (Arjovsky et al., 2019). We add the digit colors and background colors to allow for the domain shifts. Specifically, we set the invariant/non-causal/dummy features to digit/digit color/background color respectively. We randomly select ten colors as the digit colors and five colors as the background colors. 10C-CMNIST contains 60000 datapoints of dimension (3,28,28) from 10 digit classes.

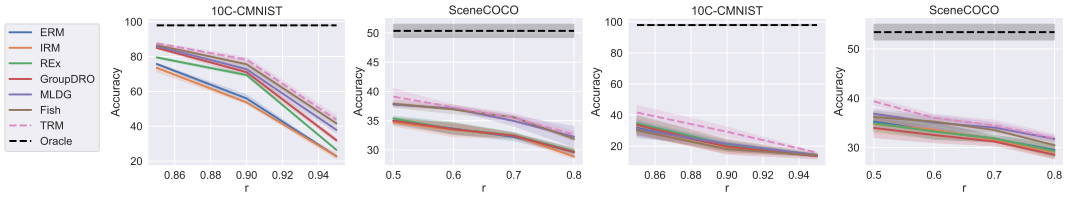
SceneCOCO superimposes the objects from the COCO datasets (Lin et al., 2014) on the background scenes from the Places datasets (Zhou et al., 2018). Following Ahmed et al. (2021), we select 10 objects and 10 scenes from above two datasets. We set the invariant/non-causal/dummy features to object/background scene/object color. This dataset consists of 10000 datapoints of dimension $(3, 64, 64)$ from 10 object classes.

In addition, we define a measurement of the correlations between the label and the non-causal features. Note that there is a one-to-one corresponding between the label and the non-causal features, *e.g.*, "2" $\leftrightarrow$ "blue digit color" in 10C-CMNIST and "boat" $\leftrightarrow$ "beach scene" in SceneCOCO. For each environment, we define the *bias degree* to be the ratio of the data that obeys this relationship. Those data which don't follow this relationship are then assigned with random non-causal features. In each training environment, the data is generated by environmental-specific combination of features and bias degree. This setting is commonly adopted in existing literature (Arjovsky et al., 2019; Krueger et al., 2020; Ahmed et al., 2021). The label y is set as the class where the invariant feature lies.

5.1.2 Controlled Scenarios

Next, we consider two scenarios with distinct combinations of latent features.

Label-correlated shift In this scenario, each training environment is assigned with a different non-zero bias degree. The dummy feature is set to a constant, *e.g.* black background color in 10C-CMNIST. The bias degree is set to zero in the test environment for evaluating how much extent the model has learn the invariant feature.



(a) Label-correlated shift (b) Combined shift

Figure 4: Test accuracy on 10C-CMNIST and SceneCOCO datasets in (a) Label-correlated shift and (b) Combined shift, using various bias degrees r.

Combined shift In this scenario, training environments have varying non-zero bias degrees and prior distributions of dummy feature. The test environment is unbiased. It simulates the joint effects of the shifts of non-causal and dummy features.

Fig. 3 visualizes the label-correlated and combined shifts. We use two training environments in the controlled experiments. To better evaluate the robustness of algorithms, we vary the bias degrees in the second training environment. We use $100\%/r\%$ bias degrees in 10C-CMNIST for the first/second training environment respectively, and $90\%/r\%$ in SceneCOCO. $r\%$ is ranging from 85% to 95% in 10C-CMNIST and 50% to 80% in SceneCOCO. We use different configurations for the two datasets because SceceCOCO is more complex. The biased degree is 0% in the test environment.

We adopt a 4-layer CNN/Wide ResNet (Zagoruyko & Komodakis, 2016) as the feature extractor for 10C-CMNIST/SceneCOCO, following prior work (Ahmed et al., 2021). We train for 10/100 epochs on 10C-CMNIST/SceneCOCO, both using batch size 128 and SGD with 0.1 initial learning rate and 0.9 momentum. For more details of datasets, hyperparameter selection and training, please refer to Appendix B.

5.1.3 Results

In Fig. 4, we report the accuracy on the test environment under label-correlated shift and combined shift. The x-axis in Fig. 4 stands for the varying biased degree $r\%$ of the second training environment. We observe a consistent performance drop of all the methods as the training environments become more biased. Our main finding is that the TRM algorithm achieves a better test accuracy at most bias degrees on both datasets and competitive performance with Fish and MLDG on SceneCOCO when the bias degrees are large. The results show that the model trained by the TRM algorithm depends more on the invariant features to make predictions.

We also observe that non-causal features combined with the distribution shifts of dummy features degrade the performance of all the methods (Combined shift). In Appendix C.1.3, we show that all the methods have similar performance to the Oracle when only changing the dummy feature distribution. The experiments suggest that when non-causal features exist, distribution shifts on dummy features can further hurt the out-of-distribution generalization. Besides, we show in Appendix C.1.2 that TRM-trained models transfer faster to target environments with limited data for fine-tuning.

5.2 Experiments on PACS and Office-Home

5.2.1 Setups

We further evaluate our methods on two real-world datasets, PACS and Office-Home.

PACS (Li et al., 2017) comprises four environmental data, namely arts, cartoons, photos, and sketches. This dataset contains 9991 datapoints of dimension (3,224,224) from 7 classes.

Office-Home (Venkateswara et al., 2017) includes four environments, namely art, clipart, product and real. It contains 15588 datapoints of dimension (3,224,224) from 65 classes.

These two datasets are widely used in domain generalization literature. We follow the standard valuation protocol (Li et al., 2017), which reports the test accuracy on each hold-out environment when training on the other three environments. We use the ImageNet-pretrained ResNet18 as the backbone of feature extractors. We use the SGD optimizer with a momentum of 0.9, a weight decay of 1e-4, and a fixed learning rate of 1e-4. The batch size is set to 32.

5.2.2 RESULTS

In Table 1 and 2, we report the test accuracy on PACS and Office-Home dataset. The proposed TRM algorithm achieves superior average accuracy on these datasets. We observe that TRM has better generalization ability on most hold-out environments in the two datasets, except the Photo environment, where all the methods have comparable performance. Further, we test TRM without the weighted gradient-matching term (TRM w/ GM) by setting $\lambda = 0$. TRM w/ GM still outperforms other baselines with only the direct transfer term. We also show in Appendix C.2 that TRM consistently improves over other methods with different architectures and validation set configurations.

Table 1: Test accuracy on PACS dataset

Algorithm	Art	Cartoon	Photo	Sketch	Average
ERM	73.7 ± 1.0	65.7 ± 2.3	94.8 ± 0.7	62.5 ± 2.4	74.1
IRM	73.3 ± 1.0	65.5 ± 1.9	94.7 ± 0.6	62.9 ± 1.5	74.1
REx	74.0 ± 1.0	66.8 ± 2.5	94.6 ± 0.8	63.7 ± 2.8	74.8
GroupDRO	74.1 ± 0.8	67.3 ± 2.0	94.7 ± 0.7	63.5 ± 3.4	74.9
MLDG	76.0 ± 1.8	69.2 ± 0.9	**95.0 ± 0.4**	64.3 ± 3.4	76.4
Fish	75.0 ± 1.2	69.0 ± 1.7	94.7 ± 0.5	64.3 ± 1.6	76.0
TRM w/ GM	77.5 ± 2.3	**70.3 ± 0.9**	94.4 ± 0.7	65.5 ± 1.7	76.9
TRM	**80.6 ± 2.1**	68.7 ± 1.4	93.7 ± 1.5	**67.1 ± 2.5**	**77.5**

Table 2: Test accuracy on Office-Home dataset

Algorithm	Art	Clipart	Product	Real	Average
ERM	51.1 ± 0.4	42.5 ± 0.7	65.5 ± 0.1	68.4 ± 0.7	56.8
IRM	51.6 ± 0.2	42.5 ± 0.6	65.2 ± 0.1	68.1 ± 0.7	56.8
REx	50.9 ± 1.3	42.3 ± 0.6	65.1 ± 0.4	68.1 ± 0.7	56.6
GroupDRO	51.2 ± 0.6	42.1 ± 0.9	64.9 ± 0.4	67.9 ± 0.6	56.5
MLDG	52.3 ± 0.3	44.1 ± 0.7	66.9 ± 0.8	69.4 ± 0.5	58.2
Fish	51.5 ± 0.5	43.2 ± 0.9	66.5 ± 0.3	69.1 ± 0.1	57.6
TRM w/ GM	**54.0 ± 0.5**	46.4 ± 0.9	67.9 ± 1.0	69.4 ± 1.0	59.4
TRM	53.9 ± 0.2	**46.4 ± 0.9**	**68.9 ± 1.1**	**69.6 ± 0.8**	**59.7**

6 CONCLUSION

The discrepancy between the training and test domain can degrade the performance of algorithms developed for the i.i.d setting. We propose a robust criterion termed Transfer Risk Minimization (TRM) to tackle the out-of-distribution problem. The transfer risk promotes the transferability of the per-environment predictors. The feature representation updates accordingly to support such transfer. We demonstrate that TRM better recovers the weights associated with the invariant features by an illustrative example. Due to the optimality of the per-environment predictor, TRM objective naturally decomposes into two terms, the direct transfer term and the weighted-gradient matching term. One limitation of TRM is that the inverse Hessian vector product can have a large variance with a small batch size. The better optimization of the weighted gradient-matching term is left for future work.

Experimentally, we test our approach on several controlled experiments. We show that TRM achieves better out-of-distribution performance under different combinations of features. We also demonstrate the effectiveness of TRM on the PACS and Office-Home datasets.

REFERENCES

Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine learning in linear time. *J. Mach. Learn. Res.*, 18:116:1–116:40, 2017.

Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron Courville. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=b9PoimzZFJ.

Martín Arjovsky, L. Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *ArXiv*, abs/1907.02893, 2019.

Sara Beery, Grant Van Horn, and P. Perona. Recognition in terra incognita. In *ECCV*, 2018.

A. Ben-Tal, D. D. Hertog, A. D. Waegenaere, B. Melenberg, and G. Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Manag. Sci.*, 59:341–357, 2013.

G. Blanchard, Gyemin Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *NIPS*, 2011.

Fabio Maria Carlucci, Antonio D'Innocente, S. Bucci, B. Caputo, and T. Tommasi. Domain generalization by solving jigsaw puzzles. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2224–2233, 2019.

S. Chang, Y. Zhang, M. Yu, and T. Jaakkola. Invariant rationalization. In *ICML*, 2020.

John C. Duchi, P. Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv: Machine Learning*, 2016.

Peyman Mohajerin Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171:115–166, 2018.

Yaroslav Ganin, E. Ustinova, Hana Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *ArXiv*, abs/1505.07818, 2016.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, M. Bethge, Felix Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *ArXiv*, abs/1811.12231, 2019.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *ArXiv*, abs/2007.01434, 2020.

Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *ICML*, 2018.

Wengong Jin, R. Barzilay, and T. Jaakkola. Domain extrapolation via regret minimization. *ArXiv*, abs/2006.03908, 2020.

D. Krueger, E. Caballero, J. Jacobsen, A. Zhang, Jonathan Binas, Rémi Le Priol, and Aaron C. Courville. Out-of-distribution generalization via risk extrapolation (rex). *ArXiv*, abs/2003.00688, 2020.

Y. LeCun and Corinna Cortes. The mnist database of handwritten digits. 2005.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5543–5551, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018a.

Y. Li, M. Gong, X. Tian, T. Liu, and D. Tao. Domain generalization via conditional invariant representations. In *AAAI*, 2018b.

Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

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

Krikamol Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. *ArXiv*, abs/1301.2115, 2013.

J. Müller, R. Schmier, Lynton Ardizzone, C. Rother, and U. Köthe. Learning robust models using the principle of independent causal mechanisms. *ArXiv*, abs/2010.07167, 2020.

H. Nam, Hyunjae Lee, Jongchan Park, W. Yoon, and Donggeun Yoo. Reducing domain gap via style-agnostic networks. *ArXiv*, abs/1910.11645, 2019.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19:1574–1609, 2009.

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and B. Schölkopf. Learning explanations that are hard to vary. *ArXiv*, abs/2009.00329, 2020.

Adam Paszke, S. Gross, Francisco Massa, A. Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Z. Lin, N. Gimelshein, L. Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

J. Pearl. Causality: Models, reasoning and inference. 2000.

Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6: 147–160, 1994.

J. Peters, Peter Buhlmann, and N. Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *arXiv: Methodology*, 2015.

Mateo Rojas-Carulla, B. Schölkopf, Richard E. Turner, and J. Peters. Invariant models for causal transfer learning. *J. Mach. Learn. Res.*, 19:36:1–36:34, 2018.

Elan Rosenfeld, P. Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *ArXiv*, abs/2010.05761, 2020.

Shiori Sagawa, Pang Wei Koh, T. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *ArXiv*, abs/1911.08731, 2019.

Yuge Shi, Jeffrey S. Seely, P. Torr, N. Siddharth, Awni Y. Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *ArXiv*, abs/2104.09937, 2021.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*, 2016.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5385–5394, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *ArXiv*, abs/1605.07146, 2016.

B. Zhou, Àgata Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1452–1464, 2018.

K. Zhou, Yongxin Yang, Timothy M. Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, 2020.

A PROOFS

A.1 PROOF OF THEOREM 1

In the following theorem, we show a simplified version as in Rosenfeld et al. (2020), where $\sigma_e = 1, \forall e \in \{1, 2, \ldots, E\}$. The full version can be similarly deduced.

Theorem 2 (Formal statement)**.** *Assume there exist two training environments $i, j \in \{1, 2, \ldots, E\}$ that satisfying $\| \mu_i - \mu_k \|_2 \geq 2\sqrt{2d_e}, \forall k \in \{1, 2, \ldots E\} - \{i\}$ and $-\exp(4)d_e \leq \mu_i^T \mu_j \leq -\frac{1}{\sigma_c^2} \| \mu_c \|_2^2$. We also assume $\sigma_e = 1, \forall e \in \{1, 2, \ldots, E\}$. Then there exists a classifier which achieves near optimal IRMv1 loss (Eq. (2)) and has high transfer risk (Eq. (1)) with high dimension d_e. In addition, for any test environment $E + 1$ with a non-causal mean far from the those in training:*

$$\forall e \in \{1, \ldots, E\}, \min_{y \in \{\pm 1\}} \| \mu_{E+1} - \mu_i \| \geq (\sqrt{2} + \delta)\sqrt{d_e}$$

for some $\delta > 0$. Then the classifier behaves like the ERM-trained classifier in environment i on $1 - \frac{2E}{\sqrt{\pi}\delta} \exp(-\delta^2)$ of the test distribution.

Proof. We follow the proof idea of theorem 6.1 in Rosenfeld et al. (2020) and first define $r = \sqrt{2d_e}$. We construct $\mathcal{B}_r$ as

$$\mathcal{B}_r = \left[\bigcup_{e \in \{1,\ldots,E\} - \{i\}} B_r(\mu_e) \right] \cup \left[\bigcup_{e \in \{1,\ldots,E\} - \{i\}} B_r(-\mu_e) \right]$$

where $B_r(\mu_e)$is the ℓ_2 ball centered at μ_e. We construct the classifier as follows:

$$\Phi = \begin{cases} \begin{bmatrix} z_c \\ 0 \end{bmatrix}, z_e \in \mathcal{B}_r \\ \begin{bmatrix} z_c \\ z_e \end{bmatrix}, z_e \in \mathcal{B}_r^c \end{cases} \quad \text{and} \quad w = \begin{pmatrix} \frac{2\mu_c}{\sigma_c^2} \\ 2\mu_i \end{pmatrix}$$

Φ outputs the invariant feature $\begin{bmatrix} z_c \\ 0 \end{bmatrix}$ in the ℓ_2 balls centered at non-causal means except μ_i. Note that w is the optimal predictor on environment i when using the feature extractor above, hence it automatically zero gradient penalty on environment i. By setting $\epsilon = 2, \sigma_e = 1$ in theorem D.3 (Rosenfeld et al., 2020), the IRMv1 penalty term environment other than i is upper bounded by

$$\sum_{P \in \Omega} \| \nabla_w \mathbb{E}_P[\ell(w \circ \Phi)] \|^2 \leq O\left(\exp(-\frac{d_e}{8})(2d_e \exp(4) + \bar{\mu}) \right)$$

where $\bar{\mu} = \frac{1}{E} \sum_{k=1}^{E} \| \mu_k \|_2^2$. Thus in high dimensions, the penalty term shrinks rapidly towards 0.

When $z_e \in \mathcal{B}_r$, the classifier is the invariant classifier that only uses z_c for prediction, and thus has small ERM loss. When $z_e \in \mathcal{B}_r^c$, the incurred logistic loss can be upper bounded by

$$\begin{aligned}
&\Pr(z_e \in \mathcal{B}_r^c) \max_{k \in \{1,\ldots,E\}} \mathbb{E}_{P_k}[\ell(2\frac{\mu_c^T z_c}{\sigma_c^2} + 2\mu_i z_e)] \\
&\quad \leq \exp(-\frac{d_e}{8}) \max_{e \in \{1,\ldots,E\}} \mathbb{E}_{P_i}[\ell(2\frac{\mu_c^T z_c}{\sigma_c^2} + 2\mu_i z_e)] \qquad \text{(sub-exponential tail bound)} \\
&\quad \leq \exp(-\frac{d_e}{8})(1 + \ln 2 + \max_{e \in \{1,\ldots,E\}} \mathbb{E}_{P_k}[-\min(0, (2\frac{\mu_c^T z_c}{\sigma_c^2} + 2\mu_i z_e))]) \\
&\quad < \exp(-\frac{d_e}{8})(1 + \ln 2 + \max_{e \in \{1,\ldots,E\}} \mathbb{E}_{P_k}[-\min(0, 2\mu_i \mu_k))]) \\
&\quad \leq \exp(-\frac{d_e}{8})(1 + \ln 2 + (\exp 4)d_e)
\end{aligned}$$

Together, the classifier has smalle ERM loss and IRMv1 penalty, hence it achieves nearly optimal IRMv1 loss when d_e is large. By theorem D.3 (Rosenfeld et al., 2020), for some $\delta > 0$, the classifier behaves like the optimal ERM classifier in environment i on $1 - \frac{2E}{\sqrt{\pi}\delta}\exp(-\delta^2)$ of the test distribution.

On the other hand, consider the transfer risk of the constructed classifier. The environmental optimal classifier on the worse-group j is $w_j = (\frac{2\mu_c}{\sigma_c^2}, 2\mu_j)^T$ by the construction of Φ. The incurred transfer risk is lower bounded by applying the worse-group predictor w_j on environment i:

$$\begin{aligned}
&\max_{j\in\{1,\dots,E\}-\{i\}} \mathbb{E}_{P_i}[\ell(2\frac{\mu_c^T z_c}{\sigma_c^2} + 2\mu_j z_e)] \\
&\qquad \geq \max_{j\in\{1,\dots,E\}-\{i\}} \ell(2\frac{\mu_c^T \mu_c}{\sigma_c^2} + 2\mu_j\mu_i) \qquad \text{(Jensen's inequality)} \\
&\qquad = \ell(2\frac{\mu_c^T \mu_c}{\sigma_c^2} + 2\min_{j\in\{1,\dots,E\}-\{i\}} \mu_j\mu_i) \\
&\qquad \geq \ell(0) = \ln 2 \qquad \text{(Decreasing of } \ell \text{ and } \exists j, \mu_i^T\mu_j \leq -\frac{\mu_c^T\mu_c}{\sigma_c^2})
\end{aligned}$$

Hence the transfer risk is at least $\ln 2$. □

A.2 Analysis in the 2-d case

We denote the mean of non-causal means as $\mathbb{E}_i\mu_i = \sum_{i=1}^E \mu_i/E$. We consider the setting where the feature extractor is linear, *i.e.*, , $\Phi(x) = az_c + bz_e$, and the loss function is logistic loss $f(x) = \log(1 + e^{-x})$.

For simplicity, we use the "sum-sum" version of TRM:

$$\mathcal{R}_{TRM}(\Phi) = \mathbb{E}_{i\neq j}\mathbb{E}_{(x,y)\sim P_j}[f(y\Phi(x)w(P_i))]$$

The minimal values of the objectives are scale-invariant to $a^2 + b^2$. W.l.o.g we assume $a^2 + b^2 = 1$ in the following. We show that TRM would learn a robust linear feature extractor when $\mathbb{E}_i[\mu_i] = 0$, as shown below.

Proposition 2. *Assume $a^2 + b^2 = 1$ and $\mathbb{E}_i[\mu_i] = 0$, then the minimizer of the $\mathcal{R}_{TRM}$ $(\pm 1, 0)$.*

Proof. Given feature extractor with parameter a, b, the optimal predictor $w(P_i)$ has the closed form $w(P_i) = \frac{2(a\mu_c + b\mu_i)}{a^2+b^2} = 2(a\mu_c + b\mu_i)$ in $\mathcal{R}_{TRM}$. Since the logistic loss $f(x) = \log(1 + e^{-x})$ is convex, by Jensen's inequality we have $\forall\Phi$:

$$\begin{aligned}
&\mathcal{R}_{TRM}(\Phi) = \mathbb{E}_{i\neq j}\mathbb{E}_{(x,y)\sim P_j}[f(y\Phi(x)w(P_i))] \\
&= 2\mathbb{E}_{\mu_i,\mu_j,i\neq j}\mathbb{E}_{z_c\sim\mathcal{N}(\mu_c,1),z_e\sim\mathcal{N}(\mu_j,1)} f(\frac{2(az_c + bz_e)(a\mu_c + b\mu_i)}{(a^2+b^2)}) \qquad \text{(Symmetry of } y = \pm 1) \\
&\geq 2\mathbb{E}_{z_c\sim\mathcal{N}(\mu_c,1)} f(2\mathbb{E}_{\mu_i,\mu_j}\mathbb{E}_{z_e\sim\mathcal{N}(\mu_j,1)}[2(az_c + bz_e)(a\mu_c + b\mu_i)]) \qquad \text{(Jensen's inequality)} \\
&= 2\mathbb{E}_{z_c\sim\mathcal{N}(\mu_c,1)} f(2[a^2 z_c\mu_c + b\mathbb{E}_j\mu_j(az_c + b\mathbb{E}_i\mu_i) + ab\mu_c\mathbb{E}_i\mu_i]) \\
&= 2\mathbb{E}_{z_c\sim\mathcal{N}(\mu_c,1)} f(2a^2 z_c\mu_c) \qquad (\mathbb{E}_i[\mu_i] = 0) \\
&\geq \mathcal{R}_{TRM}((\pm 1, 0)) \qquad \text{(Decreasing of } f)
\end{aligned}$$

□

A.3 Proof of Proposition 1

Proposition 1. *Denote the Hessian as $H_{w(Q)} = \frac{\partial^2\mathbb{E}_Q[\ell(w(Q)\circ\Phi)]}{\partial w(Q)^2}$. Suppose the loss $\ell(w\circ\Phi)$ is continuously differentiable and $H_{w(Q)}$ is non-singular, then we have:*

$$(\frac{\partial\mathcal{L}_P(Q)}{\partial w(Q;\Phi)})^T \frac{\mathrm{d}w(Q;\Phi)}{\mathrm{d}\Phi} = -\frac{\partial(\mathtt{sg}(v_Q)^T \frac{\partial\mathbb{E}_Q[\ell(w(Q)\circ\Phi)]}{\partial w(Q)})}{\partial\Phi}$$

where $v_Q = (\frac{\partial \mathcal{L}_P(Q)}{\partial w(Q)})^T H^{-1}_{w(Q)}$*, and is treated as a constant vector in the above equation (note the use of stop-gradient version* $w(Q)$*).*

Proof. The function $w(Q;\Phi)$ is implicitly defined by the optimality condition:

$$\frac{\partial \mathbb{E}_Q[\ell(w(Q)\circ\Phi)]}{\partial w(Q)} = 0$$

By the implicit function theorem we know that:

$$\frac{dw(Q)}{d\Phi} = -H^{-1}_{w(Q)} \frac{\partial^2 \mathbb{E}_Q[\ell(w(Q)\circ\Phi)]}{\partial w(Q)\partial\Phi}$$

Thus we have

$$\begin{aligned}(\frac{\partial \mathbb{E}_P[\ell(w(Q)\circ\Phi)]}{\partial w})^T \frac{dw(Q)}{d\Phi} &= -(\frac{\partial \mathbb{E}_P[\ell(w(Q)\circ\Phi)]}{\partial w})^T H^{-1}_{w(Q)} \frac{\partial^2 \mathbb{E}_Q[\ell(w(Q)\circ\Phi)]}{\partial w(Q)\partial\Phi} \\ &= \frac{\partial (v_Q^T \frac{\partial \mathbb{E}_Q[\ell(w(Q)\circ\Phi)]}{\partial w})}{\partial\Phi}\end{aligned}$$

where $v_Q = -(\frac{\partial \mathbb{E}_P[\ell(w(Q)\circ\Phi)]}{\partial w})^T H^{-1}_{w(Q)}$. □

B Experimental Details

B.1 Synthetic Datasets

For 10C-CMNIST and SceneCOCO, we construct 2 training environments, 1 test environments and 1 validation. Below we discuss the detailed data generation process. All the data are constructed on the publicly available dataset and do not contain personally identifiable information or offensive content. The dataset is splitted evenly across environments. For 10C-CMNIST/SceneCOCO, each environment has 16000/2600 datapoints and validation set has 12000/2200 datapoints. We repeat all the experiments ten times on one GeForce RTX 2020 GPU.

B.1.1 10C-CMNIST

10C-CMNIST dataset consists of 60000 datapoints of dimension (3,28,28) from 10 classes. We use 16000 datapoints for each train/test environment. The remaining 12000 datapoints are used for validation.

We use the digits in the public MNIST dataset (LeCun & Cortes, 2005) as the invariant (causal) feature. We use the following ten RGB colors as digit colors (non-causal feature): (0, 100, 0), (188, 143, 143), (255, 0, 0), (255, 215, 0), (0, 255, 0), (65, 105, 225), (0, 225, 225), (0, 0, 255), (255, 20, 147), (180, 180, 180). For background color (dummy feature), we randomly pick five RGB colors for each each environment in different runs.

For combined shift and label-correlated shift, the two training environments are constructed with $100\%/r$ label-digit color correlation, where r is the biased degree. For the two training environments, every digit label is randomly correlated with a digit color with $100\%/r$ correlation in different runs. We uniformly sample a digit color for every image for the test environment, indicating no correlation between label and git color.

For each image, the background color is uniformly drawn from five RGB colors. Note that the five background colors for each environment are picked separately in combined shift and label-uncorrelated shift. Hence the background colors across environments can be non-overlapped.

B.1.2 SceneCOCO

SceneCOCO dataset consists of 10000 datapoints of dimension (3,64,64) from 10 classes. We use 2600 datapoints for each train/test environment. The remaining 2200 datapoints are used for validation.

We use the following ten objects in the public COCO dataset (Lin et al., 2014) as the invariant feature: boat, airplane, truck, dog, zebra, horse, bird, train, bus, motorcycle. For the background scenes (non-causal feature), we use the following 19 scenes in the public Places dataset (Zhou et al., 2018): beach, canyon, building facade, staircase, desert (sand), crevasse, bamboo forest, broadleaf forest, ball pit, kasbah, lighthouse, pagoda, rock arch, oast house, orchard, viaduct, water tower, waterfall, zen garden. For object color (dummy feature), we randomly pick ten RGB colors for each run.

We randomly pick ten scenes as the non-causal feature for different runs for combined shift and label-correlated shift. Every object is randomly correlated with a background scene with $100\%/r$ correlation in different runs.

For each image, the object color is uniformly drawn from five RGB colors. Note that the ten object colors for each environment are picked separately in combined shift and label-uncorrelated shift. The object colors across environments can be non-overlapped.

B.2 Hyper-parameter selection

Below we list the method specific hyper-parameters:

IRMv1 (Arjovsky et al., 2019) The objective of IRMv1 is: $\sum_{P\in\Omega}\mathbb{E}_P[\ell(w\circ\Phi)] + \lambda \parallel \nabla_w\mathbb{E}_P[\ell(w\circ\Phi)] \parallel^2$. The coefficient of the gradient penality term λ is searched over a range of $\{1e-3, 1e-2, 1e-1, 1, 10\}$. The number of epochs over which to plug in the gradient penalty term is searched over 1~5 epochs for all datasets.

VREx (Krueger et al., 2020): The objective of VREx is: $\sum_{P\in\Omega}\mathbb{E}_P[\ell(w\circ\Phi)]+\lambda\text{Var}(\{\mathbb{E}_P[\ell(w\circ\Phi)]\}_{P\in\Omega})$. The coefficient of the regularization term (the variance of loss across environments) is searched over a range of $\{1e-3, 1e-2, 1e-1, 1, 10\}$. The number of epochs over which to plug in the regularization is searched over 1~5 epochs for all datasets.

GroupDRO (Sagawa et al., 2019): GroupDRO proposes an online algorithm for group distributionally robust optimization. The learning rate of the online exponential gradient descent over the simplex is search over $\{1e-3, 1e-2, 1e-1\}$.

MLDG (Li et al., 2017): MLDG proposes meta learning algorithms for domain generalization. The coefficient of the meta learning objective is search over $\{1, 0.5, 0.1, 0.05\}$.

Fish (Shi et al., 2021): Fish uses an inner-loop and outer-loop optimization, which is equivalent to the gradient matching objective. The learning rate in the outer-loop (meta learning step) is searched over $\{1, 0.5, 0.1, 0.05\}$.

TRM: The coefficient of the weighted gradient-matching term is searched over $\{1e-3, 1e-2, 1e-1, 1\}$. We unroll the Taylor expansion for 10 steps in the inverse Hessian vector product approximation, *i.e.*, $H^{-1}v \approx H_{10}^{-1}v = \sum_{i=0}^{10}(I-H)^i v$.

We evaluate all the methods both on the train-domain validation (Section 5.1.2) set and test-domain validation set (Appendix C).

B.3 Training

B.3.1 Convergence of Algorithm 1

We first introduce the result from Eq. 3.23 in Nemirovski et al. (2009) and proposition 2 in Sagawa et al. (2019). Denote the $E-1$-dimension simplex as Δ_{E-1}, and the parameter space of Φ as Θ. Consider the min-max optimization problem

$$\max_{\alpha\in\Delta_{E-1}}\min_{\Phi\in\Theta}\sum_{i=1}^{E-1}\alpha_i f_i(\Phi)$$

Assumption 1. *f_i is convex on Θ.*

Assumption 2. *We have the unbiased stochastic gradient $\nabla f_i(\Phi;\epsilon)$ of f_i, that is $\mathbb{E}_\epsilon[\nabla f_i(\Phi;\epsilon)] = \nabla f_i(\Phi)$.*

Online mirror descent yielding average iterates over T iterations $\bar{\alpha}^T$ and $\bar{\Phi}^T$, has the following guarantee.

Proposition 3. *(Nemirovski et al. (2009), Eq. 3.23). Suppose that Assumptions 1-2 hold. Then we have*

$$\mathbb{E}_\epsilon\left[\max_{\alpha\in\Delta_{E-1}}\sum_{i=1}^{E-1}\alpha_i f_i(\bar{\Phi}^T) - \min_{\Phi\in\Theta}\sum_{i=1}^{E-1}\bar{\alpha}_i^T f_i(\Phi)\right] \le 2\sqrt{\frac{10(R_\Phi^2 M_\Phi^2 + M_\alpha^2\log(E-1))}{T}}$$

where

$$\mathbb{E}_\epsilon\left[\parallel \nabla_\Phi \sum_{i=1}^{E-1}\alpha_i f_i(\Phi;\epsilon)\parallel_2^2\right] \le M_\Phi$$
$$\mathbb{E}_\epsilon\left[\parallel \nabla_\alpha \sum_{i=1}^{E-1}\alpha_i f_i(\Phi;\epsilon)\parallel_2^2\right] \le M_\alpha$$
$$R_\Phi^2 = \max_\Phi \parallel \Phi\parallel_2^2 - \min_\Phi \parallel \Phi\parallel_2^2$$

for online mirror descent with 1-strongly convex norm $\|\cdot\|_2$.

Recall that $P(Q) = \sum_{P_i\in\Omega\setminus Q}\alpha_i(Q)P_i$. The TRM risk in Eq. (6) can be formulate the as a saddle point problem:

$$\mathcal{R}(\Phi, w_{all}; Q) = \sum_{i:P_i\in\Omega\setminus Q}\alpha_i(Q)\Big(\mathbb{E}_Q[\ell(w_{all}\circ\Phi)] + \mathbb{E}_{P_i}[\ell(w(Q)\circ\Phi)]$$
$$-\lambda D\langle \mathtt{sg}(\frac{\partial\mathbb{E}_{P(Q)}[\ell(w(Q)\circ\Phi)]}{\partial w}), \frac{\partial\mathbb{E}_Q[\ell(w(Q)\circ\Phi)]}{\partial w}\rangle\Big)$$

Correspondingly, let $f_i(\Phi, w_{all}) = E_Q[\ell(w_{all}\circ\Phi)] + \mathbb{E}_{P_i}[\ell(w(Q)\circ\Phi)] - \lambda D\langle \mathtt{sg}(\frac{\partial\mathbb{E}_{P(Q)}[\ell(w(Q)\circ\Phi)]}{\partial w}), \frac{\partial\mathbb{E}_Q[\ell(w(Q)\circ\Phi)]}{\partial w}\rangle$. Denote the average iterate over T iterations as $\bar{\Phi}^T, \bar{w_{all}}^T, \bar{\alpha}(Q)^T$. We define the average per-environment regret as

$$r_T(Q) = \max_{\alpha(Q)}\mathcal{R}(\bar{\Phi}^T, \bar{w}_{all}^T; Q) - \min_{\Phi, w_{all}, \alpha=\bar{\alpha}(Q)^T}\mathcal{R}(\Phi, w_{all}; Q)$$

Algorithm 1 can be seen as an instance of online mirror descent for saddle point problem above, with the following assumptions:

Assumption 3. *$f_i(\Phi, w_{all})$ is convex for $(\Phi, w_{all})\in\Theta$.*

Assumption 4. *We have the unbiased stochastic gradient $\nabla f_i(\Phi, w_{all};\epsilon)$ of f_i in each iteration, that is $\mathbb{E}_\epsilon[\nabla f_i(\Phi, w_{all};\epsilon)] = \nabla f_i(\Phi, w_{all})$.*

Proposition 4. *Suppose that Assumptions 3-4 hold, and f_i is convex, C-Lipschitz continuous, and bounded by B_ℓ. Further assume that $\parallel(\Phi, w_{all})\parallel_2 \le B_{\Phi, w_{all}}$ for all $(\Phi, w_{all})\in\Theta$ with convex set $\Theta\subseteq\mathbb{R}^d$. Then, the average iterate of Algorithm 1 achieves an expected per-environment regret at the rate*

$$\mathbb{E}_\epsilon[r_T] \le 2\sqrt{\frac{10(B_\Phi^2 C^2 + B_\ell^2\log(E-1))}{T}}$$

Proof. We compare the correspond terms in Proposition 3.

$$\mathbb{E}_\epsilon\left[\parallel \nabla_{\Phi, w_{all}} \sum_{i=1}^{E-1}\alpha_i f_i(\Phi, w_{all};\epsilon)\parallel_2^2\right] \le C^2$$
$$\mathbb{E}_\epsilon\left[\parallel \nabla_\alpha \sum_{i=1}^{E-1}\alpha_i f_i(\Phi;\epsilon)\parallel_2^2\right] \le B_l^2$$
$$R_\Phi^2 = \max_{\Phi, w_{all}}\parallel(\Phi, w_{all})\parallel_2^2 - \min_{\Phi, w_{all}}\parallel(\Phi, w_{all})\parallel_2^2 \le B_{\Phi, w_{all}}^2$$

We arrive at the result directly by Proposition 3. □

B.3.2 TRAINING DETAILS

We adopt a 4-layer CNN as the feature extractor for 10C-CMNIST and the Wide ResNet (Zagoruyko & Komodakis, 2016) for SceneCOCO. We use ResNet18/ResNet50 as the backbone for PACS. The predictor w is a fully connected layer.

We train for 10 epochs with batch size 128 on 10C-CMNIST, 100 epochs with batch size 128 on SceneCOCO, and 50 epochs with batch size 32 on PACS. SGD with $1e-1$ initial learning rate and 0.9 momentum as the optimizer for 10C-CMNIST and SceneCOCO. We decay the learning rate with a constant 0.1 at 4-th epoch on 10C-CMNSIT and 40-th epoch for SceneCOCO. We fine-tune on PACS with constant learning rate $1e-4$.

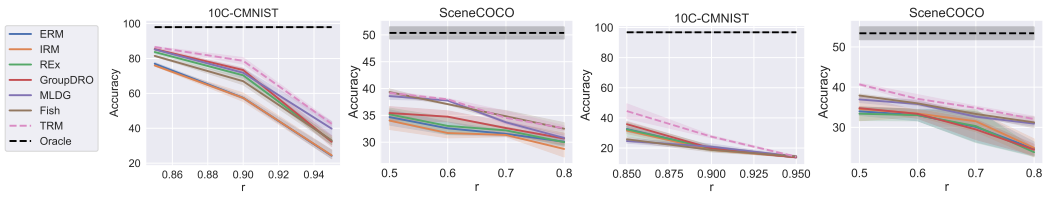
All model-specific hyper-parameters are picked on test-domain or train-domain validation set (Appendix B.2).

C EXTRA EXPERIMENTAL RESULTS

C.1 10C-CMNIST AND SCENECOCO

C.1.1 DIFFERENT VALIDATION CONFIGURATION

Fig. 5 reports the test accuracy on 10C-CMNIST and SceneCOCO with a validation set that has the same distribution as the test set.



(a) Label-correlated shift (b) Combined shift

Figure 5: Test accuracy on 10C-CMNIST and SceneCOCO datasets in (a) Label-correlated shift and (b) Combined shift, using various bias degrees r. The validation set has same distribution as the test set, *i.e.*, test-domain validation set.

C.1.2 TRANSFERABILITY OF MODELS

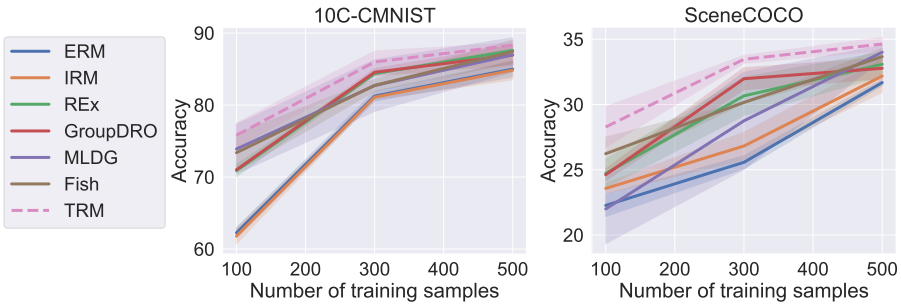


Figure 6: Test accuracy on 10C-CMNIST and SceneCOCO datasets in (a) Label-correlated shift and (b) Combined shift, using different numbers of training samples for finu-tuning on the target environment.

We evaluate the transferability of learned features by fine-tuning the model on the unbiased target environment with limited data. In Fig. 6, we report the test accuracy on the target environment after fine-tuning with the different numbers data. We use model trained

on $100\%/90\%$ and $90\%/60\%$ bias degree configurations in 10C-CMNIST and SceneCOCO respectively. Our main finding is that the feature extractors trained by TRM outperform other methods under different numbers of data using for fine-tuning. It indicates that models trained by TRM transfer faster to the target environments.

C.1.3 EFFECT OF DUMMY FEATURE

Table 3 reports the test accuracy on label-uncorrelated domain shift. We observe that all the methods have similar performance and good generalization in this scenario. The experiment suggests that without the biased effect of non-causal features, learning algorithms are robust for domain shift in general.

Table 3: Test accuracy on Label-uncorrelated shift. All the methods achieve performances comparable to the Oracle.

Algorithm	**10C-CMNIST**	**SceneCOCO**
ERM	98.0 ± 0.1	64.3 ± 0.9
IRM	98.0 ± 0.1	65.3 ± 0.7
REx	98.1 ± 0.1	65.3 ± 0.7
GroupDRO	98.1 ± 0.1	66.0 ± 1.0
MLDG	**98.4 ± 0.1**	**67.4 ± 0.9**
Fish	98.2 ± 0.1	65.4 ± 0.7
TRM	98.3 ± 0.1	65.5 ± 1.0
Oracle	98.3 ± 0.1	66.9 ± 0.6

C.2 PACS

Table 4 demonstrates the test accuracy on PACS when the validation set has the same distribution as the test set (test-domain validation set), using backbones ResNet18 and ResNet50. Table 5 reports the test accuracy on train-domain validation set, using ResNet50.

Table 4: Test accuracy on PACS dataset by ResNet18, on a test-domain validation set.

Algorithm	**Art**	**Cartoon**	**Photo**	**Sketch**	**Average**
ERM	76.2 ± 1.6	70.6 ± 1.0	95.5 ± 2.4	66.2 ± 1.1	77.1
IRM	76.0 ± 0.7	70.8 ± 1.0	95.2 ± 2.4	66.7 ± 0.7	77.2
REx	75.6 ± 1.7	64.9 ± 1.1	**95.8 ± 0.9**	63.4 ± 1.0	74.9
GroupDRO	77.6 ± 0.3	72.9 ± 1.1	94.9 ± 2.4	69.2 ± 0.9	78.6
MLDG	77.3 ± 1.2	72.0 ± 0.6	95.4 ± 1.0	**72.3 ± 0.2**	79.2
Fish	76.8 ± 1.7	72.9 ± 0.3	94.5 ± 1.1	70.6 ± 1.0	78.7
TRM	**78.7 ± 0.5**	**74.4 ± 0.6**	94.6 ± 0.9	71.4 ± 2.3	**79.8**

Table 5: Test accuracy on PACS dataset by ResNet50, on a train-domain validation set.

Algorithm	**Art**	**Cartoon**	**Photo**	**Sketch**	**Average**
ERM	83.7 ± 1.3	68.6 ± 0.4	98.0 ± 0.4	69.8 ± 1.0	80.0
IRM	81.9 ± 1.1	68.6 ± 0.4	98.0 ± 0.3	70.9 ± 1.0	79.9
REx	84.2 ± 1.5	68.4 ± 1.0	98.0 ± 0.5	69.3 ± 0.6	80.0
GroupDRO	84.3 ± 2.0	69.4 ± 0.4	98.2 ± 0.3	70.8 ± 0.3	80.7
MLDG	85.0 ± 0.7	72.8 ± 0.5	98.0 ± 0.4	**74.8 ± 0.2**	82.7
Fish	83.9 ± 0.3	71.4 ± 0.4	**98.4 ± 0.1**	72.3 ± 0.1	81.5
TRM	**85.8 ± 1.1**	**77.3 ± 0.5**	97.6 ± 0.2	70.9 ± 1.2	**82.9**

C.3 TRAINING OVERHEADS

In Table 6, we report the average wall-clock time (seconds/epoch) of different methods on SceneCOCO with batch size 128. We observe that TRM does not take significantly longer training time although it needs to compute the per-environment optimal predictor w and weighted gradient matching term in every minibatch.

Table 6: Per-epoch training time of different algorithms.

Algorithm	ERM	IRM	REx	GroupDRO	Fish	MLDG	TRM
Wall-clock time (s)	5.81	5.95	5.86	5.90	6.55	10.69	8.69

C.4 EXPERIMENTS ON GROUP DISTRIBUTIONAL ROBUSTNESS

C.4.1 SETUPS

The group distributional robustness (Hu et al., 2018; Sagawa et al., 2019) aims to minimize the *worse-group accuracy* over a set of pre-defined groups, instead of the average accuracy over these groups. Following Sagawa et al. (2019), we uses **CelebA** dataset (Liu et al., 2015) for evaluation. The hair color {blond, non-blond} is used as label and the gender {male, female} as the spurious feature. There are four groups in the dataset, namely blond-haired male, blond-haired female, non-blond-haired hair male, non-blond-haired female, with a total of 162700 datapoints and 1387 datapoints in the smallest group (blond-haired male).

We compare TRM with vanilla **ERM**, **GroupDRO**, and **Reweight**, which sets the sampling weights to the inverse of the group priors (Sagawa et al., 2019). Each group has only one class label and hence hinders the predictor transfer procedure in TRM. Thus we combine blond-haired male and non-blond-haired hair male/blond-haired female and non-blond-haired hair female as two groups for TRM training. We resize all images to $(3, 224, 224)$ and use the ImageNet-pretrained ResNet18 as the backbone of feature extractors. We use the SGD optimizer with a momentum of 0.9, a weight decay of 1e-4, and a fixed learning rate of 1e-4. The batch size is set to 32.

C.4.2 RESULTS

In Table 7, we report the average and worse-group test accuracy on CelebA dataset. The proposed TRM algorithm improves over other baselines on the worse-group test accuracy. We observe that TRM still has a competitive average accuracy with other baselines. The results show that TRM more relies on the invariant features (hair color) for prediction, leading to lower worse-group accuracy, even though the spurious feature (gender) exists in the dataset.

Table 7: Worse-group and average accuracy on CelebA dataset

	ERM	Reweight	GroupDRO	TRM
Worse-group accuracy	46.0 ± 2.9	89.3 ± 1.2	90.0 ± 1.3	$\mathbf{90.3 \pm 0.4}$
Average accuracy	$\mathbf{94.7 \pm 0.3}$	91.8 ± 0.4	91.6 ± 0.3	91.6 ± 0.4