Do as your neighbors: Invariant learning through non-parametric neighbourhood matching

Andrei Liviu Nicolicioiu [1 2] **Jerry Huang** [1 2] **Dhanya Sridhar** [1 2] **Aaron Courville** [1 2]

Abstract

Invariant learning methods aim to obtain robust features that can be used in the same way in multiple environments and can generalize out-of-distribution. This paper introduces a novel method to achieve this, called Invariant KNN. We are guided by the idea that *robust features should elicit an invariant non-parametric predictor* across domains. For this, we create a K-nearest neighbors predictor from each training environment and constrain them to be the same. We prove experimentally that this approach leads to invariant predictors which learn to use the robust features in the data and generalize out-of-distribution. We test our algorithm on a simple but popular benchmark and demonstrate that it is both competitive with other popular algorithms as well as less sensitive to hyperparameter selection.

1. Intro

Current deep-learning methods continue to have progressively better results with each new generation of models in many domains (Jumper et al., 2021; Brown et al., 2020; Alayrac et al., 2022). Nevertheless, all current methods still take shortcuts (Geirhos et al., 2020) in learning by incorporating different biases and spurious correlations (Bolukbasi et al., 2016) found in the data. With the near ubiquity of these models, there is a high risk of propagating and amplifying such biases (Bender et al., 2021).

The field of causal representation learning (Schölkopf et al., 2021; Schölkopf, 2022) gives a formal treatment of such cases and tries to offer solutions. One fundamental idea is to learn to be invariant to some changes between domains. This broad goal was successfully applied to computer vision problems, where the invariances were hand-designed, for example, most self-supervised methods learn to be invariant to some augmentations (Chen et al., 2020; Chen & He, 2020). There are many works making this idea more formal and giving proof of its utility in making causal decisions (Muandet et al., 2013; Peters et al., 2016; Arjovsky et al., 2019; Krueger et al., 2021; Wang & Veitch, 2022).

Most related to our work are methods like Invariant Risk Minimization (IRM) (Arjovsky et al., 2019). IRM method constrains the optimal predictor from different environments to be invariant across such domains, but this is challenging in both their initial formulation as well as in their IRMv1 relaxation. We use a similar idea, where instead of an optimal predictor, we create a non-parametric predictor in each environment and we constrain them to be invariant. One possible advantage of this approach is that the non-parametric predictor could be close to optimal and does not require learning, so the optimization should be easier.

Guided by this principle of *invariant non-parametric predictors*, we propose two regularizations. The first one is equivalent to matching the global average of class conditional embedding. This is related to methods like (Sun & Saenko, 2016; Long et al., 2015; Tachet des Combes et al., 2020). The second constraint is based on the idea of matching local averages between environments and is more flexible.

More specifically, our method works in a multi-environment setting, by learning a robust feature extractor $\phi(x)$ shared across domains. To compute the regularisation, we first make predictions using a K-nearest neighbors approach, using **neighbors from a different environment**. The invariance constraint is that the prediction should be the same regardless of the domain of the neighbors. We show empirically that this method is able to learn invariant predictions.

Overall, this work has the following contributions:

1. We propose a **method for invariance learning** (iKNN) based on the idea of domain invariance of non-parametric methods.

2. We show that iKNN is able to recover the robust predictor in simple standard settings.

3. We show that this method is somewhat **less sensitive**

[*] Equal contribution [1] Mila, Montreal, Canada [2] Université de Montréal, Canada. Correspondence to: Andrei Nicolicioiu <andrei.nicolicioiu@mila.quebec>.

Published at the ICML 2023 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA. Copyright 2023 by the author(s).

to hyperparameters, this being a big advantage since hyperparameter search and model selection are very challenging in out-of-distribution settings.

2. Related Work

Invariance regularisers. Recent methods learn a predictor that generalizes out-of-distribution by putting an invariance constraint on the model predictions as an additional loss during training (Arjovsky et al., 2019; Krueger et al., 2021; Wald et al., 2021; Eastwood et al., 2022; Shi et al., 2022). Arjovsky et al. (2019) optimize the features for an optimal predictor across the domains. Because the optimal IRM loss is hard to optimize as it requires a bilevel optimization, the authors propose a relation IRMv1. Subsequent work (Kamath et al., 2021) has shown that the practical IRMv1 loss does not capture the desired invariances in all cases. Krueger et al. (2021) propose the principle of risk extrapolation (REx) to achieve invariant prediction by enforcing equal risks across training domains and is obtained by constraining the variance of the risks (V-REx). Models with perfect calibration across multiple domains are shown (Wald et al., 2021) to generalize out-of-distribution. The work of Eastwood et al. (2022) show that the domain generalization problem is better viewed as solving the most probable problem (quantile risk) instead of the average (mean risk) or worst-case(maximum risk) problem. In order to match the distributions of the features between domains, Shi et al. (2022) proposes to compute the gradients of the losses with respect to the parameters on different training domains and constrain them to be orthogonal (small inner product). This is a challenging second-oder optimization, thus the authors solve it via a meta-learning approach denoted as Fish, which requires computing an inner-loop.

Distributions matching. Some methods explicitly try to match the distributions of the features between different environments. Although many assume access to unlabeled testing domain data, they can also be applied between training domains (Gulrajani & Lopez-Paz, 2021). Sun & Saenko (2016) introduces a method (deep Coral) that regularises the features of a deep network such that their covariance is the same between training and testing domains. In practice, in popular benchmarks (Gulrajani & Lopez-Paz, 2021; Koh et al., 2021) the implementation of this method optimizes the difference between both the means (first moment) and covariances (second moment) of different environments. Maximum Mean Discrepancy (MMD) is used to minimize the differences between environments in multiple methods. MMD is defined (Gretton et al., 2006) as the maximum mean difference of any scalar function of the samples. A collection of fixed functions are used in practice, by defining a set of Gaussian kernels with different bandwidths (Li et al., 2018; Long et al., 2015). Related to this, the idea of kernel mean embeddings (Muandet et al., 2017) is also used to match the distributions across training domains and learn an appropriate re-weighting of the samples as in Gretton et al. (2009); Zhang et al. (2013); Tachet des Combes et al. (2020).

Invariant Features. Ye et al. (2021) prove that learning features with small variations across the training environments is important for increasing out-of-distribution generalization. Wang et al. (2022) propose a method to recover a subspace of invariant features. Their ISR-Mean method is based on the idea of creating a set of class-conditioned embeddings, one for each training environment and extracting the invariant subspace by selecting the most stable eigenvectors given by PCA. This method is applied post-hoc to already-learned features on realistic datasets. As said in subsection 4.3 the simplified setting of our method $\mathcal{L}_{mean}$ also uses conditional mean embeddings to optimize the features, but our method has the advantage that we obtain robust features during training.

3. Invariant Non-parametric predictor

IRM (Arjovsky et al., 2019) aims to find a data representation such that the *optimal* predictor on top of that representation is the same across all environments. The main IRM objective, containing a bilevel optimization problem, is hard to use, therefore in practice it is often relaxed into IRMv1. In this work, use a related objective, of optimizing the features such that each environment defines the same non-parametric predictor. Since there is no learning involved in creating the non-parametric predictor, our objective is easy to optimize.

Key idea: We constrain the features to obtain the same non-parametric predictor across multiple domains.

Another type of useful invariance for domain generalization is the following Conditional Distribution Invariance (Wang & Veitch, 2022) between environments $\mathcal{E}_A$ and $\mathcal{E}_B$:

$$P_{\mathcal{E}_A}[\phi(X)|Y] = P_{\mathcal{E}_B}[\phi(X)|Y] \iff \phi(X) \perp\!\!\!\perp E|Y \quad (1)$$

The simplest constraint in this direction is to use the expected value of the class-conditioned features across each environment:

$$\mathbb{E}_{\mathcal{E}_A}[\phi(X)|Y] = \mathbb{E}_{\mathcal{E}_B}[\phi(X)|Y] \quad (2)$$

We show (subsection 4.3) that a simplified version of our constraint ($\mathcal{L}_{mean}$) has this type of invariance.

To gain more flexibility and move towards matching the entire distribution, we aim to obtain a data representation that produces an invariant non-parametric predictor.

Definition (Invariant Non-parametric predictor): A data representation $\phi : \mathcal{X} \to \mathcal{H}$ elicits an invariant non-parametric predictor w^* across environments $\mathcal{E}$ if the data from each environment creates the same predictor:

$$w^*(x) = w_A(x) = w_B(x), \quad \forall x \in \mathcal{E}, \forall \mathcal{E}_A, \mathcal{E}_B \tag{3}$$

where the non-parametric predictor w_i is created using data from environment $\mathcal{E}_i$. In our case the non-parametric predictor is a K-nearest neighbors classifier.

Optimizing for features satisfying this invariance has some advantages: 1) ease of learning since we don't have a bi-level or second-order optimization 2) flexibility in matching the distributions, as using different non-parametric classifiers arrive at different forms of distributional matching.

4. Method

In this work, we introduce a method called **Invariant KNN** (iKNN) based on the idea that different environments should create the same non-parametric predictor.

We work in a setting where we have access to multiple training environments (at least 2), each having some aspect of the world changed. We train our model on in-distribution data $\mathcal{D}_{\text{ID}} = \{(X_i, Y_i)|\forall X_i, Y_i \text{ in-distribution}\}$ and test it on out-of-distribution data $\mathcal{D}_{\text{OOD}} = \{(X_i^e, Y_i^e)|\forall X_i, Y_i \text{ out-of-distribution}\}$. Each of these datasets can be split into multiple ID, and OOD environments, respectively.

We use input $X \in \mathbb{R}^M$ (usually an image $M = 3 \times H \times W$) and use a function ϕ to generate some features $\phi(X) \in \mathbb{R}^D$. Then, we use a non-parametric method to make the predictions using these features. $h(\phi(X)) \in \mathbb{R}^C$. where C is the number of classes. In our implementation, we use multi-layer perceptions (MLPs) as ϕ and k-nearest-neighbours classifiers as non-parametric method.

4.1. Non-Parametric prediction

We introduce the general way to predict using a non-parametric method like K-nearest neighbors (KNN) and then show how to use it in the proposed method.

For a given query sample $X_q \in \mathbb{R}^D$, we make a prediction using samples from a support set $\mathcal{S} = \{(X_i, Y_i)|\forall i \neq q\}$ where we have access to one-hot labels $Y_i \in \mathbb{R}^C$, where C is the number of classes. First, we compute distances from X_q to every sample in the support set $\mathcal{S}$ and use them to compute similarities. The prediction will then be a weighted average of the labels in the support set, where the weight is given by these similarities.

If we use only the top-k most similar neighbors $\mathcal{N}_k(X_q) = \{X_j \in \mathcal{S}|\phi(X_j) \in \text{top-k most similar to } \phi(X_q)\} \subseteq \mathcal{S}$ from the support set, we arrive at KNN classifier:

$$p(X_q) = \frac{1}{k} \sum_{X_i \in \mathcal{N}_k(X_q)} \langle X_q, X_i \rangle Y_i \in \mathbb{R}^C \tag{4}$$

We can use cosine similarity as $\langle , \rangle$, although in practice, similar to other deep learning methods (Vaswani et al., 2017), we use only use an inner product, without normalization.

4.2. Invariant KNN

We propose an invariance condition using KNNs in a multiple training environment setting. The same feature extractor ϕ creates embeddings for each sample in each domain. Normally, we predict queries from one domain A using a support set from the same domain, but here we also use a support set from a different domain B to create a different non-parametric predictor. We enforce the constraint that regardless of the domain used, the predictions should be the same.

Practically, we only work at the mini-batch level, where a mini-batch contains samples from multiple domains. We present the method using 2 training domains, but it can be generalized to an arbitrary number of domains where the constraint is put on pairs of environments.

For a sample X from training environment A, we make a prediction using two sets of neighbors. The first set, $\mathcal{N}_k^A(X)$, is formed from the top-k most similar samples in the same environment A, while the second set $\mathcal{N}_k^B(X)$ contains the top-k neighbors from training environment B:

$$\mu_A(X) = \frac{1}{k} \sum_{X_i \in \mathcal{N}_k^A(X)} \langle \phi(X), \phi(X_i) \rangle Y_i \tag{5}$$

$$\mu_B(X) = \frac{1}{k} \sum_{X_i \in \mathcal{N}_k^B(X)} \langle \phi(X), \phi(X_i) \rangle Y_i \tag{6}$$

We propose to constrain these two predictions to be close:

$$\mathcal{L}_{\text{iKNN}}(X) = ||\mu_A(X) - \mu_B(X)|| \tag{7}$$

The $\mathcal{L}_{\text{iKNN}}$ loss is used to obtain invariant features, but we also need another signal for learning useful features. This is orthogonal to the invariant regularizer, and we can use the basic ERM applied to the KNN predictions made using queries and support sets from the same training domain. Optionally, we can also predict using a trainable linear head on top of the features, since they are constrained to be robust.

The final loss used to train the whole model is therefore:

$$\mathcal{L} = \sum_{(X,Y) \in \mathcal{E}_{\text{train}}} [\mathcal{L}_{\text{ERM}}(X, Y) + \lambda \mathcal{L}_{\text{iKNN}}(X)] \tag{8}$$

As other methods (Gulrajani & Lopez-Paz, 2021; Zhang et al., 2022), we use just the ERM loss for a small number of epochs, then we optimize both losses. In practice, to be consistent with other methods, we use the linear head to make the predictions.

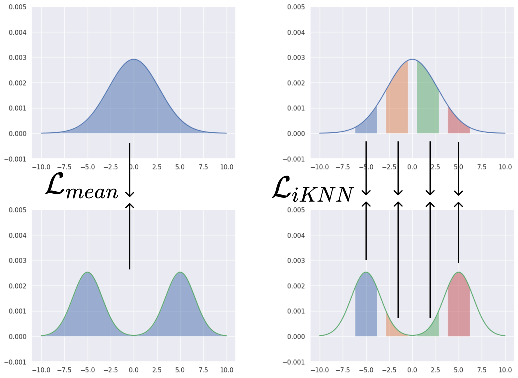

Figure 1. Our proposed $\mathcal{L}_{\text{iKNN}}$ vs the simplified version $\mathcal{L}_{\text{mean}}$. $\mathcal{L}_{\text{mean}}$ puts a constraint on the mean embeddings of features across the complete distributions, while $\mathcal{L}_{\text{iKNN}}$ constraints *local* means across all points in the two distributions. For the given distributions $\mathcal{L}_{\text{mean}}$ is zero already, while $\mathcal{L}_{\text{iKNN}}$ is not, thus it can constrain the distributions further.

4.3. Conditional Feature Invariance

First, we show that a simplified version of our penalty implies invariance between *the expected* value of features, conditioned on the target.

When using the entire dataset as neighbors, our proposed loss from Equation 7 becomes

$$\mathcal{L}_{\text{iKNN}}(X) = \Bigg\| \frac{1}{|\mathcal{E}_A|} \sum_{(X_i,Y_i)\in\mathcal{E}_A} \langle \phi(X), \phi(X_i) \rangle Y_i - \frac{1}{|\mathcal{E}_B|} \sum_{(X_i,Y_i)\in\mathcal{E}_B} \langle \phi(X), \phi(X_i) \rangle Y_i \Bigg\| \tag{9}$$

If we use the linear, un-normalized dot product used as the similarity measure, this is equivalent to:

$$\mathcal{L}_{\text{iKNN}}(X) = || \sum_{c=1}^{C} \langle \phi(X), w_A^c - w_B^c \rangle || \tag{10}$$

where w_A^c and w_A^c are class prototypes for environments A and B, i.e. they are the average embeddings of the samples with a certain class within each environment.

$$w_A^c = \frac{1}{|\mathcal{E}_A^c|} \sum_{X_i \in \mathcal{E}_A^c} \phi(X_i) \quad w_B^c = \frac{1}{|\mathcal{E}_B^c|} \sum_{X_i \in\in \mathcal{E}_B^c} \phi(X_i)$$

with $\mathcal{E}_A^c = \{X_i | (X_i, Y_i) \in \mathcal{E}_A^c, Y_i = e_c\}$ all samples from domain A with class c, e_c element of the standard basis.

We can minimize Equation 10 by alternatively minimizing the difference between the class prototypes from different environments.

$$\mathcal{L}_{\text{mean}} = \sum_c ||w_A^c - w_B^c|| \tag{11}$$

which has it's minimum when

$$\mathbb{E}_{\mathcal{E}_A}[\phi(X)|Y] = \mathbb{E}_{\mathcal{E}_B}[\phi(X)|Y] \tag{12}$$

This simplified formulation resembles the idea of kernel means embeddings (Muandet et al., 2017). In fact, w_A^c, w_B^c are the kernel mean embeddings where the kernel is explicitly defined by the inner product of the features. The idea of matching kernel mean embeddings between training and testing distributions is used by Gretton et al. (2009) and Zhang et al. (2013) to reweight the training examples. In our case, we use the simplified loss that matches the means as a signal for learning invariant features. We note that for certain types of kernels (characteristic kernels), matching the means is equivalent to matching the entire distributions (Muandet et al., 2017).

This is also related to ISR-Mean (Wang et al., 2022) that creates invariant features by applying PCA on class-conditioned mean embeddings, although their method is applied as a post-processing step. While methods like (Sun & Saenko, 2016) use invariance between *unconditional* moments (mean and covariance) as an additional loss, we use *conditional* first-moment matching in the simplified setting.

The $\mathcal{L}_{\text{iKNN}}$ loss from Equation 7 puts a constraint such that each point in domain A has a local neighborhood in domain A and a local neighborhood in domain B that would match, meaning the distributions of the features would be similar. On the other hand, the simplified loss $\mathcal{L}_{mean}$ matches the global mean of the entire distributions. This is broadly represented in Figure 1. Using local neighborhoods gives us more flexibility in matching the two distributions.

5. Experiments

We evaluate iKNN on OOD generalization tasks.

Colored MNIST. Arjovsky et al. (2019) constructs a binary classification problem of MNIST digits while using color as a spurious feature. Each digit is colored either red or green, with a strong positive correlation between color and label at training time. The goal is to learn the robust feature (digit shape) and ignore the spurious feature (color). We train on two environments with a strong correlation between color and shape and evaluate on environments with zero or strong negative correlation.

Main results. From our first experiment in Table 1, we show the performance on different testing environments of the ColorMNIST dataset. As expected, ERM performs better than the rest for the in-distribution setting (ID), but fails for the two increasingly more out-of-distribution (OOD) settings. All invariant learning methods learn stable predictors with small variations in performance across environments.

Our algorithm, iKNN performs as well if not better than the baselines in the out-of-distribution environments.

As model selection is challenging in OOD setting, we select the best hyperparameters using the performance on the environment with zero color-label correlation.

Table 1. iKNN performs comparably against other invariant methods on ColoredMNIST. Results over 3 seeds.

Algorithm	**Test Spurious correlation**		
	Strong (+) (ID)	Zero (OOD)	Strong (-) (OOD)
ERM	**88.4** $\pm$ 0.4	52.6 $\pm$ 0.2	17.5 $\pm$ 1.1
IRM	71.6 $\pm$ 2.3	70.0 $\pm$ 0.5	67.1 $\pm$ 2.1
V-REx	70.1 $\pm$ 1.3	**70.3** $\pm$ 0.7	69.0 $\pm$ 0.7
iKNN	68.7 $\pm$ 0.2	**70.1** $\pm$ 1.0	**70.1** $\pm$ 1.2

Top-K	Strong Positive (ID)	Zero (OOD)	Strong Negative (OOD)
5	72.2	68.0	64.1
10	**74.6**	68.8	62.0
32	70.3	68.9	67.1
64	68.5	69.5	**69.3**
512	70.5	**70.5**	**69.3**

Table 2. Performance of iKNN using different neighborhoods. We always consider the top-K closest neighbors in the mini-batch. Generally, we see that increasing the number of neighbors helps. We use a maximum mini-batch size of 512.

Global vs local distribution matching We ablate the effect of hyperparameter k, while keeping $\lambda = 1000$ on the Colored MNIST performance in Table 2. This parameter can be also seen as the degree of locality used for matching the distributions between environments. We see that k is rather robust and generally performs well, although larger values of k appear to perform better on the out-of-distribution settings. Nevertheless, the performance even with reasonably small k (greater than 32) is competitive with other methods.

Invariance Penalty Sensitivity. In Figure 2 we show the performance of IRM, vREx and iKNN across environments for various values of the invariance penalty weight λ. An invariant method is robust to λ if it has uniform performance across environments for a large range of λ values. To quantify this, we compute the standard deviation of the accuracy across the environments (across rows) and average them across λ values (across columns). We obtain:

Model	IRM	V-REx	iKNN
Avg std ↓	4.7	2.7	**1.8**

where lower is better. This shows that iKNN is less sensitive to choices of λ, and can work well under a larger range of λ values, helping in hyperparameter selection. Since choosing the right hyperparameters is extremely important in an out-of-distribution setting (Gulrajani & Lopez-Paz, 2021), this larger range is an important advantage.

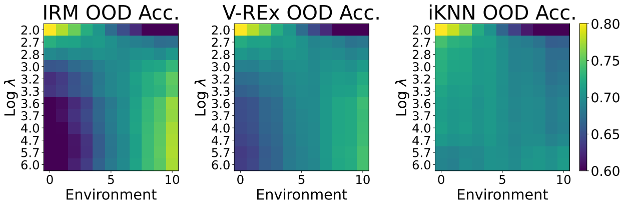

Figure 2. Performance of algorithms on Color MNIST as the λ parameter increases on differing levels of OOD testing environments. Environments go from 0 (color-label correlation 1.0) to 10 (color-label correlation -1.0) and the models are trained on environments 1 and 2. A good invariant model has stable performance across all environments. We see that iKNN is generally more stable as for a larger range of λ values.

6. Implementation details

Dataset. On ColorMNIST, we train two environments, where color and label match in 0.1 and 0.2 times cases. We evaluate on in-distribution (0.1) and out-of-distribution $(0.5$, no correlation; 0.9, negative correlation) environments. The shape of the digits always matches the labels with a probability of 0.25. We downsample images to $2 \times 14 \times 14$. We create validation sets of all environments for model selection and testing sets for reporting the final performance.

Hyperparameters. As feature extractor ϕ, we use an MLP with 2 hidden layers of size 390. We train using Adam optimizer with a learning rate of 0.0001. All invariant learning methods, IRM, V-REx and iKNN use a hyperparameter λ to control the weight of the invariance penalty. For all methods, we search over $\lambda \in [1.0, 10.0, 10^2, 5{*}10^2, 7{*}10^2, 10^3, 1.5{*}10^3, 2 * 10^3, 4 * 10^3, 5 * 10^3, 10^4, 5 * 10^4, 5 * 10^5, 10^6]$

Similar to other methods (Zhang et al., 2022) we select the best hyperparameters using OOD validation data. Model selection is an important aspect for out-of-distribution generalization and it is challenging for all existing methods (Gulrajani & Lopez-Paz, 2021). The proposed iKNN method has larger ranges of good hyperparameters λ (see Section 5), having an advantage in this regard.

7. Conclusion

We introduce iKNN, a simple method aimed to learn robust features based on an invariance constraint of non-parametric (KNN) predictors. We demonstrate that it compares favorably with other, more well-known OOD-generalization methods. While it remains to be determined how the proposed method behaves in more realistic scenarios, the positive initial results suggest that this could be a promising direction for future work.

Acknowledgment. This work was supported by CIFAR.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623, 2021.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

Eastwood, C., Robey, A., Singh, S., Kügelgen, J. V., Hassani, H., Pappas, G. J., and Schölkopf, B. Probable domain generalization via quantile risk minimization. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=6FkSHynJr1.

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., Rasch, M., Schölkopf, B., and Smola, A. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4): 5, 2009.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=lQdXeXDoWtI.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

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., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. WILDS: A benchmark of in-the-wild distribution shifts. In *Proceedings of the International Conference on Machine Learning, ICML*, 2021.

Krueger, D., Caballero, E., Jacobsen, J., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. C. Out-of-distribution generalization via risk extrapolation (rex). In *Proceedings of the 38th International Conference on Machine Learning, ICML*, 2021.

Li, H., Pan, S. J., Wang, S., and Kot, A. C. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409, 2018.

Long, M., Cao, Y., Wang, J., and Jordan, M. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18. PMLR, 2013.

Muandet, K., Fukumizu, K., Sriperumbudur, B., Schölkopf, B., et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

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.

Schölkopf, B. Causality for machine learning. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 765–804. ACM, 2022.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634, 2021.

Shi, Y., Seely, J., Torr, P. H. S., Narayanaswamy, S., Hannun, A. Y., Usunier, N., and Synnaeve, G. Gradient matching for domain generalization. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer, 2016.

Tachet des Combes, R., Zhao, H., Wang, Y.-X., and Gordon, G. J. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33:19276–19289, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wald, Y., Feder, A., Greenfeld, D., and Shalit, U. On calibration and out-of-domain generalization. *Advances in neural information processing systems*, 34:2215–2227, 2021.

Wang, H., Si, H., Li, B., and Zhao, H. Provable domain generalization via invariant-feature subspace recovery. In *International Conference on Machine Learning*, pp. 23018–23033. PMLR, 2022.

Wang, Z. and Veitch, V. A unified causal view of domain invariant representation learning. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*, 2022. URL https://openreview.net/forum?id=-l9cpeEYwJJ.

Ye, H., Xie, C., Cai, T., Li, R., Li, Z., and Wang, L. Towards a theoretical framework of out-of-distribution generalization. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=kFJoj7zuDVi.

Zhang, J., Lopez-Paz, D., and Bottou, L. Rich feature construction for the optimization-generalization dilemma. In *International Conference on Machine Learning*, pp. 26397–26411. PMLR, 2022.

Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. Domain adaptation under target and conditional shift. In *International conference on machine learning*, pp. 819–827. PMLR, 2013.