Generalized Invariant Risk Minimization: relating adaptation and invariant representation learning

Steffen Schneider[1 2 3*] **Shubham Krishna**[1*] **Luisa Eck**[4*]
Wieland Brendel[1] **Mackenzie W. Mathis**[3] **Matthias Bethge**[1]

[1]University of Tübingen [2]IMPRS for Intelligent Systems
[3]EPFL [4]LMU Munich

Abstract

If faced with new domains or environments, a standard strategy is to adapt the parameters of a model trained on one domain such that it performs well on the new domain. Here we introduce Generalized Invariant Risk Minimization (G-IRM), a technique that takes a pre-specified adaptation mechanism and aims to find invariant representations that (a) perform well across multiple different training environments and (b) cannot be improved through adaptation to individual environments. G-IRM thereby generalizes ideas put forward by Invariant Risk Minimization (IRM) and allows us to directly compare the performance of invariant representations with adapted representations on an equal footing, i.e., with respect to the same adaptation mechanism. We propose a framework to test the hypotheses that (i) G-IRM outperforms IRM, (ii) G-IRM outperforms Empirical Risk Minimization (ERM) and (iii) that more powerful adaptation mechanisms lead to better G-IRM performance. Such a relationship would provide a novel and systematic way to design regularizers for invariant representation learning and has the potential to scale Invariant Risk Minimization towards real world datasets.

1 Introduction

The ability to learn representations that generalize to changes of the data distribution is a key challenge in both domain adaptation and domain generalization. Recently, Invariant risk minimization (IRM; 1) was proposed to learn representations that are simultaneously optimal across all training domains. However, to the best of our knowledge, no efforts succeeded in scaling the original formulation of IRM to real-world datasets. In fact, Gulrajani and Lopez-Paz (5) showed that IRM does not significantly outperform empirical risk minimization (ERM) in absence of a model selection strategy. Recently, Rosenfeld et al. (22) discussed theoretical limitations of standard IRM especially in non-linear problems.

The reason behind this failure might be rooted in the particular formulation of IRM that relies only on the last linear readout. Adapting the last layer alone has been demonstrated to be insufficient in many transfer learning settings (14). More importantly, supervised domain adaptation (SDA; 23) techniques typically adapt parameters along the network depth because distribution shifts in real-world datasets cause distribution shifts throughout the network (24).

To address this problem, we here propose Generalized IRM (G-IRM) which takes a pre-specified adaptation mechanism and aims to find representations that are simultaneously optimal across the training environments, meaning the representations cannot be improved on individual domains using the given adaptation mechanism. This lets us utilize the aforementioned domain adaptation

[*] Equal contribution.

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

mechanisms as regularizers for invariant representation learning, and lets us probe the relationship between the adaptation mechanisms and their corresponding invariant representations.

Our contribution is two-fold: First, we propose G-IRM as a new member in the family of invariant representation learning techniques. Second, we describe an evaluation protocol with multiple benchmark problems for G-IRM and we investigate whether mechanisms from established domain adaptation techniques could be leveraged for obtaining better representations (which we hypothesize to be possible). If true, this would enable a principled way to design new invariant representation learning techniques, provide a novel link between adaptive and invariant representation learning, and potentially enable scaling of G-IRM to practically relevant large-scale problems.

Related work. Alternatives for domain generalization (DG) algorithms beyond IRM include risk extrapolation (REx; 16) as a practical way and extension of min-max optimization across environments. Follow up work considered variations of the regularizer, e.g. using score matching (SIL; 11). Zhao et al. (26) explore limitations of ERM for generalization and trade-offs in domain-invariant representation learning from an information-theoretic perspective, including generalization bounds based on domain mismatch. Importantly, Gulrajani and Lopez-Paz (5) show that in absence of dedicated (unsupervised) strategies for hyperparameter tuning, ERM outperforms all existing DG methods. Rosenfeld et al. (22) discuss theoretical limitations of IRM, particularly in non-linear problems.

2 Methodology

We are given data sets $\mathcal{D}_e = \{\mathbf{x}_i, y_i\}_{i=1}^{n_e}$ from multiple training environments $e \in \mathcal{E}_t$ and (unknown) test environments $e \in \mathcal{E}\backslash\mathcal{E}_t$. Consider a model $h_{\boldsymbol{\sigma},\boldsymbol{\alpha}_e} : \mathcal{X} \mapsto \mathcal{Y}$ with *shared* parameters $\boldsymbol{\sigma}$ and *adaptation* parameters $\boldsymbol{\alpha}_e$. The adaptation parameters are allowed to adapt to each environment. We train the model parameters in three different ways:

Empirical Risk Minimization (ERM). In the baseline case we do not adapt $\boldsymbol{\alpha}_e$ to each environment, but instead train $\boldsymbol{\sigma}$ and $\boldsymbol{\alpha}$ on the standard weighted empirical risk $\mathcal{R}_e(\boldsymbol{\sigma}, \boldsymbol{\alpha}) = \mathbb{E}_{\mathbf{x},y\sim\mathcal{D}_e}[\ell_{\text{cross-entropy}}(h_{\boldsymbol{\sigma},\boldsymbol{\alpha}}(\mathbf{x}), y)]$ across training environments,

$$\min_{\boldsymbol{\sigma},\boldsymbol{\alpha}} \sum_{e\in\mathcal{E}_t} \lambda_e \mathcal{R}_e(\boldsymbol{\sigma}, \boldsymbol{\alpha}) \tag{1}$$

We consider both constant weightings $\lambda_e = 1$ or treat them as tunable hyperparameters for a better baseline performance (cf. Appendix for further motivation).

Supervised Domain Adaptation (SDA). We measure the properties of different adaptation mechanisms by the improvement we can obtain by finetuning the adaptation parameters on each train and test environment. We first train $\boldsymbol{\sigma}$ and a shared $\boldsymbol{\alpha}$ as in ERM, obtaining an optimal parameter configuration $\boldsymbol{\sigma}^*, \boldsymbol{\alpha}^*$. We then adapt the network by introducing environment-specific adaptation parameters $\boldsymbol{\alpha}_e$ on each test environment:

$$\min_{\boldsymbol{\sigma},\boldsymbol{\alpha}} \sum_{e\in\mathcal{E}_t} \lambda_e \mathcal{R}_e(\boldsymbol{\sigma}, \boldsymbol{\alpha}), \qquad \min_{\boldsymbol{\alpha}_e} \mathcal{R}_e(\boldsymbol{\sigma}^*, \boldsymbol{\alpha}_e). \tag{2}$$

We alternatively consider a variant where we jointly train the shared parameters and adapt the adaptation parameters to each environment:

$$\min_{\boldsymbol{\sigma},\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_E} \sum_{e\in\mathcal{E}_t} \lambda_e \mathcal{R}_e(\boldsymbol{\sigma}, \boldsymbol{\alpha}_e). \tag{3}$$

Generalized Invariant Risk Minimization (G-IRM). In G-IRM we do not adapt $\boldsymbol{\alpha}$ to each environment, but instead aim to to find model parameters $\boldsymbol{\sigma}$ and $\boldsymbol{\alpha}$ such that $\boldsymbol{\alpha}$ is simultaneously optimal for each individual environment. This leads to the optimization problem

$$\min_{\boldsymbol{\sigma},\boldsymbol{\alpha}} \sum_{e\in\mathcal{E}_t} \mathcal{R}_e(\boldsymbol{\sigma}, \boldsymbol{\alpha}) \quad \text{s.t. } \boldsymbol{\alpha} \in \arg\min_{\boldsymbol{\alpha}'} \mathcal{R}_e(\boldsymbol{\sigma}, \boldsymbol{\alpha}') \tag{4}$$

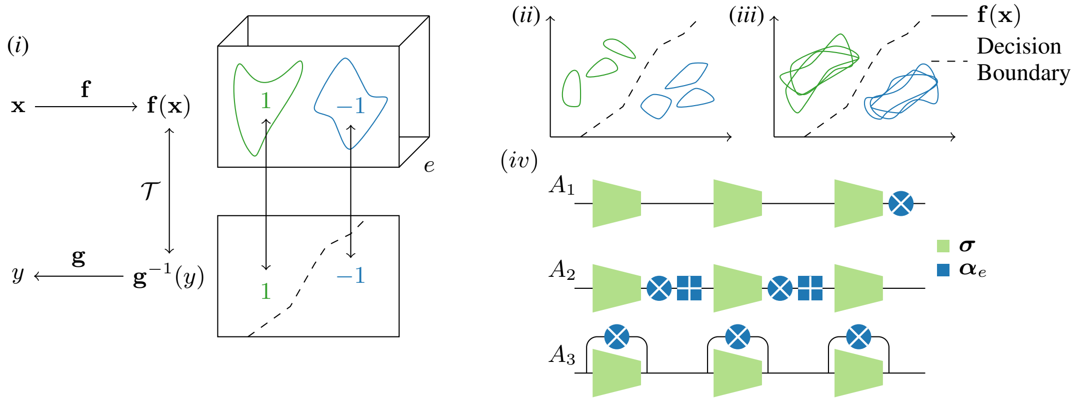


Figure 1: (*i*) G-IRM with affine adaptation ensures an *invariant alignment* between the image of $\mathbf{f}$ and the pre-image of $\mathbf{g}$. (*ii*) The alignment is possible with disjunct sets of features across environments, or (*iii*) yield directly domain invariant representations (*iii*). (*iv*) Adaptation mechanisms: A_1 reproduces IRM, A_2 tunes affine parameters after each convolutional block, and A_3 tunes residual adapters.

which we relax by using a gradient regularizer similar to Arjovsky et al. (1), yielding

$$\min_{\boldsymbol{\sigma},\boldsymbol{\alpha}} \sum_{e\in\mathcal{E}_t} \left(\mathcal{R}_e(\boldsymbol{\sigma},\boldsymbol{\alpha}) + \lambda \|\nabla_{\boldsymbol{\alpha}'|\boldsymbol{\alpha}'=\boldsymbol{\alpha}} \mathcal{R}_e(\boldsymbol{\sigma},\boldsymbol{\alpha}')\|_2^2\right). \quad (5)$$

Note that G-IRM reduces to IRM (1) if the adaptation parameters $\boldsymbol{\alpha}$ represent a linear readout at the end of the network.

2.1 Generalizing IRM allows to learn higher order invariances at fixed model capacity

We consider three families of adaptation mechanisms, as outlined in Fig. 1 (*iv*): We start with fine-tuning the last layer weights, which makes G-IRM equivalent to IRM (Fig. 1 (*iv*), A_1). Given the success of mechanisms for affine adaptation layers, we include this as the second mechanism (Fig. 1 (*iv*), A_2). Affine adaptation layers are inserted after each convolutional layer except for the final output to allow for distributed adaptation. Finally, we consider residual adapters similar to the structure proposed by Rebuffi et al. (21): In parallel to each convolutional layer, we add a set of additional adaptation parameters (Fig. 1 (*iv*), A_3). For A_3, we consider applying the adaptive weight across input or output channels only, across kernel dimensions only, or across all dimensions.

These adaptation mechanisms induce different biases regarding the network's invariance properties: Consider an example of an adaptation mechanism $\mathcal{T}$ embedded between two learnable network modules $\mathbf{f}, \mathbf{g}$. G-IRM learns an invariant mapping for aligning the image of $\mathbf{f}$ to the pre-image of $\mathbf{g}$ irrespective of the training environment (Fig. 1(i)). In particular, at its optimum, we will be unable to find a $\mathcal{T}$ which improves the risk on any of the training environments. We note that this scheme alone is not sufficient to guarantee that the *representation* is invariant: Both the (non-invariant) alignment of multiple domains in Fig. 1(ii) and the invariant representations depicted in Fig. 1(iii) are valid solutions in G-IRM. We however expect that solutions as shown Fig. 1(ii) are less practically relevant when objective functions like the cross-entropy are employed, and classifiers are not perfect.

2.2 Metrics

We focus on four types of metrics. First, we directly consider the test accuracies obtained by all models (ERM, G-IRM, SDA) on all training and test domains. We also report the test accuracies after adaptation of the (fixed) adaptation parameters after G-IRM training as a measure of invariance. As a summary statistic, we report the worst case test error. Second, the regularizer of G-IRM indicates the quality (invariance) of the representation. A value closer to zero indicates that the adaptation parameters cannot improve the loss on any of the target domains, indicating invariance. We will consider the value of the regularizer for any mechanism A_1–A_3 for both model selection and further analysis. Third, we consider options for comparing representations obtained by the three algorithms. Kornblith et al. (13) introduced a new similarity index, Centered Kernel Alignment (CKA), and showed that it can measure meaningful (higher order) similarities between high dimensional representations in DNNs; we consider the variant using a linear or RBF kernel. Using CKA, we process images with the same label, but from different environments. We compare representations in the ERM-trained models (expectation: low similarity), in the G-IRM-trained model (cf. Fig. 1, either (*ii*) low or (*iii*) high similarity is possible), and among the two models (expectation: low similarity). Finally, we quantify the invariance of the representations learned by domain and class classifiers: At multiple network depths, we extract features, and report cross-validated decoding performance for both classifiers, using logistic regression or support vector classifiers.

3 Experimentation Protocol

We investigate three hypotheses: First, we will test H_1: G-IRM with the optimal mechanism A^* performs better than IRM, against H_0: G-IRM and IRM obtain the same performance. Second, we consider H_1': G-IRM with the optimal mechanism A^* performs better than ERM, against H_0': G-IRM and ERM obtain the same performance. In contrast to Gulrajani and Lopez-Paz (5), we will use selection strategies which are tuned to G-IRM, exploiting a potential correlation between the gradient penalty and the test accuracy (Fig. 2). For the special case of IRM, we will challenge the claims by (16)**?**) Finally, we address our core hypothesis H_1'': For a given model architecture and dataset, the performance of adaptation mechanism A_j is positively correlated with the worst case test accuracy of G-IRM (Fig. 3), against H_0'' : there are no correlations between SDA performance and G-IRM performance.

Figure 2

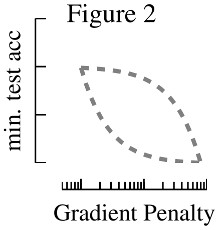
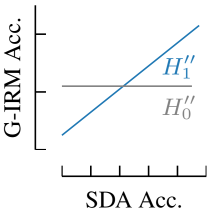


Figure 3



For each experiment, we select a dataset and a baseline network structure depending on the overall data complexity. We start with synthetic toy data, then proceed with variants of Colored MNIST(1), extended by including simultaneous variations of background and foreground (cf. Supplement). We then add other small digit datasets like SVHN (19), Synth Digits (4) and USPS (10), and consider augmented versions of the datasets (MNIST-C, 18). Eventually, we consider more naturalistic images like PACS (17), VLCS (3), ImageNet-C (7) and ImageNet-R (8).

We consider the splits into shared and adaptation parameters outlined in §2.1. We train shared parameters using ERM and G-IRM, and shared and adapted parameters using SDA. Performance is evaluated on all environments using metrics outlined in § 2.2.

Baseline and Variations. For H_1, our most important baseline is IRM; for H_1', we compare performance mainly against ERM. We also report performances of REx (16), SIL (11) and other methods implemented in DomainBed (5). Adaptation mechanisms A_2,A_3 can be varied by only inserting adaptation modules in certain network layers; in addition, we can combine approaches A_1, A_2, A_3. Finally, we will closer study alternative relaxations (and approximations) of G-IRM if we encounter optimization issues.

Architectures. Based on the complexity of tasks, we choose the appropriate model from the following discussed network architectures: First, we use the fully connected network used by Arjovsky et al. (1) with increasing number of layers for synthetic data and Colored MNIST. Each hidden layer is followed by a ReLU or ELU nonlinearity and we do not use any batch/group normalization. Second, we use Convolutional Networks with four, six or eight convolutional layers and these convolutional layers (kernel size 3 or 5) are followed by fully connected layers with ReLU or ELU nonlinearity. The number of channels for each convolution is of the form 2^x where $x \in \{5, 10\}$. The number of hidden units in the fully connected layer of both the discussed architectures is treated as a hyperparameter and varies from $\{2^5, 2^{10}\}$ We optionally use Batch or Group normalization for better convergence. Finally, on real-world image data we use well-known architectures widely used and implemented in torchvision (20), e.g. AlexNet (15), residual networks (ResNet18 and ResNet50; 6) and DenseNet121 (9). We will run a more limited hyperparameter sweep on these larger architectures.

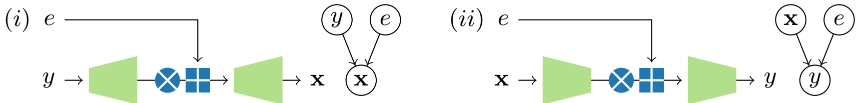


Figure 4: Data generating processes for (*i*) anti-causal prediction and (*ii*) causal prediction.

Synthetic Data Distributions. We now consider two controlled experiments with a fully known data generating process, using a student-teacher setup (Fig. 4). The teacher is a randomly initialized DNN serving as the data distribution. First, we use the teacher to define $p(\mathbf{x}|y, e)$. We specify a correlation between y and e, sample from the model, and aim to infer the original class labels, yielding an inference problem in the anti-causal direction (25). Second, we build the reverse setup to construct a classification problem in the causal direction, starting from $\mathbf{x}$ and e, and parametrizing $p(y|\mathbf{x}, e)$ with the teacher. Again, we aim to infer the class label from $\mathbf{x}$. In both cases, in contrast to real datasets, we can ensure that the data complexity and environment-specific process matches our model, and can additionally study cases where the student is over- or under-parametrized.

Hyperparameters Selection. We focus on classification tasks and in all cases optimize the negative log-likelihood, using (Stochastic) Gradient Descent (SGD; 2) or Adam (12); we use batch training for small scale experiments, and batch sizes in $\{32, 64, \ldots, 256\}$. All hyperparameters are tuned with random search (log-uniform, cf. Table 2). The most critical hyperparameter is the G-IRM penalty coefficient. We follow a linear warmup stategy (1), sampling the number of warmup iterations uniformly between 10% and 90% of the total training iterations.

G-IRM penalty	$[10^0, 10^{10}]$
learning rate	$[10^{-2}, 10^{-5}]$
weight decay	$[10^{-2}, 10^{-6}]$
hidden dim	$[2^5, 2^8]$

Tab. 2: Log-uniform sampling

We explore four model selection strategies (S_1,S_2, S_{3a}, S_{3b}): First, we estimate the topline performance by selecting hyperparameters that minimize the maximal error across all environments (S_1; 1, 16). Second, we consider different splits of the training datasets into training and validation subsets and select models based on the min-max error across all training domains (S_2; cf. 5).

Importantly, S_1 is not applicable in practice, while S_2 potentially provides an unfair advantage to ERM, which is also reflected in the results obtained by Gulrajani and Lopez-Paz (5). We therefore finally explore selection strategies more tuned to the G-IRM algorithm: We either select based on highest training accuracy s.t. to sufficiently low gradient penality (S_{3a}), or select based on low gradient penality s.t. sufficiently high training accuracy (S_{3b}). In both cases, we first subselect based on the sufficient criterion by considering all experiments in the lower quantile of this metric, and then greedily pick hyperparameters according to the primary metric.

Ablative studies. To better stress the empirical difference between IRM and G-IRM and to investigate H_1, we will always run ablative studies where we (*i*) test the novel mechanisms A_2, A_3 in isolation, (*ii*) test the adaptation mechanisms A_2, A_3 in conjuction with A_1 and (*iii*) contrast this to IRM (A_1) only. Besides, we extensively test different options for possible adaptation mechanisms and model architectures, as outlined before.

Evaluation and Results. Evidence in favor of H_1 implies that the choice of adaptation mechanism matters in the practical application of G-IRM. Leveraging CKA, the different penalty terms for A_1–A_3, and decoding performance of the domain classifier across the representational hierarchy of the DNN allows to study differences in the learned representations: G-IRM could potentially yield more invariant intermediate representations (Fig. 1) than IRM, where only invariance w.r.t. to the last layer is required. If we fail to reject H_0, this implies that the original IRM formulation is sufficient for learning invariant representations, extending the original claim (1).

Evidence in favor of H_1' implies that the adaptation parameters used as regularizer enforces the learned feature representations to be invariant in a way applicable in practical settings, in constrast to the results in favor for H_0' by Gulrajani and Lopez-Paz (5).

Evidence in favor of H_1'' suggests that good adaptation techniques can be used as regularizers for learning invariant representations. For distributed adaptation (A_2, A_3), the CKA similarity index for the G-IRM learned representations will be high both within and at the end of the network. Failure to reject H_0'' implies that invariant representations can be principally obtained without special choices for the adaptation method (in conjunction with H_0 and H_1') or, indicate that further work is needed on domain adaptation techniques beyond IRM (conjunction with H_1').

4 Conclusion

We outlined a novel experimental framework for investigating a possible link between domain adaptation and invariant representation learning. Our proposed algorithm, G-IRM, allows to leverage adaptation mechanisms as a regularizer for obtaining invariant representations.

On a broad variety of datasets, we explore the relationship between performance of an adaptation technique for supervised domain adaptation and G-IRM. Evidence for such a link will allow a rigorous new way to design algorithms for invariant representation learning, and scale IRM to real-world data, while a negative result will motivate more work on different families of domain generalization techniques beyond IRM.

Acknowledgments

We thank Roland Zimmermann, Yash Sharma and Matthias Kümmerer, for helpful discussions and feedback for improving our manuscript. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting St.S. St.S. acknowledges his membership in the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program. This work was supported by the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ: 01IS18039A).

References

[1] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019.

[2] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[3] C. Fang, Y. Xu, and D. N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.

[4] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[5] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. *CoRR*, abs/2007.01434, 2020.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] D. Hendrycks and T. G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *CoRR*, abs/1807.01697, 2019.

[8] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *CoRR*, abs/2006.16241, 2020.

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[10] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[11] D. Idnani and J. C. Kao. Learning robust representations with score invariant learning.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[13] S. Kornblith, M. Norouzi, H. Lee, and G. E. Hinton. Similarity of neural network representations revisited. In *ICML*, 2019.

[14] S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2661–2671, 2019.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] D. Krueger, E. Caballero, J. Jacobsen, A. Zhang, J. Binas, R. L. Priol, and A. Courville. Out-of-distribution generalization via risk extrapolation (rex). *CoRR*, abs/2003.00688, 2020.

[17] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[18] N. Mu and J. Gilmer. Mnist-c: A robustness benchmark for computer vision. *CoRR*, abs/1906.02337, 2019.

[19] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[21] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[22] E. Rosenfeld, P. Ravikumar, and A. Risteski. The risks of invariant risk minimization. *CoRR*, abs/2010.05761, 2020.

[23] S. Schneider, A. S. Ecker, J. H. Macke, and M. Bethge. Multi-task generalization and adaptation between noisy digit datasets: An empirical study. In *Neural Information Processing Systems (NeurIPS), Workshop on Continual Learning*, 2018.

[24] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge. Removing covariate shift improves robustness against common corruptions. *Forthcoming in Proceedings of NeurIPS 2020*, 2020.

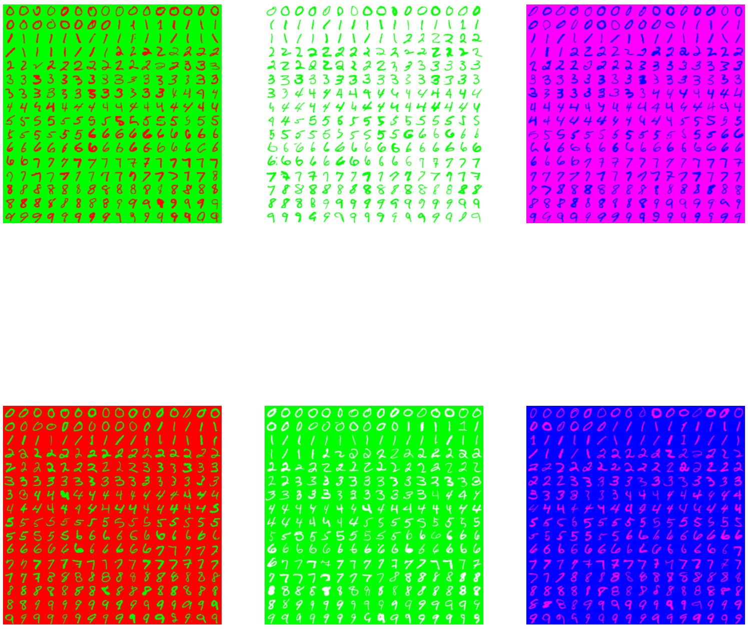
[25] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij. On causal and anticausal learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 459–466, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

[26] H. Zhao, R. T. des Combes, K. Zhang, and G. Gordon. On learning invariant representation for domain adaptation. *CoRR*, abs/1901.09453, 2019.

—SUPPLEMENTARY MATERIAL—
GENERALIZED INVARIANT RISK MINIMIZATION: RELATING ADAPTATION AND INVARIANT REPRESENTATION LEARNING

Extended Colored MNIST Task

Example training and testing environments for the extended colored MNIST task. Background and foreground can optionally be correlated with the label. We utilize this task as an intermediate dataset between the original colored MNIST dataset, and more complex data distributions like SVHN/Synth Digits, PACS or VLCS.



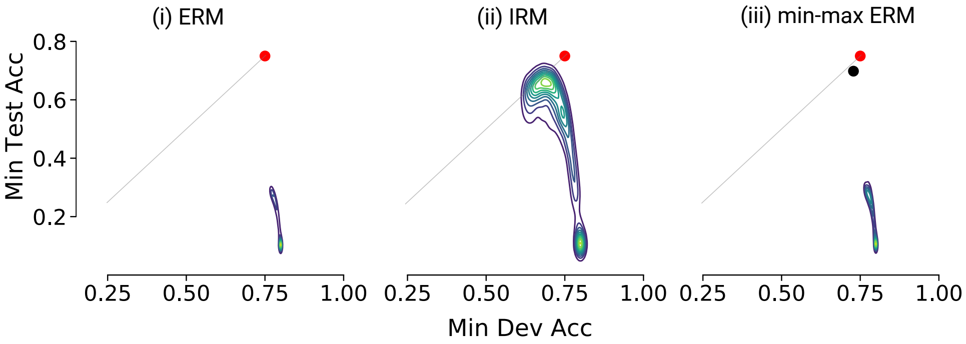
IRM Reproduction Study I

Reproduction study of the original IRM experiments with train environments $\mathcal{E}_t = \{0.1, 0.2\}$, and test on $e = 0.9$, to visualize our motivation for proposing hyperparameter selection schemes $S_{3a,b}$ in the main paper. Selecting hyperparameters based on the worst case training performance (upper table) yields only slight improvements over ERM. Selection based on the test performance is necessary (lower table) to observe the originally reported gain.

Plotting all samples using a contour plot (red dot indicates maximum possible performance at 75%; training is done with 25% label noise) demonstrate a slightly negative slope between the train and test set accuracies, making it impossible to select good hyperparameters based on the training set accuracies. This motivates alternative selection schemes based on the regularizer values of IRM as outlined in our proposal.

$N = 89889$	Epoch	worst train		best train		test	
Model (on train)	≥ 500	mean	std	mean	std	mean	std
erm	500	0.8052	0.0043	0.8988	0.0033	0.1034	0.0054
min-max erm	500	0.8055	0.0044	0.9016	0.0029	0.0978	0.0061
irm	800	0.7926	0.0014	0.8643	0.0028	**0.2520**	0.0077

$N = 89889$	Epoch	worst train		best train		test	
Model (on test)	≥ 500	mean	std	mean	std	mean	std
erm	900	0.7251	0.0606	0.7661	0.0951	0.4236	0.1477
min-max erm	500	0.7263	0.0754	0.7703	0.1122	0.4232	0.2144
irm	500	0.6933	0.0013	0.6993	0.0033	0.6840	0.0059



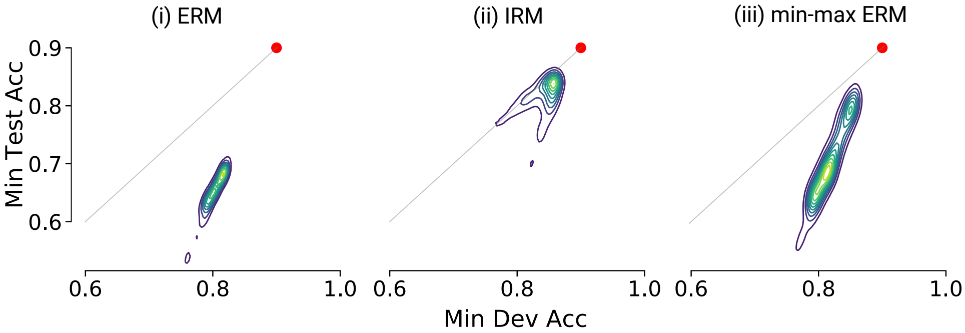
We now modify the training environments to $\mathcal{E}_t = \{0.0, 0.05, 0.075, 0.3, 0.35, 0.4\}$, and test on $e \in \{0.15, 0.5, 0.7, 0.8, 0.9\}$. The training now includes environments with lower correlation between color and label than between digit shape and label (75 %, cf. red dot); we kept all search ranges except for the maximum number of epochs (increased to 1000) according to the original IRM experiment.

Selecting hyperparameters based on the worst case training performance (upper table) now yields a comparable performance to selection based on the test set, and the contour plots of all considered samples reveal a slightly positive correlation.

Note that it is crucial to consider a better baseline than ERM in this case: Using the min-max formulation of ERM, i.e., minimizing the worst case expected error across training environments, results in effectively training the model on the environment $e = 0.4$ which the weakest correlation between color and label, improving the overall performance. In our protocol, we reflect this by considering the optimal weighting of environment risks for the ERM optimizer.

$N = 38033$	Epoch	worst train		best train		test	
Model (on train)	≥ 100	mean	std	mean	std	mean	std
erm	200	0.8349	0.0115	0.9134	0.0043	0.7307	0.0255
min-max erm	700	0.8629	0.0046	0.8987	0.0066	0.8123	0.0036
irm	800	0.8683	0.0034	0.8859	0.0058	0.8443	0.0052

$N = 38033$	Epoch	worst train		best train		test	
Model (on test)	≥ 100	mean	std	mean	std	mean	std
erm	200	0.8349	0.0115	0.9134	0.0043	0.7307	0.0255
min-max erm	200	0.8587	0.0105	0.8921	0.0078	0.8262	0.0266
irm	200	0.8605	0.0015	0.8660	0.0015	0.8540	0.0040


(i) ERM
(ii) IRM
(iii) min-max ERM
Min Test Acc
Min Dev Acc