Towards Environment-Invariant Representation Learning for Robust Task Transfer

Benjamin Eyre [1 2] **Richard Zemel** [3] **Elliot Creager** [1 2]

Abstract

To train a classification model that is robust to distribution shifts upon deployment, auxiliary labels indicating the various "environments" of data collection can be leveraged to mitigate reliance on environment-specific features. In this paper we attempt to determine where in the network the environment invariance property can be located for such a model, with the hopes of adapting a single pre-trained invariant model for use in multiple tasks. We discuss how to evaluate whether a model has formed an environment-invariant internal *representation*—as opposed to an invariant final classifier function—and propose an objective that encourages learning such a representation. We also extend color-biased digit recognition to a transfer setting where the target task requires an invariant model, but lacks the environment labels needed to train an invariant model from scratch, thus motivating the transfer of an invariant representation trained on a source task with environment labels.

1. Introduction

Domain generalization techniques aim to make use of training data coming from multiple domains (also called "environments") in order to create machine learning models which generalize to unseen domains at test time (Blanchard et al., 2011; Peters et al., 2016; Arjovsky et al., 2019; Sagawa et al., 2019). Models that are able to make use of common features (that are shared across the training domains) and ignore domain-specific features are referred to as invariant models. Given the prevalence of distribution shift in practical settings (Balagopalan et al., 2020; Koh et al., 2021) and propensity of neural nets to prefer features brittle to such shifts (Geirhos et al., 2020), invariant models present a promising path towards realizing robust machine learning.

In this work, we consider a scenario where we are provided with a source task with multiple training environments, and a target task with a single training environment. Due to the presence of spurious features in the target task, training directly from scratch on available target data is unlikely to yield an invariant model. To investigate the feasibility of robust task transfer, we develop methods for measuring, and ultimately encouraging, environment-invariance in learned representations. Our key contributions are as follows:

- Insight into how the Environment Invariance Constraint (EIC) (Arjovsky et al., 2019) provides a flexible objective that can be achieved by different components of a network, leading to a distinction between invariant classifiers and invariant representations.
- A tractable metric (EIRS) for evaluating the degree to which a representation satisfies the difficult-to-compute EIC.
- An extension of a popular color-biased digit recognition benchmark to include a novel transfer task where an invariant representation (beyond just an invariant classifier on the source task) is needed.
- A new method for learning invariant representations (LEIR) that helps with the aforementioned transfer task.

2. Environment Invariant Representations

2.1. Invariant Learning Setup

We consider data that is sampled from a series of environment-conditioned generative distributions. Formally, we assume that each instance of the target variable Y belongs to one of k distinct classes. Inputs are denoted by $X \in \mathcal{X}$, and the discrete environment variable is denoted $E \in \mathcal{E}, \mathcal{E} = \mathcal{E}_{tr} \cup \mathcal{E}_{ts}$, where $\mathcal{E}_{tr}$ and $\mathcal{E}_{ts}$ are sets of train and test environments. Our input and target variable observations are distributed according to conditional distributions: $X, Y \sim P(X, Y|E = e)$. The training data

[1] University of Toronto [2] Vector Institute [3] Columbia University. Correspondence to: Benjamin Eyre <benjamin.eyre@mail.utoronto.ca>.

Published at the ICML 2022 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA. Copyright 2022 by the author(s).

comprises N samples with environment labels: $\mathcal{D}_{tr} = \{\{(x_e^i, y_e^i, e)\}_{i=1}^N\}_{e \in \mathcal{E}_{tr}}$, where $x_e^i, y_e^i \sim P(X, Y|E = e)$. Our model consists of an encoder $h : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is our representation space and a classifier $f : \mathcal{H} \to \mathbb{R}^k$, which can be converted to a predictive distribution over the k classes using a softmax function. Our full model is described by the expression $f(h(x))$. The output of the encoder $h(x)$ is often referred to as the *representation* of x. Our goal will be to train a model with strong performance on data from all environments in $\mathcal{E}$, while only training using data from environments in $\mathcal{E}_{tr}$.

2.2. Approaches to Invariant Learning

Empirical Risk Minimization (ERM) ignores environment labels, and instead learns by reducing aggregate training loss, which can yield a model that is overly sensitive to spurious features (Geirhos et al., 2020). Environment labels can be used to learn an invariant model in several different ways. Some approaches use the environment labels to attain certain predictive properties across each environment-conditioned distribution. Wald et al. (2021) propose that an invariant classifier can be trained by training a classifier that is simultaneously calibrated in each training environment. Specifically, multi-domain calibration asks $\forall e \in \mathcal{E}_{tr}, \alpha \in [0, 1], E[Y|f(X) = \alpha, e_i] = \alpha$. However, since this constraint is described using the classifier, rather than the encoder, it is uncertain as to whether satisfying multi-domain calibration would necessarily yield a transferable representation with invariant properties.

In addition, more flexible approaches that can be applied directly to the encoder output have been proposed. The MMD approach (Veitch et al., 2021) involves matching the output distribution of a function $\phi(X)$ across each environment-conditioned input distribution. Specifically, the objective of these approaches is to minimize an estimate of the maximum mean discrepancy (Gretton et al., 2012): $min_\phi sup_{\omega \in \Omega}(E[\omega(\phi(X))|e_1] - E[\omega(\phi(X))|e_2])$ where $e_1, e_2 \in \mathcal{E}_{tr}$. Taking $\phi = h$, minimizing this quantity ensures a level of consistency across environments, as the high-dimensional representation densities will match. It is worth noting that this traditional formulation of MMD does not make use of the valuable information found in the target labels. Additionally, target label information can be used in MMD regularization by way of importance weights (Makar et al., 2022); this approach makes use of causal assumptions about the data generative process to apply the penalty on a modified encoder distribution where Y and e are decorrelated.

Other approaches to invariant representation learning use kernel-based or adversarial regularizers that seek to match support of the learned representation (conditioned on e, and possibly Y) (Edwards & Storkey, 2015; Ganin et al., 2016; Madras et al., 2018; Long et al., 2018). A potential downfall of this distribution matching approach is that it is known to fail under label shift (Zhao et al., 2019).

We aim to learn invariant representations while leveraging label information, and therefore focus our investigation on the *Environment Invariance Constraint* (EIC) (Arjovsky et al., 2019). The EIC requires that the expected target label Y conditioned on both a representation $h(x)$ and an environment e must be the same regardless of which environment it is conditioned on. Formally, the EIC asks that $\forall e_1, e_2 \in \mathcal{E}, \mathbb{E}[Y|h(x), e_1] = \mathbb{E}[Y|h(x), e_2]$ for all representations $h(x)$ that are in intersection of supports for $h(X^e), X^e \sim P(X|E = e), e \in \{e_1, e_2\}$. Satisfying this constraint in turn guarantees that for a given representation, the optimal prediction is the same across all environments. This therefore suggests that the representation is robust to changes across environment, as the representation is not encoding information that is correlated with certain values of the target label in one environment and other values of the target label in others.

2.3. Measuring Environment Invariance

Unfortunately, it is difficult to ascertain whether or not a representation has met the requirements of the EIC as it requires comparing densities in representation-spaces that are typically high dimensional. We propose a diagnostic kernel-based method for quantifying the degree to which a representation abides by the EIC. Inspired by recent works describing differentiable measures for calibration (Kumar et al., 2018; Wald et al., 2021), our *Environment-Invariant Representation Score* (EIRS) uses a Gaussian kernel as a similarity measure between two outputs to create a distribution over each output's neighbors. Formally, we define x to be a sample, $K(.,.)$ a kernel, ϕ an output function, $e \in \mathcal{E}$ an environment, and $x' \in e, x' \neq x$ a sample in that environment, and define the distribution:

$$p_K(x'; x, e, \phi) = \frac{K(\phi(x), \phi(x'))}{\sum_{x'' \in e, x'' \neq x} K(\phi(x), \phi(x''))}$$

Using this distribution, we can calculate the expected label "conditioned" on an output and an environment, and use that to calculate the EIRS for the encoder h and some dataset D of (x', y', e') input-target-environment label pairs. Here, $x', y' \in e$ refers to training examples in D with environment label e:

$$EIRS(\phi, D) = \sum_{x,y \in D} \sum_{e,e' \in \mathcal{E} \times \mathcal{E}} |E_K(y; x, e, \phi) - E_K(y; x, e', \phi)|$$

$$E_K(y; x, e, \phi) = \sum_{x', y' \in e, x' \neq x} y' * p(x'; x, e, \phi)$$

We can calculate this score by taking the output function ϕ to be either the encoder h or the classifier output $f \circ h$,

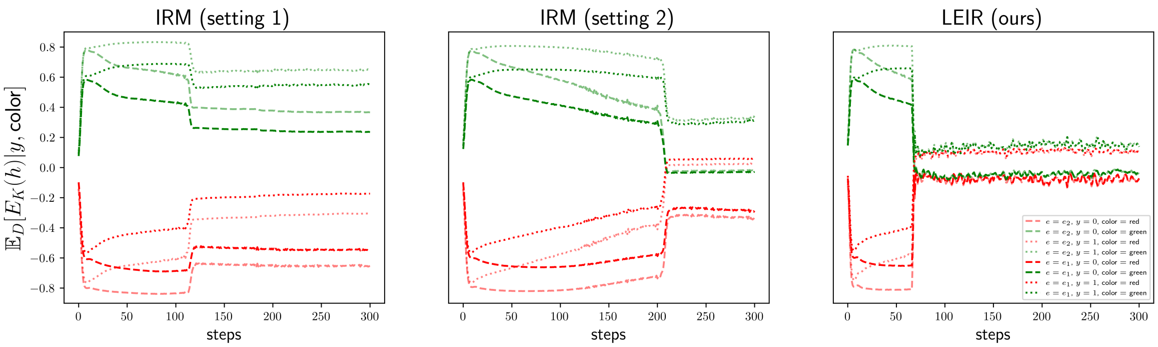


Figure 1. We compute the average $E_K(\cdot)$ (See Section 2.3), our kernel-based approximation to $\mathbb{E}[y|h(x), e]$, across various strata of the CMNIST training set. Strata are determined by the colour, uncorrupted digit class, and environment label. Note that the regularizer in all cases is incorporated after a number of training steps. An invariant representation is realized when, for a given (y, color), the expected label distribution is approximately the same across environments (e.g. the opaque dotted green line lies close to the faded dotted green line, and likewise for other color/pattern combinations). IRM sometimes learns a relatively color-invariant representation (center; $E_K(y; x, e_1, h) \approx E_K(y; x, e_2, h) \forall (y, \text{color})$), while LEIR consistently does (right). However, IRM can also learn a representation that includes color (left; $E_K(y; x, e_1, h) \neq E_K(y; x, e_2, h) \forall (y, \text{color})$). In this case, the final linear layer f projects out the color feature. All models realize an invariant classifier $f \circ h$, and thus have the same test performance.

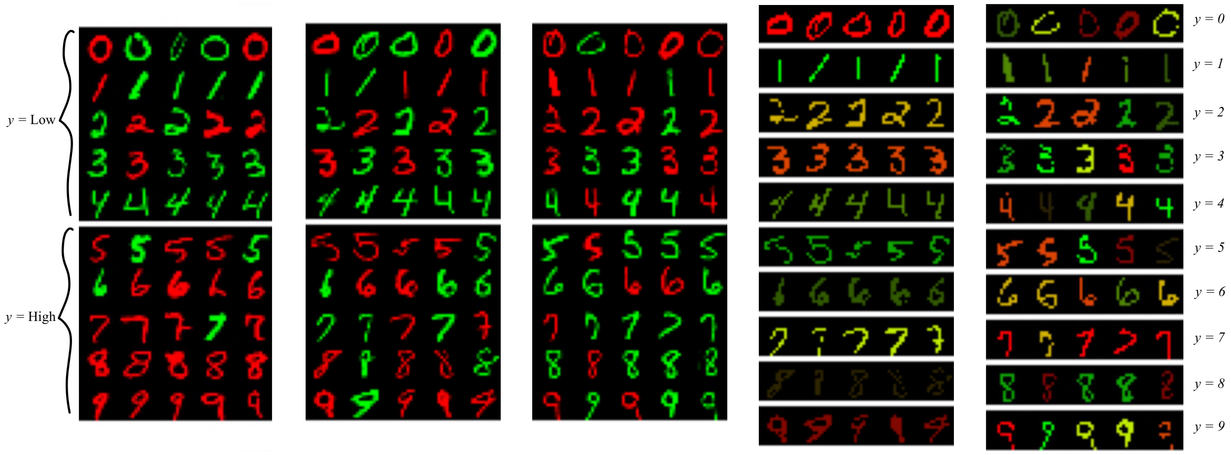
which yields logits for the predictive distribution. Models that use invariant representations in order to make invariant predictions will attain a low EIRS with respect to both the classifier $f \circ h$ as well as the encoder h on its own. In contrast, models which produce invariant predictions by finding an invariant dimension in an otherwise environment-dependent representation will have a high EIRS with respect to h but a low EIRS with respect to the classifier $f \circ h$. We note that due to the quadratic complexity of EIRS, we calculate it with respect to batches $B \subset D$ in practice.

2.4. Do Invariant Classifiers Always Use Invariant Representations?

As a more tractable substitute for a direct optimization problem involving the EIC, Arjovsky et al. (2019) propose IRMv1 (which we refer to as IRM for brevity). IRM augments combines a predictive loss with a regularizer meant to promote invariance across environments. However, this regularizer is applied to $f \circ h$, rather than h directly. As a result, one might suspect that the useful properties ensured by the EIC may only be satisfied by the classifier $f \circ h$ rather than the representation h when training with IRM. While other works have assessed the shortcomings of this approach (Ahuja et al., 2021; Li et al., 2021; Kamath et al., 2021; Rosenfeld et al., 2020; Madras & Zemel, 2021), we instead focus on understanding how models trained with IRM satisfy the EIC by measuring *where* the invariance property is located within the network.

We use the Color-MNIST (CMNIST) dataset (Arjovsky et al., 2019) to study invariant representation and classifier learning. This is a handwritten (binarized) digit recognition dataset where the target label is more strongly associated with the color of the digit than its shape due to added label noise. This correlation strength varies slightly across two training environments, and is dramatically reversed at test time, where classifiers relying on color fail catastrophically.[1]

We fit CMNIST using the following methods: IRM (Arjovsky et al., 2019), ERM, and a grayscale oracle model. We then measure EIRS w.r.t. h and $f \circ h$ for each model,[2] and report the results in Table 1. For IRM, we find two hyperparameter settings that outperform ERM to achieve roughly the same test performance, but realize qualitatively different solutions[3], which we call IRM (setting 1) and IRM (setting 2). Further investigation using EIRS reveals the difference: IRM can induce an invariant representation h, but it does not always do so. To see this, we note that for setting 2, low EIRS on h indicates a color-invariant representation similar to the invariant grayscale model, while for setting 1 high EIRS indicates that h is not color-invariant, even though the classifier logits $f \circ h$ are. This suggests that instead of minimizing the influence of the environment-dependent feature on the representation h, IRM (setting 1) creates a representation similar to that created by ERM, and induces an invariant classifier by projecting away the color dimensions of h in the final linear layer f.



(a) CMNIST train 1 (b) CMNIST train 2 (c) CMNIST test (d) 10CMNIST train (e) 10CMNIST test

Figure 2. We evaluate the ability of invariant *representations* to transfer to a related task with similar spurious features but different target labels. We pre-train using environment labels on binarized CMNIST (2a, 2b) and freeze the representation. To evaluate transfer, we then train logistic regressor on this representation to classify the full 10 digits (under color bias) (2d), finally evaluating on a 10-class test set (where color bias differs) (2e). We call the transfer task 10CMNIST.

Method	EIRS ($f \circ h, D_{tr}$)	EIRS (h, D_{tr})	Test Accuracy
ERM	943.39 ± 27.48	943.92 ± 48.19	19.66% ± 0.88%
IRM (setting 1)	209.79 ± 33.55	1292.39 ± 133.56	65.49% ± 1.22%
IRM (setting 2)	242.84 ± 33.33	592.77 ± 145.17	64.74% ± 0.65%
Grayscale (oracle)	196.25 ± 34.03	161.37 ± 46.5	71.61% ± 0.23%
LEIR (ours)	181.96 ± 22.49	154.36 ± 37.61	64.43% ± 2.1%

Table 1. EIRS (± std. dev., 10 seeds) of various methods trained on CMNIST. High EIRS of h indicates a *representation* containing information about the spurious feature (color). This may occur even if the classifier logits $f \circ h$ project out the spurious information (low EIRS of $f \circ h$) to realize an overall invariant model.

3. Towards Invariant Representation Learning

3.1. Our Proposed Regularizer

In addition to providing a diagnostic measure for helping to determine whether a technique is yielding an invariant representation and/or an invariant classifier, the EIRS also serves as a differentiable objective for regularizing the representation towards satisfying the EIC.

We experiment with using the EIRS as a regularizer, in combination with cross entropy loss and L2 regularization. We also include an additional term in our regularizer to help avoid the representation collapse that is a degenerate minimizer of EIRS. This additional term is penalty on the average value in the Gram matrix, the matrix of kernel values calculated between pairs of examples:

$$LEIR(\phi, D) = EIRS(\phi, D) + \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} K(\phi(x_i), \phi(x_j))}{N^2}.$$

Code to reproduce our experiments can be found at `https://github.com/btleyre/invariant-task-transfer`.

3.2. Experiment: CMNIST

We compare the performance of MLPs trained with ERM, IRM, MMD (Veitch et al., 2021; Makar et al., 2022), CLOvE (Wald et al., 2021), and with our LEIR regularizer on CMNIST. MMD-based invariant learning methods use kernel based estimates of the maximum mean discrepancy (Gretton et al., 2012) of environment-conditioned function output distributions. We experiment with applying an MMD-based penalty to both the encoder output (MMD-Rep) and classifier output (MMD-Logit) of the model. CLOvE uses a trainable measure for calibration (Kumar et al., 2018) to induce calibration on each environment-conditioned distribution. For model selection, we follow Arjovsky et al. (2019) and select the hyperparameters from each regime which maximize min(training accuracy, test accuracy), which in this circumstance is tantamount to selecting models with the best test accuracy. See Appendix D.1 for details.

3.3. Experiment: Classification Transfer (10CMNIST)

To demonstrate the use case for invariant representations, we investigate a classification task where learning an invariant classifier from scratch is difficult. Specifically, we take inspiration from the work of Ahmed et al. (2020), and use a variation of their color-biased 10-way digit classification dataset. Specifically, our 10CMNIST dataset differs from the colour-confounded MNIST datasets presented by Arjovsky et al. (2019) and Ahmed et al. (2020) in that the

[1] See Appendix A for details.

[2] See Appendix B for details

[3] This can be understood as an instance of "underspecification" (D'Amour et al., 2020).

Method	CMNIST (Train)	CMNIST (Test)	10CMNIST Transfer (Train)	10CMNIST Transfer (Test)
ERM - Baseline	-	-	**99.34% ± 0.05%**	12.18% ± 0.35%
ERM	**97.48% ± 1.69%**	30.44% ± 2.41%	88.46% ± 0.87%	31.8 % ± 3.57%
IRM	74.47% ± 0.83%	**65.49% ± 1.22%**	87.06% ± 0.54%	27.48% ± 1.42%
MMD-Rep	62.09% ± 9.91%	60.05% ± 8.59%	24.81% ± 3.0%	15.72 % ± 3.28%
MMD-Logit	62.94% ± 5.38%	60.43% ± 6.69%	46.23% ± 6.11%	17.72 % ± 2.42%
CLOvE	61.27% ± 3.38%	56.92% ± 1.05%	61.78% ± 9.53%	19.67% ± 2.95%
LEIR (Ours)	71.49% ± 1.48%	**64.43% ± 2.1%**	74.52% ± 1.5%	**42.63 % ± 3.39%**

Table 2. 10CMNIST test accuracy for each training regime across 10 seeds, with models selected according to their CMNIST accuracy.

training set does not contain any examples contradicting the strong colour-label correlation. Here, MNIST digits are assigned labels corresponding to their digit class. Each digit class is associated with a unique biasing color. In the training set each sample within a digit class is coloured with that digit's biasing color. This makes the shape of the digit and the biasing color equally predictive of the target label on the training set. On the test set, samples are coloured with any color that is not their own digit class's biasing color. Consequently, models which have learned to rely solely on the simpler biasing color features will perform poorly on the test set.

We experiment with transferring representations pre-trained on CMNIST (which has binary target labels) to 10CMNIST (with all ten digit classes). Specifically, models are first pre-trained with ERM, IRM, MMD, CLOvE, or LEIR on the binary CMNIST training set. We then train a linear probe on top of the encoder on the 10CMNIST training set. During this process we "freeze" the encoder, meaning we perform gradient updates on the classifier but not the encoder during this adaptation phase. Post adaptation, models are evaluated on the 10CMNIST test set. Pre-training hyperparameters are selected based on binary CMNIST test accuracy (a more fair model selection strategy than is typical in domain generalization, which usually relies on validation samples from the test domain; see Appendix E for results using the more standard model selection strategy). Additional details regarding the dataset and adaptation method can be found in Appendices C and D.2. As a baseline, we also evaluate a non-transfer model: a full model (encoder and classifier in tandem) trained solely on 10CMNIST.

3.4. Results

We first note that models trained with LEIR on the binary CMNIST dataset achieve comparable performance to IRM on the test set, with these regimes achieving a mean 64.43% and 65.49% test accuracy across 10 seeds, respectively (Table 2). The LEIR model also achieves low EIRS with respect to both $f \circ h$ and h, indicating that the representation and classifier as a whole are invariant (Table 1).

Our experiments indicate that the LEIR regularizer is able to condition the representation of the MLP model for transfer performance superior to all other training regimes. We observe this superior performance regardless if model selection is performed with a validation set as in Table 3 or using binary CMNIST test accuracy as in Table 2. Finally, the model which was fully trained from scratch on the adaptation training set performs poorly on the transfer task, achieving a mean test accuracy of 12.18% (Table 2, and 3 in the appendix). This demonstrates that transferring a useful representation is necessary when environment labels or training data from a diverse set of environments are not available. Additionally, we note that a model trained with LEIR outperforms an IRM model that achieves low EIRS with respect to both $f \circ h$ and h. This suggests that satisfying the EIC on its own is not sufficient for a representation to transfer to other tasks effectively.

4. Conclusion

In this work, we proposed a novel transfer task as well as a regularizer to enhance performance on this transfer task. This work highlights how the EIC can be satisfied by different representations with different qualities, as well as completely different parts of the model. One interesting question warranting further research concerns whether particular solutions satisfying the EIC are more useful than others, such as high-entropy representations.

References

Ahmed, F., Bengio, Y., van Seijen, H., and Courville, A. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2020.

Ahuja, K., Caballero, E., Zhang, D., Gagnon-Audet, J.-C., Bengio, Y., Mitliagkas, I., and Rish, I. Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34, 2021.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-

Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Balagopalan, A., Novikova, J., Mcdermott, M. B., Nestor, B., Naumann, T., and Ghassemi, M. Cross-language aphasia detection using optimal transport domain adaptation. In *Machine Learning for Health Workshop*, pp. 202–219. PMLR, 2020.

Blanchard, G., Lee, G., and Scott, C. Generalizing from several related classification tasks to a new unlabeled sample. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/b571ecea16a9824023ee1af16897a582-Paper.pdf.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

Edwards, H. and Storkey, A. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

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

Kamath, P., Tangella, A., Sutherland, D., and Srebro, N. Does invariant risk minimization capture invariance? In *International Conference on Artificial Intelligence and Statistics*, pp. 4069–4077. PMLR, 2021.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Kumar, A., Sarawagi, S., and Jain, U. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pp. 2805–2814. PMLR, 2018.

Li, B., Shen, Y., Wang, Y., Zhu, W., Reed, C. J., Zhang, J., Li, D., Keutzer, K., and Zhao, H. Invariant information bottleneck for domain generalization. *arXiv preprint arXiv:2106.06333*, 2021.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Madras, D. and Zemel, R. Understanding post-hoc adaptation for improving subgroup robustness. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

Madras, D., Creager, E., Pitassi, T., and Zemel, R. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pp. 3384–3393. PMLR, 2018.

Makar, M., Packer, B., Moldovan, D., Blalock, D., Halpern, Y., and D'Amour, A. Causally motivated shortcut removal using auxiliary labels. In *International Conference on Artificial Intelligence and Statistics*, pp. 739–766. PMLR, 2022.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

Rosenfeld, E., Ravikumar, P., and Risteski, A. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

Veitch, V., D'Amour, A., Yadlowsky, S., and Eisenstein, J. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *arXiv preprint arXiv:2106.00545*, 2021.

Wald, Y., Feder, A., Greenfeld, D., and Shalit, U. On calibration and out-of-domain generalization. *Advances in Neural Information Processing Systems*, 34, 2021.

Zhao, H., Des Combes, R. T., Zhang, K., and Gordon, G. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pp. 7523–7532. PMLR, 2019.

A. CMNIST Dataset Details

CMNIST consists of handwritten digits that are initially assigned binary labels Y depending on whether the digit belongs to the group 0-4 or 5-9. These binary target labels are flipped with a 25% probability of label-noise, creating the final target labels $\hat{y}$. Finally, the digit receives a color z by flipping $\hat{y}$ with probability p^e according to the environment e that these samples are being generated from. On the two training environments, $p^e \in \{0.1, 0.2\}$, making the color of the digit more predictive of the target label than the digit's shape. However, on the test environment $p^e = 0.9$, inverting the correlation between color and the target label seen on the training set. Consequently, a model must rely on the feature that maintains the same correlation with the target label across environments, shape, in order to generalize effectively. An invariant representation would therefore keep samples of the same digit super group close together, while an environment-dependent representation would keep samples sharing a color close together.

B. EIRS Experiment Details

We train MLP models on CMNIST using IRM (Arjovsky et al., 2019) with 50 different randomly selected hyperparameter settings, and select the model with the best test accuracy. Note that these are the same sets of random hyperparameters described in section D.1. We then use those best performing hyperparameters in 10 trials using different random seeds, recording the EIRS with respect to h and $f \circ h$ in each trial. We also perform 10 trials each for two models using ERM, with those best performing hyperparameters: the first is the same type of MLP used for the IRM model, and the second is a grayscale MLP that does not see color in the image and is therefore required to rely on shape. Results can be found in Table 1.

C. 10CMNIST Dataset Details

To create this dataset, we first binarize the pixel values to ensure that lighter/darker pixels do not result in biasing colors overlapping between digits. We then create the ten biasing colors, with the first two being the red and green from the binary CMNIST dataset. We then randomly generate the remaining colors such that no two colors are too alike in pixel values using the process described by (Ahmed et al., 2020). For the validation set, the ten additional colors are generated using the same random process. Samples in the validation set are coloured using a randomly selected colour from this new set, and therefore validation colours are not seen in either the train or test sets. Unlike the binary CMNIST dataset described by Arjovsky et al. (2019), this adaptation dataset does not make use of any label noise.

D. Additional Training Details

Across all experiments, the model architecture we experiment with is a simple neural network with one hidden layer and one output layer. Models are always trained with cross entropy loss and L2 weight decay in addition to the additional objectives prescribed by the individual techniques.

When training with LEIR, we use a single penalty weight λ for both the EIRS term and the average Gram matrix value term. However, we first multiply the Gram matrix term by another weight, λ_{Gram}, which is permanently set to 100. Additional experiments could be performed to see if tuning this parameter could yield improved performance.

D.1. Binary Classification (CMNIST)

For each training method we experimented with 50 randomly selected sets of hyperparameters. Randomly selected hyperparameters included the hidden dimension of the model, the weight for L2 regularization, and the learning rate. For all methods except for ERM the penalty weight was also experimented with. Also included in this set of randomly selected hyperparameters is the number of epochs for which we set the penalty weight to zero. After this number of epochs has elapsed, we set the penalty weight to whatever the randomly selected value is. Finally, we also randomly select the hyperparameter σ for our Radial Basis Function/ Gaussian kernel defined by $K(x, x') = exp(-\gamma||x - x'||^2)$, where $\gamma = \frac{1}{\sigma}$.

Additionally, across all trials models are trained for 301 epochs. All trials use a batch size of 10000, with 5000 samples coming from each of the two environments.

D.2. 10-Way Classification (10CMNIST)

To select the pre-training hyperparameters for each regime we experiment with two model selection criteria. The first hyperparameter selection method is selecting hyperparameters based on the post-adaptation models' performance on a validation set. This validation set consists of digits that are coloured with colors not seen on either the training or test sets. For the second model selection technique we simply select the hyperparameters which achieve the best test accuracy on the binary CMNIST test set prior to adaptation.

Pre-training on the binary CMNIST dataset follows the procedure described in Section D.1. Linear probes are trained for 100 epochs with a learning rate value of 0.001 and L2 regularization weight 0.001. A full search on the adaptation hyperparameters was not conducted after 50 sets of randomly selected L2 regularization weights and learning rates were tested out using the IRM and ERM models described in Table 2 as the transferred representations. Across the 50 random hyperparameter settings, the standard deviation for ten-way test accuracy was no more than 2.5%, indicating that a full search on the adaptation learning rate and L2 regularization weight would not yield substantially different results.

E. Additional Results

In Section 3.3, we describe a single method for performing model selection: choosing models which perform best on the binary CMNIST test set. However, given the binary task's differences from the 10-way classification task, it may not be a particularly effective measure of how effectively the pre-trained model will transfer to the new task. Therefore, we make use of the validation set described in Section C. Models are therefore selected based on their performance on this validation set. Calculating the accuracy of the selected models, we can see that LEIR still yields models with a far greater potential for transfer (Table 3).

Method	10CMNIST Training Accuracy	10CMNIST Test Accuracy
ERM	82.93% ± 10.01%	24.6% ± 10.49%
ERM - Baseline	**99.34%** ± **0.05%**	12.18% ± 0.35%
IRM	88.64% ± 7.18%	36.76% ± 4.35%
MMD-Rep	88.69% ± 0.99%	43.68% ± 4.06%
MMD-Logit	86.45% ± 3.4%	31.01% ± 9.18%
CLOvE	78.88% ± 6.92%	21.14% ± 2.43%
LEIR (Ours)	78.21% ± 1.3%	**48.17%** ± **2.93%**

Table 3. 10-way test accuracy for each training regime across 10 seeds, with models selected according to their validation accuracy