In Search of Robust Measures of Generalization

Gintare Karolina Dziugaite[1][*]**, Alexandre Drouin**[1][*]**, Brady Neal**[1,2,3]**, Nitarshan Rajkumar**[2,3]**,**

Ethan Caballero[2,3]**, Linbo Wang**[4]**, Ioannis Mitliagkas**[2,3]**, Daniel M. Roy**[4,5]

[1]Element AI, [2]Mila, [3]Université de Montréal, [4]University of Toronto, [5]Vector Institute

Abstract

One of the principal scientific challenges in deep learning is explaining generalization, i.e., why the particular way the community now trains networks to achieve small training error also leads to small error on held-out data from the same population. It is widely appreciated that some worst-case theories—such as those based on the VC dimension of the class of predictors induced by modern neural network architectures—are unable to explain empirical performance. A large volume of work aims to close this gap, primarily by developing bounds on generalization error, optimization error, and excess risk. When evaluated empirically, however, most of these bounds are numerically vacuous. Focusing on generalization bounds, this work addresses the question of how to evaluate such bounds empirically. Jiang et al. [9] recently described a large-scale empirical study aimed at uncovering potential causal relationships between bounds/measures and generalization. Building on their study, we highlight where their proposed methods can obscure failures and successes of generalization measures in explaining generalization. We argue that generalization measures should instead be evaluated within the framework of distributional robustness.

1 Introduction

Despite tremendous attention, a satisfying theory of generalization in deep learning remains elusive. In light of so many claims about explaining generalization in deep learning, this statement is somewhat controversial. It also raises an important question:

What does it mean to explain generalization in deep learning?

In this work, we propose empirical methodology to aid in the search of a precise mathematical theory, allowing us to leverage large-scale empirical studies of generalization, like those in recent work [8, 9]. Unlike earlier work, however, our proposal rests on the foundation of *robust prediction*, in order to catch out, rather than average out, failures.

The dominant approach to studying generalization is the frequentist framework of statistical learning theory. We focus our attention on the simplest setting within supervised classification, where the training data, S, are modeled as a sequence of n random variables, drawn i.i.d. from a distribution $\mathcal{D}$ on labeled examples (x, y). In supervised classification, learning algorithms choose a classifier h_S, based on the training data S. Ignoring important considerations such as fairness, robustness, etc., the key property of a classifier h is its probability of error, or (classification) risk,

$$R_{\mathcal{D}}(h) = \mathbb{P}_{(x,y)\sim\mathcal{D}}[h(x) \neq y]. \tag{1}$$

One of the key questions is why deep learning often produces classifiers with human-level risk in domains that stymied researchers for decades. In this work, we take an empirical perspective and

---

[*]Equal contribution. Correspondence to: {karolina.dziugaite, alexandre.drouin}@elementai.com

judge theories of generalizations by the predictions they provide when tested. In the other direction, any systematic rule for predicting generalization—whether learned or invented—can be thought of as a theory that can be tested.

We consider families of environments, defined by data distributions, architectural choices, train set sizes, learning algorithms and their tuning parameters, etc. Given a particular family of environments, a strong theory achieves a desired level of precision for its predictions, while depending as little as possible on the particular details of the environments. At one extreme, explanations based on the VC dimension of the zero–one loss class of neural networks would pin the success of deep learning on empirical risk minimization. In practice, these explanations are poor, not just because the ensuing bounds are numerically vacuous for the size of networks and datasets used in practice, but because they fail to correctly predict the effect of changes to network width, depth, etc.

At the other extreme, average risk on held-out data (i.e., a test-set bound) provides a sharp estimate of risk, yet the computation to produce this estimate is inextricably tied to every detail of the learned weights and data distribution. Viewing predictors as theories, the test-set bound is essentially silent. Any satisfactory theory of generalization in deep learning must therefore lie between these two extremes. We must necessarily exploit properties of the data distribution and/or learning algorithm, but we must also be willing to trade precision to decrease our dependence on irrelevant details.

What dependence on the data distribution and learning algorithm is necessary to explain deep learning? Even taking the data distribution into consideration, the fact that stochastic gradient descent (SGD) and its cousins often perform empirical risk minimization cannot explain generalization [29]. There is, however, a picture emerging of overparametrization and SGD conspiring to perform capacity control. In some theoretical scenarios, this control can be expressed in terms of norms. At the same time, great strides have been made towards identifying notions of capacity that can be shown to formally control generalization error, $R_{\mathcal{D}}(h) - \hat{R}_S(h)$, uniformly over h belonging to specially defined classes. (Here $\hat{R}_S(h)$ denotes the empirical risk, as estimated by the training data.)

Despite this progress, there is still a wide gulf between performance as measured empirically via held-out data and performance as predicted by existing theory. No expert would be surprised to discover that a published bound yields predictions for risk that are numerically vacuous when evaluated empirically. A standard retort is that the constants in most bounds are suboptimal or that the purpose of bounds is to identify qualitative phenomena or inspire the development of new algorithms. Even ignoring the issue of numerically vacuous bounds, many bounds demonstrably fail to account for the right dependencies. As a case in point, recent empirical work [17] identifies state-of-the-art bounds on generalization error that grow with training set size, while the generalization error actually shrinks. Indeed, many bounds are distribution- or data-dependent and so the question of whether they explain generalization in practice is an empirical question.

1.1 Large-scale empirical studies

Recent work proposes large-scale empirical investigations to study generalization [9]. (See also [8].) While it is becoming more common for theoretical work to present empirical evaluations, among recent empirical studies [2, 3, 5, 11, 12, 19, etc.], most are limited. One motivation for large-scale empirical studies is to leverage massive computing resources in the pursuit of a scientific challenge that has largely been approached mathematically. Another motivation is to go beyond simply measuring correlation towards measuring aspects of causation. (Several authors of this work—Dziugaite, Neal, and Roy—have each advocated for this publicly.)

Given how influential these proposals by Jiang et al. [9] may be, we believe they deserve critical attention. (Indeed, recent preprints have already started to integrate their methodology.) Jiang et al. [9] propose to use Kendall correlation coefficients and independence tests to evaluate a suite of so-called *generalization measures*. Many of these generalization measures are frequentist bounds, though with missing constants or lower-order terms. Others are only loosely inspired by bounds.

One of central proposals made by Jiang et al. is to *average* evaluation metrics (i.e., Kendall-τ) over a range of experimental settings, which arise from targeted changes to hyperparameters.[2] In contrast, we argue that averaging across experimental settings is *not* an appropriate summarization of the strength of a generalization measure as a theory of generalization. In particular, a satisfying

---

[2]Jiang et al. also propose a conditional-independence test, which we discuss in Appendix E.

theory should admit a generalization measure that offers reasonable predictions of generalization across the entire range of experimental settings under study. A theory—realized by a generalization measure—is as strong as its weakest part: *a satisfying theory cannot simply predict well on average*.

The study of prediction across a range of environments is the subject of *distributional robustness* [1, 4]. An extreme form of robustness is obtained when one seeks to predict well in all environments that may arise from all possible interventions to an experimental setting. This extreme form of robustness can be linked to a weak form of causality [4, 16].

Crucially, we do not aim for robustness over all possible environments. To achieve some level of generality, useful theories must necessarily have limited scope. As we demonstrate in Section 2, frequentist generalization bounds can exploit noncausal correlations that can be seen to stand in for unknown properties of the data distribution because of properties of the learning algorithm. Such bounds have an important role to play in our search for a theory of generalization in deep learning, but we cannot expect them to explain generalization under interventions that upset these noncausal correlations. Bounds that depend on properties of the data distribution have an important role to play, though one hindered by the statistical barriers of unknown distributions, accessible only through a limited pool of data. More general theories (that minimize this dependence) can pinpoint key data properties.

Contributions. Theories of generalization yield predictions: How should we evaluate these predictions empirically? In this work, we adopt the proposal of [9] to exploit large-scale empirical studies, but critique the use of averaging in the evaluation of predictions made by these theories. Based on the specific scientific goals of understanding generalization, we propose that the framework of distributional robustness is more appropriate, and suggest how to use it to evaluate generalization measures.

Besides theoretical contributions, we make empirical contributions: We collect data from thousands of experiments on CIFAR-10 and SVHN, with various values for width, depth, learning rate, and training set size. We adopt the ranking task and sign-error loss introduced by [9], but use the collected data to perform a *robust* ranking evaluation, across the 24 candidate generalization measures on over 1,600,000 pairs of experiments.

We find that *no existing complexity measure has better robust sign-error than a coin flip*. Even though some measures perform well on average, every single measure suffers from 100% failure in predicting the sign change in generalization error under some intervention. This observation is not the end of the evaluation, but the beginning.

To better understand the measures, we evaluate them in families of environments defined by interventions to a single hyperparameter. We find: (i) most, though not all, measures are good at robustly predicting changes due to training set size; (ii) robustly predicting changes due to width and depth is hard for all measures, though some PAC Bayes-based measures are robust across a large fraction of the environments tested; (iii) norm-based measures outperform other measures at learning rate interventions.

By focusing on robust evaluation, we force ourselves to dig into the data to uncover the cause of failures—failures which might otherwise go undiscovered by looking at average performance. As such, robustness provides better guidance to the scientific challenge of explaining generalization. The rest of this paper is organized as follows. In Section 2, we present a concrete example of a frequentist analysis of a learning algorithm, which reiterates some of the high-level points above. We then introduce distributional robustness in Section 3 and describe how the framework can be used to analyze large-scale empirical studies of generalization measures in Section 4. We detail our experimental setting in Section 5 and summarize our experimental findings in Section 6 before ending with a discussion.

2 A Motivating Example: SVMs, Norm-based Capacity Control, and the Role of Causality

In this section, we study support vector machines (SVMs) to demonstrate some of the challenges in understanding and explaining generalization error. This section owes much to [27]. The intuition extracted from this simple model motivates our methodological choices for the rest of this paper. In

particular, we see that frequentist generalization bounds derived under one set of conditions may rely on quantities that do not have a direct causal relationship with the generalization error under other conditions. This highlights that frequentist bounds can be expected to have limits to the predictive powers under intervention, but also that asking for causal measures of generalization may rule out measures that nonetheless work well in a range of scenarios.

Consider linear prediction, based on an embedding of inputs into $\mathbb{R}^p$. As usual, we index the space of linear predictors by nonzero vectors $\boldsymbol{w} \in \mathcal{H} = \mathbb{R}^p$, where the decision boundary associated to $\boldsymbol{w}$ is the tangent hyperplane $\{\boldsymbol{x} \in \mathbb{R}^p : \langle \boldsymbol{w}, \boldsymbol{x} \rangle = 0\}$, passing through the origin. Assuming labels take values in $\{\pm 1\}$, the zero–one classification loss of the predictor $\boldsymbol{w}$ on a labeled input $(\boldsymbol{x}, y)$ is $\ell(\boldsymbol{w}, (\boldsymbol{x}, y)) = \frac{1}{2}(1 + y \operatorname{sgn}(\langle \boldsymbol{x}, \boldsymbol{w} \rangle))$. Note that the loss is invariant to the magnitude of the vector $\boldsymbol{w}$, and so the set of hyperplanes can be put into correspondence with the unit vectors $\overline{\boldsymbol{w}} := \boldsymbol{w}/\|\boldsymbol{w}\|$. We focus on the realizable setting, i.e., data are assumed to be labeled according to some hyperplane. In this case, every finite data set admits a positive cone of empirical risk minimizers.

Let $S = \{(\boldsymbol{x}_i, y_i)\}_{i \in [n]}$ be n i.i.d. labeled data in $\mathbb{R}^p \times \{\pm 1\}$, and let $\boldsymbol{w}_S$ be chosen according to the SVM rule: $\min_{\boldsymbol{w}} \|\boldsymbol{w}\|^2$, subject to the constraint that $y_i \langle \boldsymbol{x}_i, \boldsymbol{w} \rangle \geq 1$ for all $i \in [n]$. The constraint demands that, for each data point, the functional margin, $y_i \langle \boldsymbol{x}_i, \boldsymbol{w}_S \rangle$, be greater than one. Thus the hyperplane $\overline{\boldsymbol{w}_S}$ indeed separates the data and achieves zero empirical risk. However, among the vectors that satisfy the margin constraint, $\boldsymbol{w}_S$ has the smallest L2 norm. Geometrically, the hyperplane $\overline{\boldsymbol{w}_S}$ is that with the largest *geometric* margin, $\min_i y_i \langle \boldsymbol{x}_i, \overline{\boldsymbol{w}_S} \rangle$.

Why does the SVM classifier generalize? The best explanation may depend on the situation. The VC dimension of the space of p-dimensional linear predictors is p, and so, with high probability over the sample S, uniformly over all separating hyperplanes $\boldsymbol{w}$, the difference between the empirical risk and risk is $\tilde{O}(p/n)$. If $n \gg p$, then this reason alone suffices to explain strong performance. The details of the SVM rule are irrelevant beyond it returning an empirical risk minimizer.

Suppose that we consider a family of embeddings of growing dimensionality and find that the SVM rule generalizes equally well across this family. The VC theory cannot explain this. A theory based on the maximum-margin property of the SVM rule may. To that end, assume there exists a hyperplane $\boldsymbol{w}_*$ such that $y \langle \boldsymbol{w}_*, \boldsymbol{x} \rangle \geq 1$ with probability one over the pair $(\boldsymbol{x}, y)$. To fix a scale, assume $\|\boldsymbol{x}\| \leq \rho$ with probability one. By exploiting strong convexity, and the fact that $\|\boldsymbol{w}_S\| \leq \|\boldsymbol{w}_*\|$, one can show that the risk of $\boldsymbol{w}_S$ is bounded by $\tilde{O}(\rho\|\boldsymbol{w}_*\|/n)$. Note that this bound has *no explicit dependence* on the dimension p. Instead, it depends on the quantity $\rho\|\boldsymbol{w}_*\|$, whose reciprocal has a geometric interpretation: the distance between the separating hyperplane $\overline{\boldsymbol{w}_*}$ and the nearest data point, normalized by the radius of the data. Therefore, this analysis trades dependence on dimension for dependence on the data distribution's density near the decision boundary. When $\rho\|\boldsymbol{w}_*\| \ll n$, SVM's inductive bias is sufficient to explain generalization, even if $p \gg n$.

In fact, we can always build a bound based on the norm of the learned weights: with high probability, for *every* ERM $\boldsymbol{w}_S$, the risk is bounded by $\tilde{O}(\rho\|\boldsymbol{w}_S\|/n)$. One might prefer such a bound since $\|\boldsymbol{w}_*\|$ is often presumed unknown. Even if this bound matches risk empirically, it has a strange property: The bound depends on the norm $\|\boldsymbol{w}_S\|$ even though risk is *independent of norm*. Thus, we cannot expect the bound to remain valid if we intervene on the norm after training, e.g., to test for a causal relationship between norms and risk. Norms are the effect of the data and SVM interacting.

This example highlights that there may be multiple overlapping explanations depending on the range of environments in which one wants to understand generalization. We cannot, however, expect a theory to be robust to arbitrary interventions. Identifying a theory with limitations may lead us to more general ones, once we understand those limitations. All of this motivates a careful design of experimental methodology, in order to navigate these tradeoffs. In particular, we demand that a theory is *robustly predictive* of generalization over a carefully designed family of *environments*.

3 Preliminaries on Robust Prediction

In this section, we introduce the framework of robust prediction, borrowing heavily from Bühlmann [4], Peters, Bühlmann, and Meinshausen [23], and Rothenhäusler et al. [25]. In the next section, we cast the problem of studying generalization into this framework.

Consider samples collected in a family $\mathcal{F}$ of different *environments*. In particular, let $(\Omega, \mathcal{A})$ denote a common (measurable) sample space and, in each of these environments $e \in \mathcal{F}$, assume the data

we collect are drawn i.i.d. from a distribution P^e on Ω. We will think of environments as representing different experimental settings, interventions to these experiments, sub-populations, etc. For example, each sample might be a covariate vector and binary response, i.e., $\Omega = \mathbb{R}^p \times \{0, 1\}$. A well-studied setting is where the distributions P^e all agree on the conditional mean of the response given the covariates (i.e., the regression function), but disagree on the distribution of the covariates.

Prediction is formalized by a *loss function*. In particular, a loss function for a set Φ of predictors is a map $\ell : \Phi \times \Omega \to \mathbb{R}$. The *error* or *risk* (of a predictor $\phi \in \Phi$ in an environment $e \in \mathcal{F}$) is then the expected loss, $\mathbb{E}_{\omega \sim P^e}[\ell(\phi, \omega)]$. If we focus on one environment $e \in \mathcal{F}$, it is natural to seek a predictor $\phi \in \Phi$ with small risk *for that individual environment*. However, if we care about an entire family $\mathcal{F}$ of environments, we may seek a predictor that works well simultaneously across $\mathcal{F}$. In the setting of *distribution robustness*, the performance of a predictor relative to a family $\mathcal{F}$ of environments is measured by the *robust error (or risk)*

$$\sup_{e \in \mathcal{F}} \mathbb{E}_{\omega \sim P^e}[\ell(\phi, \omega)]. \tag{2}$$

The goal of robust prediction is to identify a predictor with small robust risk.

Connection to causality.   If taken to an extreme, then robust prediction is closely related to learning causality. Specifically, suppose that (X, Y) is induced by a common causal model $Y := f(X)$. If $\mathcal{F}$ represents all possible interventions on subsets of X, then the causal predictor $f(X)$ also minimizes the robust risk. See [4, 24] for more details.

4   Studying Generalization via Distributional Robustness

We are interested in understanding the effects of changes to a complex machine learning experiment, with a focus on effects on generalization. In this section, we cast this problem into the framework of distributional robustness. In order to study generalization, we view theories of generalization as yielding predictors for generalization under a range of experimental settings. We use the term *generalization measure* to refer to such predictors.

4.1   Experimental Records and Settings

In the notation of Section 3, points $\omega \in \Omega$ represent possible samples. In our setting, each sample represents a complete record of a machine learning experiment. An environment e specifies a distribution P^e on the space Ω of complete records.

In the setting of supervised deep learning, a complete record of an experiment would specify hyperparameters, random seeds, optimizers, training (and held out) data, etc. Ignoring concerns of practicality, we assume the complete record also registers every conceivable derived quantity, not only including the learned weights, but also the weights along the entire trajectory, training errors, gradients, etc. Formally, we represent these quantities as random variables defined on the probability spaces $(\Omega, \mathcal{A}, P^e)$, $e \in \mathcal{F}$. Among these random variables, there is the empirical risk $\hat{R}$ and risk R of the learned classifier, and their difference, G, the generalization error/gap.

Each distribution P^e encodes the relationships between the random variables. Some of these relationships are common to all the environments. E.g., the generalization error G always satisfies $G = R - \hat{R}$, and the empirical risk $\hat{R}$ is always the fraction of incorrectly labeled examples in the training data. Some relationships may change across environments. E.g., in a family $\mathcal{F}$ designed to study SGD, changes to, e.g., the learning rate, affect the distribution of the trajectory of the weights.

In machine learning, environments arise naturally from learning algorithms applied to benchmarks under standard hyperparameter settings. In order to evaluate theories that explain the effect of, e.g., hyperparameter changes, we also consider environments arising from perturbations/interventions to standard settings. E.g., we may modify the hyperparameters or data, or intervene on the trajectory of weights in some way. Every perturbation e is captured by a different distribution P^e.

With respect to a family of environments $\mathcal{F}$, a generalization measure is preferred to another if it has smaller robust error (2). In Sections 5 and 6, we restrict our attention to $\mathcal{F}$ induced by varying hyperparameters, data distributions, training datasets, and dataset sizes. In this work, we do not intervene on the dynamics of SGD. However, intervening on the trajectory induced by SGD might be an interesting future direction that could allow one to tease apart the role of implicit regularization.

5

4.2 Prediction tasks

The predictions associated with a theory of generalization are formalized in terms of a map $C : \Omega \to \mathbb{R}$, which we call a *generalization measure*. We will study ad hoc generalization measures as well as ones derived from frequentist bounds. In both cases, we are interested in the ability of these measures to predict changes in the generalization.

One important aspect of a generalization measure is the set of (random) variables (i.e., covariates) it depends on. Indeed, there is an important difference between the task of predicting generalization using only the architecture and number of data and using also, e.g., the learned weights. Formally, let $V_1, \ldots, V_k$ be a finite collection of random variables. A generalization measure C is $\sigma(V_1, \ldots, V_k)$-*measurable* if there exists a map g such that $C(\omega) = g(V_1(\omega), \ldots, V_k(\omega))$ for all $\omega \in \Omega$. We may prefer one generalization measure to another on the basis of the covariates it uses. As a simple example, if a generalization measure offers comparable precision to another measure, but is measurable with respect to a strict subset of variables, then this increased generality may be preferred.

Goals of the prediction. We are broadly interested in two types of prediction tasks, distinguished by whether we train one or two networks.

In *coupled-network* experiments, we train two networks, such that they share all hyperparameters except one. We are interested in trying to predict which network has smaller generalization error.

Some of the generalization measures we consider are based on generalization bounds from the literature. Given that generalization bounds are often numerically vacuous, it would not be informative to evaluate their predictions directly at this stage. It is, however, reasonable to evaluate whether they capture the right dependencies. Indeed, one desirable property of evaluating generalization measures by the rankings they induce in coupled-network experiments is that the rankings are invariant to monotonically increasing transformations of the measure.

In *single-network* experiments, we try to predict the numerical value of the generalization error for that network based on a linear or affine function of a generalization measure. Generalization measures that perform well in such a task would serve as accurate predictors of generalization, and could be used for, e.g., model selection. However, such measures would not necessarily serve to be useful in generalization bounds. We describe the experimental details and results of *single-network* experiments in Appendix B due to space limitations.

5 Experimental methodology

In coupled-network experiments, we evaluate the *ranking* that the generalization measure induces on training networks. The approach we describe here is a robust analogue of the Kendall-τ-based approach advocated by Jiang et al. [9].[3] This change is deceptively minor. We highlight the very different conclusions drawn using our methodology in Section 6.

Evaluation criterion. In more detail, recall that a coupled-network environment e determines a distribution P^e on pairs (ω, ω') of *variable assignments*, each representing a full record of an experiment. We evaluate a generalization measure, C, and the realized generalization error, G, on both assignments, ω and ω'. We use the ranking of C values to predict the ranking of G values. Then, the *sign-error of a generalization measure C* for this task[4] is given by

$$\mathrm{SE}(P^e, C) = \tfrac{1}{2}\mathbb{E}_{(\omega,\omega')\sim P^e}\left[1 - \mathrm{sgn}\left(G(\omega') - G(\omega)\right) \cdot \mathrm{sgn}\left(C(\omega') - C(\omega)\right)\right]. \qquad (3)$$

Given a family $\mathcal{F}$ of coupled-network environments, the *robust sign-error of a generalization measure C* is $\sup_{e\in\mathcal{F}} \mathrm{SE}(P^e, C)$. The Ψ summary proposed by Jiang et al. [9] is analogous to the average sign-error, $|\mathcal{F}|^{-1} \sum_{e\in\mathcal{F}} \mathrm{SE}(P^e, C)$.[5]

In our experiments, we use a modification of the loss in Eq. (3) in order to account for Monte Carlo variance in empirical averages. We use a weighted empirical average, where the weight for a sample

---

[3]See Appendix E for a comparison to Jiang et al. [9]'s conditional-independence-based approach.

[4]In order to match [9], in all of our experiments, we train to a fixed level of cross entropy loss that also yields zero training error. Since $G = R - \hat{R}$, and $\hat{R} = 0$, a prediction task that depends on changes in generalization error G is equivalent to one that depends on changes in risk R.

[5]We apply a $\frac{1-\Psi}{2}$ transformation to obtain values in $[0, 1]$, where 1 is achieved if $\Psi = -1$ and 0 if $\Psi = 1$.
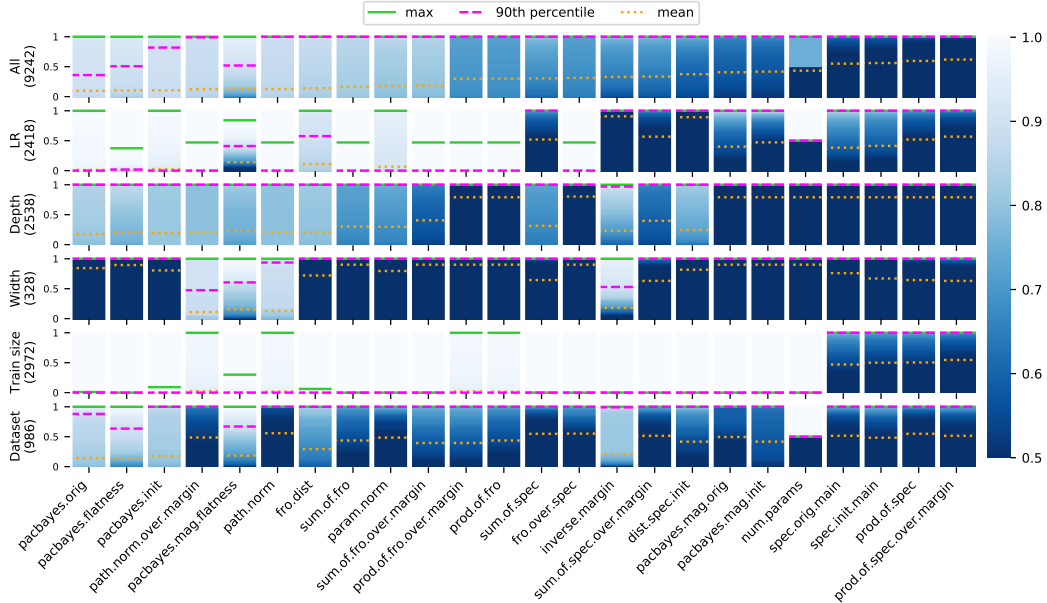
Figure 1: Cumulative distribution of the sign-error across subsets of environments for each generalization measure. The measures are ordered based on the mean across 'All' environments. A completely *white* bar indicates that the measure is perfectly robust, whereas a *dark blue* bar indicates that it completely fails to be robust.

(ω, ω') is calculated based on the difference in generalization error $|G(\omega) - G(\omega')|$. We discard samples for which the difference in generalization error is below the Monte Carlo noise level. In effect, we control the precision to which we want our generalization measure to predict changes: when the difference is insignificant, we do not predict the sign. See Appendix A for the details on how we use the Monte Carlo variance to choose what environments are being considered. Other details of data collection are described in Appendix C.

Environments. In our experiments, variable assignments (ω) are pairs (H, σ) of hyperparameter settings and random seeds, respectively. The hyperparameters are: learning rate, neural network width and depth; dataset (CIFAR-10 or SVHN), and training set size. (See Appendix C for ranges.)

Each environment e is a pair (H, H') of hyperparameter settings that differ in the setting of *one* hyperparameter (e.g., depth changes from $2 \rightarrow 3$ between H and H' and the remaining hyperparameters are identical). The distribution P^e for a pair $e = (H, H')$ is the distribution of $(\omega, \omega') = ((H, \sigma), (H', \sigma'))$, where the random seeds σ, σ' are chosen uniformly at random. That is, the expectation in Eq. (3) is taken only over a random seed.

6   Empirical Findings

In Fig. 1, we present a visualization of 1,600,000 ranking evaluations on 24 generalization measures derived from those used in [9]. A full description of these measures can be found in Appendix C.6. Motivated by the discussion in the introduction, we seek strong predictive theories: generalization measures that increase monotonically with generalization error and for which this association holds across a range of environments. Such a measure would achieve zero robust sign-error (Eq. (3)).

As described in Section 5, each environment contains a pair of experiments that share all hyperparameters but one (learning rate, depth, width, train set size, dataset). In each environment, we calculate the weighted empirical average version of the sign-error over 100 samples from P^e (10 networks runs with different seeds per ω. Note that we discard environments where too many samples have differences in generalization error below the Monte Carlo noise level (see Appendix A.2

7

for details). This is in contrast with the protocol proposed by Jiang et al. [9] where such noise is not filtered and can significantly undermine the estimation of sign-error (see Appendix A.3).

In the remainder of this section we interpret the results of Fig. 1, highlight some significant shortcomings of the generalization measures, and point out cases where these shortcomings would have been obscured by non-robust, average-based summary statistics like those used by Jiang et al. [9].

How to read Fig. 1. This figure presents the empirical cumulative distribution function (CDF) of the sign-error across all environments and generalization measures. **Every row** shows the CDF over a subset of environments (e.g., those where only depth is varied). The 'All' row shows the same but over all environments. The number of environments in each subset is given on the left of each row. **Each bar** in the figure is the empirical CDF of all sign-errors in the set of environments. A bar's y-axis corresponds to the range of possible sign-errors and the internal coloring depicts the distribution (starting at the median value for improved readability). We annotate the bars with the max (i.e., robust sign-error; **green**), the 90th percentile (**magenta**), and the mean (**orange**). The latter statistics do not measure robustness over all environments. However, a low 90th percentile value means the measure would have had low empirical robust sign-error restricting to some 90% of the environments tested. If the max is at 1.0, then there exists at least one environment where the measure fails to predict the sign of the change in generalization on all random seeds. If the max is below 0.5, then the measure is more likely than not to predict the correct sign on *all environments* in the set. *Identifying subfamilies in which a measure is robust is one of our primary objectives.*

No measure is robust. As illustrated in the 'All' row, for every one of the 24 measures, there is at least one environment in which the measure *always incorrectly predicts* the direction of change in generalization. Nonetheless, some measures have low robust error over large fractions of environments, as reflected by the 90th percentiles of the sign-error distributions. Notice how the average-based summaries proposed by Jiang et al. [9] do not reflect robustness, which implies their inability to detect the causal associations that they seek. Given these poor results, we must dig deeper to understand the merits and shortcomings of these generalization measures. Therefore, we study their performance in natural subfamilies $\mathcal{E} \subseteq \mathcal{F}$ of environments. Our analyses of the 'Train Size', 'Depth', and 'Width' rows below are examples of this. While no measure is robust across the CIFAR-10 and SVHN datasets considered here, we find measures that are quite robust over a 90% fraction of environments for SVHN only (see Appendix D.1).

Robustness to train set size changes is not a given. In the 'Train size' row, most measures correctly predict the effect of changing the train set size. (In general, generalization error decreases with train set size.) It may seem a foregone conclusion that a bound of the form $\tilde{O}(\sqrt{c/n})$ would behave properly, but, for most of these measures, the complexity term c is a random variable that can grow with more training data. In fact, while many measures do achieve a low robust sign-error, some measures fail to be robust. In particular, some bounds based on Frobenius norms (e.g., prod.of.fro; Appendix C.6.4 and [20]) increased with train set size in some cases. Such corner cases arose mostly for shallow models (e.g., depth 2) with limited width (e.g., width 8) and were automatically identified by our proposed method. Note that the same finding was recently uncovered in a bespoke analysis [17], and we may have missed this looking only at average sign-errors, which are usually low.

Robustness to depth. In the 'Depth' row, we depict robust sign-error for interventions to the depth. Again, robust sign-error is maxed out for every measure. Digging deeper, these failures are not isolated: many measures actually fail in most environments. However, there are exceptions: a few measures based on PAC-Bayes analyses show better performance in some environments. In Fig. 2, we dig into the performance of pacbayes.mag.flatness (Appendix C.6.6) by looking at the subset of environments where it performs well (e.g., varying depth $3 \rightarrow 4$), fails but shows signs of robustness (e.g., $3 \rightarrow 6$), completely fails (e.g., $4 \rightarrow 5$), and those were a conclusion cannot be reached (e.g., $5 \rightarrow 6$). Looking into the data, we found that almost all environments where the measure fails are from the CIFAR-10 dataset, where the smaller networks we test suffer from significant overfitting. This illustrates how our proposed methodology can be used to zero-in on the limited scope where a measure is robust.

Robustness to width is surprisingly hard. In the 'Width' row, all measures have robust sign-error close to 1. Looking into the data, we discover that generalization error changes very little in response to interventions on width because the networks are all very overparametrized. In fact, of the 4,000 available width environments, only 328 remain after accounting for Monte Carlo noise.

Comparison to Jiang et al. [9] Our contribution is primarily a methodological refinement to the proposals in [9]. We describe how to discover failures of generalization measures in specific environments by looking at worst-case rather than average performance. We note that there are several reasons that even our *average-case* results are not directly comparable with those in [9]. First, their analysis considers the CIFAR-10 and SVHN datasets in isolation, whereas we combine models trained on both datasets. Second, they do not account for Monte Carlo noise, which we found to significantly alter the distribution of sign-errors (see Appendix A.3). This is important since we found that many environments had to be discarded due to high noise (e.g., only 8.2% of the width environments remain after filtering out noise in our analysis). Third, the hyperparameters and ranges that they consider are different from ours and, consequently, both studies look at different populations of models. For example, the majority of models in [9] use dropout, whereas our models do not. Such differences can alter how generalization measures and gaps vary in response to interventions on some hyperparameters and lead to diverging conclusions. For instance, in our results, no measure has an average-case performance much better than a coin-flip in the 'Depth' environments for CIFAR10, while Jiang et al. [9] find measures that perform well in this context. Nevertheless, there are some general findings that persist across both
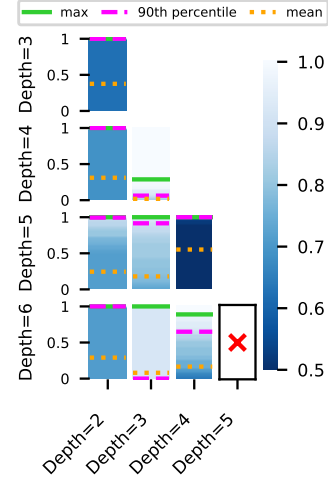

Figure 2: CDFs of sign-errors separated by environments where depth varies between two values for `pacbayes.mag.flatness`. The red X indicates that no environments remained after accounting for Monte Carlo noise.

studies; for instance, we see the good average-case performance of the path-norm (Appendix C.6.5) and PAC-Bayes-flatness-based (Appendix C.6.6) measures in contrast to the poor performance of spectral measures (e.g., `prod.of.spec`; Appendix C.6.3). We also find more specific similarities, such as the poor average-case performance of most measures in 'Width' environments for CIFAR-10 (Appendix D.2), in contrast to the better performance of `path.norm` (Appendix C.6.5), `path.norm.over.margin` (Appendix C.6.5), and `pacbayes.mag.flatness` (Appendix C.6.6).

7 Discussion

The quest to understand and explain generalization is one of the key scientific challenges in deep learning. Our work builds on recommendations in [9] to use large-scale empirical studies to evaluate generalization bounds. At the same time, we critique some aspects of these recommendations. We feel that the proposed methodology in [9] based on taking averages of sign-errors (or independence tests, which we have not pursued) can obscure failures. Indeed, for a long time, empirical work on generalization has not been systematic, and as a result, claims of progress outpace actual progress.

Based on an understanding of the desired properties of a theory of generalization, we propose methodology that rests on the foundation of distributional robustness. Families of environments define the range of phenomena that we would like the theory to explain. A theory is then only as strong as its worst performance in this family. In our empirical study, we demonstrated how a family can be broken down into subfamilies to help identify where failures occur. While the present work focused on the analysis of existing measures of generalization, future work could build on the robust regression methodology of Appendix B and attempt to formulate new robust measures via gradient-based optimization.

The development of benchmarks and quantitative metrics has been a boon to machine learning. We believe that methodology based on robustness with carefully crafted interventions will best serve our scientific goals.

Broader Impact

Our work aims to sharpen our understanding of generalization by improving the way that we evaluate theories of generalization empirically. The proposed methodology is expected to aid in the quest to understand generalization in deep neural networks. Ultimately, this could lead to more accurate and reliable models and strengthen the impact of machine learning in critical applications where accuracy must be predictable. We believe that this work has no direct ethical implications. However, as with all advances to machine learning, long-term societal impacts depend heavily on how machine learning is used.

References

[1] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. "Invariant risk minimization" (2019). arXiv: 1907.02893.

[2] S. Arora, R. Ge, B. Neyshabur, and Y. Zhang. "Stronger Generalization Bounds for Deep Nets via a Compression Approach". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, Oct. 2018, pp. 254–263.

[3] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. "Spectrally-normalized margin bounds for neural networks". In: *Advances in Neural Information Processing Systems*. 2017, pp. 6241–6250.

[4] P. Bühlmann. *Invariance, Causality and Robustness*. 2018. arXiv: 1812.08233.

[5] G. K. Dziugaite and D. M. Roy. "Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data". In: *Proceedings of the 33rd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. 2017. arXiv: 1703.11008.

[6] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. "Array programming with NumPy". *Nature* 585.7825 (Sept. 2020), pp. 357–362.

[7] J. D. Hunter. "Matplotlib: A 2D graphics environment". *Computing in Science & Engineering* 9.3 (2007), pp. 90–95.

[8] Y. Jiang, D. Krishnan, H. Mobahi, and S. Bengio. "Predicting the Generalization Gap in Deep Networks with Margin Distributions". In: *International Conference on Learning Representations*. 2019.

[9] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio. "Fantastic Generalization Measures and Where to Find Them". In: *International Conference on Learning Representations*. 2020.

[10] A. Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

[11]   T. Liang, T. Poggio, A. Rakhlin, and J. Stokes. "Fisher-Rao Metric, Geometry, and Complexity of Neural Networks". In: *Proceedings of Machine Learning Research*. Ed. by K. Chaudhuri and M. Sugiyama. Vol. 89. Proceedings of Machine Learning Research. PMLR, 16–18 Apr 2019, pp. 888–896.

[12]   Q. Liao, B. Miranda, A. Banburski, J. Hidary, and T. Poggio. *A Surprising Linear Relationship Predicts Test Performance in Deep Networks*. 2018. arXiv: 1807.09659 [cs.LG].

[13]   M. Lin, Q. Chen, and S. Yan. *Network In Network*. 2013. arXiv: 1312.4400 [cs.NE].

[14]   D. J. MacKay and D. J. Mac Kay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[15]   W. McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by S. van der Walt and J. Millman. 2010, pp. 56–61.

[16]   N. Meinshausen. "Causality from a distributional robustness point of view". In: *2018 IEEE Data Science Workshop (DSW)*. IEEE. 2018, pp. 6–10.

[17]   V. Nagarajan and J. Z. Kolter. "Uniform convergence may be unable to explain generalization in deep learning". In: *Advances in Neural Information Processing Systems (33)*. 2019. arXiv: 1902.04742.

[18]   Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. "Reading Digits in Natural Images with Unsupervised Feature Learning" (2011).

[19]   B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro. "The role of over-parametrization in generalization of neural networks". In: *International Conference on Learning Representations*. 2019.

[20]   B. Neyshabur, R. Tomioka, and N. Srebro. "Norm-based capacity control in neural networks". In: *Proceedings of the Eighth Annual Conference on Learning Theory*. COLT 2016. 2015, pp. 1376–1401. arXiv: 1503.00036v2 [cs.LG].

[21]   A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035.

[22]   F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[23]   J. Peters, P. Bühlmann, and N. Meinshausen. "Causal inference by using invariant prediction: identification and confidence intervals". *J. R. Stat. Soc. Ser. B. Stat. Methodol.* 78.5 (2016). With comments and a rejoinder, pp. 947–1012. ISSN: 1369-7412.

[24]   M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters. "Invariant models for causal transfer learning". *The Journal of Machine Learning Research* 19.1 (2018), pp. 1309–1342.

[25]   D. Rothenhäusler, N. Meinshausen, P. Bühlmann, and J. Peters. *Anchor regression: heterogeneous data meets causality*. 2018. arXiv: 1801.06229.

[26]   H. Sedghi, V. Gupta, and P. M. Long. "The Singular Values of Convolutional Layers". In: *International Conference on Learning Representations*. 2019.

[27]   S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[28]   Y. Yoshida and T. Miyato. *Spectral Norm Regularization for Improving the Generalizability of Deep Learning*. 2017. arXiv: 1705.10941 [stat.ML].

[29]   C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. "Understanding deep learning requires rethinking generalization". In: *International Conference on Representation Learning (ICLR)*. 2017. arXiv: 1611.03530v2 [cs.LG].

Appendix

Table of Contents

Contents

A   Importance sampling schemes to account for Monte Carlo noise

Generalization error $G(\omega)$ is estimated from a held-out test set, since we do not have access to the data distribution. The size of the test set determines the precision at which true generalization error can be approximated. Let G, G' denote true generalization errors for ω, ω', respectively, and similarly $\hat{G}, \hat{G}'$ estimates of generalization made on a test set of size m. Let $\epsilon = |\hat{G} - \hat{G}'|/2$. Then

$$\mathbb{P}(\mathrm{sgn}(G - G') = \mathrm{sgn}(\hat{G} - \hat{G}')) \geq \mathbb{P}(|G - \hat{G}| \leq \epsilon) \cap (|\hat{G}' - G'| \leq \epsilon)) \tag{4}$$

$$\geq (1 - \mathbb{P}(|G - \hat{G}| \geq \epsilon))(1 - \mathbb{P}(|\hat{G}' - G'| \geq \epsilon)). \tag{5}$$

We can now apply Hoeffding inequality to bound $\mathbb{P}(|G - \hat{G}| \geq \epsilon) \leq 2e^{-2m\epsilon^2}$. Let $\chi(\epsilon, m) = (1 - 2e^{-2m\epsilon^2})^2$. If $\hat{G}$ and $\hat{G}'$ are computed using m samples, then accepting samples only when $\chi(\epsilon, m) > p$ would mean that with probability at least p, $\mathrm{sgn}(G - G') = \mathrm{sgn}(\hat{G} - \hat{G}')$.

In our experiments, we only keep the samples with $\chi((\hat{G}(\omega') - \hat{G}(\omega))/2, m) > 0.5$. The expectation in the sign-error in Eq. (3) is approximated by a weighted average of the loss for each sample, where the weight is equal to a rescaled version of

$$\kappa(\omega, \omega') = \max(0, \chi((\hat{G}(\omega') - \hat{G}(\omega))/2, m) - 0.5). \tag{6}$$

A.1 Filtering environments

The weighting scheme proposed in Eq. (6) allows to downweigh (and in some cases discard) pairs of experiments for which the generalization errors do not differ significantly. Consequently, some environments may be left with very few samples. Let the ith sample have a weight κ_i. To avoid calculating the expected sign-error on too few data points, we discard environments where the *effective sample size*, defined as

$$n_{\text{eff}} = \frac{(\sum_i \kappa_i)^2}{\sum_i \kappa_i^2}, \tag{7}$$

is smaller than 12. In our case, the weights are as defined in Eq. (6). The choice of 12 samples means we estimate the expected loss to a precision of around standard deviation divided by 3 ([14][Ch. 29, p. 380]).

In Fig. 3, we show the number of environments remaining at various n_{eff} cutoffs. Notice that very few environments are included for the width hyperparameter, even at $n_{\text{eff}} \geq 12$. This is because the variations in generalization error due to width are often negligible in our data. Therefore, many of the environments where the width is varied are automatically discarded.
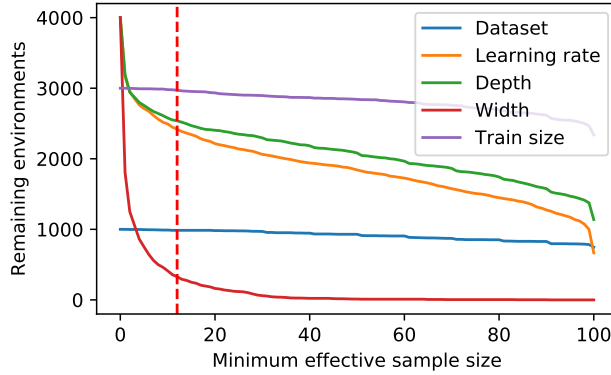

Figure 3: Number of environments remaining (per hyperparameter) at various effective sample size thresholds. The red vertical bar marks $n_{\text{eff}} = 12$, which is the minimum effective sample size that we consider. Notice how most environments where width is varied have small effective sample sizes.

A.2 Calculating the sign-error in empirical evaluation

Given the weighing and filtering schemes described above, we present the exact formulation of the sign-error used in the coupled-network experiments of Section 6. For each environment e, we sample seeds $\{\sigma_i, \sigma_i'\}_{i \in \{1,..,10\}}$, yielding a total of 100 samples $\{\omega_i, \omega_j'\}_{(i,j) \in \{1,..,10\}^{\otimes 2}}$. For the sake of computation, we then use $\mathcal{E} = \{(\omega_i, \omega_i')\}_{(i,j) \in \{1,..,10\}^{\otimes 2}}$ as our samples from P^e. We calculate the empirical sign-error as:

$$\widehat{\text{SE}}(\mathcal{E}, C) = \frac{1}{2} \mathbb{1}\Big(n_{\text{eff}} \geq 12\Big)$$
$$\sum_{(\omega,\omega') \in \mathcal{E}} \bar{\kappa}(\omega, \omega') \big[1 - \text{sgn}\left(\hat{G}(\omega') - \hat{G}(\omega)\right) \cdot \text{sgn}\left(C(\omega') - C(\omega)\right)\big], \tag{8}$$

where the effective sample size n_{eff} is defined in Eq. (7), and $\bar{\kappa}(\omega, \omega') = \dfrac{\kappa(\omega, \omega')}{\sum_{(\omega_i, \omega_i') \in \mathcal{E}} \kappa(\omega_i, \omega_i')}$ are normalized weights.

A.3 Ablation study: What is the effect of Monte Carlo noise?

In this section, we investigate how filtering Monte Carlo noise using the aforementioned procedure affects the estimated sign-errors. This is done by running an ablation study where the sign-errors
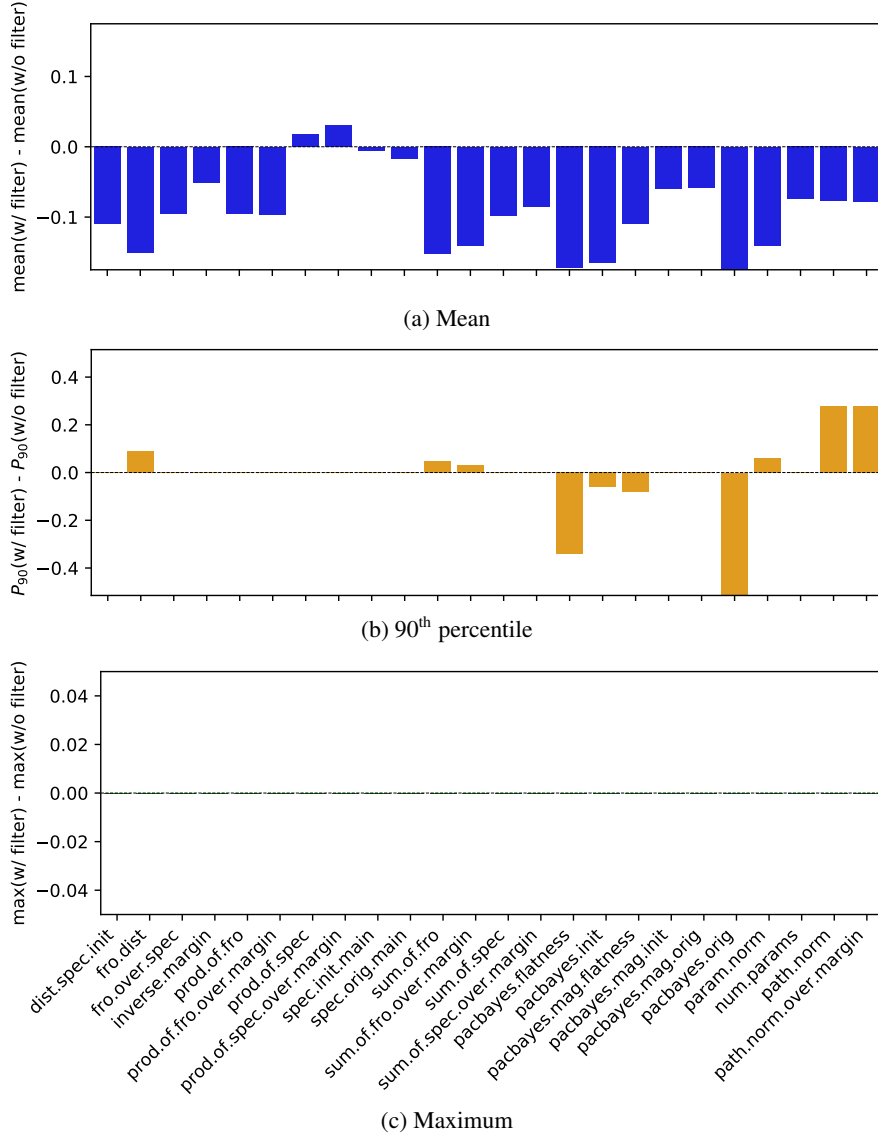
Figure 4: Effect of Monte Carlo noise filtering on various statistics of the sign-error distribution for each measure.

are computed with and without Monte Carlo noise filtering and the resulting error distributions are compared. Specifically, for the *with filtering* case, pairs (ω, ω') in each environment are weighted using $\kappa(\omega, \omega')$ as described at Eq. (6). Conversely, for the *without filtering* case, pairs are weighted with $\kappa(\omega, \omega') = 1$, allowing pairs with very small differences in generalization gap to be included in the expectation. Intuitively, Monte Carlo noise should increase the sign-error in environments where it occurs, since it can randomly push $\text{sgn}(\hat{G}(\omega) - \hat{G}(\omega'))$ to be $+1$ or -1, making it unpredictable.

The results reported in Fig. 4 support this hypothesis: including noisy ranking pairs generally leads to larger sign-errors. Indeed, for many generalization measures, the mean sign-error decreases with noise filtering (Fig. 4a). Moreover, extreme values such as the maximum do not seem to be affected by noise filtering (Fig. 4c), which was expected since noise is likely not prevalent in each of the considered environments. In light of these results, we conclude that the procedure that we propose to account for Monte Carlo noise (Appendix A) is beneficial and that it should be implemented in studies, such as ours and Jiang et al. [9], that rely on ranking comparisons of generalization gaps.

14

B Evaluating robust prediction of the numerical value of generalization error

In single-network experiments, we evaluate the ability of *generalization measures* to predict the exact numerical value of the generalization error.

Evaluation criterion.   We rely on a robust mean squared error (MSE) objective. For a transformation $f_\theta(\cdot)$ of a generalization measure C, the robust MSE is

$$\inf_\theta \sup_{e \in \mathcal{F}} \text{MSE}(P^e, f_\theta(C)) \text{ where } \text{MSE}(P^e, C') = \mathbb{E}_{\omega \sim P^e}(C'(\omega) - G(\omega))^2. \tag{9}$$

Choosing $f_a(x) = ax$ in Eq. (9), we recover *robust risk of the linear oracle transformation of a generalization measure C*. Similarly, choosing $f_{(a,b)}(x) = ax + b$, we get *robust risk of the affine oracle transformation of a generalization measure C*.

Environments.   In this setting, each environment $e \in \mathcal{F}$ is defined by a single hyperparameter configuration H. The data points in each environment are acquired by training a model with hyperparameters H and varying the random seed. For example, an environment could be composed of multiple experimental records where learning rate is 0.01, the model depth is 2, model width is 10, the dataset is CIFAR-10, and the training set size is 50 000. The hyperparameter values considered are those given at Appendix C and we consider ten random seeds, resulting in 1000 environments with ten data points each.

B.1 Experiments and results

In our experiments, we fit an *affine oracle* for each generalization measure by minimizing the robust mean squared error. This evaluates the ability of each measure to predict generalization error in each environment, up to a common linear rescaling and an additive constant. Note that we constrain the linear coefficients to be non-negative (i.e., $a \geq 0$), since we expect these measures to upper bound generalization error. We compare the performance of the affine oracles to that of a baseline that ignores the generalization measures and only fits a bias parameter (i.e., $a = 0$). The results are reported in Fig. 5.
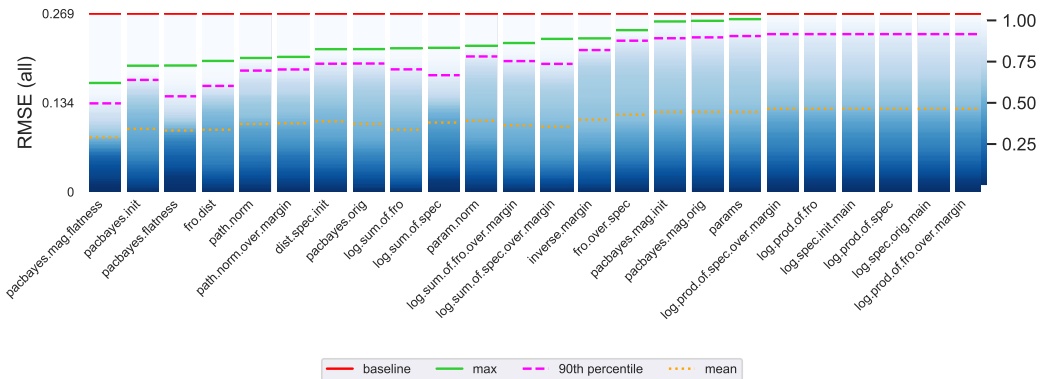

Figure 5: Cumulative distribution of robust root mean squared error (RMSE) of affine oracles trained on the 1000 environments defined at Appendix B.1. The maximum and average RMSE over all environments are shown in black and orange, respectively. A measure leading to perfect prediction of generalization error would have a white column and a black line at zero. The RMSE of a baseline oracle that only fits a bias parameter is shown in red.

Observe that many generalization measures achieve lower robust mean squared error than the baseline oracle. This suggests that these measures do carry meaningful information about generalization error that transfers across all environments. In addition, notice that many measures that show good robust performance in this setting often perform well in the coupled-network experiments (see Section 6 and Appendix D). For instance, the PAC-Bayes measures based on `flatness` show signs

15

of robustness (although not perfect) in both types of experiments. Furthermore, measures based on `path.norm`, which show strong signs of robustness in the SVHN-only coupled-network experiments (see Appendix D.1) are also among the best performers in this setting. Finally, notice how the average mean squared error tends to be very similar across measures, while their robust mean squared errors differ more. This provides additional evidence that averaging can mask failures in robustness and that a worst-case analysis should be preferred.

B.2 Exploring weaker families of environments

In this section, we consider families of environments $\mathcal{F}$ which are intermediate between the robust regression described above and empirical risk minimization (where the average MSE is minimized). Minimizing the robust risk should be an easier task in these environments, due to increasing levels of averaging which mask robustness failures.

Varying a single hyperparameter. In this setting, each environment $e \in \mathcal{F}$ is composed of runs where a single variable $H_i \in H$ is allowed to vary. All other variables $\mathcal{V} \setminus H_i$ are fixed, though the random seed varies. For example, an environment could be composed of all runs where the model depth is 2, model width is 10, the dataset is CIFAR-10, the training set size is $50\,000$, and the learning rate takes any of the 5 values considered. When considering the hyperparameter values described at Appendix C, we obtain 1350 environments with between 20 to 50 points each (due to 10 repeats per hyperparameter setting). The results for all experiments over this family of environments are reported in Fig. 6.

As expected, the range of robust MSE values attained per measure is lower than in Fig. 5, since the added averaging makes the task easier. As in the previous section, we can see that the mean MSE over environments obscures a clear ordering over measures that is present when using the robust MSE. Also, notice that the ordering induced by the robust MSEs is similar to that in Fig. 5, with some PAC-Bayesian measures and `path.norm` measures performing best.

Varying all but one hyperparameter. In this setting, each environment $e \in \mathcal{F}$ is composed of runs where a single variable $H_i \in H$ is fixed. All other variables $\mathcal{V} \setminus H_i$ and the random seed vary. For example, an environment could be composed of all runs where the model depth is 2 and the other parameters take on every possible value. When considering the hyperparameter values described at Appendix C, we obtain 21 environments with 2000 to 5000 points each. The results for all experiments over this family of environments are reported in Fig. 7.

The gap between the mean and robust RMSEs is much narrower here, caused by averaging the MSE over much more points. Nevertheless, we are still able to see a very similar ordering to that of Fig. 5 and Fig. 6 preserved here.
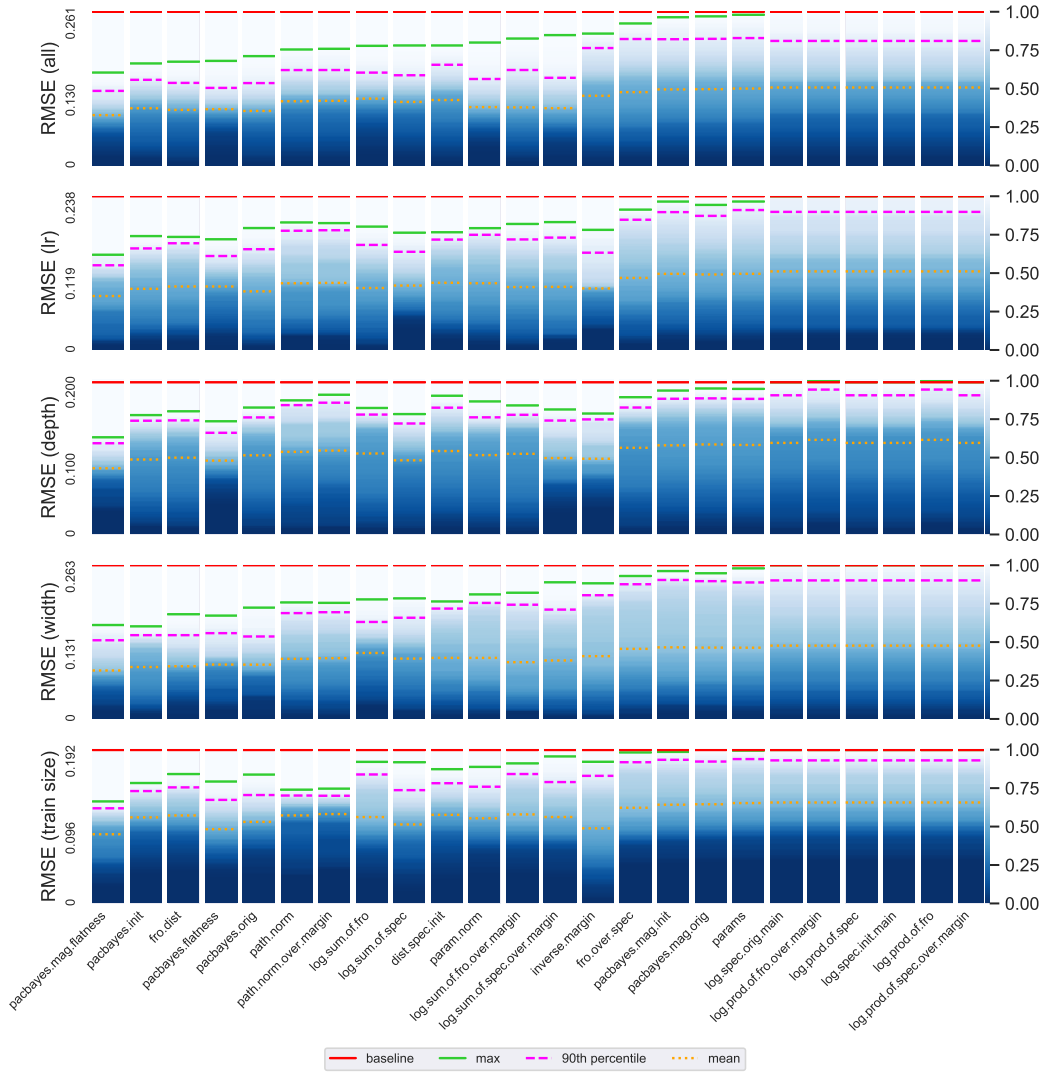
Figure 6: Cumulative distribution of robust root mean squared error (RMSE) of affine oracles trained on the 1350 environments where a single hyperparameter is varied and the others are fixed. Results are shown over all environments (top row) and separately for the subset of environments where each hyperparameter is varied (other rows). The maximum and average RMSE over all environments in each row are shown in black and orange, respectively. A measure leading to perfect prediction of generalization error would have a white column and a black line at zero. The RMSE of a baseline oracle that only fits a bias parameter is shown in red.
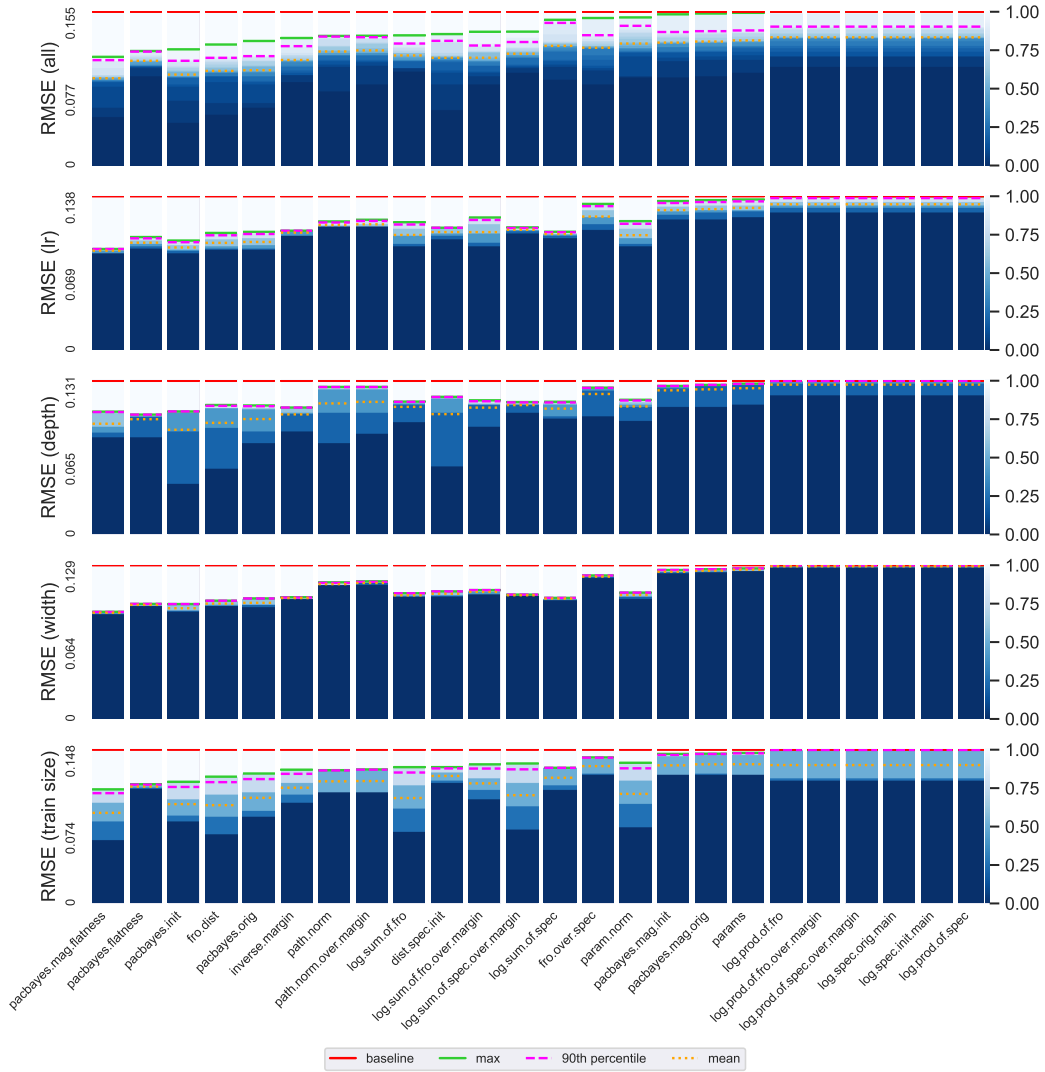
17

Figure 7: Cumulative distribution of robust root mean squared error (RMSE) of affine oracles trained on the 21 environments where one hyperparameter is kept fixed and all the others are varied. Results are shown over all environments (top row) and separately for the subset of environments where each hyperparameter is kept fixed (other rows). The maximum and average RMSE over all environments in each row are shown in black and orange, respectively. A measure leading to perfect prediction of generalization error would have a white column and a black line at zero. The RMSE of a baseline oracle that only fits a bias parameter is shown in red.

C   Experimental details, code, and data

Our experimental protocol is inspired by that used in the large-scale study of Jiang et al. [9]. We proceed in two steps: 1) data collection, where many models are trained with various hyperparameter configurations, 2) data analysis (e.g. coupled-network experiments; see Section 5). Our code relies on PyTorch [21], NumPy [6], Pandas [15], Scikit-Learn [22] and Matplotlib [7].

Availability: Our code is open-source and available online, along with the data used in the experiments: https://github.com/nitarshan/robust-generalization-measures

C.1   Hyperparameters

Each data point in our analysis is obtained by training a model with a given hyperparameter configuration. Between data points, we vary 5 hyperparameters that alter the model, the learning procedure, and the data distribution. These are:

1. Learning rate: $H_{lr} \in \{10^{-3}, 10^{-2.8}, 10^{-2.5}, 10^{-2.2}, 10^{-2}\}$
2. Model depth: $H_{depth} \in \{2, 3, 4, 5, 6\}$
   This corresponds to the number of "blocks" used in our model architecture. Each block corresponds to 3 convolutional layers.
3. Model width: $H_{width} \in \{8 \times 25, 10 \times 25, 12 \times 25, 14 \times 25, 16 \times 25\}$
   This corresponds to the filter width used in convolutional layers of our model architecture.
4. Dataset: $H_{dataset} \in \{\text{CIFAR-10}, \text{SVHN}\}$
5. Training set size: $H_{size} \in \{50\,000, 25\,000, 12\,500, 6\,250\}$

C.2   Models

We use a fully convolutional "Network-in-Network" architecture similar to that described in [13] and used for the study in [9]. A full specification of our model can be found in our codebase.

While the most successful model architectures of today employ residual connections between blocks of convolutional layers, we are unable to make use of those here due to the unclear applicability of many bounds to models using skip-connections.

C.3   Datasets

We make use of two common vision datasets: CIFAR-10 [10] and SVHN [18]. Both datasets are composed of 32x32 RGB images with 10 classes of natural images, with CIFAR-10's classes corresponding to animals and vehicles, and SVHN's classes corresponding to digits cropped from Street View images. We make use of the full training (50k images) and testing (10k images) splits of CIFAR-10, and randomly sample (without replacement) a subset of the larger training and testing sets of SVHN to match the split sizes of CIFAR-10. We also sample without replacement when generating the smaller training sets of size 25k, 12.5k and, 6.25k, but always make use of the same testing sets of size 10k across all experiments.

We do not make use of data augmentation when passing these images into our models, following the observation in [9] that doing so negatively affects the ability of these models to consistently reach low cross-entropies.

C.4   Training procedure

We use SGD with a momentum parameter of 0.9 for all experiments. We do not use learning rate decay or weight decay. As in [9] we use a cross-entropy stopping criterion, which we set to 0.01 for all experiments, and calculate over the entire training dataset.

C.5   Data collection

We run 10 repeats with different random seeds for each of the 1000 possible hyperparameter combinations, providing a total of 10, 000 experimental runs. Of these, 300 runs failed to meet the

cross-entropy criterion as well as an additional training accuracy criterion of being greater than 99%. These data points were filtered out before any analysis.

As in [9] we "fuse" the batch-norm layers of our model with their preceding convolutional kernels before calculating the value of the generalization measures.

C.6   Measures

We look at the following 24 generalization measures for convolutional networks, which are modifications of a subset of those studied in [9]. A key difference is that, while those original measures do not account for dataset size, we correct for this through a normalization of the form $\sqrt{C/m}$ for all measures C, where m is the size of the training dataset.

While we provide the mathematical expressions for all our measures here, we direct the reader to [9, Appendix D] for more details. It is worth noting, however, that while many of these expressions are derived from generalization bounds, there is no requirement that generalization measures correspond to bounds. Even among the measures that correspond to bounds, direct comparison is not necessarily meaningful. Some are generalization bounds for the networks learned by SGD. Some, in particular those derived from PAC-Bayes methods, are bounds on stochastic classifiers, e.g., obtained by randomizing the weights of a neural network in some way. Some of these quantities control the generalization error between the risk and empirical risk, while others relate to the difference between surrogate risks, e.g., based on margins.

We calculate the spectral norm of convolutional layers using the exact FFT-based method of [26]. We had initially used the approximation proposed in Yoshida and Miyato [28], but our findings were not noticeably different between the approximation and exact methods.

For reasons of numerical stability, we apply log transformations to some of these measures. **As this is a monotonic transformation, it does not affect the ranking results covered in the main paper.**

C.6.1   VC Measures

For a convolutional network of d layers, with a $k_i \times k_i$ kernel and c_i filters at depth i:

$$C_{params} = \sqrt{\frac{\sum_i^d k_i^2 c_{i-1}(c_i + 1)}{m}} \tag{10}$$

C.6.2   Output Measures

Let γ be the 10th-percentile of margin values over the training dataset.

$$C_{inverse.margin} = \sqrt{\gamma^2 m}^{-1} \tag{11}$$

20

C.6.3 Spectral Measures

Let $\mathbf{W}_i$ denote the ith convolutional layer's weight tensor, and $\mathbf{W}_i^0$ it's initial value. Let $\|\mathbf{W}_i\|_2$ denote it's spectral norm. Let $\|\mathbf{W}_i\|_F = \|\text{vec}(\mathbf{W}_i)\|$ denote it's Frobenius norm.

$$C_{log.spec.init.main} = \log \sqrt{\frac{\prod_{i=1}^{d} \|\mathbf{W}_i\|_2^2 \sum_{j=1}^{d} \frac{\|\mathbf{W}_j - \mathbf{W}_j^0\|_F^2}{\|\mathbf{W}_j\|_2^2}}{\gamma^2 m}} \tag{12}$$

$$C_{log.spec.orig.main} = \log \sqrt{\frac{\prod_{i=1}^{d} \|\mathbf{W}_i\|_2^2 \sum_{j=1}^{d} \frac{\|\mathbf{W}_j\|_F^2}{\|\mathbf{W}_j\|_2^2}}{\gamma^2 m}} \tag{13}$$

$$C_{log.prod.of.spec.over.margin} = \log \sqrt{\frac{\prod_{i=1}^{d} \|\mathbf{W}_i\|_2^2}{\gamma^2 m}} \tag{14}$$

$$C_{log.prod.of.spec} = \log \sqrt{\frac{\prod_{i=1}^{d} \|\mathbf{W}_i\|_2^2}{m}} \tag{15}$$

$$C_{fro.over.spec} = \sqrt{\frac{\sum_{i=1}^{d} \frac{\|\mathbf{W}_i\|_F^2}{\|\mathbf{W}_i\|_2^2}}{m}} \tag{16}$$

$$C_{log.sum.of.spec.over.margin} = \log \sqrt{\frac{d \left(\frac{\prod_{i=1}^{d} \|\mathbf{W}_i\|_2^2}{\gamma^2} \right)^{1/d}}{m}} \tag{17}$$

$$C_{log.sum.of.spec} = \log \sqrt{\frac{d \left(\prod_{i=1}^{d} \|\mathbf{W}_i\|_2^2 \right)^{1/d}}{m}} \tag{18}$$

C.6.4 Frobenius Measures

$$C_{log.prod.of.fro.over.margin} = \log \sqrt{\frac{\prod_{i=1}^{d} \|\mathbf{W}_i\|_F^2}{\gamma^2 m}} \tag{19}$$

$$C_{log.prod.of.fro} = \log \sqrt{\frac{\prod_{i=1}^{d} \|\mathbf{W}_i\|_F^2}{m}} \tag{20}$$

$$C_{log.sum.of.fro.over.margin} = \log \sqrt{\frac{d \left(\frac{1}{\gamma^2} \prod_{i=1}^{d} \|\mathbf{W}_i\|_F^2 \right)^{1/d}}{m}} \tag{21}$$

$$C_{log.sum.of.fro} = \log \sqrt{\frac{d \left(\prod_{i=1}^{d} \|\mathbf{W}_i\|_F^2 \right)^{1/d}}{m}} \tag{22}$$

$$C_{fro.dist} = \sqrt{\frac{\sum_{i=1}^{d} \|\mathbf{W}_i - \mathbf{W}_i^0\|_F^2}{m}} \tag{23}$$

$$C_{dist.spec.init} = \sqrt{\frac{\sum_{i=1}^{d} \|\mathbf{W}_i - \mathbf{W}_i^0\|_2^2}{m}} \tag{24}$$

$$C_{param.norm} = \sqrt{\frac{\sum_{i=1}^{d} \|\mathbf{W}_i\|_F^2}{m}} \tag{25}$$

C.6.5   Path Measures

Define the parameter vector as $\mathbf{w} = \text{vec}(\mathbf{W}_1, \ldots, \mathbf{W}_d)$. Below, $f_{\mathbf{w}^2}(\mathbf{1})[i]$ denotes the ith logit output of a network using squared weights where the input is a vector of ones.

$$C_{path.norm.over.margin} = \sqrt{\frac{\sum_i f_{\mathbf{w}^2}(\mathbf{1})[i]}{\gamma^2 m}} \tag{26}$$

$$C_{path.norm} = \sqrt{\frac{\sum_i f_{\mathbf{w}^2}(\mathbf{1})[i]}{m}} \tag{27}$$

C.6.6   Flatness Measures

Let ω denote the number of weights. Let $\epsilon = 1 \times 10^{-3}$. We use the search procedure for σ described in [9], where it is chosen to be the largest number such that $\mathbb{E}_{\mathbf{u} \sim \mathcal{N}(0,\sigma^2 I)}\left[\hat{L}(f_{\mathbf{w}+\mathbf{u}}) \leq 0.1\right]$. Similarly, we choose the magnitude-aware σ' to be the largest number such that $\mathbb{E}_{\mathbf{u}}\left[\hat{L}(f_{\mathbf{w}+\mathbf{u}}) \leq 0.1\right]$, where $u_i \sim \mathcal{N}(0, \sigma'^2 |w_i|^2 + \epsilon^2)$.

$$C_{pacbayes.init} = \sqrt{\frac{\frac{\|\mathbf{w}-\mathbf{w}^0\|_2^2}{4\sigma^2} + \log\left(\frac{m}{\sigma}\right) + 10}{m}} \tag{28}$$

$$C_{pacbayes.orig} = \sqrt{\frac{\frac{\|\mathbf{w}\|_2^2}{4\sigma^2} + \log\left(\frac{m}{\delta}\right) + 10}{m}} \tag{29}$$

$$C_{pacbayes.flatness} = \sqrt{\frac{1}{\sigma^2 m}} \tag{30}$$

$$C_{pacbayes.mag.init} = \sqrt{\frac{\frac{1}{4}\sum_{i=1}^{\omega} \log\left(\frac{\epsilon^2 + (\sigma'^2+1)\|\mathbf{w}-\mathbf{w}^0\|_2^2/\omega}{\epsilon^2 + \sigma'^2|w_i - w_i^0|^2}\right) + \log\left(\frac{m}{\delta}\right) + 10}{m}} \tag{31}$$

$$C_{pacbayes.mag.orig} = \sqrt{\frac{\frac{1}{4}\sum_{i=1}^{\omega} \log\left(\frac{\epsilon^2 + (\sigma'^2+1)\|\mathbf{w}\|_2^2/\omega}{\epsilon^2 + \sigma'^2|w_i - w_i^0|^2}\right) + \log\left(\frac{m}{\delta}\right) + 10}{m}} \tag{32}$$

$$C_{pacbayes.mag.flatness} = \sqrt{\frac{1}{\sigma'^2 m}} \tag{33}$$

D Coupled-network: Additional experimental results

D.1 Restricting the analysis to the SVHN dataset

In Section 6, we observed that some measures failed to be robust across changes in depth for CIFAR-10 environments (e.g., `pacbayes.mag.flatness`). It is therefore reasonable to ask what would happen if we restricted the study to SVHN. Thus, we replicate the experiments reported in Fig. 1 of the main text, but leave out all CIFAR-10 environments. The results are reported in Fig. 8.

Notice how all measures still achieve a robust sign-error of 1.0 overall, but that many measures now have a 90th percentile much closer to zero. This indicates that the error distributions of some measures would have been judged to be robust on some large subfamily of environments. Furthermore, observe how some measures now achieve perfect robustness on 'Depth' environments (e.g., `path.norm`), while none had achieved a sign-error lower than 1.0 in Fig. 1.

Our methodology allows to dig deeper into these results. For instance, we can try to understand where `pacbayes.orig` fails to be robust by looking at the CDF of sign-errors for every pair of values of depth (Fig. 9a) and width (Fig. 9b). We observe that most failures in 'Depth' environments occur when varying depth from $2 \rightarrow 3$. The sources of non-robustness in width are slightly harder to interpret, but we still observe that the measure is significantly more robust for some pairs of width than others.

Digging even deeper, we can look at the distribution of hyperparameter values in the environments where this measure fails to be robust. For 'Depth' environments, we observe that there are 10 environments with a sign-error greater than 0.01 and that these all correspond to very wide networks ($\{14, 16\}$) with small learning rates ($\{0.001, 0.0016\}$) trained on a small dataset (6250 examples). For 'Width' environments, we observe 16 environments with a sign-error greater than 0.01 and that these all correspond to shallow models of depth 2, trained with large learning rates ($\{10^{-2.5}, 10^{-2.2}, 10^{-2}\}$) on datasets of less than $50,000$ examples.

This example clearly illustrates how our proposed methodology allows to zero-in on cases where generalization measures fail to be robust. We expect that studying the shortcomings of measures at such a detailed level will aid in elaborating new, more robust, theories of generalization.
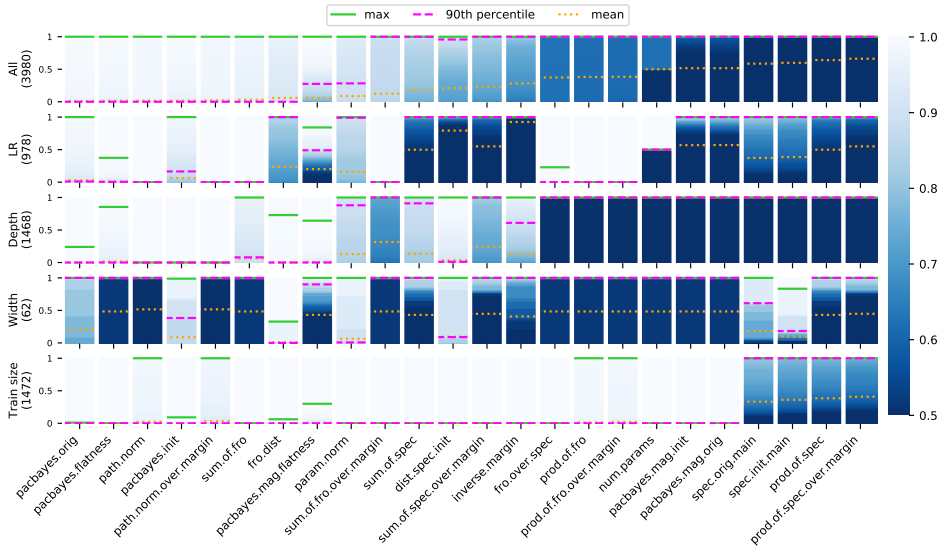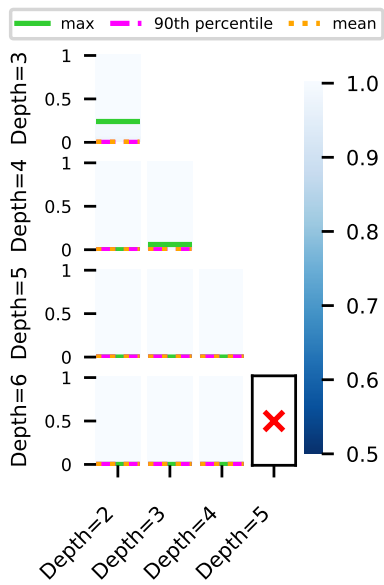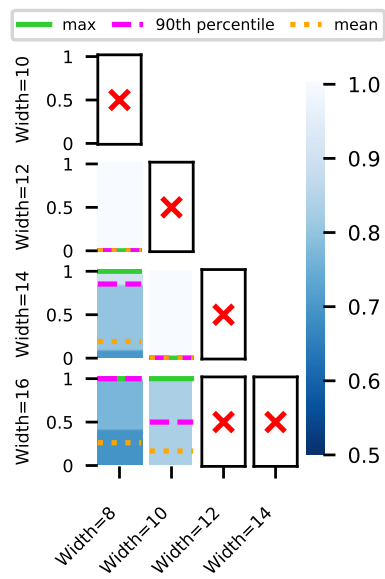

Figure 8: Cumulative distribution of the sign-error across subsets of environments for each generalization measure (in SVHN only). The measures are ordered based on the mean across 'All' environments. A completely *white* bar indicates that the measure is perfectly robust, whereas a *dark blue* bar indicates that it completely fails to be robust.

(a) Interventions on depth

(b) Interventions on width

Figure 9: CDFs of sign-errors separated by environments where depth and width vary between two values for `pacbayes.orig` (on SVHN only). The red Xs indicate that no environments remained after accounting for Monte Carlo noise.

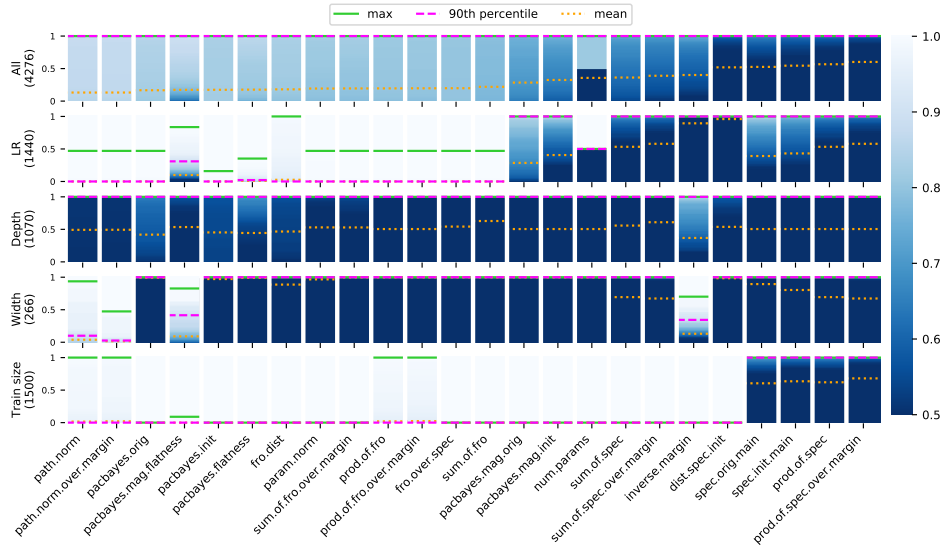## D.2 Restricting the analysis to the CIFAR-10 dataset

Figure 10: Cumulative distribution of the sign-error across subsets of environments for each generalization measure (in CIFAR-10 only). The measures are ordered based on the mean across 'All' environments. A completely *white* bar indicates that the measure is perfectly robust, whereas a *dark blue* bar indicates that it completely fails to be robust.

D.3 Exploring a weaker family of environments

We further evaluate the performance of generalization measures in a family of environments that is intermediate between the one considered in Section 5 and by Jiang et al. [9]. As described in Section 5, each environment contains pairs of hyperparameter settings where, within a pair, one hyperparameter is varied between two specific values (e.g., depth 2 → 3). However, here, the value of the other hyperparameters is allowed to change between the pairs. For example, assuming only two hyperparameters (width (w) and depth (d)), the pairs $\{[(d = 2, w = 10), (d = 3, w = 10)], [(d = 2, w = 5), (d = 3, w = 5)]\}$ could belong to the same environment (depth 2 → 3), whereas this would not be allowed in Section 5. Hence, one environment in this setting corresponds to the union of multiple environments in the setting described at Section 5. Achieving robustness in this family of environments may be significantly easier due to further averaging, which can mask failures of robustness in some hyperparameter configurations.

As illustrated in Fig. 11, this setting appears to be less challenging than the one described in Section 5. In fact, in Fig. 1, we observe that all measures achieve a robust sign-error (max value) of 1. However, in Fig. 11, we observe that some measures, such as `pacbayes.mag.flatness` and `path.norm.over.margin`, achieve a maximum sign-error much lower than 1. It is thus apparent that the further averaging that occurs in this setting leads to an easier task and prevents some failures in robustness of being exposed.

We therefore confirm that the set of environments considered in the main text is more challenging and that it constitutes a more relevant setting for studying the robustness of generalization measures.
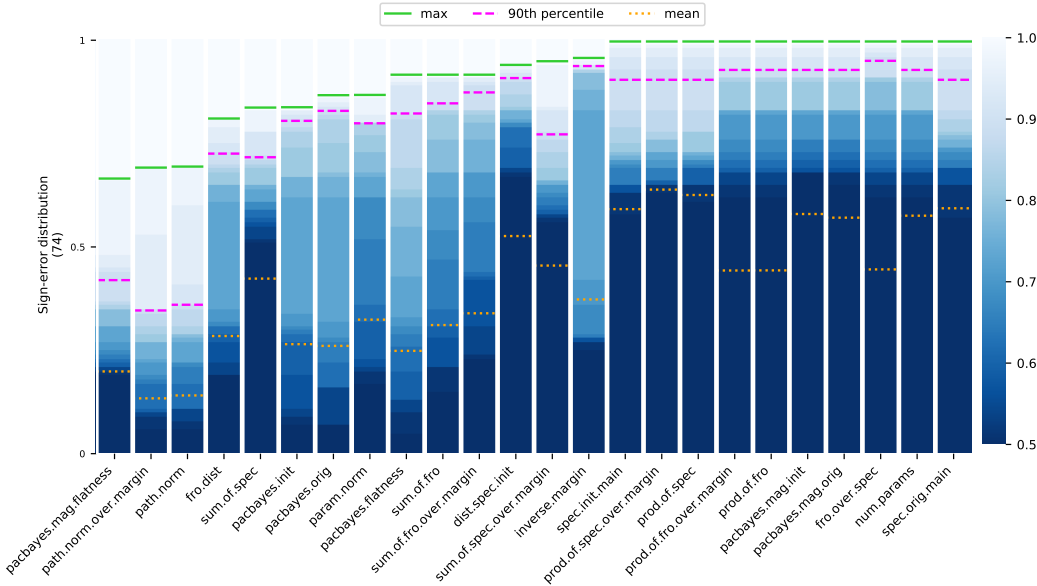

Figure 11: Cumulative distribution of the sign-error across an alternative family of environments where the values of only one pair of variables are fixed and the error is averaged over the rest of variable settings. The number of such environments is shown in parenthesis. The measures are ordered based on their robust sign-error (max) across 'All' environments. A completely *white* bar indicates that the measure is perfectly robust, whereas a *dark blue* bar indicates that it completely fails to be robust.

E Methodological comparison to the conditional-independence testing method of Jiang et al. [9]

In addition to their Kendall-τ-based Ψ measure, Jiang et al. [9] propose a measure based on conditional-independence testing (Section 2.2.3 in [9]). This measure attempts to identify the existence of a causal relationship (edge in a postulated causal graph) between a generalization measure and the generalization gap. While the concept of robustness (that our work builds on) and conditional-independence testing can both be tied to the causal inference literature, our methods are fundamentally different. Notably, we do not claim, nor seek, to identify causal relationships (see the discussion in Section 2). Below, we highlight some key differences between our approaches.

It can be tempting to see a similarity in the fact that both methods look at extreme values (min and max). Our method looks at the maximum sign-error over all environments, which is analogous to taking the max over every possible intervention on each hyperparameter. The method of Jiang et al. [9] considers the minimum normalized conditional mutual information ($\hat{\mathcal{I}}$; Eq. (13) of [9]) over all conditioning sets of two hyperparameters (Eq. (14) of [9]). However, as described in their Eq. (11) and (12), the calculation of $\hat{\mathcal{I}}$ involves *averaging over all values* of the hyperparameters in the conditioning set (i.e., $\sum_{U_\mathcal{S}} p(U_\mathcal{S})$). Therefore, while they may appear similar, both approaches are fundamentally different in that one averages over multiple values of the same hyperparameter, while the other (ours) does not.

Furthermore, in the limit where we observe every possible intervention on the HPs, our method identifies measures that may have a causal relationship to generalization (all causal explanations are necessarily robust in this extreme case). However, this is not necessarily true for the IC-based method of Jiang et al. [9]. The reason is that their conditioning sets are of size 2, which may leave open confounding paths in the graph if more than 2 hyperparameters act as confounders, resulting in non-causal mutual information. This means that the method is not guaranteed to detect a causal edge from the generalization measure to generalization gap unless they condition on all hyperparameters. However, if they were to condition on all hyperparameters, their conditional mutual information would collapse to zero, suggesting that there is no causal edge.