Evading the Simplicity Bias: 
Training a Diverse Set of Models Discovers 
Solutions with Superior OOD Generalization

Damien Teney\textsuperscript{1,2} Ehsan Abbasnejad\textsuperscript{2} Simon Lucey\textsuperscript{2} Anton van den Hengel\textsuperscript{2,3} 
\textsuperscript{1}Idiap Research Institute \textsuperscript{2}Australian Institute for Machine Learning, University of Adelaide \textsuperscript{3}Amazon

damien.teney@idiap.ch,\{ehsan.abbasnejad, simon.lucey, anton.vandenhengel\}@adelaide.edu.au

Abstract

Neural networks trained with SGD were recently shown to rely preferentially on linearly-predictive features and can ignore complex, equally-predictive ones. This simplicity bias can explain their lack of robustness out of distribution (OOD). The more complex the task to learn, the more likely it is that statistical artifacts (i.e. selection biases, spurious correlations) are simpler than the mechanisms to learn.

We demonstrate that the simplicity bias can be mitigated and OOD generalization improved. We train a set of similar models to fit the data in different ways using a penalty on the alignment of their input gradients. We show theoretically and empirically that this induces the learning of more complex predictive patterns.

OOD generalization fundamentally requires information beyond i.i.d. examples, such as multiple training environments, counterfactual examples, or other side information. Our approach shows that we can defer this requirement to an independent model selection stage. We obtain SOTA results in visual recognition on biased data and generalization across visual domains. The method – the first to evade the simplicity bias – highlights the need for a better understanding and control of inductive biases in deep learning.

1. Introduction

Inductive biases in deep learning. At the core of every learning algorithm are a set of inductive biases \cite{45}. They define the learned function outside of training examples and they allow extrapolation\textsuperscript{1} to novel test points. Deep neural networks are remarkably effective because their inductive biases happen to reflect properties of real-world data, although the reasons are still poorly understood \cite{87}. In particular, the simplicity bias \cite{27, 47, 51, 57, 70} has been proposed as a reason of their success. It makes networks trained with SGD\textsuperscript{2} represent preferentially simple, approximately piecewise linear functions.\textsuperscript{3} But the simplicity bias can also prevent the learning of complex patterns that are the actual mechanisms of the task of interest. This effect is problematic when the learned simple patterns correspond to spurious correlations a.k.a. statistical shortcuts \cite{17}. In image recognition, an example of a shortcut is to use the

\textsuperscript{1} Contrary to popular belief, deep neural networks rarely perform interpolation even in i.i.d. settings. In high dimensions, test points are extremely unlikely to lie within the convex hull of training points \cite{24, 34}.

\textsuperscript{2} The simplicity bias is not a property of neural networks themselves, but also of their training with SGD, since it is possible to manually construct networks with arbitrarily poor generalization \cite{87} i.e. no simplicity bias.

\textsuperscript{3} We adopt the definition of simplicity of a feature from \cite{70}: it is the minimum number of linear pieces in the decision boundary that achieves optimal classification accuracy using this feature. The definition naturally extends to the simplicity of a model implementing this decision boundary.
background rather than the shape of the object. In natural language understanding, an example is to use the presence of certain words rather than the overall meaning of a sentence. These shortcuts are inevitable byproducts the data collection process e.g. from selection biases. They are increasingly problematic as tasks tackled with deep learning grow in complexity. The mechanisms to learn are more and more likely to be overshadowed by simpler spurious patterns.

**Role of inductive biases in OOD generalization.** OOD Generalization or strong generalization is the capability of making accurate predictions under arbitrary covariate shifts. To achieve this, a model must learn and reflect the intrinsic (i.e. causal) mechanisms of the task of interest. For example, recognizing objects in arbitrary scenes requires a model to learn about their shape and details. It cannot rely solely on the background or contextual cues (Figure 1a). OOD Generalization fundamentally requires extra information beyond i.i.d. training examples [6, 68]. Existing methods use side information such as multiple training environments [1, 12, 56], counterfactual examples [25, 73], or non-stationary time series [23, 30, 58]. Importantly, OOD generalization is not achievable only through regularizers, network architectures, or unsupervised control of inductive biases [6]. To make this limitation intuitive, consider the task of image recognition in Figure 1. Should a bird label result from a bird shape or from a blue sky? If shape and background are equally predictive of training labels, the data simply lacks the information to prefer one over the other (i.e. the task is underspecified: the same data could support a task where the labels relate to the background and not the objects). This is where a learning algorithm’s inductive biases come into play, possibly detrimentally. The simplicity bias favors the most linearly-predictive patterns, but these may be spurious. While existing methods attempt to integrate extra information during training, we show this can be deferred to a model selection stage.

**This study.** We seek to control the simplicity bias of neural networks and investigate benefits in OOD generalization. Variations of architectures, hyperparameters, or random seeds have no effect on the simplicity bias. Instead, we train a collection of similar models to fit the training data in different ways. Each model is optimized for standard empirical risk minimization (ERM) while a regularizer encourages diversity across the collection. It pushes each model to rely on different patterns in the data, including complex ones that are otherwise ignored because of the simplicity bias. Identifying a model with good OOD performance is reduced to an independent model selection step that can use any type of side information such as those mentioned above.

**Applicability of our method.** We use three image recognition datasets to demonstrate improvements in generalization relevant to computer vision. Issues with OOD generalization are also root causes of adversarial vulnerabilities [26], some model biases [49, 66], and poor cross-domain/dataset transfer [76]. Potential benefits in these areas remain to be investigated. Kariyappa et al. [32] already demonstrated improved adversarial robustness by increasing diversity in an ensemble with a method similar to ours.

**Summary of contributions.**

1. We review the fundamental requirements for OOD generalization and derive a rationale for addressing generalization during model selection rather than training.
2. We describe a method to overcome the simplicity bias by learning a collection of diverse predictors.
3. We demonstrate these benefits on existing benchmarks.

(a) A new capability to learn multiple predictive patterns otherwise ignored because of the simplicity bias (multi-dataset collages [70]).

(b) Improved activity recognition after training on visually-biased data (Biased Activity Recognition dataset [48]).

(c) Improved object recognition across visual domains (PACS dataset [36]).

Far from a complete solution to OOD generalization, this paper highlights the need for a better understanding and control of inductive biases in deep learning.

**2. Background**

**Simplicity bias.** Deep learning is actively studied to understand reasons for its successes and failures. The simplicity bias [51, 70], gradient starvation [57], and the learning of functions of increasing complexity [57] help explain the lack of robustness of deep neural networks and why their performance degrade under minor distribution shifts and adversarial perturbations. Shah et al. [70] showed that neural networks trained with SGD are biased to learn the simplest predictive features in the data while ignoring others. Worryingly, approaches like ensembles and adversarial training – believed to improve generalization and robustness – are ineffective at mitigating the simplicity bias.

**Shortcut learning** [17, 35] is synonymous with poor OOD generalization. It happens when a model learns predictive

---

4 OOD Generalization goes beyond the in-domain (ID) generalization of classical learning theory. Perfect ID generalization (i.e. reaching Bayes error rate on a test set from the same distribution as the training data) is achievable with infinite training data, but the predictions may rely entirely on spurious correlations (e.g. recognizing birds from blue skies).

5 Model selection for OOD performance cannot be achieved with a standard (in-domain, ID) validation set [6]: high ID performance can be attained by relying on spurious patterns, which says nothing about the model’s capabilities OOD. An OOD validation set is a valid option that makes this step similar to the cross-validation routinely used to select architectures and hyperparameters.
patterns that do not correspond to the task of interest. For example in object recognition (Figure 1), the model uses the background rather than the shape of an object [5, 18]. The model is accurate on in-domain (ID) data (i.e. from the same distribution as the training set) but is correct for the wrong reasons. Failures are apparent on OOD test data where the spurious patterns learned during training cannot be relied on. The more complex the task, the larger the space of possible spurious patterns that are simpler than the mechanisms of task, and the more likely is a case of shortcut learning. The simplicity bias exacerbates shortcut learning.

OOD Generalization (i.e. avoiding shortcut learning) is fundamentally not attainable solely with ERM [68]. ERM learns any pattern predictive of training labels. OOD generalization requires knowing which patterns correspond to causal mechanisms of the task (Figure 1). This information is lost by sampling i.i.d. training examples from the joint distribution produced by the data-generating process [6]. Current approaches to recover the missing information use multiple training environments [1, 12, 56], counterfactual examples [25, 73], or non-stationary time series [23, 30, 58]. Other options to improve OOD generalization rely on ad hoc task-specific knowledge [3, 11, 41, 48, 71, 77].

Ensembles. This paper is not about building ensembles. Ensembling means that multiple models are combined for inference. Rather, we train a collection of models and identify one for inference (experiments include ensembles for comparison). The goal of ensembling is to combine models with uncorrelated errors into one of lower variance. Our comparison. The goal of ensembling is to combine models with uncorrelated errors into one of lower variance. Our goal is to discover predictive patterns normally missed by a learning algorithm because of its inductive biases.

See Appendix D for an extended literature review.

3. Proposed method

Method overview. We train a collection of models in parallel (see Figure 2). A diversity regularizer encourages them to represent different functions. They share the same architecture and data. The regularizer is required because trivial options such as training models with different initial weights, hyperparameters, architectures, or shuffling of the data do not prevent converging to very similar solutions affected by the simplicity bias as demonstrated in [70].

Setup. We consider a supervised learning task, where one model is typically trained on a training set of examples $T = \{x^k, y^k\}_{k=1}^K$. The vectors $x$ represent input data such as images and $y$, in the case of a classification task, vectors of ground truth scores $[0,1]^C$ over $C$ classes. The standard practice is to train a model (typically a neural network) for empirical risk minimization (ERM) on $T$. A model implements a function $F : \text{supp}(x) \rightarrow \text{supp}(y)$. We represent it as a composition $F = g \circ f$ of a feature extractor $f_\theta(^\cdot)$ and classifier $g_\phi(^\cdot)$ parametrized by weights $\theta$ and $\phi$ respectively. We further define the hidden representation $h = f_\theta(x)$. The model $F$ is typically optimized to minimize the risk $\mathcal{R}$ of a predictive loss $\mathcal{L}_{\text{classification}}$ on $T$ by solving

$$\min_{\theta, \phi} \mathcal{R}(F_{\theta, \phi}) \quad (1)$$

with the risk

$$\mathcal{R}(F_{\theta, \phi}) = \sum_{k=1}^K \mathcal{L}_{\text{classification}}(\hat{y}^k, y^k) \quad (2)$$

and predictions $\hat{y}^k = F_{\theta, \phi}(x^k) = g_\phi(f_\theta(x^k))$. (3)

In the following, we call a predictor any function $F_{\theta^*, \phi^*}$ from the chosen hypothesis space (e.g. neural networks of a certain architecture) where $(\theta^*, \phi^*)$ is a solution to (1).

Why we sometimes need more complexity. The simplicity bias exposed in [70] implies that a neural network trained with SGD for (1) relies on the simplest features predictive of labels in $T$. It ignores more complex ones even if equally predictive. The simplicity of a feature is defined in [70] as the minimum number of linear pieces in the decision boundary achieving optimal classification using this feature. We further assume that a predictor relying this feature implements its corresponding simple decision boundary (although not stated explicitly in [70] it seems supported by their experiments). If a simple spurious pattern exists in the data, the simplicity bias will prevent from learning any more complex mechanism of the task. In such cases, it is desirable to force learning a more complex predictor.

How diversity can induce complexity. By assumption of the simplicity bias, the default predictor learned by solving (1) with SGD is the simplest. In other words, the model learned by default lies at one end of the space of solutions. A diverse set of solutions departing from the default one will necessarily include more complex models, that represent more complex decision boundaries and rely on different features of the data.

How to quantify diversity. We compare the functions implemented classifiers using their input gradients i.e. the gra-
dient of their output with respect to their input. Given any two classifiers \( g_{\phi_1} \) and \( g_{\phi_2} \), we quantify their similarity at a point \( h \in \text{preimage}(g) \) with

\[
\delta_{g_{\phi_1}, g_{\phi_2}}(h) = \nabla_h g_{\phi_1}^*(h) \cdot \nabla_h g_{\phi_2}^*(h)
\]

where the dot product measures the alignment of gradients. Since \( g \) is vector-valued, we denote with \( \nabla g^* \) the gradient of its largest component (top predicted score). We apply (4) below to encourage diversity over a collection of models.

**Complete proposed method.** Instead of training one model, we train a collection of models \( \{ F_i \} \) in parallel, where \( F_i = g_{\phi_i} \circ f \). They share an optional feature extractor \( f \) (e.g., a ResNet) for computational reasons, whereas the model-specific classifiers \( g_{\phi_i} \) are small multi-layer perceptrons (MLPs) in our experiments. We replace the training objective (1) with

\[
\min_{(\theta_i, (\phi_i)_i)} \sum_i \mathcal{R}(g_{\phi_i} \circ f_\theta) + \lambda \sum_{i \neq j} \sum_k \delta_{g_{\phi_i}, g_{\phi_j}}(h^k)
\]

where the scalar \( \lambda \) controls the strength of the regularizer. The first term is the ERM objective. It ensures a low training error and usual asymptotic guarantees for in-domain data. The second term is the diversity regularizer. It minimizes the alignment of input gradients over pairs of models at all training points. We solve (5) by SGD, with \( \{ \theta_i \} \) initialized differently to break the initial symmetry.

### 3.1. Additional considerations

**Rationale for input gradients.** Intuitively, we want each model to rely on different features in the data. And input gradients are indicative of the features used by the model [69]. Input gradients have the advantage of being implementation-invariant [72] (applicable to any differentiable model) and directly relevant to OOD predictions. The predictions of a classifier \( g_{\phi_i} \) at a test point of features \( h \) are denoted \( g_{\phi_i}(h) \). Assuming \( g \) continuous and differentiable, the approximation with a first-order Taylor expansion about a nearby training point of features \( h^u \) gives

\[
g_{\phi_i}(h) = g_{\phi_i}(h^u) + (h - h^u) \nabla_h g_{\phi_i}^u(h^u).
\]

The ERM objective clamps the value at training points which makes the first term identical \( \forall i \). But the diversity regularizer makes the second term different, causing predictions to diverge more and more across models as one moves away from training points. Simple alternatives in weight space (e.g., pushing parameters apart) would not guarantee learning different functions, since two networks can be equivalent under permutations and scalings of weights.

**Evaluating the diversity of learned predictors.** Our method produces a collection models, one of which has to be selected for inference. The selection is necessary, just like the cross-validation routinely performed to select hyperparameters and architectures, or the ubiquitous practice of early stopping. The choice of a selection method (see below) is orthogonal to our contribution of alleviating the simplicity bias. Therefore, the goal of our experiments is to demonstrate an increase in diversity (in terms of OOD performance) of the models that the selection can then operate on. Therefore we report the mean, ensemble, and maximum accuracy (oracle selection) of all models of a collection. We also perform a cross-dataset evaluation (Section 4.3) to verify that the maximum accuracy is meaningful and not merely an example of “overfitting to the test set”.

**Model selection for OOD performance.** It is important to remember that selection for OOD performance cannot, by definition, be performed with a validation set from the same distribution as the training data [74]. OOD generalization fundamentally requires additional information beyond i.i.d. data [6, 68] (see Section 2). Our approach uses standard i.i.d. data during training, which means that extra information has to be brought in during model selection. It can be as simple as a small OOD validation set, and other options include any technique used to evaluate OOD performance: contrast sets, counterfactual examples [16], inspection through with explainability techniques and expert knowledge [62], etc. This flexibility of options is possible because we only require the extra information for model selection, compared to existing OOD methods that require extra information attached to every training example e.g. in multiple training environments [1, 12, 56], counterfactual examples [25, 73], or non-stationary time series [23, 30, 58].

**Computational cost.** Feeding all classifiers with the same mini-batches keeps the training cost small. We found no difference with feeding different mini-batches. Memory and compute scale linearly with the number of classifiers, but these are small MLPs with a tiny footprint compared to the shared feature extractor. With enough memory, the operations of all models can even be parallelized, hence no decrease in throughput. The computation of the diversity regularizer reuses the input gradients that are byproducts of the backpropagation necessary to train the classifiers. The only added cost is in the computation of second derivatives to optimize the regularizer. All our experiments were run on a single laptop (!) with a GeForce GTX 1050 Ti GPU.

We include an FAQ with past reviews in Appendix A. See Appendix A for FAQs from readers and reviewers.

### 4. Experiments

We designed a set of experiments to answer two questions.

1. Can we learn predictive patterns otherwise ignored by standard SGD and existing regularizers? (Section 4.1).
2. Are these patterns relevant for OOD generalization in computer vision tasks? (Sections 4.2 and 4.3).
Class 0
Zero, pullover
automobile, zero.

Class 1
One, coat
truck, one.

Figure 3. Training examples of collages. Each block features one of two pre-selected classes from MNIST, Fashion-MNIST, CIFAR-10, SVHN. All four blocks are predictive of the labels. Because of the simplicity bias, a standard classifier latches on MNIST and ignores others.

4.1. Multi-dataset collages

Dataset. We extend the collages of MNIST/CIFAR (see Figure 1a) used in previous investigations of the simplicity bias [70]. We also use Fashion-MNIST [82] and SVHN [50] to form four-block collages for a binary classification task (see Figure 3). Each block features one of two pre-selected classes from the corresponding dataset (details in Appendix E). In the training data, the contents of all four blocks are predictive of the labels. Because of the simplicity bias however, a standard model systematically relies on the MNIST digit and completely ignores other parts of the image. The dataset simulates the absence of prior preference for any image region, typical in vision tasks. Therefore the goal is to learn predictive patterns from all four blocks. This is evaluated with four test sets, in which the contents of all blocks but one are randomized to either of its two classes. The dataset available at https://github.com/dteney/collages-dataset.

Results of baselines. To test the simplest possible implementation of our method, we use a fully-connected 2-hidden layer MLP classifier (details in Appendix E). We obtained upper bounds on the accuracy attainable with this architecture with four training sets where all blocks but one are randomized (Table 1, top row). This provides a ranking of learning difficulty: MNIST, SVHN, Fashion-MNIST, CIFAR. We trained the baseline with popular regularizers. In all cases (32 models per experiment, repeated 5 times) the models use the MNIST digit exclusively and never perform above chance (50%) on the other test sets.

Results of our method. We then trained a collection of the same number of models (32) with our diversity regularizer. In every case (five runs) the collection contains models that use all four parts of the images. We determined that the models specialize but do not overlap: a model is typically good on one of the four test sets at a time (see Table 5 in the appendix). A larger number of models also seems beneficial (see discussion in Section 5). This is partly explained with the observation that, even with a large number of models, a larger fraction relies on the simpler MNIST and

<table>
<thead>
<tr>
<th>Collages dataset (accuracy in %)</th>
<th>Best model on</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MNIST</td>
</tr>
<tr>
<td>Upper bounds: one predictive block in tr.</td>
<td>99.7 ± 0.0</td>
</tr>
<tr>
<td>Baseline, 32 models with different seeds</td>
<td>99.7 ± 0.0</td>
</tr>
<tr>
<td>With dropout (best rate: 0.5)</td>
<td>98.7 ± 0.0</td>
</tr>
<tr>
<td>With penalty on L1 norm of gradients</td>
<td>98.9 ± 0.0</td>
</tr>
<tr>
<td>With Jacobian regularization [28]</td>
<td>98.8 ± 0.0</td>
</tr>
<tr>
<td>With spectral decoupling [57]</td>
<td>99.1 ± 0.0</td>
</tr>
<tr>
<td>Proposed, 8 models</td>
<td>97.3 ± 0.0</td>
</tr>
<tr>
<td>Proposed, 16 models</td>
<td>96.6 ± 0.0</td>
</tr>
<tr>
<td>Proposed, 32 models</td>
<td>95.6 ± 0.0</td>
</tr>
<tr>
<td>Proposed, 64 models</td>
<td>95.5 ± 0.0</td>
</tr>
<tr>
<td>Proposed, 96 models</td>
<td>95.8 ± 0.0</td>
</tr>
</tbody>
</table>

Table 1. Results on collages. The upper bounds are obtained by training the baseline four times on data where all blocks but one are randomized. Other rows correspond to the training of 32 models, of which we report the best one on each test set. All existing methods fixate on the MNIST block. Ours discovers predictive signals from all four blocks (mean and std. dev. over five runs).

Baseline, 16 models

Proposed, 16 models

Best model on region: MNIST SVHN Fashion-M CIFAR

Figure 4. Visualization of input gradients \( \nabla h g^*(h) \) (abs. val. averaged over 10 test images; brighter means higher value). Each model specializes in different image regions.

Fashion-MNIST blocks than on the others. There is still room for improvement since our best models do not quite reach the upper bounds. Finally, a manual inspection of input gradients similar to some interpretability methods [69] is an easy way to assess which part of the image is used (see Figure 4). Combined with expert task knowledge, this could serve for model selection in some applications.

4.2. Biased activity recognition (BAR)

Figure 5. Training and test (in red) examples from BAR.

Dataset. The BAR dataset [48] was recently introduced
to evaluate debiasing methods for image recognition. The task is to classify photographs into six activities (see Figure 5). Training images were sampled from six pairs of actions/places in imSitu [85] e.g. climbing/rock. Test images were sampled from the same actions in different places e.g. climbing/ice. The goal is to learn a model that relies more on a person’s appearance than on the background to recognize actions in arbitrary places. This is challenging because both are predictive of the action training labels. This represents an ubiquitous setting where training data is affected by selection biases but an image recognition model is still expected perform well on novel scenes.

Results. We follow [48] and implement a classifier on ResNet-50 features (details in Appendix E). We first tune a strong baseline classifier (almost equating the method in [48]). We then train a collection of these classifiers with our regularizer. The accuracy of the best model improves from 64.9 to 67.1 (Table 2). The average accuracy over the collection also increases, perhaps surprisingly. Indeed, an increase in diversity could induce as many worse models than better ones. But remember that the space of predictors ranked by complexity is one-sided (Section 3). The baseline is at the “simplest” end. Those learned with our method are more complex. With BAR, complex models happen to be better, as analyzed in [48]. Thanks to this, a simple ensemble (summing all predicted scores) improves over the same ensemble of models trained without our regularizer (63.1 → 66.1) with no model selection. We insist however that this is not a universal benefit of our method.

The BAR dataset contains no information to prefer backgrounds or persons’ appearance. With the same training data and annotations, the task could as well be to recognize places rather than actions! This reinforces our insistence that side information is necessary for OOD generalization. Debiasing methods like [48] rely on task-specific design choices. Our approach is of more general purpose.

The existing Jacobian regularizer [28] and spectral decoupling [57] produce models almost as good as our best one, although worse on average as seen with a lower ensemble accuracy. The advantage of our method is to produce a collection of diverse models, whereas any specific regularizer is either suited or not to the given task. If not, the practitioner has to manually find another one.

4.3. Domain generalization (PACS)

Dataset. PACS is a standard benchmark for visual domain generalization (DG) [36]. It contains images from seven classes and four visual domains: art paintings, cartoons, photographs, and sketches. It is used in a leave-one-out manner with three training domains and the remaining one for testing. The standard baseline uses images from all domains indistinctively whereas DG methods use domain labels to try and identify common features that should also be reliable in the test domain.

Results. We report an ablative study in Table 3 (see Appendix E for implementation details). We obtain a better model by training a collection with our regularizer, compared to the baseline where differences within the collection result only from different random seeds. The improvements are modest but they clearly result from our regularizer, as seen from a heatmap of accuracy as a function of the number of models and regularizer strength (also see Figure 6, left). A larger number of models also seem beneficial. We provide the training curves of all models in a collection in Figure 6. Looking at the OOD accuracy as training progresses, we see that the regularizer induces vastly more variance, producing both worse and better models as expected.

As observed with BAR, the average accuracy within a collection improves slightly with the regularizer – although not to the point that a naive ensemble would be beneficial.

| Table 2. Evaluation on BAR (mean accuracy and std. dev. over five runs). Each row corresponds to the training of 64 models, unless otherwise noted. The heat map (right; accuracy of best model) shows that the diversity regularizer clearly improves over a classical ensemble of independently-trained models (bottom row). |
This corroborates the explanations given above for BAR.

We include an additional cross-dataset evaluation i.e. zero-shot transfer (Table 3, last column). We use the test images from VLCS [19] for a detection task of classes both in PACS and VLCS (horse and person; other VLCS classes serve as negatives). We report the area under curve (AUC) averaged over these two classes. We observe that the best model selected on PACS also brings substantial improvements on VLCS. This verifies that the accuracy of the best model on PACS is meaningful and not merely an example of “overfitting to the test set”.

We evaluate regularizers previously proposed to improve generalization. We spent substantial effort tuning these alternatives to their best and trying multiple variants (see Appendix E for details). As seen in Table 3, a Jacobian regularizer provides small improvements. So does minimizing the squared L2 norm of the gradient. Our regularizer mainly minimizes the alignment between pairs of gradients but it can also reduce their absolute norm as a side effect, just like these two alternatives. The ablation shows this to be beneficial but also that it does not entirely explain the benefits of our method. Indeed on VLCS, only our regularizer provides a substantial improvement (74.57 → 79.66).

We provide a comparison with state-of-the-art DG methods in Table 4. Most methods rely on labels of training domains, which we do not use. Most were demonstrated on top of weak baselines, as pointed out in [21]. We use the same ResNet-18 feature extractor, we highly optimized it, and we still get substantial improvements. See Section 5 for more discussion about the comparison with DG methods.

5. Discussion

The above results show that (1) we now have a tool to expand the set of solutions learned by a neural network through SGD and (2) some solutions are relevant to computer vision as evidenced by improved OOD generalization.

Limitations of the method. The main hyperparameters are the regularizer strength and the number of models learned. Any number > 1 obviously gives more options than the single model learned by default. Empirically, larger numbers (>64) seem beneficial, but we did not derive guarantees that a robust predictor will be found. The chosen number of models may not induce the optimal “granularity”. Too small a number could produce a model that relies on multiple entangled features including robust and spurious ones. And too large a number could produce multiple models that each implement a different part of a robust solution.

An interesting direction for future work would be to encourage learning a “basis” of elemental predictors (suggested in [63]) for a given dataset by promoting notions of sparsity or complementarity. The model selection could thereafter search for their optimal combination as attempted in [53]. The approach would resemble the disentanglement methods of representation learning [39] but it would operate in the space of predictors optimized for ERM, rather than the space of features optimized for reconstruction.

The need for model selection. The utility of evading the simplicity bias is in improving OOD performance because this is where the simplicity bias shows its detri-
On the opposite, we show that the two methods for datasets like PACS (Section 4.3) include a selection step. This allows more flexibility in the type and sophistication. The key novelty is to require no extra information during training contrary to existing methods that require upfront task-specific knowledge (e.g., debiasing methods [3, 11, 41, 48, 71, 77]) or additional annotations [11, 56, 73]. The requirement for side information (besides a training set of i.i.d. examples) is fundamental for OOD performance (see [6, 68] and additional background in Appendix B) but we defer its use to an independent model selection step. This allows more flexibility in the type and quantity of side information used.

Limitations of the evaluation. In a review on domain generalization (i.e. OOD generalization across visual styles), Gulrajani and Lopez-Pas [21] note that model fitting and model selection are equally hard. They recommend that methods for datasets like PACS (Section 4.3) include a selection strategy. On the opposite, we show that the two steps can be completely decoupled. Our analysis focuses on the best learned model (denoted oracle selection in [21] and best in [53]). This optimistic choice is justified because it is the performance achievable with an optimal selection strategy. This accounts for existing and future selection strategies e.g. the calibration-based method of Wald et al. [80] that came out after the writing of this paper.

Is the comparison unfair with methods for PACS that train a single model? We do not think so: existing methods were selected at the paper level. Methods with no improvement on the test set did not get published. It is unlikely that authors never peeked at test-set performance until getting published! It was even showed in [21] that all tested methods lost their benefits when deprived of heavy hyperparameter tuning and early stopping based on OOD performance. Our approach makes the selection more explicit.

Heuristics about simplicity. Our improvements support previous claims [51, 70] that more complex models are sometimes preferable. Works in NLP have even argued that any simple correlation in a dataset is likely to be spurious [11, 78, 88]. We point out that such heuristics are necessarily task- or dataset-specific. However, we also remark that the tasks addressed with deep learning are increasingly complex (take visual question answering for example [75]). This implies that the space of potentially-spurious patterns that are simpler than the task in any given dataset is also growing. The above heuristics may therefore have a practical utility. It remains crucial to study their limits of applicability. They cannot be universal [45] and cannot obviate the need for extra information (or task-specific knowledge) in our model selection step.

Universality of inductive biases. The inductive biases of any learning algorithm cannot be universally superior to another’s [81]. For example, weight decay, Jacobian regularization, or even data augmentation are only as good as they are tuned to a particular task. In comparison, our method does not affect inductive biases in a directed way. It only increases the variety of the learned models, so it could be seen as a “meta-regularizer”. Our results also show that intuitions notions behind classical regularizers like smoothness (Jacobian regularization), sparsity (L1 norm), or simplicity (L2 norm) are sometimes detrimental.

Any quest for universal architectures, regularizers, data augmentations, or even dataless selection strategies [46, 88] is known to be futile. Benefits can only apply to a subset of learning tasks [81]. Questions remain: how small or vast is the subset of learning tasks that humans care about? Which properties of naturally-produced data make some inductive biases generally useful? Deep learning has proven surprisingly successful. Studying its inductive biases will help understanding its limits of applicability. And methods to control these biases will help expanding these limits.

<table>
<thead>
<tr>
<th>PACS Dataset</th>
<th>Art</th>
<th>Cartoon</th>
<th>Photo</th>
<th>Sketch</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test style</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(leave-one-out)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-SAM baseline [15]</td>
<td>77.9</td>
<td>75.9</td>
<td>95.2</td>
<td>69.3</td>
<td>79.6</td>
</tr>
<tr>
<td>D-SAM*</td>
<td>77.3</td>
<td>72.4</td>
<td>95.3</td>
<td>77.8</td>
<td>80.7</td>
</tr>
<tr>
<td>Epi-FCR baseline [37]</td>
<td>77.6</td>
<td>73.9</td>
<td>94.4</td>
<td>74.3</td>
<td>79.1</td>
</tr>
<tr>
<td>Epi-FCR*</td>
<td>82.1</td>
<td>77.0</td>
<td>93.9</td>
<td>73.0</td>
<td>81.5</td>
</tr>
<tr>
<td>DMG baseline [9]</td>
<td>72.6</td>
<td>78.5</td>
<td>93.2</td>
<td>65.2</td>
<td>77.4</td>
</tr>
<tr>
<td>DMG*</td>
<td>76.9</td>
<td>80.4</td>
<td>93.4</td>
<td>75.2</td>
<td>81.5</td>
</tr>
<tr>
<td>DecAug baseline [4]</td>
<td>78.4</td>
<td>78.3</td>
<td>94.2</td>
<td>72.1</td>
<td>80.8</td>
</tr>
<tr>
<td>DecAug*</td>
<td>79.0</td>
<td>79.6</td>
<td>95.3</td>
<td>75.6</td>
<td>82.4</td>
</tr>
<tr>
<td>JiGen baseline [8]</td>
<td>77.9</td>
<td>74.9</td>
<td>95.7</td>
<td>67.7</td>
<td>79.1</td>
</tr>
<tr>
<td>JiGen</td>
<td>79.4</td>
<td>75.3</td>
<td>96.0</td>
<td>71.4</td>
<td>80.5</td>
</tr>
<tr>
<td>Latent domains</td>
<td>78.3</td>
<td>75.0</td>
<td>96.2</td>
<td>65.2</td>
<td>78.7</td>
</tr>
<tr>
<td>baseline [44]</td>
<td>81.3</td>
<td>77.2</td>
<td>96.1</td>
<td>72.3</td>
<td>81.8</td>
</tr>
</tbody>
</table>

Table 4. Comparison with existing methods on PACS. For each method, we mention the accuracy of the baseline reported by its authors, and of the method itself. All methods are based on a ResNet-18. "Require labels of tr. domains. Our method discovers additional predictive features in the data. It returns a collection of models, among which some clearly have a better OOD performance (last row) than the baseline simply trained with different random seeds (82.9 → 84.2).
References

[24] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning: data min-
ing, inference, and prediction. Springer Science & Business Media, 2009. 1


[50] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5


