Can Subnetwork Structure be the Key to Out-of-Distribution Generalization?

Dinghuai Zhang [1] **Kartik Ahuja** [1] **Yilun Xu** [2] **Yisen Wang** [3] **Aaron Courville** [1]

Abstract

Can models with particular structure avoid being biased towards spurious correlation in out-of-distribution (OOD) generalization? Peters et al. (2016) provides a positive answer for linear cases. In this paper, we use a functional modular probing method to analyze deep model structures under OOD setting. We demonstrate that even in biased models (which focus on spurious correlation) there still exist unbiased functional subnetworks. Furthermore, we articulate and demonstrate the functional lottery ticket hypothesis: full network contains a subnetwork that can achieve better OOD performance. We then propose Modular Risk Minimization to solve the subnetwork selection problem. Our algorithm learns the subnetwork structure from a given dataset, and can be combined with any other OOD regularization methods. Experiments on various OOD generalization tasks corroborate the effectiveness of our method.

1. Introduction

Despite the remarkable progress we have witnessed in neural-network-based machine learning, the stories of failures continue to accumulate (Geirhos et al., 2020). Many of these failures are attributed to models exploiting spurious correlations or shortcuts (i.e. factors that are not used to generate the label). A colloquial example comes from (Beery et al., 2018) where the authors show how a neural network trained to distinguish cows from camels exploits shortcut such as background color for prediction. In a much more concerning example, (DeGrave et al., 2020) show how machine learning systems trained to detect COVID-19 exploited the data source (e.g., hospital) to artificially boost inference performance.

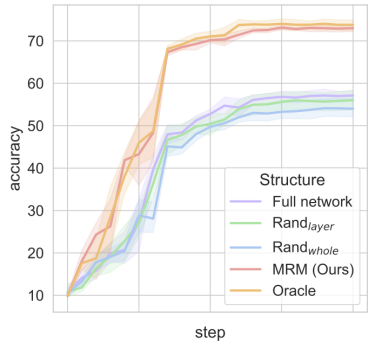


Figure 1. OOD performance of models with different structures when trained with ERM algorithm on FULLCOLOREDMNIST. The oracle subnetwork and our MRM method significantly surpass the performance of full network. See details in Section 4.

What causes these failures? Recent works (Peters et al., 2016; Arjovsky et al., 2019) argue the principle of empirical risk minimization (ERM) is at fault: if the data is generated from a fully observed causal bayesian network (CBN), then ERM would typically use all the features in the Markov blanket including those which are not the causes of the label. It may consequently fail to perform well under distribution shifts. This is known as the out-of-distribution (OOD) generalization problem. In an effort to alleviate the problem, Peters et al. (2016) proposes to first identify the target's causal parents, and constrain the model structure by only updating the parameters for the parents. Nevertheless, their approach is only applicable for linear problem. Sagawa et al. (2020) also analyze the problem from a model structure perspective, but focusing on deep neural networks and showing that overparameterization will hurt OOD performance through data memorization and overfitting. Rather than focusing model structure, most recent works (Arjovsky et al., 2019; Sagawa et al., 2019; Ahuja et al., 2020; Krueger et al., 2020; Jin et al., 2020; Koyama & Yamaguchi, 2020; Creager et al., 2020) mainly target improvements in the objective function over ERM.

In this work, we set out to study the effect of the model structure in OOD generalization beyond simple considerations of model capacity. We begin by demonstrating that even already trained models that exploit spurious correlation can contain subnetworks that capture invariant features. We then

[1]Mila - Quebec AI Institute [2]CSAIL, Massachusetts Institute of Technology [3]Key Lab of Machine Perception (MoE), School of EECS, Peking University. Correspondence to: Dinghuai Zhang <dinghuai.zhang@mila.quebec>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

turn to investigate whether the choice of structure matters in the training process. To this end, we propose a functional lottery ticket hypothesis – a full network contains a subnetwork that can possibly achieve *better* performance for OOD generalization than full network. We confirm this hypothesis by experiments on a manually crafted dataset (Figure 1) with our "oracle" subnetwork that uses information from OOD examples. As a practical method to that avoids the use of OOD information, we propose the Modular Risk Minimization (MRM) approach. MRM is a simple algorithm to address OOD tasks via structure learning. Our approach hunts for subnetworks with a better OOD inductive bias and can also combine with other OOD algorithms, bringing consistent performance improvement. We summarize our contributions as follows:

- We show that large trained networks that exploit spurious correlations contain subnetworks that are less susceptible to these spurious shortcuts.
- We propose a novel functional lottery ticket hypothesis: there exists a subnetwork that can achieve better OOD and commensurate in-distribution accuracy in a comparable number of iterations when trained in isolation.
- We propose Modular Risk Minimization (MRM), a straightforward and effective algorithm to improve OOD generalization. MRM helps select subnetworks and can be used in conjunction with other methods (*e.g.*, IRM) and boosts their performance as well.

2. Invariant Prediction

2.1. Out-of-distribution (OOD) generalization problem

Consider a supervised learning setting where the data is gathered from different environments and each environment represents a different probability distribution. Let $(X^e, Y^e) \sim \mathbb{P}^e$, where $X^e \in \mathcal{X}, Y^e \in \mathcal{Y}$ stands for the feature random variable and the corresponding label, $e \in \mathcal{E} = \{1, ..., E\}$ is the index for environments, and the set $\mathcal{E}$ corresponds to all possible environments. The set $\mathcal{E}$ is divided into two sets: seen environments $\mathcal{E}_{\text{seen}}$ and unseen ones $\mathcal{E}_{\text{unseen}}$ ($\mathcal{E} = \mathcal{E}_{\text{seen}} \cup \mathcal{E}_{\text{unseen}}$). The training dataset comprises samples from $\mathcal{E}_{\text{seen}}$. The dataset from environment e is given as $D_e = \{x_i^e, y_i^e\}_{e=1}^{n^e}$, where each point (x_i^e, y_i^e) is an independently identically distributed (IID) sample from $\mathbb{P}^e$ and n^e is the number of samples in environment e. We write the training dataset as $D_{\text{train}} = \cup_{e \in \mathcal{E}_{\text{seen}}} D_e$. In the rest of the work, we interchangeably use the term *domain* and *environment*, and we will use *in*-distribution or *in*-domain to refer to seen environmental data, and *out*-distribution or *out*-domain for unseen environmental data.

Let $f_\theta : \mathcal{X} \to \mathcal{Y}$ denote the parametrized model with parameters $\theta \in \Theta$. Define the risk achieved by the model as $\mathcal{R}^e(\theta) = \mathbb{E}^e\big[\ell(X^e, Y^e)\big]$ where ℓ is the loss per sample (*e.g.*, cross-entropy, square loss). The goal of out-of-distribution (OOD) generalization problem is to learn a model that solves

$$\min_{\theta \in \Theta} \max_{e \in \mathcal{E}} \mathcal{R}^e(\theta). \tag{1}$$

Since we only have access to data from D_{train} and do not see samples from the unseen environments, the above problem can be challenging to solve.

Data generation process. We assume X^e is generated from latent variables $Z^e = (Z^e_{\text{inv}}, Z^e_{\text{sp}})$. Consider an illustrative example where X^e could be the pixels in images, while Z_{inv} denotes invariant features (*e.g.*, foreground) and Z_{sp} denotes spurious features (*e.g.*, background). We write $X^e = G(Z^e_{\text{inv}}, Z^e_{\text{sp}})$, where G is a map from the latent space to the pixel space. Y^e is the label for the object and it is determined based on the following map $Y_e = F(Z^e_{\text{inv}})$. The combination pattern of Z^e_{inv} and Z^e_{sp} varies across domains, hence generating different environmental distributions. In our description of the data generation, we do not use noise variables to keep things simple (Y is related to Z deterministically and X is related to G deterministically). Suppose that we can recover Z^e_{inv} and Z^e_{sp} from X^e and we write these inverse maps as $Z^e_{\text{inv}} = G^\dagger_{\text{inv}}(X^e)$ and $Z^e_{\text{sp}} = G^\dagger_{\text{sp}}(X^e)$. The ideal function that the model wants to learn is $F \circ G^\dagger_{\text{inv}}$ as it yields zero error and only relies on invariant latents. However, as we explain next that due to selection biases the model can often find it hard to learn a model that only relies on Z^e_{inv}.

Bias. To explain why the datasets have a bias, let us consider a simple example, where $Z^e_{\text{inv}} \in \{-1, 1\}^{D_{\text{inv}}}$, $Z^e_{\text{sp}} \in \{-1, 1\}^{D_{\text{sp}}}$ and $Y^e \in \{-1, 1\}$. Suppose each component of Z^e_{inv} is Y^e and each component of Z^e_{sp} independently takes a value equal to Y^e with a probability p^e and $-Y^e$ with a probability $1 - p^e$. If p^e is close to 1 and $G^\dagger_{\text{sp}}$ is an easier function to learn than $G^\dagger_{\text{inv}}$, then it is intuitive that the model can instead learn Z^e_{sp} and predict the label Y_e. However, this can be catastrophic as the correlation between the spurious feature and the label only holds in the training environments and does not translate to the test environments where $p_e = \frac{1}{2}$. Even if p^e is small, as long as Z^e_{sp} is high dimensional ($D_{\text{sp}} \gg D_{\text{inv}}$), the model can be shown to significantly rely on Z^e_{sp} (Nagarajan et al., 2020). The above example uses binary valued latents for ease of exposition, but the same biases can occur in more general settings where the same problems plague the models.

2.2. A Motivating Example

In this section, we use a simple example to motivate the constraints we impose in our approach. Consider the data setting described in the previous section, $Z^e_{\text{inv}} \in \{-1, 1\}$ ($D_{\text{inv}} = 1$) and $Z^e_{\text{sp}} \in \{-1, 1\}^D$ ($D_{\text{sp}} = D$). We take G to be

the identity map as in Tsipras et al. (2019); Rosenfeld et al. (2020) and thus $X^e = (Z^e_{\text{inv}}, Z^e_{\text{sp}})$. Suppose the model f_θ is a linear predictor; we refer to the components associated with invariant feature as w_{inv} and those associated with the spurious feature as $\boldsymbol{w}_{\text{sp}}$.

Learning a sparse classifier: Find a maximum margin classifier that satisfies the following sparsity constraint: the number of non-zero coefficients $\leq d$. We denote such a classifier as f^d_{sparse}.

In the next proposition, we compare the behavior of the sparse classifier that we defined above with a classifier that relies only on spurious features. We construct a regular classifier f_{reg} (with unit norm) that purely relies on the spurious features, i.e., $w_{\text{inv}} = 0$ and $\boldsymbol{w}_{\text{sp}} = \mathbf{1}\frac{1}{\sqrt{D_{\text{sp}}}}$ and thus has poor OOD performance. We denote the average error rate of the classifier h on seen (or unseen) environments as $\text{Err}_{\text{seen}}(h)$ (or $\text{Err}_{\text{unseen}}(h)$.) Here the error for binary classification is defined to be $\text{Err}^e(h) = \frac{1}{2}\mathbb{E}_{(X^e,Y^e)\sim\mathbb{P}^e}\left[1 - Y^e h(X^e)\right]$. We denote the margin of classifier for data in environment e as Margin^e.

Proposition 1. *Consider the dataset in Section 2.1 with $Z^e_{\text{inv}} \in \{-1, 1\}$ ($D_{\text{inv}} = 1$) and $Z^e_{\text{sp}} \in \{-1, 1\}^D$ ($D = D_{\text{sp}}$). Let n be the number of training samples in D_{train}, c be a constant in $(0, 1)$ such that for all $e \in \mathcal{E}_{\text{seen}}$, $p^e > \frac{1}{2} + \frac{c}{2}$ and $p^e = \frac{1}{2}$ for $e \in \mathcal{E}_{\text{unseen}}$. For sparsity constraint $d = 2$, we have:*

- *Compare margin for in-distribution sample: for any $\delta \in (0, 1)$, if $D \geq \frac{1}{2c}\sqrt{2ln(n)/\delta}$, then with a probability at least $1 - \delta$, $\text{Margin}^e_{\text{seen}}(f^d_{\text{sparse}}) < \text{Margin}^e_{\text{seen}}(f_{\text{reg}})$;*
- *Similar in-distribution performance $\forall e \in \mathcal{E}_{\text{seen}}$, $\text{Err}^e_{\text{seen}}(f^d_{\text{sparse}}) = 0$, $\text{Err}^e_{\text{seen}}(f_{\text{reg}}) \leq 2e^{-2c^2 D}$;*
- *Better out-distribution performance: $\forall e \in \mathcal{E}_{\text{unseen}}$, $\text{Err}^e_{\text{unseen}}(f^d_{\text{sparse}}) = 0$ and $\text{Err}^e_{\text{unseen}}(f_{\text{reg}}) = 0.5$.*

From the above Proposition, we can conclude that if c or D is high, then the train accuracy of the sparse classifier and the regular classifier are similar but the OOD accuracy of the two classifiers are different with sparse classifier being much better. The algorithm is likely to select the regular classifier over the sparse classifier as it has a much higher margin than the sparse classifier.

Proposition 1 compares the optimal sparse classifier with a purely spurious one. Both have same in-distribution performance, but the former has a better OOD performance. We compare the margins to show that if we use a gradient descent on logistic loss, it will be biased towards the spurious classifier (Soudry et al., 2018). We clarify that Proposition 1 is not intended to show a tradeoff between OOD performance and margin. Consider the experiment of spiral vs. linear boundary of Sec 3.1 in Parascandolo et al. (2020). In the experiment, the spiral boundary is associated with invariant features and the linear boundary is associated with spurious ones. The authors set the margin for linear boundary to be larger than the that of the spiral boundary. In this case, ERM learns a model that uses spurious features. Even if we were to reduce the margin of the linear boundary to be smaller than the spiral boundary, ERM continues to rely on the spurious features as it prefers to use a simpler margin (Shah et al., 2020).

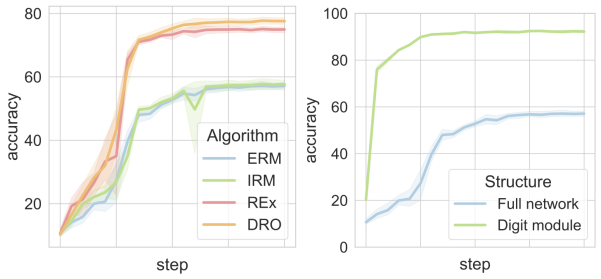
For this linear setting that we discussed above, we can learn a constrained max-margin classifier by adding ℓ_1 constraints. This is a tractable problem to solve as the problem remains convex. However, as we move to neural networks, learning sparse classifiers with good OOD performance is significantly more challenging owing to the non-convexity. This issue is the subject of later sections. Before we address this issue of learning sparse networks with good OOD properties, there is another important question to be answered. In the setting of the above proposition, we rely on the fact that a sparse model exists that relies on invariant features only and yields better OOD performance. How do we do know that this is a proper assumption for real datasets used for neural network training? In the next section, we analyze neural networks via modular subnetwork introspection to show that such a sparse model exists.

3. A Functional Modularity Based Analysis

3.1. Preliminaries

Technical approach. The modularity property of neural network has long been considered as an essential foundation of systematical generalization (Ballard, 1987; Marcus, 1998; Csordás et al., 2020). Consider a task that can be compositionally separated into different independent subtasks, we aim to probe a functional module subpart of the full neural network that can solve one particular subtask. Following Zhou et al. (2019); Csordás et al. (2020), we identify different subnetworks which perform different functions, from a given pretrained network.

Specifically, we deem functional modules to be particular subsets of the weights inside a neural network. For a L layer neural network model $f(\mathbf{w}_1, \cdots, \mathbf{w}_L; \cdot)$ where $\theta = \{\mathbf{w}_1, \cdots, \mathbf{w}_L\}$, we model the subnetwork with a set of binary masks $\mathbf{m}_l \in \{0, 1\}^{n_l}$ on the l-th layer weight tensor $\mathbf{w}_l \in \mathbb{R}^{n_l}$, where n_l is the number of dimensionality of the l-th layer network parameters. The subnetwork is then given by $f(\mathbf{m}_1 \odot \mathbf{w}_1, \cdots, \mathbf{m}_L \odot \mathbf{w}_L; \cdot)$. Further, in order to make this subnetwork structure learnable, we assume each entry of the mask to be independent Bernoulli random variables, and model their logits as $\boldsymbol{\pi}_l \in \mathbb{R}^{n_l}$. Hence, in this probabilistic modeling setting, the l-th layer subnetwork structure $\mathbf{m}_l$ is generated by performing Bernoulli



(a) Accuracy of baselines. (b) Accuracy of module in ERM.

Figure 2. Left: OOD accuracy for four algorithms. *Right*: OOD accuracy for ERM algorithm and its digit module. The plot shows that a highly biased model can contain an unbiased subnetwork.

sampling with parameters $\text{sigmoid}(\boldsymbol{\pi}_l)$. We adopt Gumbel-sigmoid trick (Jang et al., 2016) to enable an end-to-end training process, together with a logit regularization term to promote subnetwork sparsity (Csordás et al., 2020). For each particular subtask, our analysis will output a logits tensor for each neuron in the form of $\boldsymbol{\pi} = \{\boldsymbol{\pi}_1, \cdots, \boldsymbol{\pi}_L\}$, and thereby uncover the corresponding functional module within the neural network in the form of binary tensor $\mathbf{m} = \{\mathbf{m}_1, \cdots, \mathbf{m}_L\} = \{\text{sigmoid}(\boldsymbol{\pi}_l) > 0.5 \mid l = 1, 2, \cdots\}$. We then use the term *modularity probing* method to refer to this technique subsequently. We will interchangeably use the term of *module* and *subnetwork* due to their consistency in our context.

Dataset construction. We take the intuition from Arjovsky et al. (2019); Nam et al. (2020); Ahuja et al. (2021); Ahmed et al. (2021) to design a biased variant of the original MNIST dataset (LeCun et al., 1998). A discussion about the difference between ours and theirs is deferred to supplementary materials. The digit shape semantics are considered as Z_{inv} while color semantics as Z_{sp}. We choose ten different kinds of color and define a one-to-one corresponding bias relationship with ten digit class (*e.g.*, "2" $\leftrightarrow$ "green", "4" $\leftrightarrow$ "yellow"). For each domain, we define the bias coefficient to be the ratio of the data that obeys this relationship. Those images which don't follow this relationship are then assigned with random colors. The bias coefficient for two in-domains is $(1.0, 0.9)$ respectively, which means the first domain is completely biased and 90 percent of the second domain is biased. For the out-domain, all images are assigned a random color for evaluating to how much extent the model has learned the invariant feature. The out-domain will serve as a tool environment for module learning in this section, representing a thorough disentanglement of two attributions. It will then act as the test distribution in a realistic setting in Section 5. Unless otherwise specified, the label is set as the class where the invariant attribution lies. We use the term FULLCOLOREDMNIST to refer to this task to distinguish with the binary colored mnist dataset in Arjovsky et al. (2019).

Algorithms analyzed. We study four OOD generalization algorithms in this paper: Empirical Risk Minimization (ERM) (Vapnik, 1999), Invariant Risk Minimization (IRM) (Arjovsky et al., 2019), Risk Extrapolation (REx) (Krueger et al., 2020) and group Distributional Robust Optimization (DRO) (Sagawa et al., 2019). More details about them are left to supplementary materials. Figure 2(a) plots the generalization performance of these algorithms w.r.t. the training process. REx (76.17%) and DRO (78.56%) methods surpass ERM baseline by a large margin, while IRM (59.55%) only gets slightly better results than ERM (58.04 %). The failure of IRM in realistic problems has been analyzed in Jin et al. (2020); Nagarajan et al. (2020); Rosenfeld et al. (2020); Ahuja et al. (2021) and attributed to the overparameterization regime and curse of dimensionality, hence we omit related discussion here.

3.2. Modular subnetwork introspection

Departing from previous approaches, in this section we think of learning the digit and color semantics as different functional subtasks of the original task, rather than opposite non-spurious / spurious features. We split the out-domain into two parts and refer to them as the *in*-split and *out*-split of the *out*-domain (terminology from Gulrajani & Lopez-Paz (2020)). We define two subtasks, identification of digit and identification of color. For each subtask, we assume that we have access to respective semantic labels. It's important to note that the semantic color label is used here for analysis and is not a part of our main method described later.

In order to study the functional module for the two subtasks, we apply the modularity probing method to diagnose given pretrained models. Specifically, we separately train and get a digit and a color subnetwork for each model across different algorithms and training steps. We evaluate the obtained digit modules' behaviors on the out-split of out-domain (as the in-split has been taken for module searching). Figure 2(b) suggests a significant evidence that, even for biased models such as ERM trained ones, there exist unbiased invariant subnetworks (digit modules) with good OOD generalization ability. We also explore this property for other modules and algorithms and defer these results to supplementary materials.

Discussion about the sparsity of digit weights / features. We additionally visualize the Bernoulli probability of learned subtask modules. Figure 3 displays the first layer of model trained on two in-domains with ERM. We can see that the color feature is more pervasive than digit feature, spreading over a broader range across the neurons. Although the sparsity of weights is not exactly the sparsity of features, the discovery is aligned with the assumption in Proposition 1 that D_{inv} has a small number of dimensionality. The visualization results of other layers are similar to

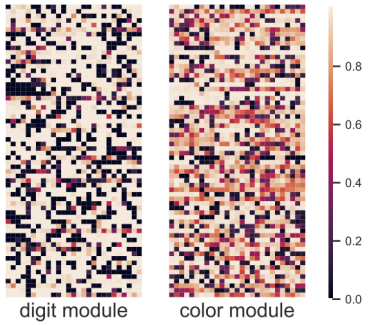


Figure 3. The visualization of the Bernoulli probability of digit and color functional module for the first (convolutional) layer. The weight tensor is reshaped to two dimension for display convenience. The probability takes value from 0 to 1.

this, and can be found in the supplementary materials.

In this section we confirm in large *trained* models, there lie invariant functional modules, *viz*. subnetwork that behaves well for target invariant function (*e.g.*, digit classification). However, it's more worthwhile to find out about whether an appropriate subnetwork structure can help improve in the sense of OOD generalization *during training*. We investigate this problem in the next section.

4. Structure Matters: Towards Functional Lottery Tickets Hypothesis

Frankle & Carbin (2018) proposes the lottery ticket hypothesis from a pruning perspective, suggesting that among all different subnetworks, there exists a so-called "winning ticket" that can *reach* the generalization ability of the full network with faster training speed. In this original lottery ticket hypothesis, the data distribution remains unchanged across training and testing. Whereas in our OOD context, we seek a model subnetwork whose functional predictions are invariant with respect to the change in distribution.

The functional lottery ticket hypothesis: A randomly initialized, dense neural network contains a subnetwork that is initialized such that — when trained in isolation — it can achieve *better* out-of-distribution performance w.r.t. the given function (*e.g.*, digit identification in our context) than the original full network after training for the same number of iterations.

Concretely, our functional lottery ticket hypothesis claims that for a dense neural network model $f(\mathbf{w};\mathbf{x})$ with initialization parameter $\mathbf{w}_0$, there exists a module $\mathbf{m}$ enabling a subnetwork $f(\mathbf{m}\odot\mathbf{w};\mathbf{x})$ to *surpass* the OOD performance of full network when trained from $f(\mathbf{m}\odot\mathbf{w}_0;\mathbf{x})$ on in-distribution data. Note that this is a stronger statement than the original lottery ticket hypothesis, which only requires the winning ticket to reach similar performance to the full network in the IID setting.

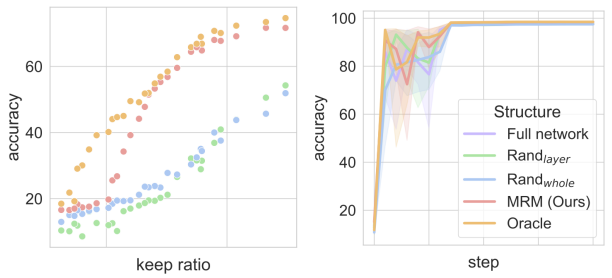


Figure 4. Left: Performance of different networks for various levels of sparsity. Here the *keep ratio* is defined to be 1− sparsity and left side of figure means smaller keep ratio. *Right:* In-distribution generalization performance of different subnetworks in Section 4.

Demonstration. How do we identify the functional winning tickets? How should one search for a structure that is best for OOD generalization? To unravel the possible best result one can reach, we design an "oracle" subnetwork. After obtaining an ERM trained model, the structure of oracle module is found with the aid of the in-split data part of out-domain. Namely, we use the modularity probing technique introduced in Section 3.1 with these "oracle" data from the out-distribution, and then deploy the resulting subnetwork back onto the previous initialization. We then evaluate all methods on the out-split.

According to Sagawa et al. (2020), the underparameterized regime can keep the model from overfitting to spurious features. Therefore, we choose the random subnetwork as another option to investigate whether sparsity / underparametrization alone can achieve an unbiased solution. "rand$_{\text{whole}}$" method keeps the ratio of full network same as the oracle subnetwork. In other works it has been claimed that the sparsity per layer is the only working factor for pruning (Su et al., 2020; Frankle et al., 2020), we also experiment with the "rand$_{\text{layer}}$" method, where we randomly sample subnetworks with the same per-layer-sparsity.

Figure 1 and 4(b) show the OOD and in-distribution generalization results for these subnetworks with ERM training on the FULLCOLOREDMNIST dataset respectively. The oracle subnetwork beats the original ERM by a large margin for OOD and maintains indistinguishable performance for the in-distribution examples, confirming that a good module structure can indeed surpass the full network in terms of this digit function. We show their performance with greater levels of sparsity in Figure 4(a) and see a considerable consistent accuracy gap between oracle and random baselines for all level of sparsity. The validity of our functional lottery ticket hypothesis is thereby empirically affirmed, and we thus propose that *appropriate structure induction can impose a needed inductive bias to prevent the model from fitting the spurious correlation*. We also conclude that sparsity constraint imposed cannot help alone, as two random

methods don't yield non-trivial benefit than ERM (both under 60% accuracy). Additionally, we demonstrate our hypothesis is also applicable for other OOD algorithms in the supplementary materials.

Discussion. One can rightly criticize this investigation as unfair in that it compares a method using out-domain data to baselines without such privileged access. We acknowledge this issue and, for now, only seek out this "oracle subnetwork" to highlight the importance of structure. We now turn to the question of how we can design a practical structure searching algorithm to overcome this limitation.

4.1. Modular risk minimization

The motivation behind the Modular Risk Minimization (MRM) method is to get rid of spurious features by hunting for a desired functional winning ticket. Since we have shown in previous subsection that contrary to Sagawa et al. (2020), only sparsity constraints imposed at the beginning of training cannot do the magic, we propose the following criterion:

A good structure should balance the predictiveness for invariant feature and sparsity well.

Our procedure first trains the model with ERM resulting in a potentially biased classifier. At this time, the functional lottery ticket hypothesis suggests that the model has already learned a promising functional module within. Hence, we simply apply the subnetwork probing technique with *training* data to learn the potential advantageous structure. The structure learning objective takes a combination of cross entropy loss and sparsity regularization to balance the two desiderata mentioned above. We then simply train from scratch again only with the weights in the obtained subnetwork and fix the other weights to zero. We summarize this procedure in Algorithm 1, where i, c, l are respectively the index for a datum, label class and network layer. It's notable that in Figure 1, our proposed MRM algorithm successfully reaches a very close accuracy to the oracle optimal structure.

Structure learning by invariance capturing. Notably, MRM *does not impose any invariance across domains* and is thus orthogonal to the advantage of other OOD algorithms. Unlike heuristic structure searching paradigms (Lee et al., 2018; Wang et al., 2020), our method can incorporate *any* OOD generalization approach to help improve the structure learning and model training process. This enables MRM to act as a plug-in method to boost other algorithms by supplying a good subnetwork learned from their respective objectives. We simply replace the cross entropy loss $\mathcal{L}_{\text{CE}}$ in Algorithm 1 with recently developed OOD losses: $\mathcal{L}_{\text{IRM}}$, $\mathcal{L}_{\text{REx}}$ and $\mathcal{L}_{\text{DRO}}$ by IRM, REx and DRO. These new methods are therefore referred by Modular Invariant Risk Minimization (ModIRM), Modular Risk Extrapolation (ModREx) and Modular Distributionally Robust Optimization (ModDRO). With these explicitly OOD learning algorithms, the cross domain variance is taken into account and thereby ameliorates the invariance property of the subnetwork. We note that more flexible combinations can be explored (*e.g.*, use a different loss design for subnetwork and model learning), but we leave this for future work and only study these three variants in this paper.

Algorithm 1 Modular Risk Minimization

Input: Data $\{(x_i^e, y_i^e)\}_{i,e}$, neural network $f(\mathbf{w}; \cdot)$, subnetwork logits $\boldsymbol{\pi}$, the coefficient of sparsity penalty α, number of steps for model and subnetwork structure training N_1, N_2.

Stage 1: full model (pre-) train
Get model initialization $\mathbf{w}_0$.
for n=1 **to** N_1 **do**
  Update f with $\mathcal{L}_{\text{CE}}(\mathbf{w}) \coloneqq \sum_{i,c} y_{i,c} \log f(\mathbf{w}; x_i)_c$.
end for
Stage 2: module structure probing
for n=1 **to** N_2 **do**
  Sample subnetwork $\mathbf{m} \sim \text{sigmoid}(\boldsymbol{\pi})$.
  Update module $\boldsymbol{\pi}$ with
    $\mathcal{L}_{\text{MOD}} = \mathcal{L}_{\text{CE}}(\mathbf{m} \odot \mathbf{w}) + \alpha \sum_{l,j} \boldsymbol{\pi}_{l,j}$.
end for
Stage 3: subnetwork retrain
Obtain the module by hard thresholding:
    $\mathbf{m} = \{\boldsymbol{\pi}_l > 0 \mid l = 1, 2, \cdots\}$.
Set model parameters back to $\mathbf{w}_0$.
for n=1 **to** N_1 **do**
  Update f with $\mathcal{L}_{\text{CE}}(\mathbf{m} \odot \mathbf{w})$.
end for

4.2. Ablation for winning tickets learning

To better understand the crucial succeeding reasons for two kinds of winning tickets – oracle subnetworks and MRM subnetworks, we conduct corresponding ablation studies.

Importance of initialization. One of the main argument of Frankle & Carbin (2018) is that the winning tickets cannot be learned effectively without its original initialization. We verify this for our hypothesis as well. Figure 5(a) depicts the failure of functional winning tickets when random *re*-initialization is performed before the training of subnetworks. At this time, both subnetworks achieves similar OOD performance to full network ERM. This ablation study confirms the importance of reusing initialization.

Effects of bias relationship. Our FULLCOLOREDMNIST keeps one fixed color-digit relationship for all biased data. Will rearranging this bias relationship defined in Section 3.1 before the subnetwork is trained destroy the tickets? We then

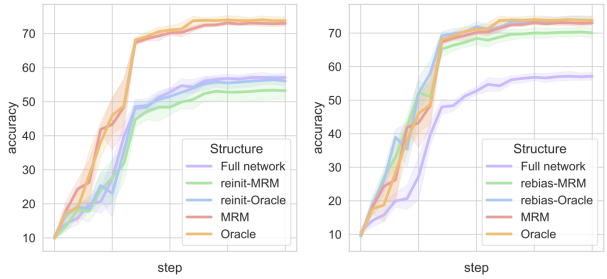
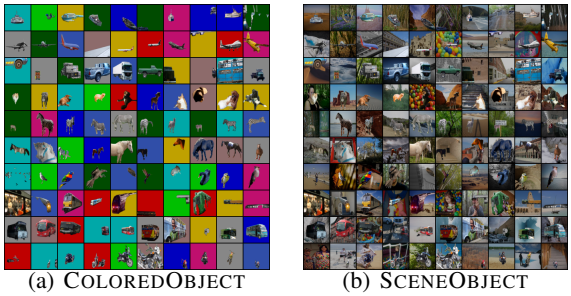


Figure 5. Left: Ablation for the importance of initialization. With re-initialized model weights, the winning tickets fail to win the jackpot. *Right:* Rearrange the color-digit relationship slightly reduces performance.



(a) ColoredObject (b) SceneObject

Figure 6. The visualization of ColoredObject (left) and SceneObject (right) datasets. We keep the same object (invariant feature) for each row and assign random backgrounds (spurious feature) to the images across different columns.

apply this to both winning tickets in Figure 5(b) and witness only a small accuracy drop of MRM after resetting the bias. This suggests our method indeed finds a subnetwork with a robust inductive bias for the invariant function, instead of only memorizing the bias relationship.

5. Experiments

In this section, we demonstrate the effectiveness of our modular risk minimization algorithm on a variety of datasets. We compare our algorithm and its OOD variants with recent methods aiming at robust predictions across environments. For all methods we keep the same model architectures and training settings. We build three OOD datasets according to the bias protocol introduced in Section 2.1: FullColoredMNIST, ColoredObject and SceneObject. For all datasets we design two training in-domains and one out-domain for evaluating OOD generalization capability. We defer all other experimental details to the supplementary materials.

FullColoredMNIST. Details of the construction are in Section 2.1. We summarize the results in Table 1. We also use ERM trained with completely unbiased data whose bias coefficient is $(0.0, 0.0, 0.0)$ to serve as an upper bound

Table 1. Generalization performance on FullColoredMNIST.

Methods	Train Accuracy	Test Accuracy
ERM	98.10 ± 0.09	57.75 ± 1.84
MRM	98.90 ± 0.05	**72.98** ± 0.58
IRM	98.18 ± 0.09	59.30 ± 1.88
ModIRM	98.77 ± 0.12	**70.86** ± 2.12
REx	98.86 ± 0.10	75.61 ± 1.26
ModREx	99.28 ± 0.04	**82.06** ± 0.73
DRO	98.96 ± 0.09	78.25 ± 1.31
ModDRO	99.39 ± 0.04	**85.53** ± 0.61
Unbias	99.07 ± 0.04	99.03 ± 0.08

Table 2. Generalization performance on ColoredObject.

Methods	Train Accuracy	Test Accuracy
ERM	87.56 ± 2.52	43.74 ± 2.11
MRM	94.01 ± 0.82	**54.85** ± 2.11
IRM	88.68 ± 2.11	45.4 ± 2.40
ModIRM	93.01 ± 0.36	**52.35** ± 1.28
REx	89.85 ± 1.50	47.20 ± 3.43
ModREx	93.55 ± 1.45	**55.51** ± 2.76
DRO	91.73 ± 0.40	51.95 ± 1.62
ModDRO	92.67 ± 0.92	**55.20** ± 1.40
Unbias	95.00 ± 0.70	72.37 ± 2.53

(coined as "Unbias" in the tables). Our method can consistently promote the OOD performance on this task, bringing around 10% accuracy promotion. The best behaved algorithm, ModDRO, reaches 85.53% accuracy, contrary to the 78.25% of top-grade baseline DRO and 99.0% achieved by unbiased solution.

ColoredObject. We take inspiration from Ahmed et al. (2021) to build this biased dataset together with the following SceneObject one. Ten classes of objects extracted from MSCOCO dataset (Lin et al., 2014) are put onto ten kinds of color backgrounds. Figure 6(a) displays 100 samples from this crafted biased dataset. Like FullColoredMNIST, we also set a one-to-one object-color relationship and set the bias coefficient differently as $(0.8, 0.6, 0.0)$. Results in Table 2 demonstrate the advantages of our methods: all our four methods all achieve accuracy above 50%, boosting their different baselines towards the optimal "unbias" solution.

SceneObject. Ten classes of objects extracted from MSCOCO dataset are put onto ten kinds of scenery backgrounds from Places dataset (Zhou et al., 2018). These scenery backgrounds make this task a more complex one than ColoredObject. Figure 6(b) displays 100 samples

Table 3. Generalization performance on SCENEOBJECT.

METHODS	TRAIN ACCURACY	TEST ACCURACY
ERM	98.87 ± 0.23	37.29 ± 2.74
MRM	99.61 ± 0.04	**39.44** ± 0.77
IRM	98.68 ± 0.27	37.19 ± 2.58
MODIRM	99.39 ± 0.01	**39.14** ± 1.34
REX	92.91 ± 1.11	38.84 ± 1.39
MODREX	96.71 ± 0.53	**41.04** ± 1.46
DRO	98.89 ± 0.35	36.34 ± 1.67
MODDRO	99.41 ± 0.13	**39.14** ± 1.60
UNBIAS	95.25 ± 2.21	56.46 ± 0.75

from this crafted dataset. Like FULLCOLOREDMNIST, we set a one-to-one object-scenery relationship and set the bias coefficient to be $(0.9, 0.7, 0.0)$, making it a even more biased and thus more difficult one than the previous task. This can also be shown with only 56.46% accuracy of unbias solution. Corresponding results in Table 3 shows that for this highly biased task, MRM and its variants can still accordingly improve out-distribution generalization performance in this highly bias setting, where previous OOD algorithms bring very limited benefit.

6. Related Work

Out-of-distribution generalization. Machine learning beyond IID assumption is a very important problem and many research areas such as domain adaptation (Crammer et al., 2008; Ben-David et al., 2010) and domain generalization (Muandet et al., 2013; Motiian et al., 2017) have received much attention (Gulrajani & Lopez-Paz, 2020). To get stable prediction for new unseen data distribution, it is desired to only rely on invariant features among the causal factorization of physical mechanisms of problem settings (Schölkopf et al., 2012). Peters et al. (2016) (ICP) claims that the residual of invariant method should remain IID and thus proposes to adopt statistical tests for mining invariant feature set. Rojas-Carulla et al. (2018) generalizes this approach to nonlinear settings.

Recently, since Arjovsky et al. (2019) brings invariant prediction into a more practical scenario, a large amount of works has made solid progress for alleviating spurious correlation and shortcut exploitation (Geirhos et al., 2020; Koh et al., 2020): Sagawa et al. (2019) proposes to use group DRO when attribution information is provided; Chang et al. (2020) incorporates this invariant inference idea into selective rationalization area; Ahuja et al. (2020) studies the IRM formulation from a game theory and bilevel optimization formulation; Krueger et al. (2020) propose REx to enforce the variance of losses across distribution, which is further analyzed by Xie et al. (2020a); Koyama & Yamaguchi (2020) (IGA) also has a similar contribution with different theoretical analysis; Jin et al. (2020) (RGM) proposes another training objective from regret minimization viewpoint; Pezeshki et al. (2020) studies the gradient starvation phenomenon which is connected with spurious correlation and proposes an insightful solution; Creager et al. (2020) (EIIL) points out that invariant prediction shares the same spirit with fair representation learning; Parascandolo et al. (2020) (ILC) proposes to focus second order landscape information; Ahmed et al. (2021) adopts a divergence term to match the output distribution spaces of different domains; Müller et al. (2020) achieves invariance from an information theory start point and enforces conditional invariance with HSIC terms. Some other works also point out the pitfalls of current approaches, showing only in very limited situations can Arjovsky et al. (2019) (*e.g.*, low dimension settings) really capture invariance: Rosenfeld et al. (2020) proves the validity of IRM for linear cases but gives a negative example for nonlinear cases; Nagarajan et al. (2020) analyzes different failure modes of OOD generalization; Ahuja et al. (2021) analyze the sample efficiency properties of IRM; Kamath et al. (2021) investigates the success and failure cases of IRM and IRMv1 on simple but insightful settings, and claims the community might need a better invariance notion.

Another line of works study a related but different topic named *debiasing*, where there is no explicit multiple environments setting provided. Bias in realistic datasets are usually exploited in a spurious way, such as the texture-bias of Imagenet-trained models (Geirhos et al., 2018). Subsequent works (Wang et al., 2019; Bahng et al., 2020; Shi et al., 2020; Nam et al., 2020; Li et al., 2021; Sauer & Geiger, 2021) focus on addressing the bias problem with explicit debiasing procedure.

Modularity. Modularity (Ballard, 1987; Fodor et al., 1988; Newman, 2006) has been considered as a crucial part of intelligent systems. Lots of works focus on imposing explicit module level modularity (Clune et al., 2013; Andreas et al., 2016; Chang et al., 2018; Goyal et al., 2021), while others also explore weight level modularity in a more fine-grained way (Mallya & Lazebnik, 2018; Watanabe et al., 2019; Filan et al., 2020; Csordás et al., 2020). Our work also belongs to the latter category.

Pruning. We mainly focus on unstructured pruning literature. This line of model compression literature dates back to Mozer & Smolensky (1989); LeCun et al. (1989); Hassibi & Stork (1993) with more recent pruning methods (Han et al., 2015; Molchanov et al., 2016; Dong et al., 2017). Recently, the lottery ticket hypothesis (Frankle & Carbin, 2018) sheds more light into this field, showing the importance of initialization. (Liu et al., 2018) also propose another viewpoint that for practical settings the inherited weights are not important.

7. Discussion

Data settings. The seminal work (Arjovsky et al., 2019) proposes to use color in digit identification as a spurious correlation. In order to exposit the effectiveness of IRM, the authors enforce a 25% label noise in the binary classification data and assign color a *larger* correlation than the true digit shape. In this way, ERM exploits color feature to predict. While there is controversy surrounding whether one should still treat digit as desired learning target under this situation, we choose to impose no label noise in FULLCOLOREDMNIST as is the case in Nam et al. (2020); Ahmed et al. (2021). This choice enables the structure learning procedure could mine the true invariant feature. On the other hand, our work is limited as we haven't considered the data settings such as group attribution available ones (Sagawa et al., 2019; Xie et al., 2020b; Khani & Liang, 2021) and we shall fill this gap in future work. More about datasets can be found in Section C.1.

Success of MRM. There are several reasons for why MRM can improve OOD performance without invariance constrain. The first reason is related to our label noise free setting discussed above. This makes the invariant feature itself perfectly predictive of the label, thus containing all information about the desired target function. Then the problem would be how to exploit this information effectively. MRM becomes competent for OOD tasks by providing a novel and helpful parameterization method for the original optimization problem with extra parameters. One notable thing is that more structure parameters actually don't increase the expressive power of the neural network, since every weight can take zero value in nature. Another reason is we adopt an explicit approach to zero out the "spurious part" of the model weights, hence achieving a not-so-biased solution. Notice this cannot be achieved with random sparse model, revealing the structure is a key element for OOD generalization. Therefore, a positive answer is given to the title of this work. We further refer to Section C.3 for empirical results of the importance of a proper sparsity level in structure learning.

Acknowledgement

Kartik Ahuja acknowledges the support provided by IVADO postdoctoral fellowship funding program. Yilun Xu is supported by the MIT HDTV Grand Alliance Fellowship. Yisen Wang is partially supported by the National Natural Science Foundation of China under Grant 62006153, CCF-Baidu Open Fund (OF2020002), and PKU-Baidu Fund (2020BD006). Aaron Courville acknowledges the funding from CIFAR Canadian AI Chair and Hitachi. The authors would also like to thank Róbert Csordás, David Krueger, Faruk Ahmed, Mohammad Pezeshki, Baifeng Shi, Sara Hooker and anonymous reviewers for insightful discussion and feedbacks.

References

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

Ahuja, K., Shanmugam, K., Varshney, K., and Dhurandhar, A. Invariant risk minimization games. In *International Conference on Machine Learning*, pp. 145–155. PMLR, 2020.

Ahuja, K., Wang, J., Dhurandhar, A., Shanmugam, K., and Varshney, K. R. Empirical or invariant risk minimization? a sample complexity perspective. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jrA5GAccy_.

Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 39–48, 2016.

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

Bahng, H., Chun, S., Yun, S., Choo, J., and Oh, S. J. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pp. 528–539. PMLR, 2020.

Ballard, D. H. Modular learning in neural networks. In *AAAI*, pp. 279–284, 1987.

Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision*, pp. 456–473, 2018.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Bengio, Y., Deleu, T., Rahaman, N., Ke, R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

Brendel, W. and Bethge, M. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019.

Chang, M. B., Gupta, A., Levine, S., and Griffiths, T. L. Automatically composing representation transformations as a means for generalization. *arXiv preprint arXiv:1807.04640*, 2018.

Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. Invariant rationalization. In *International Conference on Machine Learning*, pp. 1448–1458. PMLR, 2020.

Clune, J., Mouret, J.-B., and Lipson, H. The evolutionary origins of modularity. *Proceedings of the Royal Society b: Biological sciences*, 280(1755):20122863, 2013.

Crammer, K., Kearns, M., and Wortman, J. Learning from multiple sources. *Journal of Machine Learning Research*, 9(8), 2008.

Creager, E., Jacobsen, J.-H., and Zemel, R. Exchanging lessons between algorithmic fairness and domain generalization. *arXiv preprint arXiv:2010.07249*, 2020.

Csordás, R., van Steenkiste, S., and Schmidhuber, J. Are neural nets modular? inspecting functional modularity through differentiable weight masks. *arXiv preprint arXiv:2010.02066*, 2020.

DeGrave, A. J., Janizek, J. D., and Lee, S.-I. Ai for radiographic covid-19 detection selects shortcuts over signal. *medRxiv*, 2020.

Dong, X., Chen, S., and Pan, S. J. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *arXiv preprint arXiv:1705.07565*, 2017.

Filan, D., Hod, S., Wild, C., Critch, A., and Russell, S. Neural networks are surprisingly modular. *arXiv preprint arXiv:2003.04881*, 2020.

Fodor, J. A., Pylyshyn, Z. W., et al. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28 (1-2):3–71, 1988.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Frankle, J., Dziugaite, G. K., Roy, D. M., and Carbin, M. Pruning neural networks at initialization: Why are we missing the mark? *arXiv preprint arXiv:2009.08576*, 2020.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

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.

Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., and Schölkopf, B. Recurrent independent mechanisms. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=mLcmdlEUxy-.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Hassibi, B. and Stork, D. G. *Second order derivatives for network pruning: Optimal brain surgeon*. Morgan Kaufmann, 1993.

Hooker, S., Courville, A., Clark, G., Dauphin, Y., and Frome, A. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

Hooker, S., Moorosi, N., Clark, G., Bengio, S., and Denton, E. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*, 2020.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Jin, W., Barzilay, R., and Jaakkola, T. Domain extrapolation via regret minimization. *arXiv preprint arXiv:2006.03908*, 2020.

Kamath, P., Tangella, A., Sutherland, D. J., and Srebro, N. Does invariant risk minimization capture invariance? *arXiv preprint arXiv:2101.01134*, 2021.

Khani, F. and Liang, P. Removing spurious features can hurt accuracy and affect groups disproportionately. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2021.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Beery, S., et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020.

Koyama, M. and Yamaguchi, S. Out-of-distribution generalization with maximal invariant predictor. *arXiv preprint arXiv:2008.01883*, 2020.

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

LeCun, Y., Denker, J. S., Solla, S. A., Howard, R. E., and Jackel, L. D. Optimal brain damage. In *NIPs*, volume 2, pp. 598–605. Citeseer, 1989.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, N., Ajanthan, T., and Torr, P. H. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

Li, Y., Yu, Q., Tan, M., Mei, J., Tang, P., Shen, W., Yuille, A., and cihang xie. Shape-texture debiased neural network training. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Db4yerZTYkz.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l0 regularization. *ArXiv*, abs/1712.01312, 2018.

Mallya, A. and Lazebnik, S. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Marcus, G. F. Rethinking eliminative connectionism. *Cognitive psychology*, 37(3):243–282, 1998.

Mocanu, D. C., Mocanu, E., Stone, P., Nguyen, P. H., Gibescu, M., and Liotta, A. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9 (1):1–12, 2018.

Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

Mostafa, H. and Wang, X. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning*, pp. 4646–4655. PMLR, 2019.

Motiian, S., Piccirilli, M., Adjeroh, D. A., and Doretto, G. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5715–5725, 2017.

Mozer, M. C. and Smolensky, P. *Skeletonization: A Technique for Trimming the Fat from a Network via Relevance Assessment*. Morgan Kaufmann Publishers Inc., 1989.

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

Müller, J., Schmier, R., Ardizzone, L., Rother, C., and Köthe, U. Learning robust models using the principle of independent causal mechanisms. *arXiv preprint arXiv:2010.07167*, 2020.

Nagarajan, V., Andreassen, A., and Neyshabur, B. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: Training debiased classifier from biased classifier. *arXiv preprint arXiv:2007.02561*, 2020.

Newman, M. E. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

Parascandolo, G., Neitz, A., Orvieto, A., Gresele, L., and Schölkopf, B. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

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)*, pp. 947–1012, 2016.

Pezeshki, M., Kaba, S.-O., Bengio, Y., Courville, A., Precup, D., and Lajoie, G. Gradient starvation: A learning proclivity in neural networks. *arXiv preprint arXiv:2011.09468*, 2020.

Priol, R. L., Harikandeh, R. B., Bengio, Y., and Lacoste-Julien, S. An analysis of the adaptation speed of causal models. *arXiv preprint arXiv:2005.09136*, 2020.

Ritter, S., Barrett, D. G., Santoro, A., and Botvinick, M. M. Cognitive psychology for deep neural networks: A shape bias case study. In *International conference on machine learning*, pp. 2940–2949. PMLR, 2017.

Rojas-Carulla, M., Schölkopf, B., Turner, R., and Peters, J. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

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 for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pp. 8346–8356. PMLR, 2020.

Sauer, A. and Geiger, A. Counterfactual generative networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=BXewfAYMmJw.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020.

Shi, B., Zhang, D., Dai, Q., Zhu, Z., Mu, Y., and Wang, J. Informative dropout for robust representation learning: A shape-bias perspective. In *International Conference on Machine Learning*, pp. 8828–8839. PMLR, 2020.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Su, J., Chen, Y., Cai, T., Wu, T., Gao, R., Wang, L., and Lee, J. D. Sanity-checking pruning methods: Random tickets can win the jackpot. *arXiv preprint arXiv:2009.11094*, 2020.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy, 2019.

Vapnik, V. N. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Wang, C., Zhang, G., and Grosse, R. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*, 2020.

Wang, H., He, Z., Lipton, Z. C., and Xing, E. P. Learning robust representations by projecting superficial statistics out. *arXiv preprint arXiv:1903.06256*, 2019.

Watanabe, C., Hiramatsu, K., and Kashino, K. Understanding community structure in layered neural networks. *Neurocomputing*, 367:84–102, 2019.

Xie, C., Chen, F., Liu, Y., and Li, Z. Risk variance penalization: From distributional robustness to causality. *arXiv preprint arXiv:2006.07544*, 2020a.

Xie, S. M., Kumar, A., Jones, R., Khani, F., Ma, T., and Liang, P. In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. *arXiv preprint arXiv:2012.04550*, 2020b.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Zeng, W. and Urtasun, R. Mlprune: Multi-layer pruning for automated neural network compression. 2018.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Trans Pattern Anal Mach Intell*, pp. 1–1, 2018.

Zhou, H., Lan, J., Liu, R., and Yosinski, J. Deconstructing lottery tickets: Zeros, signs, and the supermask. In *Advances in Neural Information Processing Systems*, pp. 3597–3607, 2019.

Zhu, M. and Gupta, S. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.