Title
Beyond Worst-Case Generalization in Modern Machine Learning

Author
Theisen, Ryan Christopher

Publication Date
2023

Peer reviewed|Thesis/dissertation

Beyond Worst-Case Generalization in Modern Machine Learning

by

Ryan Christopher Theisen

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael W. Mahoney, Chair
Professor Aditya Guntuboyina
Professor Song Mei

Summer 2023

Beyond Worst-Case Generalization in Modern Machine Learning

Abstract

Beyond Worst-Case Generalization in Modern Machine Learning

by

Ryan Christopher Theisen

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Michael W. Mahoney, Chair

This thesis is concerned with the topic of *generalization* in large, over-parameterized machine learning systems—that is, how models with many more parameters than the number of examples they are trained on can perform well on new, unseen data. Such systems have become ubiquitous in many modern technologies, achieving unprecedented success across a wide range of domains. Yet, a comprehensive understanding of exactly why modern machine learning models work so well has alluded fundamental understanding. Classical approaches to this question have focused on characterizing how models will perform in the worst-case. However, recent findings have strongly suggested that such an approach is unlikely to yield the desired insight. This thesis is concerned with furthering our understanding of this phenomenon, with an eye towards moving beyond the worst-case. The contents of this thesis are divided into six chapters.

In Chapter 1, we introduce the problem of generalization in machine learning, and briefly overview some of the approaches—both recent and classical—that have been taken to understand it.

Chapter 2 introduces a novel analyses of deep neural networks with positive homogeneous activation functions. We develop a method to provably sparsify and quantize the parameters of a model by sampling paths through the network. This directly leads to a covering of this class of neural networks, whose size grows with a measure of complexity we call the "path norm". Using standard techniques, we can derive new worst-case generalization bounds that improve on previous results appearing in the literature.

In Chapter 3, we take a critical look at the worst-case approach to understanding generalization. To do this, we develop a methodology to compute the full distribution of test errors for interpolating linear classifiers on real-world datasets, and compare this distribution to the performance of the worst-case classifier on the same tasks. We consistently find that, while truly poor, worst-case classifiers indeed exist for these tasks, they are exceedingly rare—so

much so that we expect to essentially never encounter them in practice. Moreover, we observe that as models become larger, test errors undergo a concentration around a critical threshold, with almost all classifiers achieving nearly the same error rate. These results suggest that the worst-case approach to generalization is unlikely to describe practical performance of large, over-parameterized models, and that new approaches are needed.

In light of our findings in Chapter 3, in Chapter 4 we ask a complementary question: if modern machine learning systems perform nowhere near the worst-case, how close might their performance be to the *best-case*? For an arbitrary classification task, we can quantify the best possible error attainable by any model with the Bayes error rate, though it is generally intractable to estimate for realistic datasets. To address this intractibility, we first prove that the Bayes error rate is invariant under invertible transformation of the input features. We then use normalizing flows to estimate an invertible map between the target data distribution and a simple base distribution, for which the Bayes error can be easily computed. We then evaluate a variety of state-of-the-art architectures against this estimated Bayes error rate, and find that in some (but not all) cases, these models achieve very close to optimal error rates.

In Chapter 5, we investigate average-case generalization via ensembling—a popular method wherein the predictions of multiple models are aggregated into a single, often more powerful predictor. In classical settings, the effectiveness of ensembling is well-understood; however, for ensembles comprised of deep neural networks, the benefits of this method are surprisingly inconsistent. To understand why, we first prove a new set of results relating the *ensemble improvement rate* (a measure of how much ensembling decreases error relative to the average error rate of a single model) to the ratio of model disagreement to the average error rate. This results in new oracle bounds on the error rate of ensemble classifiers, significantly improving on prior results in the literature. We then investigate ensembling experimentally and find, most notably, a distinct and consistent transition in the rate of ensemble improvement (and the disagreement-error ratio) occuring at the interpolation threshold—the point at which individual classifiers achieve exactly zero training error. Our findings suggest that ensembling is significantly less effective in the modern, over-parameterized regime than it is in more classical settings.

Finally, in Chapter 6 we conclude with some reflections on the state of the field, and outlook for how it might advance in the coming years.

Contents

Acknowledgments

This thesis and my doctorate would not have been possible without many incredible mentors, collaborators, and friends.

First and foremost, I would like to thank my advisor Michael Mahoney. From our first meeting, Michael has opened my eyes to many new perspectives, and taught me to think about problems creatively and independently, without obsessing over the ideas that are in fashion at the present moment.

Throughout my Ph.D., I have had the privilege of working with many incredible collaborators. Jason Klusowski has been an irreplaceable mentor to me, especially in my first years as a graduate student. From the minute he replied to my first (completely cold) email, he was extremely patient and open-minded with my ideas, and made even the smallest progress feel rewarding. I was also fortunate to spend two amazing summers at Salesforce research, working with a number of incredible scientists. Specifically, I want to thank Huan Wang, Nitish Keskar, and Lav Varshney for inspiring and working with me on two separate projects. They made working in a new environment easy, and I will always be grateful for their support and friendship. Finally, I want to thank my collaborators at Berkeley, namely Yaoqing Yang, Liam Hodgkinson, and Hyunsuk Kim. Our regular meetings have been a great joy, and kept spontaneous discussion in my life, even when we could not all be together in person.

In addition to the mentors and collaborators I've worked with academically, this thesis would never have been possible without the close kinship of many friends outside of my work. At Berkeley, Taejoo and Dan have always kept me entertained and well-fed. Though impossible to name them all, my friends outside of academia have helped keep me sane through many difficult times, and allowed me to recharge when my mind desperately needed it. Though I couldn't name them all here, I have to thank Tommy, Dayton, Alex (×2), Tyler, Will, Riley, Austin (×2), Tierney, Sanj, Vig, and Vin for always being there for me.

Finally, nothing I have accomplished would ever have been possible without the endless love and support of my family.

Chapter 1

Introduction

1.1 The generalization problem in modern machine learning

A hallmark of modern (supervised) machine learning is the training of large, *over-parameterized* models, that is, models with many more parameters than the number of examples on which they are fit. These methods have yielded tremendous successes across a range of applications; for example, computer vision [KSH12, HZRS16], natural language processing [VSP+17, DCLT19], biology [BB21, JEP+21], and many others.

In Table 1.1 we give a sampling of a variety of benchmark tasks, the corresponding state-of-the-art models and their (approximate) number of parameters, and the number of labeled samples available in the training dataset. Across tasks and application areas, it is consistently observed that large, over-parameterized models perform extremely well. This phenomenon seems to defy classical reasoning: given sufficiently many parameters, such models should be capable of easily over-fitting the training data, and performing poorly on new, unseen data.

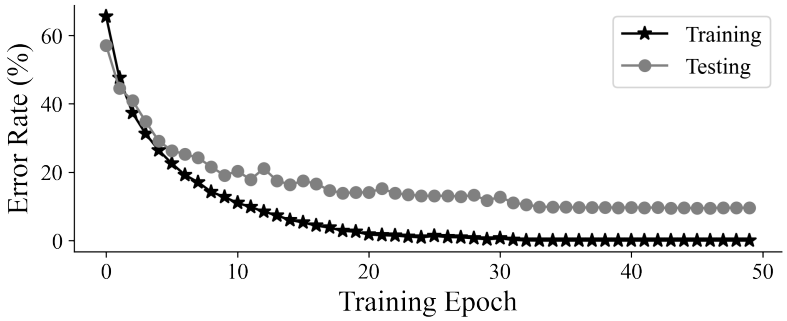


Figure 1.1: Training and testing error curves for a ResNet18 model trained on the CIFAR-10 dataset using stochastic gradient descent with *no* explicit regularization employed.

Dataset	Task	Model	# Parameters	# Examples
ImageNet	Image Classification	BASIC-L [CLH+23]	2.44B	1.3M
WMT2014 English-German	Machine Translation	T5-11B [RSR+20]	11B	4.5M
SST-2	Sentiment Analysis	T5-11B [RSR+20]	11B	12K
QM9	Drug Discovery	MXMNet [ZLX20]	3.8M	110K

Table 1.1: Current state-of-the-art (or near-state-of-the-art) models across a sampling of domains, their number of parameters, and the number of labeled training examples available for each task. Data retrieved from [Pap].

A natural refrain would be that this phenomenon is due to explicit regularization enforced during model fitting. However, this notion is quickly dispelled with a simple experiment. We run the following: we train a popular residual neural network architecture, ResNet18 [HZRS16], on a standard 10-class image classification benchmark task called CIFAR-10, [KH+09]. The model itself has approximately 17 million total parameters, and the training dataset contains 50,000 labeled examples. We train the model using standard batched stochastic gradient descent, with *no explicit regularization* employed. In Figure 1.1, we plot the learning curves (training and testing errors during training) for this model. We observe that even when the training error rate reaches *exactly zero*, the testing error remains stable, and even continues to decrease. Why do we not observe over-fitting, even with a highly over-parameterized model, and no explicit regularization?

In what follows, we provide a brief overview of some of the approaches to understanding the problem of generalization in large-scale machine learning, both mathematically and scientifically. Though many of the concepts we discuss here also apply to models besides neural networks, these will serve as our primary motivation.

1.2 The classical approach: worst-case analysis of learning

We start with a review of the classical approach to understanding generalization – the *worst-case* or *uniform* approach.

To formalize this approach we require the following: a hypothesis space of possible models $\mathcal{F}$ (e.g. a set of parametric neural networks of a prescribed architecture), a loss function $\ell(f, \boldsymbol{x}, y)$ operating on a function $f \in \mathcal{F}$ and an input-output pair $\boldsymbol{x}, y$, a training dataset $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ of size n, and a testing distribution $\mathcal{D}$ over pairs $(\boldsymbol{x}, y)$. Given these, the ***training error*** is defined as

$$\hat{L}_S(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f, \boldsymbol{x}_i, y_i)$$

and the ***testing error*** is

$$L_{\mathcal{D}}(f) = \mathbb{E}_{\boldsymbol{x}, y \sim \mathcal{D}}[\ell(f, \boldsymbol{x}, y)].$$

The ***generalization gap*** is simply the difference between performance on the training and testing sets:

$$\Delta(f) = L_{\mathcal{D}}(f) - \hat{L}_S(f).$$

We remark that throughout this thesis, we will describe results related to generalization as any bound or other characterization of either the generalization gap or the testing error itself.

The uniform (or worst-case) generalization gap is typically defined as largest possible generalization gap among models $f \in \mathcal{F}$:

$$\Delta_{\text{unif}} = \sup_{f \in \mathcal{F}} \Delta(f).$$

Over the years, a few powerful and now standard techniques have emerged to provide bounds on the uniform error Δ_{unif}. Typically, bounds are derived under the "probably approximately correct" (PAC) framework. The statement of a PAC generalization bound will most commonly take the following form: with high probability over random draws of the training data S,

$$\Delta_{\text{unif}} \leq \widetilde{O}(\sqrt{M(\mathcal{F})/n}),$$

where $M(\mathcal{F})$ is some complexity term, and the notation $\widetilde{O}(\cdot)$ hides various constant terms and log factors. Here the complexity term $M(\mathcal{F})$ could one of a number of measures, for example the The Vapnik–Chervonenkis (VC) dimension, the Rademacher complexity, or a log covering number (also called the metric entropy). The general recipe for obtaining a worst-case generalization bound is to 1) define a suitable function class $\mathcal{F}$, and 2) estimate (upper bound) $M(\mathcal{F})$, for some suitable complexity measure M. An alternative (though closely related) set of techniques are used to prove PAC-Bayes bounds, which instead relies on a prior Q over models, though the functional form of the bounds remains largely the same. A great number of bounds proved using these techniques are available in the literature. We provide a brief overview of some of these in what follows, though we note that this set of references is in no way exhaustive.

Intuition from simple models suggests that the VC dimension of a parametric model class should be closely related to the number of parameters of a given model in that class; this is formalized in [BHLM19], which shows that the VC dimension of classes of neural networks with piece-wise linear activation functions is essentially proportional to the number of parameters in the model. Thus the uniform bound based on the VC dimension scales (roughly) as $\sqrt{d/n}$, where d is the number of parameters and n is again the number of training examples. In light of the results in Table 1.1, most approaches avoid the use of the VC dimension to describe modern neural networks.

One approach that has been quite popular is to obtain norm-based generalization bounds, that is, bounds in terms of the norms of the parameter matrices in a neural network. The motivation for norm-based bounds is dates back to classical analyses of linear models, wherein

the norm of the parameters is often used as explicit regularization. [BFT17] use covering numbers to obtain a generalization bound in terms of the product of the norms of the weight matrices, $M(\mathcal{F}) \propto \prod_{l=1}^{L} \|W_l\|$, of an L-layer neural network. Similar bounds were obtained via a PAC-Bayes approach in [NBS18]. The approach of [GRS18] used a "peeling" technique to bound the Rademacher complexity, and ultimately attain similar-looking bounds. Other examples in the literature include [NTS15, NSS15], which obtain bounds in terms of a "path norm" of a neural network. In Chapter 2, we show how a new sparsification technique can be used to prove bounds that strictly improve on these.

Other examples of bounds include approaches based on compression [AGNZ18, LFK+22], control of the Lipschitz constants or Jacobians of models [WM19], "sharpness" and/or sensitivity [NBMS17], and many others. For a more exhaustive taxonomy of results, many modern reviews are available, for example [VPL20].

There's one very significant problem with the worst-case/uniform approach: it largely doesn't work. While the bounds themselves are of course mathematically correct, they simply do not describe practical performance of large, modern models. One basic issue with uniform bounds is that they are typically vacuous[1] when computed numerically (meaning that if the loss ℓ is, say, upper bounded by B, then the bounds are $> B$). However, one could argue that vacuousness is not fundamental to the uniform approach, and that these issues could be alleviated if only one were able to obtain better bounds, or somehow define better function classes $\mathcal{F}$ over which to compute a uniform bound. However, this argument has also been challenged in a number of works. Indeed, it is provably the case that uniform generalization bounds will fail in many reasonable cases [ZBH+17, NK19, GWWM23]. Notably, [NK19] shows that the general rate $1/\sqrt{n}$ appearing in many bounds is numerically wrong for many practical examples, and, worse still, that many of the best bounds from the literature can actually *grow* with the number of training examples. As we will argue in Chapter 3, these problems are in many ways inherent to over-parameterization. By computing the full distribution of test errors for interpolating classifiers, we show that worst-case models are extremely rare, and do not reflect average case performance.

1.3 Generalization beyond the worst-case

In light of the known issues with the worst-case approach, many studies have endeavored to find alternative ways to describe generalization for over-parameterized models. In what follows, we provide a rough taxonomy of these approaches, organized into three categories. We remark that this topic has been the subject of a tremendous amount of research over the last decade, and the related literature is vast; consequently, the following discussion is inevitably incomplete.

[1] Though this is not necessarily always the case, as shown in [DR17, ZVA+19].

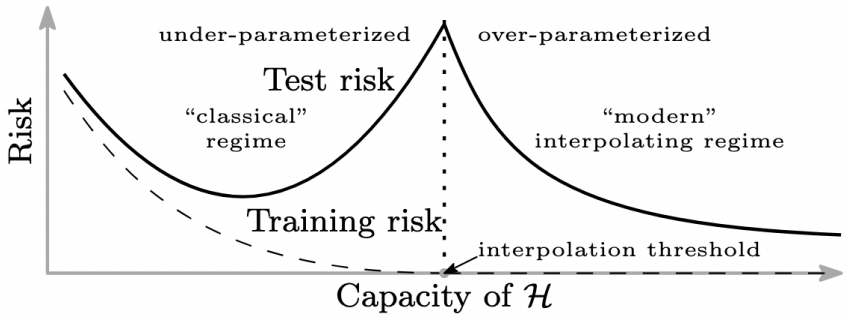


Figure 1.2: Illustration of the *double descent* phenomenon. Figure taken from [BHMM19b].

The exact approach

We refer to the exact approach as any means of studying generalization via precise analysis, usually of simple (or even toy) models, with the hope that conclusions drawn for these settings may lend insight into the performance of more complex models encountered in practice.

One of the most important results in recent years has been the identification of the so-called "double descent" phenomenon, wherein, as a function of model capacity, the testing error exhibits first a standard "U-shape", governed by the classical bias-variance trade-off, followed by a surprising *decrease* in the test error beyond the ***interpolation threshold*** (i.e. the capacity at which the model can achieve exactly zero training error). The term double descent was first coined, and popularized in the machine learning community, in a sequence of works [BHMM19b, BHX19] exhibiting the phenomenon across a wide variety of settings (though the underlying ideas can be traced back significantly farther, e.g. [VCR89, Dui00, SGd+19]). In a long line of subsequent works, the double descent phenomenon has been proven to exist for a number of different models, ranging from linear and ridge regression [BLLT20, DLM19, HMRT19, TB23], random feature models [LCM20, LSC22], logistic regression [DKT20, DKT19, CL20], nearest neighbors [XSC19], and many others. In Chapter 5, we observe double descent for ensembles of deep networks. The general message of these works is that over-parameterization need not always lead to over-fitting, i.e. a large generalization gap. In particular, it is possible (perhaps even common) to perfectly fit the training data, add no or little explicit regularization, and still achieve low (even minimal) testing error. Importantly, these results tend to rely on very precise analyses; worst-case upper bounds generally do not suffice to capture behavior like double descent.

Another important family of results are related to the derivation of the so-called neural tangent kernel (NTK) [JGH18]. The NTK is a deterministic kernel arising in the limit of infinite width (but fixed, finite depth and number of training samples) for neural networks. Specifically, it has been proven under very general conditions that as the number of the hidden units in a neural network grows to infinity, when trained using e.g. stochastic gradient descent with sufficiently small step size, the trained network is a linear model $\boldsymbol{x} \mapsto \langle \boldsymbol{w}, \phi_0(\boldsymbol{x}) \rangle$ for a

feature map $\phi_0(\boldsymbol{x})$ associated with the kernel function $k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\theta_0}[\langle \nabla_\theta f_{\theta_0}(\boldsymbol{x}), \nabla_\theta f_{\theta_0}(\boldsymbol{x}') \rangle]$, where $f_\theta(\boldsymbol{x})$ is a parameterized neural network, and θ_0 are parameters drawn at random at initialization (e.g. from a properly scaled Gaussian) [ADH+19, AZLS19]. The limit obtained here has a close analogue in the Bayesian setting, wherein the limit results in a neural network Gaussian process [LSdP+18]. The analysis in this limit reduces to that of a significantly simpler linear model, facilitating more precise results, which one might hope would result in a correspondingly precise theory for over-parameterized neural networks of large but finite width. However, empirical investigations have suggested that the linear models obtained in the NTK limit differ non-trivially from practically large models used in practice [FDP+20]. Recent work has shown exciting progress towards avoiding these issues by considering the simultaneous limit of infinite width *and* depth [LNR22].

Finally, we mention approaches utilizing techniques from statistical physics. These approaches have significantly influenced many of the ideas present in this thesis (in particular, the contents of Chapter 3). Many physical systems (e.g. large systems of interacting particles), share conceptual similarities with large machine learning systems, wherein an analogy is drawn between randomly distributed particles and the parameters of a statistical model fit to randomly sampled data; this connection has been formalized since at least the early 90s [SST92, WRB93, HKSST96, EVdB01]. Rather than studying the worst-case estimator $f \in \mathcal{F}$, the statistical mechanics approach seeks to understand the behavior of the *typical* function f. This typicality can be characterized in a number of ways. A natural measure, from the statistical physics perspective, would be the free energy (or entropy), from which large-scale behavior can be deduced. Analyses following the statistical mechanics approach usually obtain in a "thermodynamic" limit, wherein two or more parameters are sent to infinity together. For example, most commonly, this means letting some measure of the number of parameters d and the number of examples n go to infinity together such that the ratio $n/d \to \alpha > 0$. Note that this limit is in contrast to the limit used in the analysis of the NTK, which considers n fixed, and allows the width d to grow to infinity. It also contrasts with the usual limit considered relevant for uniform convergence, wherein d is presumed fixed, and we are interested in the behavior when n is large. There are many examples of the statistical mechanics style of analysis; some quite old (e.g. [HKSST96, OH91, EVdB01]), and some much more modern. For example, this approach has recently been used to demonstrate the existence of phase transitions in learning behavior in logistic regression [CS18] and generalized linear models [BKM+19].

Average-case bounds

While the exact approach has provided many important insight into how large, over-parameterized models generalize, there is a still a desire to have theory that applies to real-world systems. Towards this end, a number of efforts have been made to obtain bounds on the *average-case* generalization error of realistic systems. The first challenge in this approach is defining a distribution under which the "average" error can be estimated. There

are multiple ways this can be done, though typically approaches have considered two distinct sources of randomness: randomness in the drawing of the training data, and randomness in the learning algorithm used to fit a model. The latter randomness is much harder to characterize in general, as it requires a study of the dynamics during training using, most commonly, variants of stochastic gradient descent.

One approach that has proved successful in providing very generic bounds on the generalization error is the *information theoretic* approach. Recently, a large number of works have obtained average-case generalization bounds under very general conditions, for a variety of stochastic learning algorithms. Perhaps the simplest of these, proved in [XR17], holds for a σ-sub-Gaussian loss function ℓ, in which case it can be proven that the average generalization gap $\mathbb{E}_{\theta,S}[\Delta(f_\theta)]$ is upper bounded by $\sqrt{2\sigma^2 I(S;\theta)/n}$, where $I(S;\theta)$ is the mutual information between the training data S and trained parameters θ. Subsequently, a number of results have improved upon this initial bound, for example by refining the analysis via the conditional mutual information, or the chaining of mutual information [HNK+20, SZ20, HDMR21, HRSG21, ZTL22]. One shortcoming of this approach is that the information theoretic quantities typically appearing in the bounds, such as $I(S;\theta)$, are generally intractable to compute, and hence it's unclear the extent to which they describe behavior encountered in practice.

The scientific approach

In a parallel line of work to much of the theory that has been developed around generalization, significant progress has been made via a "scientific" approach to understanding generalization, i.e. through careful experimentation testing specific hypotheses. Like in many sciences, many works in this category conduct empirical studies testing hypotheses arising out of rigorous analyses in simpler settings. For example, [NKB+20] investigates the presence of the double descent phenomenon in large, practical models used in practice, finding that, surprisingly, many of the findings derived analytically for simpler models hold much more generally in practice. Specifically, they report the existence of non-trivial phase transitions occurring at the interpolation threshold. This finding has since been corroborated in more recent studies, e.g. [YHT+21]. In Chapter 5, we find similar behavior arises when measuring the rate of ensemble improvement for ensembles of interpolating models. Another interesting example of this approach is in [FDP+20], which rigorously investigates the validity of the neural tangent kernel analysis in practical settings. They find that in practice, the linearization obtained in the infinite-width limit differs non-trivially from what is found in practice, even for very wide models, suggesting that this limit is not appropriate for studying realistic models.

A separate line of work has recently endeavored on large-scale empirical studies in search of generalization measures (i.e. computable metrics derived from theory or practice), that exhibit a robust ability to predict generalization performance in realistic settings [SQDC21, JNM+20] (this goal inspired a popular competition at NeurIPS 2020 [JFY+20]). Other work has gone even further, attempting to design studies capable of deducing *causal* relationships between

various metrics and generalization [DDN+20]. Towards the goal of finding robust and effective generalization metrics, an interesting line of work has observed a strong relationship between the spectrum of the weight matrices in trained neural networks and their generalization performance [MM17, MM18, MM20, MM22, YTH+23]. In particular, this work suggests that models with good generalization performance exhibit heavy tails in the spectra of their weight matrices. Interestingly, [YTH+23] suggests that studying the generalization *gap* may be misleading; in practice, most users are interesting in predicting the testing error itself, rather than the gap between testing and training errors. They show that many metrics exhibiting strong (rank) correlation with the generalization gap have surprisingly poor correlation with the test error itself.

As previously discussed, it has been observed empirically that the "standard" $1/\sqrt{n}$ rate arising out of uniform convergence analyses is often incorrect in practical settings. Following the scientific approach, exciting recent work has experimentally derived effective "scaling laws" that empirically govern model performance as model size and number of training examples grows [KMH+20, HBM+22]. In general, the scientific approach has seen great success in providing a practically useful understanding of how and when large machine learning systems can be expected to work. We suspect that while purely theoretical efforts to understand generalization will surely continue to progress, the field will rely more heavily on carefully designed, large-scale experimental studies to help guide it forward—much like any other science.

1.4 This thesis

This thesis will touch on aspects of many different approaches to understanding generalization, using both theoretical and empirical studies to derive new insights.

Organization and credits

The contents of this thesis are adapted from four separate works, of which I was the lead author. It is organized as follows.

Chapter 2 is based on develops a method for sparsifying and quantizing the parameters of a neural network with positive homogeneous activation function by sampling *paths* through the network. By approximating a model in such a way, we are able to obtain a covering of the space of such neural networks, which we use to derive novel worst-case generalization bounds in terms of a family of path norms, improving on similar bounds appearing previously in the literature. This chapter showcases a template of how uniform generalization bounds can be derived.

In Chapter 3, we take a critical look at the worst-case approach to understanding generalization. To do this, we develop a methodology to compute the full distribution of test errors for interpolating linear classifiers on real-world datasets, and compare this distribution to the

performance of the worst-case classifier on the same tasks. We consistently find that, while truly poor, worst-case classifiers indeed exist for these tasks, they are exceedingly rare—so much so that we expect to essentially never encounter them in practice. Moreover, we observe that as models become larger, test errors undergo a concentration around a critical threshold, with almost all classifiers achieving nearly the same error rate. These results suggest that the worst-case approach is unlikely to describe practical performance of large, over-parameterized models, and supports the case for new approaches to understanding generalization.

Chapter 4 addresses a complementary question: if modern models don't perform nearly as bad as the worst-case, how close might they be to achieving the *best possible* error rate? For classification tasks, the lowest attainable error rate can be quantified with the Bayes error rate—a quantity which is generally intractable to estimate for realistic datasets. To address this, we first prove that the Bayes error rate is invariant under invertible transformation of the input features. We then use normalizing flow techniques—a class of generative models which explicitly learn an invertible mapping—to estimate a map between the target data distribution a simple base distribution (e.g. a multivariate Gaussian), for which the Bayes error can be easily computed. We then evaluate a variety of state-of-the-art architectures against this estimated Bayes error rate, and find that in some (but not all) cases, these models achieve very close to optimal error rates.

In Chapter 5, we study the average-case behavior of classification models f as it relates to the practice of ensembling—a popular method wherein the predictions of multiple models are aggregated into a single, often more powerful predictor. In classical settings, the effectiveness of ensembling is well-understood; however, for ensembles comprised of deep neural networks, the benefits of this method are surprisingly inconsistent. To understand why, we first prove a new set of results relating the *ensemble improvement rate* (a measure of how much ensembling decreases error relative to the average error rate of a single model) to the ratio of model disagreement to the average error rate. This results in new oracle bounds on the error rate of ensemble classifiers, significantly improving on prior results in the literature. We then investigate ensembling experimentally and find, most notably, a distinct and consistent transition in the rate of ensemble improvement (and the disagreement-error ratio) occurring at the interpolation threshold. Our findings suggest that ensembling is significantly less effective in the modern, over-parameterized regime than it is in more classical settings.

Other works not included

In addition to the works included fully here, I was also fortunate during my time as a graduate student to be a co-author on other works closely related to the content presented herein. There are two of particular relevance: the first is [YHT+21], appearing in NeurIPS 2021, in which we conduct an extensive study on modern neural network architectures and training methods, and connect the generalization of these models to a large number of metrics related to the (training) loss landscape. The second is [YTH+23], to appear in KDD 2023, in which we evaluate the various theoretically-proposed generalization metrics across modern natural

language processing (NLP) models and tasks. NLP provides an interesting deviation from other popular benchmark areas for machine learning, as the discrete nature of the input domain often precludes the possibility of interpolation. We show that, correspondingly, many standard quantitative measures of generalization fail to "generalize" to this domain, and that superior metrics measuring the spectral properties of the weight matrices perform much better.

Chapter 2

Worst-Case Generalization for ReLU Networks via Path Sampling

The contents of this chapter are partially based on the technical report "Global Capacity Measures for Deep ReLU Networks via Path Sampling", co-authored with Jason Klusowski, Huan Wang, Nitish Sirish Keskar, Caiming Xiong and Richard Socher [TKW+19].

2.1 Introduction

For classes of linear models, including deep linear networks, it is well known that the norm of the weights $\|w\|$ is an important capacity measure that governs much of their statistical behavior. As a consequence, many algorithms have been developed for these problems that explicitly regularize on such norms. For more complex function classes, such as deep neural networks, various generalizations of this capacity measure have been proposed. Such analyses have commonly identified the product of norms $\prod \|W_\ell\|$ as a complexity measure, e.g. [BFT17, NBS18, GRS18]. In this chapter, we show that for a large class of deep networks possessing a positive homogeneity property, including ReLU networks and convolution networks with max or average pooling, we can obtain bounds that are instead in terms of a norm of a product $\|\prod |W_\ell|\|$, which we show often lower bounds the former product of norms. These quantities arise out of a path-based analysis of positive homogeneous networks, and are in fact closely related to the path norms appearing in previous work, such as [NSS15, NTS15, KKB19].

Our method for proving such bounds generalizes a sampling technique recently proposed in [BK18], wherein a positive homogeneous network is parameterized explicitly in terms of a path distribution, which is subsequently sampled from in order to produce a sparse

approximant of the original network. This technique allows us to prove that any given positive homogeneous network f admits a sparse approximant $\tilde{f}$ which belongs to a small representer set of functions, whose cardinality we show to be governed by various norms. These results immediately imply covering number bounds, which can be used to control various statistical performance measures, including generalization error.

Our sampling approach is an example of the *probabilistic method*, which, interestingly, appears frequently in previous work on generalization. For example, [BFT17] prove covering number bounds via the use of Maurey sparsification, wherein each layer of a network is sparsified individually using a probabilistic sampling argument. [AGNZ18] introduce a compression-based approach to generalization, and prove a bound by compressing each layer of a deep network via a random projection. [BLG+19] recently introduced an edge-wise sampling procedure for compressing networks, which likewise can be used to obtain generalization bounds via the compression approach of [AGNZ18]. Notably, however, in almost all existing works, compression and/or sparsification of deep networks is conducted layer-wise. In the context of norm-based bounds, operating in such a manner generally leads to bounds in terms of the product of norms of the weights, as is the case in [BFT17, NBS18].

In contrast, our path-based approach allows us to sample from a distribution over all parameters at once, without the need to work in a layer-wise fashion. More specifically, we define a distribution over paths through the network, obtained by normalizing with various quantities of the form $\| \prod_1^L |W_\ell| \|$. These quantities subsequently appear in our error bounds. In contrast to the product of weight norms, the norm of the product captures a notion of *global* variation in networks. Indeed, the product of weight norms measures the size of weights *within* layers, but fails to capture the strength of connections and interactions *across* successive layers. On the other hand, the norm of the product of weights incorporates both aspects.

We remark that a path-based approach to studying neural networks has appeared in several other works, for example in the design of optimization algorithms [NSS15] for neural networks, as well the study of their loss surfaces [CHM+15a]. The path norms studied in [NTS15] are closely related to the quantities arising in our analysis, and in fact, as we discuss in Section 2.4, our results can be seen as improvements of the bounds given therein, by avoiding exponential dependence on network depth. Other path-based capacity measures have been considered as well, notably by [KKB19], though the resulting bounds depend critically on the (unknown) data distribution. Recent work has also proposed the Fisher-Rao norm as a global, norm-based capacity metric for deep networks, though this has only been shown to control generalization error for the case of linear networks [LPRS19].

Organization and contributions

The chapter is organized as follows.

- In Section 2.2, we outline our general setting and notation, and review a sampling

technique proposed by [BK18], which they use to prove approximation and covering bounds for single output, fully-connected networks in terms of ℓ_1-type norms.

- In Section 2.3, we extend the sampling scheme and analysis to the multi-class and convolutional setting, and show that it can be generalized to obtain bounds in terms of a much broader class of norms. We further show with a lower bound that our analysis of the sampling scheme is nearly optimal.
- In Section 2.4, we exploit certain permutation invariances in deep networks to bound the number of networks that are realized by the sampling method. This results in covering number and metric entropy bounds. We provide as a consequence of these results a new margin-based generalization bound for multi-class classification. We compare this bound to existing results in the literature, and find that our bound is comparable to, and often improves upon, existing norm-based bounds.
- In Section 2.5, we investigate empirically the sampling strategy studied theoretically above, and find that compressibility of networks correlates well with generalization performance. An analysis of certain normalized margin distributions suggests that the quantities appearing in our bounds do indeed capture this behavior.
- Finally, in Section 2.6, we suggest directions for future research; namely, we outline one potential approach to extending our technique and results to the analysis of residual networks.

2.2 Setup and Background

In this work, we consider a standard setting of multi-class classification, wherein a network $f(x;W) : \mathcal{X} \subseteq \mathbb{R}^d \mapsto \mathbb{R}^k$ makes a classification decision $\hat{y} = \arg\max_j f(x;W)_j$. We use S to denote a training set $\{x^1, ..., x^n\}$ of n points, each of which has a corresponding label $\{y^1, ..., y^n\}$, with $y^i \in \{1, ..., k\}$. For a vector $v \in \mathbb{R}^k$ and $y \in \{1, ..., k\}$, we define the margin operator to be $\mathcal{M}(v, y) = v_y - \max_{j \neq y} v_j$. We denote the classification loss

$$\ell(f(x;W), y) = \mathbb{1}\big(\mathcal{M}(f(x;W), y) \leq 0\big) \tag{2.1}$$

and, for $\gamma > 0$, the γ-margin loss

$$\ell_\gamma(f(x;W), y) = \mathbb{1}\big(\mathcal{M}(f(x;W), y) \leq \gamma\big). \tag{2.2}$$

For the empirical loss, we denote $\hat{\ell}(f) = \frac{1}{n}\sum_{(x,y):x\in S} \ell(f(x;W), y)$ and for the population loss we use $\ell(f) = \mathbb{E}_{(x,y)}[\ell(f(x;W), y)]$, and likewise for $\hat{\ell}_\gamma(f)$ and $\ell_\gamma(f)$. We will also use the ramp function which, for any $\gamma > 0$, is given by

$$R_\gamma(z) = \begin{cases} 0 & z < -\gamma, \\ 1 + z/\gamma & z \in [-\gamma, 0] \\ 1 & z > 0 \end{cases}. \tag{2.3}$$

R_γ importantly satisfies the following:

$$\ell(f) \le \mathbb{E}_{(x,y)}[R_\gamma(-\mathcal{M}(f(x;W),y)] \le \ell_\gamma(f) \tag{2.4}$$

with analogous inequalities holding for $\hat{\ell}(f), \hat{\ell}_\gamma(f)$ with the empirical distribution on S.

We will use the notation $\mathbb{B}_q(r) = \{x \in \mathbb{R}^d : \|x\|_q \le r\}$ to denote the ℓ_q balls in $\mathbb{R}^d$. For an $m \times n$ matrix A, we define the matrix q-norm induced by the ℓ_q norm by

$$\|A\|_q = \sup_{z \neq 0} \frac{\|Az\|_q}{\|z\|_q}. \tag{2.5}$$

We will be particularly interested in $\|\cdot\|_2$, which is the spectral norm (also denoted $\|\cdot\|_\sigma$), and $\|\cdot\|_\infty$, which is also equal to the $(1,\infty)$ norm appearing in the analysis of [GRS18]. We will also use the matrix $(q,1)$ norm, which is given by

$$\|A\|_{q,1} = \sum_{j=1}^{n} \Big(\sum_{i=1}^{m} |a_{ij}|^q \Big)^{1/q}. \tag{2.6}$$

We consider networks of the form

$$f(x;W) = W_L \phi(W_{L-1}\phi(\cdots\phi(W_1 x))) \tag{2.7}$$

where $W_\ell[j_\ell, j_{\ell-1}] = w_{j_\ell, j_{\ell-1}}$ are the $d_\ell \times d_{\ell-1}$ weight matrices ($d_0 = d$ and $d_L = k$) and $\phi(z)$ is an activation which is positive homogeneous[1], 1-Lipschitz and satisfies $\phi(0) = 0$. The most common example is the ReLU activation $\phi(z) = \max(z, 0)$, though it also applies to other activations, such as the 'leaky ReLU' or the identity. Since compositions of positive homogeneous functions are also positive homogeneous, our theory applies to max or average pooling operations followed by a positive homogeneous activation, as in convolutional networks. For each output unit $j_L \in \{1, ..., k\}$, the subnetwork terminating at node j_L may be expressed as

$$f(x;W)_{j_L} = \sum_{j_{L-1}} w_{j_L, j_{L-1}} \phi\Big(\cdots\phi\Big(\sum_{j_0} w_{j_1, j_0} x_{j_0}\Big)\Big). \tag{2.8}$$

Each unit of the network outputs $x_{j_\ell} = \phi(z_{j_\ell})$ for a corresponding input

$$z_{j_\ell} = \sum_{j_{\ell-1}} w_{j_\ell, j_{\ell-1}} x_{j_{\ell-1}} = \sum_{j_{\ell-1}} w_{j_\ell, j_{\ell-1}} \phi(z_{j_{\ell-1}}). \tag{2.9}$$

[1]The property that for all $\alpha > 0$, $\phi(\alpha z) = \alpha\phi(z)$.

ℓ_1 normalization and the path distribution

A crucial observation made in [BK18] is that by doubling the number of nodes per layer and relabeling the indices, we can assign the absolute weights $|w_{j_\ell,j_{\ell-1}}|$ into one of two pre-specified groups, each of size $d_{\ell-1}$: (I) if $w_{j_\ell,j_{\ell-1}}$ is negative, we associate it with $-\phi(z_{j_{\ell-1}})$ and (II) if $w_{j_\ell,j_{\ell-1}}$ is positive, we associate it with $+\phi(z_{j_{\ell-1}})$. By doing this, we can assume all the weights are nonnegative. For notational convenience, we do not explicitly account for these sign differences in the activation function when we describe the network. Instead, without loss of generality, we simply write ϕ with the understanding that it is a placeholder for either $-\phi$ or $+\phi$. Likewise, we use W_ℓ with the understanding that the weights are taken to be nonnegative, which results in quantities that are in terms of the absolute values of the original weights.

With this convention, we may exploit the positive homogeneity of ϕ, and move all the (non-negative) weights to the inner layer sum to get

$$f(x;W)_{j_L} = \sum_{j_{L-1}} \phi\Big(\sum_{j_{L-2}} \phi\Big(\cdots\phi\Big(\sum_{j_0} w_{j_0,j_1,\ldots,j_L} x_{j_0}\Big)\Big)\Big) \tag{2.10}$$

where

$$w_{j_0,j_1,\ldots,j_L} = w_{j_L,j_{L-1}} w_{j_{L-1},j_{L-2}} w_{j_{L-2},j_{L-3}} \cdots w_{j_1,j_0}. \tag{2.11}$$

Here we think of each $(j_0,\ldots,j_L)$ as indexing a single *path* through the network. It is this representation of the network in terms of the paths that facilitates our analysis.

Remark 2.1 (Path representation for networks with pooling)**.** We remark that a similar expression can be used to study convolutional networks with max and/or average pooling, by recalling that 2D convolution can be expressed as matrix multiplication with respect to a particular class of Toeplitz matrices and that the (max or average) pooling operator $\mathcal{P}$ is positive homogeneous. For simplicity, in what follows we omit the use of $\mathcal{P}$, hence considering feed-forward networks or convolution networks without pooling, though we address the details of the convolutional case with pooling in Appendix A.2.

We now turn our attention to *normalizing* the path weights $(w_{j_0,j_1,\ldots,j_L})_{j_0,j_1,\ldots,j_L}$ in such a way that they may form a probability distribution. Since the $w_{j_0,\ldots,j_L}$ are non-negative, the simplest way to do this would be to normalize by their sum, which is also equal to the 1-norm of the product of the (non-negative) weights:

$$\mathcal{V}_1 = \sum_{j_0,j_1,\ldots,j_L} w_{j_0,j_1,\ldots,j_L} = \Big\| \prod_1^L |W_\ell| \Big\|_{1,1}. \tag{2.12}$$

Now by construction we see that we can equally well express the function as

$$f(x;W)_{j_L} = \mathcal{V}_1 \sum_{j_{L-1}} \phi\Big(\sum_{j_{L-2}} \phi\Big(\cdots\phi\Big(\sum_{j_0} p_{j_0,j_1,\ldots,j_L} x_{j_0}\Big)\Big)\Big), \tag{2.13}$$

where $p_{j_0,\ldots,j_L} = \frac{1}{\mathcal{V}_1} w_{j_0,\ldots,j_L}$. We see that by design, $p_{j_0,\ldots,j_L} \geq 0$ and $\sum_{j_0,\ldots,j_L} p_{j_0,\ldots,j_L} = 1$. Hence we can view $(p_{j_0,\ldots,j_L})_{j_0,\ldots,j_L}$ as a discrete distribution over the multi-indices $(j_0,\ldots,j_L)$, which we interpret as a path (a sequence of nodes) through the network. We call $(p_{j_0},\ldots,p_{j_L})_{j_0,\ldots,j_L}$ the 1-*path distribution* and the normalizing factor $\mathcal{V}_1$ the 1-*path variation* of the network $f(x;W)$. As we discuss in Section 2.4, in the single output case, $\mathcal{V}_1$ is in fact the same as the 1-path norm, studied in [NTS15].

Another quantity that will arise in our analysis is related to the 1/2-*Renyi entropy* of the marginal distributions p_ℓ (obtained by marginalization of the path distribution $p_{j_0,j_1,\ldots,j_\ell}$). We define the 1-path complexity to be

$$\zeta_1 = \frac{1}{L}\Big(1 + \sum_{\ell=1}^{L-1} e^{\frac{1}{2}H_{1/2}(p_\ell)}\Big) \tag{2.14}$$

Since $0 \leq H_{1/2}(p_\ell) \leq \log(d_\ell)$, we have $1 \leq \zeta_1 \leq \frac{1}{L}(1 + \sum_{\ell=1}^{L-1}\sqrt{d_\ell})$, though this quantity can be substantially smaller when the marginal distributions p_ℓ are non-uniform over units in the network. Hence ζ_1 can be thought of as a measure of the average *effective* square-root width of the intermediate layers.

Importantly, the path distribution p can be shown to possess a Markov structure (see [BK18]), allowing us to write

$$p_{j_0,\ldots,j_L} = p_{j_L} p_{j_{L-1}|j_L} p_{j_{L-2}|j_{L-1}} \cdots p_{j_0|j_1} \tag{2.15}$$

and the network correspondingly as

$$\mathcal{V}_1 f(x;P) = \mathcal{V}_1 P_L \phi(P_{L-1}\phi(\cdots\phi(P_1 x))) \tag{2.16}$$

where P_ℓ is a transition matrix for the Markov distribution p, $P_\ell[j_\ell, j_{\ell-1}] = p_{j_{\ell-1}|j_\ell}$ for $\ell < L$ and $P_L[j_L, j_{L-1}] = p_{j_L,j_{L-1}} = p_{j_L} p_{j_{L-1}|j_L}$.

Constructing sparse approximants from the path distribution

The representations (2.13) suggests an approach for constructing an approximant $\tilde{f}$ of f, by taking $\tilde{f} = f(x;\tilde{p})$ for some estimate $\tilde{p}$ of p. Since p is a probability distribution, a natural candidate for an approximant $\tilde{p}$ is an empirical distribution which arises from taking M independent samples from the path distribution p. We refer to such an empirical distribution as $\tilde{p}_M$, or simply $\tilde{p}$, when the number of samples M is clear. If one can then bound $\mathbb{E}_{\tilde{p}}[\|f(x;p) - f(x;\tilde{p})\|] \leq \delta_M$, for some δ_M, then since the average over $\tilde{p}$ is always more than the minimum over $\tilde{p}$, one can deduce the existence of some $\tilde{p}$ for which $\|f(x;p) - f(x;\tilde{p})\| \leq \delta_M$. This type of reasoning is known as the *probabilistic method*, and appears in many results in the literature, as we discuss in Section 2.3. It is also employed in the main result of [BK18], which we now review.

Consider sampling $K = (K_{j_0,j_1,\ldots,j_L})_{j_0,j_1,\ldots,j_L} \sim \text{Multinomial}(M,p)$, where $K_{j_0,j_1,\ldots,j_L}$ is the number of times the path $(j_0,j_1,\ldots,j_L)$ appeared in the M samples. One could then take

an approximant to be $\tilde{p} = K/M$. However, this $\tilde{p}$ would not necessarily factor into matrices, which is favorable both for practical reasons and for the sake of analysis. Instead, [BK18] construct $\tilde{p}_{j_0,j_1,\ldots,j_L} = \tilde{p}_{j_L}\tilde{p}_{j_{L-1}|j_L}\cdots\tilde{p}_{j_0|j_1}$ as the empirical Markov distribution on the paths $(j_0, j_1, \ldots, j_L)$, where

$$\tilde{p}_{j_\ell} = \frac{K_{j_\ell}}{M}, \quad \tilde{p}_{j_\ell,j_{\ell+1}} = \frac{K_{j_\ell,j_{\ell+1}}}{M}, \quad \tilde{p}_{j_\ell|j_{\ell+1}} = \frac{\tilde{p}_{j_\ell,j_{\ell+1}}}{\tilde{p}_{j_{\ell+1}}} \tag{2.17}$$

with the convention that $0/0 = 0$. Here $K_{j_\ell}, K_{j_\ell,j_{\ell+1}}$ are the marginal and pairwise counts, respectively, obtained by summing out $K_{j_0,\ldots,j_\ell,j_{\ell+1},\ldots,j_L}$ over unspecified indices.

Another more principled reason to favor a network built from the above quantities is that, within the class of Markov distributions, $\tilde{p}$ is the (restricted) maximum likelihood estimator (MLE) of p from the empirical counts K. We state this formally in our first theorem. At a high-level, it says that, among plug-in approximants of the original network, the one using the empirical Markov distribution is 'optimal'.

Theorem 2.1.

$$\tilde{p} = \arg\max_{p \text{ Markov}} \mathcal{L}(p),$$

where $\mathcal{L}(p) = M! \prod_{(j_0,j_1,\ldots,j_L)} \frac{p_{j_0,j_1,\ldots,j_L}^{K_{j_0,j_1,\ldots,j_L}}}{K_{j_0,j_1,\ldots,j_L}!}$ *is the likelihood of the count vector* K*.*

Proof. This can be shown by combining the fact that the (unrestricted) MLE of a multinomial distribution is the empirical class proportion vector K/M, together with the invariance property of MLEs. □

Throughout, we think of the number M as a parameter which controls the level of compression of $f(x;\tilde{p}_M)$ relative to $f(x;p)$. Intuitively, it is clear that as M gets large, $f(x;\tilde{p}_M)$ more closely approximates $f(x;p)$. Moreover, M controls the sparsity and precision of the parameters $\tilde{p}_M$, which is demonstrated by the following facts: first, the number of nonzero parameters $\tilde{p}_M$ is upper bounded deterministically by LM, and, second, the (base-10) precision of the $\tilde{p}_M$ is upper bounded by $\log_{10}(M)$. Hence, we can think of $\tilde{p}_M$ as also a natural quantization of the weights p.

In the single output case, [BK18] prove the following L_2 bound (adapted slightly to match our notation), when $\mathcal{X} = [-1,1]^d$.

Theorem 2.2 ([BK18], Theorem 1)**.** *Let* $f(x;W)$ *be a single output ReLU network with 1-path variation* $\mathcal{V}_1$ *and 1-path complexity* ζ_1*, and let* $\mathbb{P}$ *be probability measure on* $[-1,1]^d$*. Then*

$$\mathbb{E}_{\tilde{p}}\Big[\int |f(x;W) - f(x;\widetilde{W})|^2 \mathbb{P}(dx)\Big] \le \Big(\frac{\mathcal{V}_1\zeta_1 L}{\sqrt{M}}\Big)^2, \tag{2.18}$$

where $f(x;\widetilde{W}) = \mathcal{V}_1 f(x;\tilde{p})$*.*

In the next section, we extend this result in the following ways: first, we show that we can obtain path distributions by normalizing by a much broader class of path variations than the 1-path variation $\mathcal{V}_1$. We also extend the bound to the multi-output setting, where we obtain a bound on the ℓ_2 norm of outputs. This result allows us to later study multi-class classification. Finally, we show that similar results can also be obtained for networks with pooling layers, though we defer the details of this case to the Appendix.

2.3 Path Sampling and Sparse Approximation

Path sampling with general norms

In this section, we show how the sampling scheme summarized in the previous section can be generalized to norms besides $\|\cdot\|_{1,1}$. To see how this is possible, notice that for any w_{j_0}, we can express $f(x;W)_{j_L}$ as

$$\sum_{j_{L-1}} \phi\Big(\sum_{j_{L-2}} \phi\Big(\cdots \phi\Big(\sum_{j_0} w_{j_0} w_{j_0,j_1,\ldots,j_L} x'_{j_0} \Big)\Big)\Big) \tag{2.19}$$

where $x'_{j_0} = x_{j_0}/w_{j_0}$. Now for

$$\mathcal{V} = \sum_{j_0,\ldots,j_L} w_{j_0} w_{j_0,\ldots,j_L}, \tag{2.20}$$

we can define a path distribution $p_{j_0,\ldots,j_L} = \frac{1}{\mathcal{V}} w_{j_0} w_{j_0,\ldots,j_L}$.

Now let $1 \le q \le \infty$ and let q^* be its conjugate exponent (so that $\frac{1}{q} + \frac{1}{q^*} = 1$). For a dataset S, we consider

$$w_{j_0}^{(q)} = \begin{cases} (n^{-1} \sum_{x \in S} |x_{j_0}|^{q^*})^{1/q^*} & 1 < q \le \infty \\ \max_{x \in S} |x_{j_0}| & q = 1 \end{cases} \tag{2.21}$$

which gives rise to the *q-path variation*

$$\mathcal{V}_q = \sum_{j_0,\ldots,j_L} w_{j_0}^{(q)} w_{j_0,\ldots,j_L} \tag{2.22}$$

The value in these definitions is captured in the following lemma, which shows that we can bound $\mathcal{V}_q$ in terms of norms $\| \prod_1^L |W_\ell| \|$.

Lemma 1. *Let $1 \le q \le \infty$, and let q^* be its conjugate exponent. Then*

$$\mathcal{V}_q \le (\max_{x \in S} \|x\|_{q^*}) k^{1-1/q^*} \Big\| \prod_1^L |W_\ell| \Big\|_{q^*} \tag{2.23}$$

and

$$\mathcal{V}_q \le (\max_{x \in S} \|x\|_{q^*}) \Big\| \prod_1^L |W_\ell| \Big\|_{q,1}. \tag{2.24}$$

Proof. The proof is several simple applications of Hölder's inequality. See A.2 for details. □

With the above in mind, we introduce the following definitions.

Definition 2.1. For $1 \le q \le \infty$, we define the q-path distribution by

$$p^{(q)}_{j_0,\dots,j_L} = \frac{w^{(q)}_{j_0} w_{j_0,\dots,j_L}}{\mathcal{V}_q}. \tag{2.25}$$

We define the q-path complexity by

$$\zeta_q = \frac{1}{L}\Big(1 + \sum_{\ell=1}^{L-1} e^{\frac{1}{2}H_{1/2}(p^{(q)}_\ell)}\Big). \tag{2.26}$$

Notice that for $q = 1, r = 1$, the above definitions reduce to the setting of the Section 2.2, with $p^{(1)}$ obtained by normalizing by $\|\prod_1^L |W_\ell|\|_{1,1}$ and $x \in [-1,1]^d$. In the next section, we use the path distributions $p^{(q)}$ to obtain sparsification results for vector-valued neural networks.

Bounds for Path Sampling with Deep ReLU Networks

In this section, we extend the analysis of Theorem 2.2 to the multi-output and convolution setting, and show that similar bounds may be obtained in terms of $\mathcal{V}_q$, for $q \in [1,2]$.

Theorem 2.3. *Let $f(x;W)$ be an L-layer ReLU network, S a dataset, and let $1 \le q \le 2$. If $\tilde{p}$ is the Markov distribution formed from M samples from $p^{(q)}_{j_0,j_1,\dots,j_L}$, then*

$$\mathbb{E}_{\tilde{p}}\Big[\frac{1}{n}\sum_{x\in S} \|f(x;\widetilde{W}) - f(x;W)\|_2^2\Big] \le \Big(\frac{\mathcal{V}_q \zeta_q L}{\sqrt{M}}\Big)^2, \tag{2.27}$$

where $f(x;\widetilde{W}) = \mathcal{V}_q f(x;\tilde{p})$.

Proof. See A.1. □

Using Lemma 1, Theorem 2.3 can be used to give bounds, for example, in terms of the matrix $(2,1)$ norm, the spectral norm, and the $(1,\infty)$ norm.

Since the minimum over $\tilde{p}$ is always less than the expected value, the above results imply, for example, the existence of representer $f(x;\widetilde{W}) = \mathcal{V}_q f(x;\tilde{p}_M)$ such that

$$\sqrt{\frac{1}{n}\sum_{x\in S} \|f(x;\widetilde{W}) - f(x;W)\|_2^2} \le \frac{\mathcal{V}_q \zeta_q L}{\sqrt{M}}. \tag{2.28}$$

It turns out that, in the case of $q = 1$, the error analysis in Theorem 2.3 is optimal for single output, two layer networks.

Theorem 2.4. *There exists a dataset $S \subseteq [-1,1]^d$, a single output, two layer network $f(x;W)$, and an integer M_0 such that for all $M \geq M_0$, we have*

$$\mathbb{E}_{\tilde{p}}\Big[\frac{1}{n}\sum_{x\in S}|f(x;\widetilde{W}) - f(x;W)|^2\Big] = \Omega\Big(\frac{\mathcal{V}_1\zeta_1}{\sqrt{M}}\Big)^2. \tag{2.29}$$

Proof. See A.2. □

Comparison to existing techniques

It is worth taking a moment to compare our technique and results to existing work. The probabilistic method, interestingly, appears frequently.

In [BFT17], layers are sparsified individually, using a technique known as *Maurey sparsification*. This type of reasoning in the context of function approximation is due to [PM80] and was later applied to single output, single hidden layer networks (i.e., $k = 1$ and $L = 2$) in the seminal work of [Bar91, Bar93]. [BFT17] use a generalization of this technique given in [Zha02] to establish the existence of an approximant $\widetilde{U}$ of a matrix U by defining a distribution over $\widetilde{U}$ and bounding $\mathbb{E}\|\widetilde{U} - U\|^2$, though using this approach to analyze multilayer networks results in error bounds that scale with $\prod_1^L \|W_\ell\|$, which arises from a worst case analysis of error propagation between layers. In contrast, our technique takes advantage of *global* structure in the network, and as a consequence instead scales with a quantity $\|\prod_1^L |W_\ell|\|$.

Other examples of sampling techniques include the recent work of [AGNZ18] and [BLG+19]. The former uses a Johnson-Lindenstrauss-type random projection to compress each layer of a deep network, which deduces the existence of a compression by showing that the probability of sampling an approximant at the desired level of accuracy is nonzero. [BLG+19] instead use an edge-wise sampling approach which is similar to existing matrix sparsification techniques (e.g. [AKL13, KD14, DZ11]), but notably improves these methods by using a held-out set of data to measure the sensitivity of each layer's output to certain edges. In both cases, however, the error analysis does not involve norm quantities, and instead involves strongly data dependent quantities which are harder to compute and interpret. As a consequence of the stronger dependence on the function and dataset S, these techniques can only be used to prove a slightly weaker notion of generalization; they prove only the generalization of the compressed network, rather than the original network. It is nonetheless a fascinating direction for future research to study if stronger data-dependence may be incorporated into our path-based approach to obtain better error bounds.

2.4 Covering, Metric Entropy, and Implications for Generalization

In this section, we show that the sampling results given in the previous section can be used to obtain covering number bounds, which imply generalization bounds for multi-class

classification. The approach is similar to that used to prove the results of [BFT17]. Throughout this section, we use $\mathcal{F}(\mathcal{V}_q, \zeta_q)$ to denote the class of positive homogeneous networks with q-path variation at most $\mathcal{V}_q$ and q-path complexity at most ζ_q, and for any $\gamma > 0$, we denote $\mathcal{F}_\gamma(\mathcal{V}_q, \zeta_q) = R_\gamma \circ (-\mathcal{M}) \circ \mathcal{F}(\mathcal{V}_q, \zeta_q)$.

We first recall a few definitions.

Definition 2.2. For a class of real-valued functions $\mathcal{F}$ and $\epsilon > 0$, a set $\mathcal{G}_\epsilon$ is an *ϵ-covering* of $\mathcal{F}$ (with respect to $\|\cdot\|_{2,S}$) if for all $f \in \mathcal{F}$, there exists $g \in \mathcal{G}_\epsilon$ such that $\|f - g\|_{2,S} = \sqrt{\frac{1}{n}\sum_{x\in S} |f(x) - g(x)|^2} \le \epsilon$. We define the *$\epsilon$-covering number* $\mathcal{N}_2(\epsilon, \mathcal{F}, S)$ to be the minimum cardinality of covering sets $\mathcal{G}_\epsilon$. Finally, the *ϵ-metric entropy* of $\mathcal{F}$ is defined to be $\log \mathcal{N}_2(\epsilon, \mathcal{F}, S)$.

To use the sampling bounds from Theorem 2.3 to get covering number bounds, we need to bound the cardinality of the set of functions $\tilde{f}$ arising from M samples. The below gives such a bound.

Theorem 2.5. *The number of networks $f(x; \tilde{p})$ that arise from the sampling scheme is at most $8^{ML}(de)^M$. Thus, the log-cardinality of the representor set is bounded by $M(\log(de) + L\log(8))$.*

Proof. See A.3. □

We remark that a more naïve bound may be obtained by simply counting the total number of possible samples $K = K_M$ that can arise from sampling Multinomial(M, p). This can be shown by a standard combinatorial argument to be $\binom{M+D-1}{M}$, where $D = d_0 d_1 \cdots d_L$. Theorem 2.5 improves this bound considerably by recognizing that there are many samples K_M which result in the same function $f(x; \tilde{p}_M)$. Exploiting this observation, the proof takes advantage of the permutation invariance of units in deep networks, which implies that the number of functions that can be obtained from the sampling scheme depends only on the number of ways we can partition the integer M into pairwise counts $K_{j_\ell, j_{\ell+1}}$. As a consequence, the cardinality is *completely* independent of the intermediate layer dimensions d_ℓ for $\ell = 1, 2, \ldots, L$, except for mild logarithmic dependence on the input dimension d. It turns out that in the setting of the 2-path variation, we can use a trick from [Zha02] to remove dependence on d altogether, though we defer the details of this to Appendix A.1.

Using the fact that $\mathcal{M}$ is 2-Lipschitz with respect to $\|\cdot\|_2$ (see appendix of [BFT17] for details), and R_γ is $\frac{1}{\gamma}$ Lipschitz, we may use this result together with Theorem 2.3 to obtain the following metric entropy bounds.

Corollary 2.1. *Let $\epsilon, \gamma > 0$, $1 \le q \le 2$. Then*

$$\log \mathcal{N}_2(\epsilon, \mathcal{F}_\gamma(\mathcal{V}_q, \zeta_q), S) \le \frac{9\mathcal{V}_q^2 \zeta_q^2 L^2 (L + \log(de))}{\gamma^2 \epsilon^2} \tag{2.30}$$

Using standard techniques, Corollary 2.1 implies the following generalization guarantees.

Theorem 2.6. *Let $f(x;W)$ be an L-layer positive homogeneous network and let $\delta \in (0,1)$. For any $1 \le q \le 2$ and $\gamma > 0$, with probability at least $1-\delta$ over the training set S, the generalization error $\ell(f) - \hat{\ell}_\gamma(f)$ is bounded by*

$$\widetilde{\mathcal{O}}\Big(\frac{\mathcal{V}_q \zeta_q L\sqrt{L+\log(d)}}{\gamma\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{n}}\Big), \tag{2.31}$$

where $\mathcal{V}_q, \zeta_q$ are the q- path variation and path complexity of f.

Proof. See A.4. □

Plugging in the bounds from Lemma 1, this implies a host of norm-based generalization bounds, summarized in the following Corollary.

Corollary 2.2. *Let $f(x;W)$ be an L-layer positive homogeneous network and let $\delta \in (0,1)$. For any $1 \le q \le 2$ and $\gamma > 0$, with probability at least $1-\delta$ over the training set S, the generalization error $\ell(f) - \hat{\ell}_\gamma(f)$ is bounded by*

$$\widetilde{\mathcal{O}}\Big(\frac{\max_{x\in S}\|x\|_{q^*}\big\|\prod_1^L |W_\ell|\big\|_{q,1}\zeta_q L\sqrt{L+\log(d)}}{\gamma\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{n}}\Big), \tag{2.32}$$

as well as

$$\widetilde{\mathcal{O}}\Big(\frac{\max_{x\in S}\|x\|_{q^*} k^{1-1/q^*}\big\|\prod_1^L |W_\ell|\big\|_{q^*}\zeta_q L\sqrt{L+\log(d)}}{\gamma\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{n}}\Big). \tag{2.33}$$

Comparison to Existing Generalization Bounds

Before comparing the bounds from Theorem 2.6 to existing norm-based bounds, we remark that by arranging for the weights to be positive, the matrix products we obtain above are in terms of *absolute values* of the original weight matrices. However, we can still compare our bounds to those that use products of matrix norms. For example, for entry-wise norms such as $(q,1)$ norms, the $(1,\infty)$ norm, and the Frobenius norm, it is always the case that $\|\prod_\ell |W_\ell|\| \le \prod_\ell \|W_\ell\|$. On the other hand, note that $\||A|\|_\sigma \ge \|A\|_\sigma$ and so the term $\|\prod_1^L |W_\ell|\|_\sigma$ is not directly comparable to $\prod_\ell \|W_\ell\|_\sigma$. Nevertheless, there are many examples of (non-positive) weight matrices such that $\|\prod_\ell |W_\ell|\|_\sigma \le \prod_\ell \|W_\ell\|_\sigma$.

Several results have identified product of weight norms as a complexity measure; we detail a few here. For example, [BFT17] use covering numbers to obtain the generalization bound

$$\widetilde{\mathcal{O}}\Big(\frac{\max_{x\in S}\|x\|_2 L^{3/2}\overline{R}^{3/2}\prod_1^L \|W_\ell\|_\sigma}{\gamma\sqrt{n}}\Big), \tag{2.34}$$

where $\overline{R} = \frac{1}{L}\sum_1^L \left(\frac{\|W_\ell\|_{2,1}}{\|W_\ell\|_\sigma}\right)^{2/3}$, which, similar to our ζ quantities, admits an interpretation as an average *effective* width. This is naturally contrasted with our bound in terms of $\|\prod_1^L |W_\ell|\|_\sigma$ and ζ_2, though as we mention above, the quantities $\|\prod_1^L |W_\ell|\|_\sigma$ and $\prod_1^L \|W_\ell\|_\sigma$ are not directly comparable. However, there are examples of matrices for which our bound is superior. Similar bounds were likewise obtained via a PAC-Bayes approach in [NBS18], though these are known to be strictly weaker than the above bound from [BFT17].

The approach of [GRS18] (which addressed the single output case) used a more direct bound on Rademacher complexities, via 'peeling', to get generalization bounds of order

$$\frac{\max_{x\in S}\|x\|_2\sqrt{L+\log(d)}\prod_1^L\|W_\ell\|}{\gamma\sqrt{n}}, \tag{2.35}$$

where the associated norm is $\|\cdot\|_{1,\infty}$ or $\|\cdot\|_F$. Notably, these bounds avoid the correction factors $\overline{R}$ or ζ appearing in our results and [BFT17, NBS18], and have slightly more mild (explicit) dependence on the number of layers. However, since these bounds are in terms of entry-wise norms, our capacity constants can be shown to lower bound these quantities. For example, in the single output case, $\mathcal{V}_2$ will lower bound the product $\prod_1^L\|W_\ell\|_F$. Similarly, taking $q=1$ in Corollary 2.2, we get a bound in terms of $\|\prod_1^L|W_\ell|\|_{1,\infty}$, which lower bounds $\prod_1^L\|W_\ell\|_{1,\infty}$.

Other norm-based bounds have identified more global quantities as complexity measures for deep networks. Of particular relevance to our work is the path norm ϕ_p of a network $f(x;W)$ studied in [NTS15, NSS15], which is defined as[2]

$$\phi_p = \sum_{j_L}\Big(\sum_{j_0,\ldots,j_{L-1}}|w_{j_0,\ldots,j_L}|^p\Big)^{1/p}. \tag{2.36}$$

We observe immediately that $\phi_1 = \mathcal{V}_1$. In [NTS15], generalization bounds of order

$$\frac{\max_{x\in S}\|x\|_\infty \mathcal{V}_1 2^L}{\gamma\sqrt{n}}, \tag{2.37}$$

were given for the case of $p=1$ and $k=1$. While this bound avoids involving products of norms, it has explicit exponential dependence on the depth of order 2^L, which our bound improves to the low order polynomial $L^{3/2}$. Furthermore, the following lemma shows that $\mathcal{V}_2$ may be upper bounded by ϕ_2. Hence we can view our results also as an improvement on this line of work.

Lemma 2. *We have*

$$\mathcal{V}_2 \le \sum_{j_L}\Big(\sum_{j_0,j_1,\ldots,j_{L-1}} w^2_{j_0,j_1,\ldots,j_L}\Big)^{1/2} \tag{2.38}$$

[2]Note this is in fact a generalization of the definition in [NTS15], which considered only the single output case, and hence did not have the sum over j_L.

where in the single output case, the right-hand side is equal to the 2*−path norm* ϕ_2 *from [NTS15].*

Proof. See A.2. □

Finally, we remark that while more direct analysis of Rademacher complexities, such as [GRS18, NTS15], avoid the correction factors such as ζ and $\overline{R}$, these works seem to address only the single output case. It is therefore unclear if extending these analyses to the multi-class setting would involve more direct dependence on the number of classes k.

2.5 Empirical Investigation

In the previous section, we used a sampling procedure as a technical tool to derive generalization bounds. Intuitively, the ability to express a large network with many parameters as a network with few parameters of low precision indicates lower complexity. In this section, we investigate this relationship empirically, using the sampling procedure employed theoretically above. For simplicity, we work with $\mathcal{V}_1$ and the 1-path distribution. For each network $\mathcal{V}_1 f(x; p)$ considered, we will draw Multinomial(M, p) samples and compute the corresponding estimates $\tilde{p}_M$. Here, as is justified in Section 2, we will use the number of trials M as a proxy for compression, and investigate the number of samples M required to obtain a given level of accuracy. We note that working directly with the full path distribution p quickly becomes unwieldy as the network grows in depth, as it involves storing a (potentially dense) L-tensor. Fortunately, by exploiting the Markov structure of p and storing only the conditional distributions $p_{j_\ell|j_{\ell+1}}$, and sampling forward through the Markov chain, this issue can be avoided.

We study three different problems, which range from easy (generalize very well) to hard (generalize poorly). Namely, we study a basic `mnist` network, a `cifar10` network, and a `cifar10` network with labels chosen uniformly at random. We use a four layer, feed-forward network with hidden layer dimensions $600, 400$ and 200. Throughout our experiments, plots demonstrating the performance of the `mnist` (easy; test accuracy $\approx 98\%$) network will be shown in orange ●, the `cifar10` (medium; test accuracy $\approx 60\%$) network in green ●, and the `cifar10` with random labels (hard; test accuracy $\approx 10\%$) network in blue ●. Each network is trained to 100% training accuracy using stochastic gradient descent with momentum set to 0.9 and no additional regularization.

We observe that compressibility does indeed correlate with generalization: networks with higher test accuracy can be represented with fewer samples. For example, we see from Figure 2.1 that with $M = 10^6$ samples, we can obtain 100% accuracy from the `mnist` model, 80% from the `cifar10` model, and only 40% from the `cifar10` model with random labels.

According to our theory, this accuracy should be governed in part by the trade-off of $\mathcal{V}_1/\gamma$ with $\hat{\ell}_\gamma(f)$, where γ is some chosen value of margin. To study this, we look at the *path*

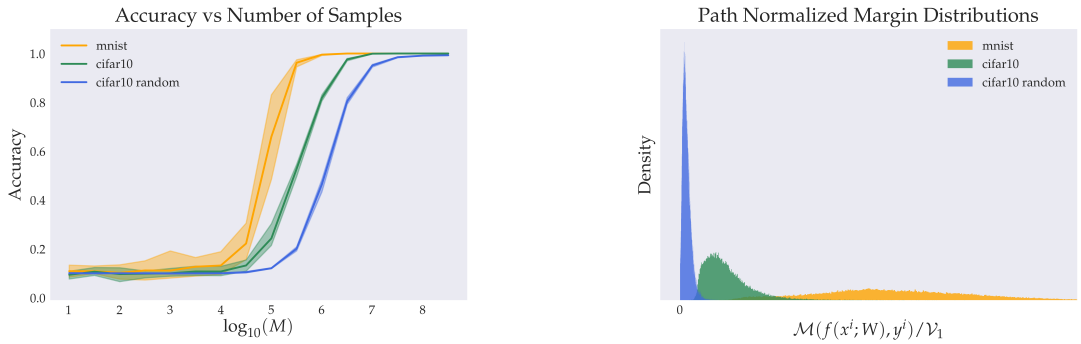


Figure 2.1: **Left**: Comparison of sampling with `mnist`, `cifar10`, and `cifar10` with random labels. Solid line indicates mean classification accuracy on the training set over 10 trials; shaded region indicates range over these trials. **Right**: Comparison of path normalized margin distributions. We observe that the `mnist` model has considerably larger normalized margins than the `cifar10` model, which itself has larger normalized margins than `cifar10` with random labels.

normalized margin distribution, which for a network $f(x;W)$ and dataset $\{(x^i,y^i)\}_{i=1}^n$, is the histogram of values

$$\frac{\mathcal{M}(f(x^i,W),y^i)}{\mathcal{V}_1}. \tag{2.39}$$

This is analogous to the normalized margins studied in [BFT17], though with the path variation serving as the normalizing constant, rather than the spectral complexity (which is related to the network's Lipschitz constant). In Figure 2.1, we see that these distributions do indeed significantly distinguish the three models. Intuitively, it seems natural that larger margin classifiers would be easier to sparsely approximate, as correspondingly larger perturbations to the function's output do not change the function's classification decision. It is also well documented, and is suggested by Theorem 2.6, that large (normalized) margins play an important role in generalization behavior of neural network classifiers. Thus we see that, in this sense, model sparsification and compression are strongly related to generalization.

2.6 Conclusions and Future Directions

In this paper, we exploited the Markov structure of positive homogeneous networks to analyze and implement a sampling scheme for network sparsification, which we then used to obtain covering number and generalization bounds. Our analysis identified the path variations $\mathcal{V}_q$, which we show to be bounded by various norms $\|\prod_1^L |W_\ell|\|$, as important quantities controlling approximation rates and generalization error, which we then verified empirically. In what follows, we briefly highlight some potential directions for further work building on the present results.

Sampling with residual networks. Residual networks have been shown to be a powerful network architecture, used to obtain state-of-the-art performance on many difficult classification tasks. However, our analysis does not immediately apply to networks with skip connections. Here we present one potential direction for extending our techniques to the analysis of residual networks.

Recently, [VWB16] proposed an 'unravelled' view of residual networks as a collection of paths with different lengths. We show how to utilize this perspective to develop a sampling strategy. Each unique path $P = (j_{\ell_0}, j_{\ell_1}, \ldots, j_{\ell_m})$ through the residual network can be assigned a binary code $b(P) \in \{0,1\}^L$ where $b_t(P) = 1$ if the input flows through residual module t and 0 if it is skipped (i.e., a skip connection). In this case, the path distribution of a residual network can be seen as a mixture of path distributions for fully-connected networks, namely, $p = 2^{-L} \sum_{b \in \{0,1\}^L} p^{(b)}$, where $p^{(b)}$ is the path distribution of the fully-connected subnetwork induced by all paths P such that $b = b(P)$.

Thus, the marginal distribution of paths leading up to residual module t is a mixture of 2^{t-1} different path distributions generated from every possible configuration of the previous $t-1$ residual modules. Note also that p generates paths with lengths that are distributed $\mathrm{Binom}(1/2, L)$. This coincides with the model of path lengths proposed in [VWB16], who empirically show that they are distributed $\mathrm{Bin}(1/2, L)$ and concentrate around $L/2$. Samples from p can easily be generated by first sampling $\tilde{b}$ from the uniform mixing distribution on $\{0,1\}^L$ and then sampling a path from the Markov distribution $p^{(\tilde{b})}$. Counts K of indices can be used to form the empirical Markov distribution $\tilde{p}$ as before.

Removing the path complexity. A notable difference between our results and those of [GRS18] is the presence of the path complexity ζ. As a similar term also appears in [BFT17], it seems as though such correction factors may be consequences of the covering approach. By using a 'peeling' argument, and bounding the Rademacher complexity directly, [GRS18] are able to avoid such factors. It is an interesting open question whether the path complexity term can be removed from our bounds using similar techniques.

Chapter 3

The Pitfalls of the Worst-Case Approach in the Overparameterized Regime

The contents of this chapter are partially based on the paper "Good Classifiers are Abundant in the Interpolating Regime", appearing in AISTATS 2021, co-authored with Jason Klusowski and Michael M.W. Mahoney [TKM21].

3.1 Introduction

The phenomenon of good generalization in highly over-parameterized models, including neural networks, has largely eluded theoretical understanding. Recently, however, progress has been made towards understanding over-parameterization in several simpler settings. Important examples include the variety of results demonstrating "double descent" phenomena in linear regression [BHMM19b, BLLT20, HMRT19, DLM19] (and, in particular, how it is essentially a consequence of a transition between two different phases of learning [LCM20]), nearest neighbors models [XSC19], and binary classification [CL20, DKT20]. These results are typically derived by defining a specific estimator (e.g., the least-norm estimator in linear regression), and carefully examining its test risk. This approach presents a challenge when extending these analyses to the setting of neural networks, where no such estimator can easily be defined. In these situations, almost all results rely, in one way or another, on the framework of *uniform convergence*; that is, results which bound a quantity of the form

$$\varepsilon_{\text{unif}} := \sup_{f \in \mathcal{F}} \left| \widehat{\mathcal{E}}_n(f) - \mathcal{E}(f) \right|, \tag{3.1}$$

where $\mathcal{F}$ is a given function class, $\widehat{\mathcal{E}}_n$ is the training error on a dataset of n points, and $\mathcal{E}$ is the population error.

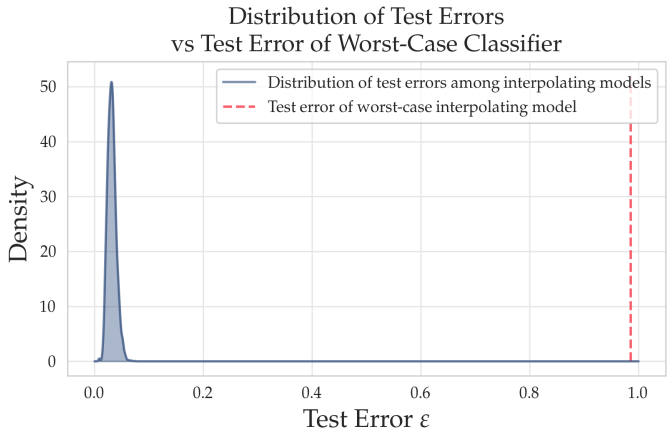


Figure 3.1: Test error distribution of MNIST 0 vs 1 interpolating classifiers, using $N = 1000$ random ReLU features, with $n = 500$ training samples, as well as test error of worst-case interpolating classifier. Here, for illustrative purposes, we plot the PDF (fit from a histogram using a kernel density estimate); in the remainder of the paper, we instead plot the CDFs, which can be more accurately estimated.

Recently, it has been drawn into question whether this approach is fine-grained enough to capture the good generalization properties observed in deep learning [MM17, NK19]. One issue that arises when using the uniform convergence framework is that for any given training set $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, and a sufficiently complex function class $\mathcal{F}$, the worst-case estimator $f \in \mathcal{F}$ fitting the training data may indeed perform quite poorly—thus dooming quantities like (3.1)—even if we are extremely unlikely to encounter such models in practice. One line of work has attempted to tackle this problem by studying the implicit biases of the algorithms used to train modern machine learning models [GWB+18, MWCC20, SHS18] (by using what may be called implicit regularization in non-exact approximation algorithms [Mah12]). Still, such results are mostly limited to simplified settings, and a comprehensive understanding of the relationship between optimization and generalization remains elusive.

In another line of work [WZ17, CHM+15b], it has been observed that, at least in practice for deep networks, it is not particularly important which model we obtain at the end of training; most models tend to have roughly the same test error. Reconciling this phenomenon with the worst-case theory must then require one of a few things to be true: i) that most models have nearly worst-case test error; ii) that models with nearly worst-case error are very rare; or iii) that worst-case bounds are simply too loose to capture the actual worst-case error. In this chapter, we investigate these possibilities rigorously in the setting of linear and random feature classification, and we find that worst-case models with very high test error do in fact exist, but that they are exceedingly rare.

Our approach builds conceptually on several old ideas originating out of the statistical physics literature. c(Such a perspective, while less common in statistical learning theory today,

has a long history [MM17, SST92, WRB93, HKSST96, EVdB01].) Rather than studying the *worst-case* estimator $f \in \mathcal{F}$, the statistical mechanics approach seeks to understand the behavior of the *typical* function f. This typicality can be characterized in a number of ways. A natural measure, from the statistical physics perspective, would be the *entropy* (or log density of states), which captures the number of models at any given test error value. Analyses of learning problems have been conducted using the entropy method in a variety of simplified settings, including the case of finite $\mathcal{F}$ as well as linear classification under various simplifying assumptions on the data [HKSST96, OH91, EVdB01]. Similar approaches have also been used to demonstrate the existence of phase transitions in learning behavior in logistic regression [CS18] and generalized linear models [BKM+19]. In the deep learning literature, [CHM+15b] used the theory of spin glasses to argue that poor local minima on the training surface are rare. While insightful (and often technically impressive), many of these theoretical results rely on very specific assumptions on the data generating process, and hold only in the asymptotic regime.

In this chapter, we study the behavior of test errors on real-world datasets used in practice, in a non-asymptotic regime, and without any assumptions on the data generating process. To do this, in Section 3.2, we formally define and develop a methodology to compute precisely the full distribution of test errors among interpolating classifiers from several model classes. In Sections 3.3 and 3.4, we then apply this methodology to compute these distributions for several real and synthetic datasets, and for both linear and random feature classification models, respectively. We furthermore develop a method to estimate the worst-case test errors of these classification models on the same datasets. Our investigation yields the following key insights:

1. Good classifiers are abundant: an overwhelming proportion of interpolating models have very small test error, relative to the worst-case error.

2. Test errors tend to concentrate: as the size of models grow, test errors concentrate sharply around a critical value ε^*.

3. There exist worst-case classifiers that are very poor: much worse than the typical classifier.

These findings are illustrated in Figure 3.1.

To understand these observations mathematically, in Section 3.5, we provide theoretical results in a simple setting in which we characterize the full (asymptotic) distribution of test errors, and we show that these indeed concentrate around a value ε^*, which we also identify exactly. We then formalize a more general conjecture, supported by our empirical findings, which we hope will motivate further research. Finally, in Section 3.6, we offer some concluding thoughts, and provide several promising directions for future work. Proofs and additional empirical results can be found in the appendix.

3.2 Efficiently Computing the Distribution of Test Errors for Interpolating Classifiers

Notation and Setup

We begin with some notation that will be used throughout the chapter.

We consider the setting of binary classification, and denote a training dataset by

$$S_n = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\},$$

with samples $\boldsymbol{x}_i \in \mathbb{R}^d$ and labels $y_i \in \{-1, 1\}$. We let $\mathcal{F}$ be a class of functions $f : \mathbb{R}^d \to \{-1, 1\}$, and we define the *version space* to be the following subset of $\mathcal{F}$:

$$\mathsf{VS}(S_n) = \{f \in \mathcal{F} : f(\boldsymbol{x}_1) = y_1, \dots, f(\boldsymbol{x}_n) = y_n\}. \tag{3.2}$$

That is, the version space is the set of "interpolating" functions, i.e., those which perfectly fit the dataset S_n. Note that if $\mathcal{F}$ is a linear family, then one element of the version space is the max-margin solution. We also use $\mathbb{P}$ to denote a probability measure defined over $\mathcal{F}$. We use $S_{\text{test}} = \{(\boldsymbol{x}_{n+1}, y_{n+1}), \dots, (\boldsymbol{x}_{n+m}, y_{n+m})\}$ to denote a set of m testing points, and $\Pr_{\boldsymbol{x},y}$ to denote a testing distribution over the data $(\boldsymbol{x}, y)$. Using these, we define the empirical and population testing errors:

$$\mathcal{E}_m(f) = \frac{1}{m} \sum_{h=1}^{m} \mathbb{1}(-y_{n+h} f(\boldsymbol{x}_{n+h}) > 0), \tag{3.3}$$

$$\mathcal{E}(f) = \Pr_{\boldsymbol{x},y}(-y f(\boldsymbol{x}) > 0). \tag{3.4}$$

With these definitions in place, we can now formally define the test error distribution of interpolating classifiers.

Definition 3.1. Given a function class $\mathcal{F}$, a measure $\mathbb{P}$ over $\mathcal{F}$, and a training set S_n, let

$$R_{n,m}(\varepsilon) := \frac{\mathbb{P}(\{\mathcal{E}_m(f) \le \varepsilon\} \cap \mathsf{VS}(S_n))}{\mathbb{P}(\mathsf{VS}(S_n))}, \tag{3.5}$$

and

$$R_n(\varepsilon) := \frac{\mathbb{P}(\{\mathcal{E}(f) \le \varepsilon\} \cap \mathsf{VS}(S_n))}{\mathbb{P}(\mathsf{VS}(S_n))}. \tag{3.6}$$

That is, the quantities $R_{n,m}(\varepsilon)$ and $R_n(\varepsilon)$ are the cumulative distribution functions (CDFs) of the errors $\mathcal{E}_m$ and $\mathcal{E}$, conditioned on perfectly fitting the training data. Intuitively, these quantities measure the fraction of interpolating classifiers $f \in \mathsf{VS}(S_n)$ that have test error at most ε.

Efficient Estimation of $R_{n,m}$

An advantage of our definition of $R_{n,m}(\varepsilon)$ is that it is defined only relative to fixed training and testing sets, S_n and S_{test}. This means that, at least in principle, $R_{n,m}(\varepsilon)$ can be computed exactly (without explicit knowledge of the training and testing distributions). To do this naïvely would require computing the ratio of two (in general very small) high-dimensional volumes, which would be costly and also lead to issues with numerical instability. Instead, a natural estimator for $R_{n,m}(\varepsilon)$ can be generated as follows: sample $\hat{f}_1, \dots, \hat{f}_M \sim \mathbb{P}(\cdot \mid \mathsf{VS}(S_n))$, and compute

$$\widehat{R}_{n,m}(\varepsilon) = \frac{1}{M} \sum_{j=1}^{M} \mathbb{1}(\mathcal{E}_m(\hat{f}_j) \le \varepsilon).$$

Standard Gilvenko-Cantelli-type results can be used to guarantee that $\sup_\varepsilon |R_{n,m}(\varepsilon) - \widehat{R}_{n,m}(\varepsilon)| = O(\frac{1}{\sqrt{M}})$. Hence, assuming we have the ability to sample from $\mathbb{P}(\cdot \mid \mathsf{VS}(S_n))$, the distribution $R_{n,m}(\varepsilon)$ can be estimated to arbitrary precision.

For the remainder of this section, we show how we can generate samples $\hat{f} \sim \mathbb{P}(\cdot \mid \mathsf{VS}(S_n))$ for any function class of the form $\mathcal{F}_\phi = \{f(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}^\top \phi(\boldsymbol{x})) : \boldsymbol{w} \in \mathbb{R}^N\}$, where $\phi : \mathbb{R}^d \to \mathbb{R}^N$ is any mapping. In this paper, we will address the following important examples:

$$\begin{aligned} &\phi(\boldsymbol{x}) = \boldsymbol{x}, && \text{(linear classification)} \\ &\phi(\boldsymbol{x}) = \sigma(\mathbf{U}\boldsymbol{x}). && \text{(random features)} \end{aligned}$$

Notice that for these classes of functions, a probability measure $\mathbb{P}$ over $\mathcal{F}$ is simply a distribution over $\mathbb{R}^N$. Throughout this paper, we will assume that $\mathbb{P}$ is the uniform distribution on the sphere $\mathbb{S}^{N-1} = \{\boldsymbol{w} \in \mathbb{R}^N : \|\boldsymbol{w}\| = 1\}$. This choice is made so as to obtain results that are agnostic to the choice of optimization algorithm: since any reasonable measure on the sphere will be absolutely continuous with respect to $\mathbb{P}$, we do not expect our main conclusions to be qualitatively changed by choosing a different base distribution. For the sake of computation, it will be convenient to make use of the equivalence (up to scaling) of the uniform distribution with the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, which is a consequence of the spherical symmetry of the Gaussian.

Let us define the function

$$\mathcal{L}_n(\boldsymbol{w}) = \prod_{i=1}^{n} \mathbb{1}(y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) \ge 0), \tag{3.7}$$

and notice that $\mathbb{P}(\cdot \mid \mathsf{VS}(S_n)) = \mathbb{P}(\cdot \mid \mathcal{L}_n = 1)$. Therefore, we are interested in drawing samples from a linearly constrained Gaussian distribution. Fortunately, the recent work [GKH20a] developed the LIN-ESS algorithm (an extension of Elliptical Slice Sampling [MPDM10]) specifically for this purpose. Using traditional Monte Carlo methods, this task would be computationally infeasible in high dimensions, since if we naïvely drew samples from $\mathbb{P}$ and rejected those not lying in the domain $\{\mathcal{L}_n(\boldsymbol{w}) = 1\}$, then drawing a reasonable number of

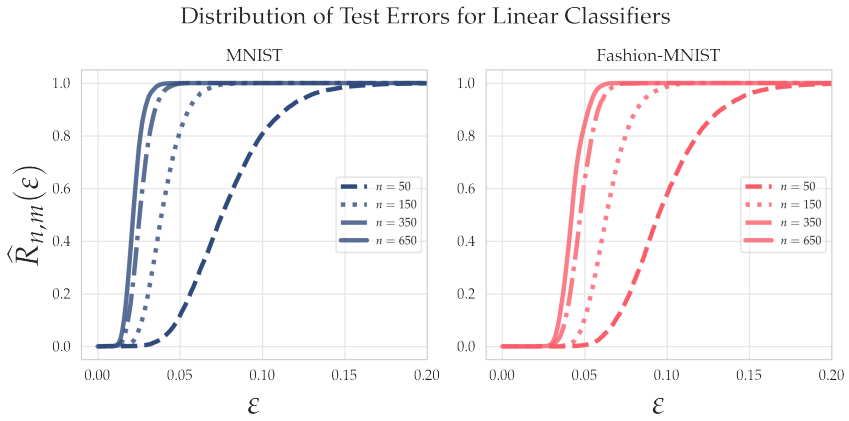


Figure 3.2: Estimated test error distribution $\widehat{R}_{n,m}(\varepsilon)$ for interpolating linear classifiers on the MNIST (0 vs 1) dataset (**blue**) and FASHION-MNIST (shirt vs pants) dataset (**red**).

samples could take an exponential amount of time. In contrast, LIN-ESS is able to exploit special properties of the linear constraints $y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) \geq 0$ to draw samples *without rejection*. In particular, in our setup, LIN-ESS can be used to generate samples $\widehat{\boldsymbol{w}}_1, \ldots, \widehat{\boldsymbol{w}}_M \sim \mathbb{P}(\cdot \mid \mathcal{L}_n = 1)$, which we can then use to compute the estimator $\widehat{R}_{n,m}(\varepsilon)$. As is the case with most MCMC algorithms, LIN-ESS is only guaranteed to produce independent samples from the posterior $\mathbb{P}(\cdot \mid \mathcal{L}_n = 1)$ asymptotically; we mitigate this issue in practice by using 1,000 warm-up samples, and keeping only every 10th sample thereafter.

3.3 Linear Classification

In this section, we compute the estimated test error distributions $\widehat{R}_{n,m}(\varepsilon)$ and $\widehat{R}_n(\varepsilon)$ on both real benchmark data as well as illustrative synthetic data, for the class $\mathcal{F}_{\text{LIN}} = \{f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x}) : \boldsymbol{w} \in \mathbb{R}^d\}$ of linear classifiers.

Evaluation on Image Datasets

For our first set of evaluations, we compute $\widehat{R}_{n,m}(\varepsilon)$ for high-dimensional image datasets used in modern machine learning. In particular, we focus on the MNIST and FASHION-MNIST datasets, which consist of images in $d = 784$ dimensional space. Thus, throughout this section, we only consider values of $n < 784$. Since we are specialized to the binary classification setting, we focus on the MNIST 0 vs 1 task, and on the shirt vs pants task for FASHION-MNIST. For both of these tasks, the data has been centered and scaled, so as to have mean 0 and variance 1.

In Figure 3.2, we plot the $\widehat{R}_{n,m}(\varepsilon)$ for various values of n. For each of the plots in this section, estimators $\widehat{R}_{n,m}(\varepsilon)$ are formed with $M = 10,000$ samples from $\mathbb{P}(\cdot \mid \mathcal{L}_n = 1)$ using the LIN-ESS algorithm, and they are evaluated on $m = 5000$ testing points.

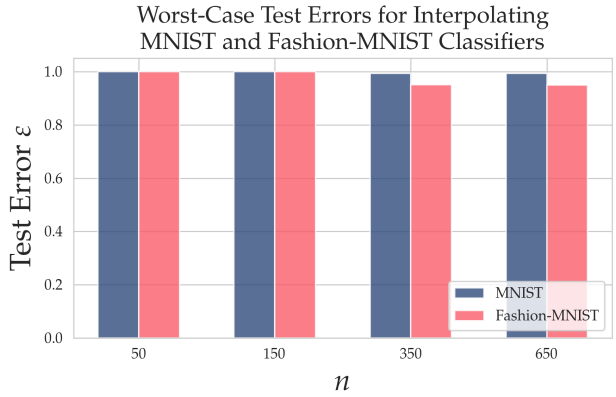


Figure 3.3: Test errors of interpolating classifiers with fit to n "good" training samples and $n_b = (d-1)-n$ "bad" training samples. The classifiers constructed here have extremely poor test set performance, in contrast to results shown in Figure 3.2.

Observation 1: Good classifiers are abundant. Our first observation is that, for reasonable n, most interpolating classifiers have good[1] test set performance. For example, for the MNIST dataset, we see that at $n = 350$, nearly 100% of the models that perfectly fit the training data achieve at least 95% ($\varepsilon = 0.05$) test accuracy. This indicates that, for this particular training set, bad classifiers (with error $> 5\%$) make up a set with very small measure. On the other hand, for the FASHION-MNIST task, only about 60% of classifiers perfectly fitting the training data get 95% test performance at $n = 350$ samples, but nearly 100% of such classifiers get 92% accuracy.

Observation 2: Existence of bad classifiers. A natural question that may arise out of these results is whether or not bad interpolating classifiers even exist for these tasks, at least for the parameter settings we consider. Here, we demonstrate a simple method for finding bad classifiers which, together with the previous results, shows that bad classifiers exist and constitute a tiny fraction of the version space. Given a dataset S_n, with $n < d$, we can append up to $n_b \leq (d-1)-n$ "bad" samples, to form a new dataset S'_n with $n' = n + n_b$ samples. Notice that any model $\boldsymbol{w} \in \mathsf{VS}(S'_n)$ must also belong to $\mathsf{VS}(S_n)$, since $\mathsf{VS}(S'_n) \subseteq \mathsf{VS}(S_n)$. Here, we construct $n_b = (d-1)-n$ "bad" points lying in the span of the set $\{-y_1\boldsymbol{x}_1, \ldots, -y_n\boldsymbol{x}_n\}$. In Figure 3.3, we plot the test error of interpolating classifiers constructed in this manner, fit using gradient descent with a logistic loss, for varying levels of n. We see that this method finds classifiers with test error that is nearly 1 for all values of n considered.

We are therefore left with an insightful contrast: in Figure 3.2, we observe that, for example, at $n = 350$, the set of interpolating MNIST classifiers with test accuracy $\geq 95\%$ comprise a set of measure essentially 1; while in Figure 3.3, we have demonstrated that there *exist* interpolating classifiers for this task with test accuracy nearly 0%. Thus, we see that the

[1] Of course, one could fit a model from a more complicated function class and obtain even better test performance.

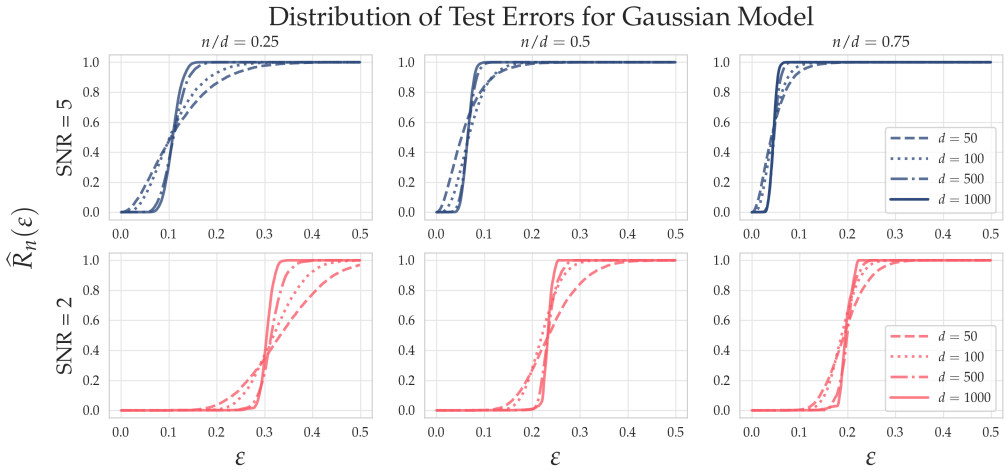


Figure 3.4: Plotting $\widehat{R}_n(\epsilon)$ for the Gaussian model (B.12) at various levels of d. **Blue** curves correspond to SNR = 5, **red** curves correspond to SNR = 2.

performance of the worst-case classifier gives basically no insight into the performance of the typical classifier, indicating that a uniform convergence-type analysis is not appropriate in this setting. This is also information that cannot be gleaned by looking at a summary statistic, like the *expected* test error of interpolating classifiers, i.e., $\mathbb{E}[\mathcal{E}_m(\boldsymbol{w}) \mid \mathsf{VS}(S_n)]$, alone—it is necessary to consider the full distribution.

Evaluation on Synthetic Datasets

For our next set of evaluations, we compute $R_n(\varepsilon)$ for synthetic data generated from the Gaussian mixture distribution

$$(\boldsymbol{x}, y) \sim \frac{1}{2}(N_+, 1) + \frac{1}{2}(N_-, -1)., \tag{3.8}$$

where $N_+ \sim \mathcal{N}(\mu, \Sigma)$, $N_- \sim \mathcal{N}(-\mu, \Sigma)$ and $\mu \in \mathbb{R}^d$, $\Sigma \in \mathcal{S}_+^d$. The purpose of this synthetic model is twofold. First, it allows us to demonstrate the ubiquity of the phenomena observed on the MNIST and FASHION-MNIST tasks. Second, it allows us to investigate the effect of varying the dimension d, which we could not do on the datasets studied in the previous section, as this was fixed at $d = 784$. This reveals that test errors begin to concentrate around a value ε^* as the dimension d increases.

For this model, we have that $y\boldsymbol{x} \sim \mathcal{N}(\mu, \Sigma)$, so we can characterize the set $\{\boldsymbol{w} : \mathcal{E}(\boldsymbol{w}) \le \varepsilon\}$ with the condition

$$\mathcal{E}(\boldsymbol{w}) \le \varepsilon \iff \frac{\boldsymbol{w}^\top \mu}{\sqrt{\boldsymbol{w}^\top \Sigma \boldsymbol{w}}} \ge -\Phi^{-1}(\varepsilon), \tag{3.9}$$

where $\Phi(\cdot)$ is the CDF of a $\mathcal{N}(0,1)$ distribution. Given a training set S_n and samples $\widehat{\boldsymbol{w}}_1, \ldots, \widehat{\boldsymbol{w}}_M \sim \mathbb{P}(\cdot \mid \mathsf{VS}(S_n))$, this expression allows us to compute an estimate $\widehat{R}_n(\varepsilon) = \frac{1}{M}\sum_{j=1}^M \mathbb{1}(\mathcal{E}(\widehat{\boldsymbol{w}}_j) \le \varepsilon)$ in a straightforward manner.

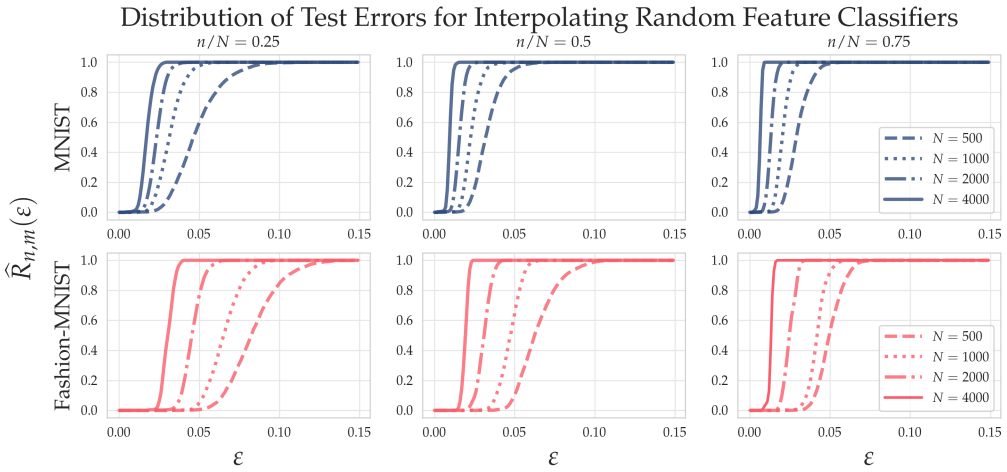


Figure 3.5: Plotting $\widehat{R}_{n,m}(\epsilon)$ for the random ReLU feature models on MNIST (0 vs 1) dataset (**blue**) and FASHION-MNIST (shirt vs pants) dataset (**red**).

As with many Gaussian models, the signal-to-noise ratio (SNR), which we define as $\sqrt{\mu^\top \Sigma^{-1} \mu}$ (or simply $\|\mu\|/\sigma$ when $\Sigma = \sigma^2 I$), controls much of the complexity of this task. In Figure 3.4, we plot $\widehat{R}_n(\varepsilon)$ for $d = 50, 100, 500, 1000$, and with SNR $= 2, 5$. For these experiments, we take $\Sigma = I$ and, to keep the SNR constant as we vary the dimension, we set $\mu = (\text{SNR}/\sqrt{d}, \ldots, \text{SNR}/\sqrt{d})^\top$.

Observation 3: Concentration at critical value ε^*. Our main observation here is the existence of a critical value ε^* around which test errors eventually concentrate. Indeed, we see in Figure 3.4 that as d grows, the distributions $R_n(\varepsilon)$ seem to approach the threshold function $\mathbb{1}(\varepsilon \geq \varepsilon^*)$ at a critical value ε^*, which depends on the aspect ratio $\alpha = n/d$. Therefore, in the large d regime, almost all interpolating classifiers have test error exactly ε^*, and so this critical value almost completely characterizes the distribution of test errors for interpolating classifiers. We also observe that this value is largely determined by the value of the SNR. In fact, we can derive a simple lower bound on the value of ε^*:

$$\varepsilon^* \geq \Phi(-\sqrt{\mu^\top \Sigma^{-1} \mu}). \tag{3.10}$$

This corresponds to the error of the optimal Bayes classifier $\boldsymbol{w}^\star = \Sigma^{-1}\mu$. In the next section, we observe a similar phenomenon for image classification tasks with random feature models.

3.4 Random ReLU Features

In this section, we consider the class of random ReLU feature classifiers $\mathcal{F}_{\text{RRF}} = \{f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^\top \phi(\boldsymbol{x})) : \boldsymbol{w} \in \mathbb{R}^N\}$, where $\phi(\boldsymbol{x}) = \sigma(\mathbf{U}\boldsymbol{x}) : \mathbb{R}^d \mapsto \mathbb{R}^N$. Here the rows $\mathbf{u}_1, \ldots, \mathbf{u}_N$ of $\mathbf{U}$ are drawn from the uniform distribution on the sphere $\mathbb{S}^{d-1}$ and $\sigma(z) = \max(z, 0)$ is the ReLU activation function. These can be viewed as one-layer ReLU networks with the weights of the first layer fixed, and they are known to enjoy universal approximation properties [SGT19].

The benefit in studying such a model is that we can examine the behavior of the test error distributions as the number of hidden features N grows large, with $\alpha = n/N$ fixed. This allows us to observe the critical value behavior seen in linear classification with the Gaussian model (B.12), but this time with the image datasets MNIST and FASHION-MNIST.

In Figure 3.5, we plot the test error distributions for interpolating random ReLU classifiers on the MNIST and FASHION-MNIST tasks, for various number of hidden features N and ratios $\alpha = n/N$. Our main observation from these experiments is that, similar to the Gaussian model, as the number of features N grows, the test errors begin to concentrate around values $\varepsilon^* \equiv \varepsilon^*(\alpha)$. Like in the Gaussian model, the critical value depends on i) the difficulty of the task (it is larger for FASHION-MNIST than for MNIST) and ii) the aspect ratio $\alpha = n/N$. This finding indicates that the concentration phenomenon observed in Section 3.3 is quite general, and holds for both real and synthetic datasets.

We remark that the same technique used in Section 3.3 demonstrates that very poor classifiers also exist for the random ReLU classification models, and hence again verifies that the worst-case analysis of test errors is inappropriate for these models and datasets.

3.5 Characterizing the Distribution of Test Errors in a Simple Model

In this section, we present a simple model, and we prove that it exhibits the main qualitative properties we observed in Sections 3.3 and 3.4.

A full mathematical characterization of $R_{n,m}(\varepsilon)$ and/or $R_n(\varepsilon)$ is a challenging task. To see why, let us define the random variables $\zeta_i = y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i)$ for $(\boldsymbol{x}_i, y_i) \in S_n$ and $\zeta_{n+h} = y_{n+h} \boldsymbol{w}^\top \phi(\boldsymbol{x}_{n+h})$ for $(\boldsymbol{x}_{n+h}, y_{n+h}) \in S_{\text{test}}$ (where we emphasize that the randomness is due to $\boldsymbol{w}$). Then, for example, the normalization term $\mathbb{P}(\mathsf{VS}(S_n))$ can be expressed as

$$\begin{aligned} \mathbb{P}(\mathsf{VS}(S_n)) &= \int \prod_{i=1}^{n} \mathbb{1}(y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) \geq 0) \mathbb{P}(\mathrm{d}\boldsymbol{w}) \\ &= \mathbb{P}(\zeta_1 \geq 0, \zeta_2 \geq 0, \ldots, \zeta_n \geq 0). \end{aligned} \tag{3.11}$$

That is, $\mathbb{P}(\mathsf{VS}(S_n))$ can be seen as an orthant probability under the distribution $\mathbb{P}$. When $\mathbb{P} = \mathcal{N}(\mathbf{0}, \mathbf{I})$, we find that $\zeta = (\zeta_1, \zeta_2, \ldots, \zeta_n) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}\mathbf{A}^\top)$, where $\mathbf{A}$ is the $n \times N$ matrix whose i^{th} row is $(y_i \phi(\boldsymbol{x}_i))^\top$ and whose $(i,j)^{\text{th}}$ entry is $y_i y_j \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$. Computing such a Gaussian orthant probability for a general covariance matrix is a classical problem, and explicit formulae for them are known only in dimensions ≤ 5 and in a few other special cases [DS55, Ste62, Abr64].

Hence, to present a model we can analyze, here we consider a simplified setting where the testing and training samples have a fixed positive correlation with each other, i.e., for fixed $\rho \in (0, 1]$,

$$(\mathbf{A}\mathbf{A}^\top)_{ij} = y_i y_j \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j) = \rho, \tag{3.12}$$

for each pair of indices $i \neq j$ in $S_n \cup S_{\text{test}}$ (where here we assume $\phi(\boldsymbol{x}_i)$ are normalized to have unit ℓ^2 norm, without loss of generality).[2] Under this assumption, we can leverage implicit expressions for the normalizing term $\mathbb{P}(\mathsf{VS}(S_n))$, which makes the problem more amenable to analysis.

We remark that to derive asymptotically valid expressions for $R_n(\varepsilon)$ and $R_{n,m}(\varepsilon)$, one may be tempted to approximate (3.11) using off-the-shelf techniques for approximating high-dimensional integrals, e.g., Laplace's method. However, there are a number of pitfalls with this approach. First, it is difficult to quantify the approximation errors, and results that do exist are not precise enough for our purposes. Second, certain conditions for Laplace's method or other standard integral expansions do not hold in our setting.[3] Nevertheless, we can leverage special properties of the Gaussian distribution and quantile functions to prove several non-trivial results. Henceforth, for sequences $\{a_n\}$ and $\{b_n\}$, the notation $a_n \sim b_n$ means $a_n = b_n(1 + o(1))$ as $n \to \infty$.[4]

Our first result considers the setting of a single testing point $(\boldsymbol{x}_{n+1}, y_{n+1})$, and it demonstrates the effect of a larger correlation ρ on the probability of correctly classifying a new test point. Furthermore, it shows that, at least for this simple setting, we can expect the probability of correctly classifying a testing point to converge to 1 at a $O(1/n)$ rate.

Theorem 3.1. *Suppose we have a single testing point $(\boldsymbol{x}_{n+1}, y_{n+1})$, which together with the training data satisfies the correlation structure (3.12). Then, as $n\rho \to \infty$,*

$$\mathbb{P}(y_{n+1} = \operatorname{sign}(\boldsymbol{w}^\top \phi(\boldsymbol{x}_{n+1})) \mid \mathsf{VS}(S_n)) \sim 1 - \frac{1-\rho}{n\rho}. \tag{3.13}$$

The proof of Theorem 3.1 relies mainly on a new asymptotic formula for the orthant probability of equicorrelated Gaussian random variables. To the best of our knowledge, this is the first of its kind, and it may be of independent interest. We state this result below in the following Lemma.

Lemma 3. *Let $\rho \in [0, 1)$ and $(X_1, ..., X_n) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma}_{ij} = \rho$ for $i \neq j$ and $\Sigma_{ii} = 1$ for all i. Then as $n\rho \to \infty$,*

$$\mathbb{P}(X_1 \geq 0, X_2 \geq 0, \ldots, X_n \geq 0) \sim \sqrt{\frac{1-\rho}{\rho}}\,\Gamma\left(\frac{1-\rho}{\rho}\right)(4\pi \log(n))^{\frac{1}{2}(\frac{1-\rho}{\rho}-1)} n^{-\frac{1-\rho}{\rho}}.$$

Theorem 3.1 then follows by carefully evaluating the ratio of the above expression at $n+1$ and n.

Before stating our next result, we provide a formal definition of a critical value ε^* which we will reference therein.

[2] By correlation between data points, we mean $y_i y_j \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$ for $i \neq j$.

[3] For example, the maximum of the function in the exponent of the integrand occurs at infinity.

[4] That is, it should not be confused with "has the probability distribution of" which uses the same notation.

Definition 3.2. We say that ε^* is a critical value if, for each $c > 0$, $R_n(\varepsilon^* - c) = 0$ and $R_n(\varepsilon^* + c) \to 1$ as $n \to \infty$.

Our next result provides a connection between the critical value ε^*, the number of training samples, and the correlation ρ.

Theorem 3.2. *Suppose the testing and training data satisfies the correlation structure (3.12). Let U be a gamma random variable with shape and scale parameters $(1-\rho)/\rho$ and 1, respectively, i.e., $U \sim \text{Gamma}(\frac{1-\rho}{\rho}, 1)$. Then, as $n\rho \to \infty$,*

$$R_n(\varepsilon) \sim \mathbb{P}(U \le n\varepsilon). \tag{3.14}$$

In particular, as $n\rho \to \infty$,

$$\varepsilon^* = \frac{1-\rho}{n\rho} \tag{3.15}$$

is a critical value.

In this simple setting, n and ρ completely determine the distribution $R_n(\varepsilon)$: if ρ is close to 1, then the data points are nearly parallel, and we will have that the test errors sharply concentrate around the critical value ε^*, even for n small. Of course, in practice, there will be a more subtle and complicated relationship between the correlations and the full distribution $R_n(\varepsilon)$, which will likely be difficult to characterize precisely. Nonetheless, we believe that it may be possible to prove concentration in the general case, without explicitly characterizing the full distribution $R_n(\varepsilon)$. This is captured by the following conjecture.

Conjecture 3.1. *For any model class $\mathcal{F}_\phi$, datasets S_n, testing distribution $\text{Pr}_{\boldsymbol{x},y}$ (each potentially satisfying some regularity conditions) and scaling $0 < \alpha < 1$, there exists a critical value $\varepsilon^*(\alpha)$ such that $\lim_{n,N\to\infty,n/N\to\alpha} R_n(\varepsilon) = \mathbb{1}(\varepsilon \ge \varepsilon^*(\alpha))$ almost surely.*

Theorem 3.2 provides such a result in the case when the data is equicorrelated. Previous work using the statistical mechanics framework also prove similar results under different simplifying assumptions, namely when the features $\boldsymbol{x}_{ik} \overset{\text{i.i.d.}}{\sim} \text{Unif}(\{-1, 1\})$, $k = 1, \dots, d$, and the labels y_i are generated via a teacher model $\boldsymbol{w}_\star$ s.t. $y_i = \text{sign}(\boldsymbol{w}_\star^\top \boldsymbol{x}_i)$ (see, e.g., Chapter 2 of [EVdB01]). However, these results typically only focus on the $n > d$ case, which is less relevant to the modern machine learning regime.

3.6 Discussion and Conclusion

In this chapter, we built on previous literature on the statistical mechanics of learning to develop a framework to study the *typical* test error of a classifier, and we propose this as an alternative to the more standard uniform convergence approach. We formally define the full distribution of test errors among interpolating classifiers and introduce a method to

compute this distribution accurately on real datasets. One of the most important findings of our investigation is that, given a particular training and testing setup, there exists a critical value ε^* around which almost all interpolating classifiers' test errors eventually concentrate. This will not come as a surprise to the statistical physicist: such typical values commonly appear in physical systems. However, as we have demonstrated, this critical value can differ significantly from the error $\varepsilon_{\text{unif}}$, which one would obtain via a uniform convergence analysis, especially in the interpolating/over-parameterized regime, and which may be more familiar to the machine learner.

Our results should motivate further research into alternatives to the uniform convergence framework, either through the lens of statistical physics or some other (likely related) perspective, and they should ultimately help resolve questions surrounding the good performance of over-parameterized machine learning models. As a first step, we state a few potential directions for future work building off of the results presented here.

More general function classes. While encompassing many models of interest, the function classes $\mathcal{F}_\phi$ of course do not include general neural network architectures. In this paper, we studied random feature models, which can be interpreted as neural networks with internal weights fixed at a random initialization. Another interesting setting which may be more tractable to study would be that of linearized networks of the form

$$f(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}^\top \nabla F(\boldsymbol{x}; \boldsymbol{w}_0)) \tag{3.16}$$

where F is an arbitrary neural network with random initialization $\boldsymbol{w}_0$. A variety of results have shown that these models coincide with neural networks in the large-width limit via the neural tangent kernel [JGH18, ADH+19]. While our approach would, in theory, work out-of-the-box for these models, in practice, these involve a very large number of features (approximately $O(LN^2)$, where L is the number of layers, and N is the width of each layer). We found that even with the LIN-ESS algorithm, sampling from $\mathbb{P}(\cdot \mid \mathsf{VS}(S_n))$ was impractical for these models. However, developing other methods for computation in this setting could yield interesting insights into the advantages (and disadvantages) of various network architectures.

Beyond the interpolating regime. The motivation for our studying interpolating classifiers comprising the version space $\mathsf{VS}(S_n)$ was previous work in the statistical mechanics literature, as well as the well-known worst-case results for these models given by, e.g., Vapnik–Chervonenkis theory. However, this is not the only method one could use to study the distribution of test errors. A promising alternative would be to consider the distribution over weights $\boldsymbol{w}$ induced by some optimization algorithm, such as stochastic gradient descent (SGD). Indeed, previous work has shown that under various assumptions, SGD produces a Gaussian stationary distribution over weights $\boldsymbol{w}$ [MHB17]. Under other (probably more realistic) assumptions, it leads to heavy-tailed structure in the weights [HM20, GSZ20]. An intriguing direction for future work would be to study the distribution over test errors $\mathcal{E}(\boldsymbol{w})$ induced by such a stationary distribution. It is possible that this may even simplify the

theoretical investigation: whereas we studied weights drawn from $\mathbb{P}(\cdot \mid \mathsf{VS}(S_n))$ (a rather complicated distribution), it may be easier to study weights drawn from a Gaussian (or some other tractable) distribution.

Acknowledgments

MM would like to acknowledge DARPA, NSF, and ONR for providing partial support of this work. JK would like to acknowledge funding from NSF DMS-1915932 and NSF HDR TRIPODS DATA-INSPIRE DCCF-1934924. We also thank the authors of [GKH20a] for sharing their implementation of the LIN-ESS algorithm.

Chapter 4

Generalization in the Best Case: Estimating the Bayes Error Rate

The contents of this chapter are partially based on the paper "Evaluating State-of-the-Art Classification Models Against Bayes Optimality", appearing in NeurIPS 2021, co-authored with Huan Wang, Lav R. Varshney, Caiming Xiong and Richard Socher [TWV+21].

4.1 Introduction

Benchmark datasets and leaderboards are prevalent in machine learning's common task framework [Don19]; however, this approach inherently relies on relative measures of improvement. It may therefore be insightful to be able to evaluate state-of-the-art (SOTA) performance against the optimal performance theoretically achievable by *any* model [VKS19]. For supervised classification tasks, this optimal performance is captured by the Bayes error rate which, were it tractable, would not only give absolute benchmarks, rather than just comparing to previous classifiers, but also insights into dataset hardness [HB02, ZWN+20] and which gaps between SOTA and optimal the community may fruitfully try to close.

Suppose we have data generated as $(X, Y) \sim p$, where $X \in \mathbb{R}^d$, $Y \in \mathcal{Y} = \{1, \dots, K\}$ is a label and p is a distribution over $\mathbb{R}^d \times \mathcal{Y}$. The **Bayes classifier** is the rule which assigns a label to an observation $\boldsymbol{x}$ via

$$y = C_{\text{Bayes}}(\boldsymbol{x}) := \arg\max_{j \in \mathcal{Y}} p(Y = j \mid X = \boldsymbol{x}). \tag{4.1}$$

The **Bayes error** is simply the probability that the Bayes classifier predicts incorrectly:

$$\mathcal{E}_{\text{Bayes}}(p) := p(C_{\text{Bayes}}(X) \neq Y). \tag{4.2}$$

The Bayes classifier is optimal, in the sense it minimizes $p(C(X) \neq Y)$ over all possible classifiers $C : \mathbb{R}^d \to \mathcal{Y}$. Therefore, the Bayes error is a natural measure of 'hardness' of a particular learning task. Knowing $\mathcal{E}_{\text{Bayes}}$ should interest practitioners: it gives a natural benchmark for the performance of any trained classifier. In particular, in the era of deep learning, where vast amounts of resources are expended to develop improved models and architectures, it is of great interest to know whether it is even theoretically possible to substantially lower the test errors of state-of-the-art models, cf. [CF07].

Of course, obtaining the exact Bayes error will almost always be intractable for real-world classification tasks, as it requires full knowledge of the distribution p. A variety of works have developed estimators for the Bayes error, either based on upper and/or lower bounds [BWHS16] or exploiting exact representations of the Bayes error [NXH19, Nie14]. Most of these bounds and/or representations are in terms of some type of *distance* or *divergence* between the class conditional distributions,

$$p_j(\boldsymbol{x}) \coloneqq p(X = \boldsymbol{x} \mid Y = j), \tag{4.3}$$

and/or the marginal label distributions $\pi_j \coloneqq p(Y = j)$. For example, there are exact representations of the Bayes error in terms of a particular f-divergence [NXH19], and in a special case in terms of the total variation distance [Nie14]. More generally, there are lower and upper bounds known for the Bayes error in terms of the Bhattacharyya distance [BWHS16, Nie14], various f-divergences [MH14], the Henze-Penrose (HP) divergence [MSGH18, MHD15], as well as others. Once one has chosen a desired representation and/or bound in terms of some divergence, estimating the Bayes error reduces to the estimation of this divergence. Unfortunately, for high-dimensional datasets, this estimation is highly inefficient. For example, most estimators of f-divergences rely on some type of ε-ball approach, which requires a number of samples on the order of $(1/\varepsilon)^d$ in d dimensions [NXH19, PXS11]. In particular, for large benchmark image datasets used in deep learning, this approach is inadequate to obtain meaningful results.

Here, we take a different approach: rather than computing an approximate Bayes error of the exact distribution (which, as we argue above, is intractable in high dimensions), we propose to compute the *exact Bayes error of an approximate distribution*. The basics of our approach are as follows.

- We show that when the class-conditional distributions are Gaussian $q_j(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma})$, we can efficiently compute the Bayes error using a variant of Holmes-Diaconis-Ross integration proposed in [GKH20b].
- We use normalizing flows [PNR+21, KD18, FJGZ20] to fit approximate distributions $\hat{p}_j(\boldsymbol{x})$, by representing the original features as $\boldsymbol{x} = T(\boldsymbol{z})$ for a learned invertible transformation T, where $\boldsymbol{z} \sim q_j(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma})$, for learned parameters $\boldsymbol{\mu}_j, \boldsymbol{\Sigma}$.

- Lastly, we prove in Proposition 4.1 that the Bayes error is invariant under invertible transformation of the features, so computing the Bayes error of the approximants $\hat{p}_j(\boldsymbol{x})$ can be done *exactly* by computing it for the Gaussians $q_j(\boldsymbol{z})$.

Moreover, we show that by varying the *temperature* of a single flow model, we can obtain an entire class of distributions with varying Bayes errors. This recipe allows us to compute the Bayes error of a large variety of distributions, which we use to conduct a thorough empirical investigation of a benchmark datasets and SOTA models, producing a library of trained flow models in the process. By generating synthetic versions of standard benchmark datasets with known Bayes errors, and training them on SOTA deep learning architectures, we are able to assess how well these models perform compared to the Bayes error, and find that in some cases they indeed achieve errors very near optimal. We then investigate our Bayes error estimates as a measure of objective difficulty of benchmark classification tasks, and produce a ranking of these datasets based on their approximate Bayes errors.

We should note one additional point before proceeding. In general the hardness of classification tasks can be decomposed into two relatively independent components: i) hardness caused by the lack of samples, and ii) hardness caused by the internal data distribution p. The focus of this work is about the latter: the hardness caused by p. Indeed, even if the Bayes error of a particular task is known to be a particular value $\mathcal{E}_{\text{Bayes}}$, it may be highly unlikely that this error is achievable given a model trained on only N samples from p. The problem of finding the minimal error achievable from a given dataset of size N has been called the optimal experimental design problem [Rit00]. While this is not the focus of the present work, an interesting direction for future work is to use our methodology to investigate the relationship between N and the SOTA-Bayes error gap.

4.2 Computing the Bayes error of Gaussian conditional distributions

Throughout this section, we assume the class conditional distributions are Gaussian: $q_j(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$. In the simplest case of binary classification with $K = 2$ classes, equal covariance $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$, and equal marginals $\pi_1 = \pi_2 = \frac{1}{2}$, the Bayes error can be computed analytically in terms of the CDF of the standard Gaussian distribution, $\Phi(\cdot)$, as:

$$\mathcal{E}_{\text{Bayes}} = 1 - \Phi\left(\tfrac{1}{2}\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\|_2\right). \tag{4.4}$$

When $K > 2$ and/or the covariances are different between classes, there is no closed-form expression for the Bayes error. Instead, we work from the following representation:

$$\mathcal{E}_{\text{Bayes}} = 1 - \sum_{k=1}^{K} \pi_k \int \prod_{j \neq k} \mathbb{1}(q_j(\boldsymbol{z}) < q_k(\boldsymbol{z})) \mathcal{N}(d\boldsymbol{z}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{4.5}$$

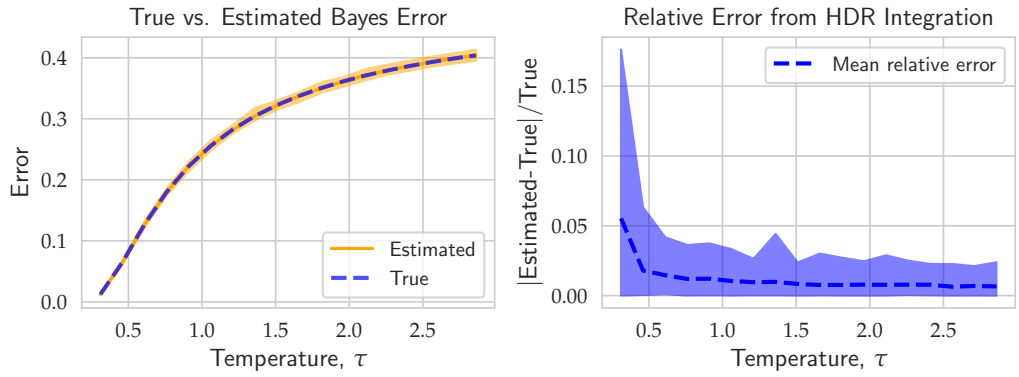


Figure 4.1: We compare the Bayes error estimated using HDR integration [GKH20b] with the exact error in the binary classification with equal covariance case given in (4.4). On the right we show the relative error from numerical integration. Shaded region on both plots shows the range over 100 runs. We see the integration routine gives highly accurate estimates. Here we use dimension $d = 784$, and take $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ to be randomly drawn unit vectors, and $\boldsymbol{\Sigma} = \tau^2 \mathbf{I}$ where τ is the temperature.

In the general case, the constraints $q_j(\boldsymbol{z}) < q_k(\boldsymbol{z})$ are quadratic, with $q_j(\boldsymbol{z}) < q_k(\boldsymbol{z})$ occurring if and only if:

$$-(\boldsymbol{z} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{z} - \boldsymbol{\mu}_j) - \log\det \boldsymbol{\Sigma}_j < -(\boldsymbol{z} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{z} - \boldsymbol{\mu}_k) - \log\det \boldsymbol{\Sigma}_k. \tag{4.6}$$

As far as we know, there is no efficient numerical integration scheme for computing Gaussian integrals under general quadratic constraints of this form. However, if we further assume the covariances are equal, $\boldsymbol{\Sigma}_j = \boldsymbol{\Sigma}$ for all $j = 1, \ldots, K$, then the constraint (4.6) becomes linear, of the form

$$\mathbf{a}_{jk}^\top \boldsymbol{z} + b_{jk} > 0, \tag{4.7}$$

where $\mathbf{a}_{jk} := 2\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_j - \boldsymbol{\mu}_k)$ and $b_{jk} := \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \boldsymbol{\mu}_j^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j$. Thus expression (4.5) can be written as

$$\mathcal{E}_{\text{Bayes}} = 1 - \sum_{k=1}^{K} \pi_k \int \prod_{j \neq k} \mathbb{1}(\mathbf{a}_{jk}^\top \boldsymbol{z} + b_{jk} > 0) \mathcal{N}(d\boldsymbol{z}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}). \tag{4.8}$$

Computing integrals of this form is precisely the topic of the recent paper [GKH20b], which exploited the particular form of the linear constraints and the Gaussian distribution to develop an efficient integration scheme using a variant of the Holmes-Diaconis-Ross method [DH95]. This method is highly efficient, even in high dimensions[1]. In Figure 4.1, we show the estimated Bayes error using this method on a synthetic binary classification problem in

[1]Note that the integrals appearing in (4.8) are really only $(K-1)$-dimensional integrals, since they only depend on $K-1$ variables of the form $\mathbf{a}_{jk}^\top \boldsymbol{x} + b_{jk}$.

$d = 784$ dimensions, where we can use closed-form expression (4.4) to measure the accuracy of the integration. As we can see, it is highly accurate.

This method immediately allows us to investigate the behavior of large neural network models on high-dimensional synthetic datasets with class conditional distributions $q_j(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma})$. However, in the next section, we will see that we can use normalizing flows to estimate the Bayes error of real-world datasets as well.

4.3 Normalizing flows and invariance of the Bayes error

Normalizing flows are a powerful technique for modeling high-dimensional distributions [PNR+21]. The main idea is to represent the random variable $\boldsymbol{x}$ as a transformation T_ϕ (parameterized by ϕ) of a vector $\boldsymbol{z}$ sampled from some, usually simple, base distribution $q(\boldsymbol{z}; \psi)$ (parameterized by ψ), i.e.

$$\boldsymbol{x} = T_\phi(\boldsymbol{z}) \qquad \text{where} \qquad \boldsymbol{z} \sim q(\boldsymbol{z}; \psi). \tag{4.9}$$

When the transformation T_ϕ is invertible, we can obtain the exact likelihood of $\boldsymbol{x}$ using a standard change of variable formula:

$$\hat{p}(\boldsymbol{x}; \theta) = q(T_\phi^{-1}(\boldsymbol{x}); \psi) \left| \det J_{T_\phi}(T_\phi^{-1}(\boldsymbol{x})) \right|^{-1}, \tag{4.10}$$

where $\theta = (\phi, \psi)$ and J_{T_ϕ} is the Jacobian of the transformation T_ϕ. The parameters θ can be optimized, for example, using the KL divergence:

$$\mathcal{L}(\theta) = D_{\mathrm{KL}}(p(\boldsymbol{x}) \parallel \hat{p}(\boldsymbol{x}; \theta)) \approx -\frac{1}{N} \sum_{i=1}^{N} \log q(T_\phi^{-1}(\boldsymbol{x}_i), \psi) + \log \left| \det J_{T_\phi^{-1}}(\boldsymbol{x}_i) \right| + \text{const.} \tag{4.11}$$

This approach is easily extended to the case of learning class-conditional distributions by parameterizing multiple base distributions $q_j(\boldsymbol{z}; \psi_j)$ and computing

$$\hat{p}_j(\boldsymbol{x}; \theta) = q_j(T_\phi^{-1}(\boldsymbol{x}); \psi_j) \left| \det J_{T_\phi}(T_\phi^{-1}(\boldsymbol{x})) \right|^{-1}. \tag{4.12}$$

For example, we can take $q_j(\boldsymbol{z}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma})$, where we fit the parameters $\boldsymbol{\mu}_j, \boldsymbol{\Sigma}$ during training. This is commonly done to learn class-conditional distributions, e.g. [KD18]. This is the approach we take in the present work. In practice, the invertible transformation T_ϕ is parameterized as a neural network, though special care must be taken to ensure the neural network is invertible and has a tractable Jacobian determinant. Here, we use the Glow architecture [KD18] throughout our experiments, as detailed in Section 4.4.

Invariance of the Bayes Error

Normalizing flow models are particularly convenient for our purposes, since we can prove the Bayes error is invariant under invertible transformation. This is formalized as follows.

Proposition 4.1. *Let $(X,Y) \sim p$, $X \in \mathbb{R}^d, Y \in \mathcal{Y} = \{1,\dots,K\}$, and let $\mathcal{E}_{Bayes}(p)$ be the associated Bayes error of this distribution. Let $T : \mathbb{R}^d \to \mathbb{R}^d$ be an invertible map and denote q the associated joint distribution of $Z = T(X)$ and Y. Then $\mathcal{E}_{Bayes}(p) = \mathcal{E}_{Bayes}(q)$.*

Proof. For convenience, denote $|\mathbf{A}|$ as the absolute value determinant of a matrix $\mathbf{A}$. Using the representation derived in [NXH19], we can write the Bayes error as

$$\mathcal{E}_{\text{Bayes}}(p) = 1 - \pi_1 - \sum_{k=2}^{K} \int \max\left(0, \pi_k - \max_{1\le i\le k-1} \pi_i \frac{p_i(\boldsymbol{x})}{p_k(\boldsymbol{x})}\right) p_k(\boldsymbol{x}) d\boldsymbol{x}. \tag{4.13}$$

Then if $\boldsymbol{z} = T(\boldsymbol{x})$, we have that $q_k(\boldsymbol{z}) = p_k(T(\boldsymbol{z}))|J_T(\boldsymbol{z})|$, and $d\boldsymbol{x} = |J_{T^{-1}}(\boldsymbol{z})|d\boldsymbol{z}$. Hence

$$\begin{aligned}
\mathcal{E}_{\text{Bayes}}(p) &= 1 - \pi_1 - \sum_{k=2}^{K} \int \max\left(0, \pi_k - \max_{1\le i\le k-1} \pi_i \frac{p_i(\boldsymbol{x})}{p_k(\boldsymbol{x})}\right) p_k(\boldsymbol{x}) d\boldsymbol{x} \\
&= 1 - \pi_1 - \sum_{k=2}^{K} \int \max\left(0, \pi_k - \max_{1\le i\le k-1} \pi_i \frac{q_i(\boldsymbol{z})|J_T(\boldsymbol{z})|}{q_k(\boldsymbol{z})|J_T(\boldsymbol{z})|}\right) q_k(\boldsymbol{z})|J_T(\boldsymbol{z})||J_{T^{-1}}|(\boldsymbol{z}) d\boldsymbol{z}.
\end{aligned}$$

By the Inverse Function Theorem, $|J_{T^{-1}}(\boldsymbol{z})| = |J_T(\boldsymbol{z})|^{-1}$, and so we get

$$\begin{aligned}
\mathcal{E}_{\text{Bayes}}(p) &= 1 - \pi_1 - \sum_{k=2}^{K} \int \max\left(0, \pi_k - \max_{1\le i\le k-1} \pi_i \frac{q_i(\boldsymbol{z})|J_T(\boldsymbol{z})|}{q_k(\boldsymbol{z})|J_T(\boldsymbol{z})|}\right) q_k(\boldsymbol{z})|J_T(\boldsymbol{z})||J_T(\boldsymbol{z})|^{-1} d\boldsymbol{z} \\
&= 1 - \pi_1 - \sum_{k=2}^{K} \int \max\left(0, \pi_k - \max_{1\le i\le k-1} \pi_i \frac{q_i(\boldsymbol{z})}{q_k(\boldsymbol{z})}\right) q_k(\boldsymbol{z}) d\boldsymbol{z} \\
&= \mathcal{E}_{\text{Bayes}}(q),
\end{aligned}$$

which completes the proof. □

This result means that we can compute the *exact* Bayes error of the approximate distributions $\hat{p}_j(\boldsymbol{x};\theta)$ using the methods introduced in Section 4.2 with the Gaussian conditionals $q_j(\boldsymbol{z};\boldsymbol{\mu}_j,\boldsymbol{\Sigma})$. If in addition the flow model $\hat{p}_j(\boldsymbol{x};\theta)$ is a good a approximation for the true class-conditional distribution $p_j(\boldsymbol{x})$, then we expect to obtain a good estimate for the true Bayes error. In what follows, we will see examples both of when this is and is not the case.

Varying the Bayes error using temperature

An important aspect of the normalizing flow approach is that we can in fact generate a whole family of distributions from a single flow model. To do this, we can vary the *temperature* τ of the model by multiplying the covariance $\boldsymbol{\Sigma}$ of the base distribution by τ^2 to get $q_{j,\tau}(\boldsymbol{z}) := \mathcal{N}(\boldsymbol{z};\boldsymbol{\mu}_j,\tau^2\boldsymbol{\Sigma})$. The same invertible map T_ϕ induces new conditional distributions,

$$\hat{p}_{j,\tau}(\boldsymbol{x};\theta) = q_{j,\tau}(T_\phi^{-1}(\boldsymbol{x});\psi_j)\left|\det J_{T_\phi}(T_\theta^{-1}(\boldsymbol{x}))\right|^{-1}, \tag{4.14}$$

as well as the associated joint distribution $\hat{p}_\tau(y = j, \boldsymbol{x}; \theta) = \pi_j \hat{p}_{j,\tau}(\boldsymbol{x}; \theta)$.

It can easily be seen that the Bayes error of $\hat{p}_\tau$ is increasing in τ.

Proposition 4.2. *The Bayes error of flow models is monotonically increasing in τ. That is, for $0 < \tau \leq \tau'$, we have that $\mathcal{E}_{Bayes}(\hat{p}_\tau) \leq \mathcal{E}_{Bayes}(\hat{p}_{\tau'})$.*

This fact means that we can easily generate datasets of varying difficulty by changing the temperature τ. For example, in Figure 4.2 we show samples generated by a flow model (see Section 4.4 for implementation details) trained on the Fashion-MNIST dataset at various values of temperature and the associated Bayes error. As $\tau \to 0^+$, the distribution $\hat{p}_{j,\tau}$ concentrate on the mode of the distributions $\hat{p}_j$, making the classification tasks easy, whereas when τ gets large, the distributions $\hat{p}_{j,\tau}$ become more uniform, making classification more challenging. In practice, this can be used to generate datasets with almost arbitrary Bayes error: for any prescribed error ε in the range of the map $\tau \mapsto \mathcal{E}_{\text{Bayes}}(\hat{p}_\tau)$, we can numerically invert this map to find τ for which $\mathcal{E}_{\text{Bayes}}(\hat{p}_\tau) = \varepsilon$.

4.4 Empirical investigation

Setup

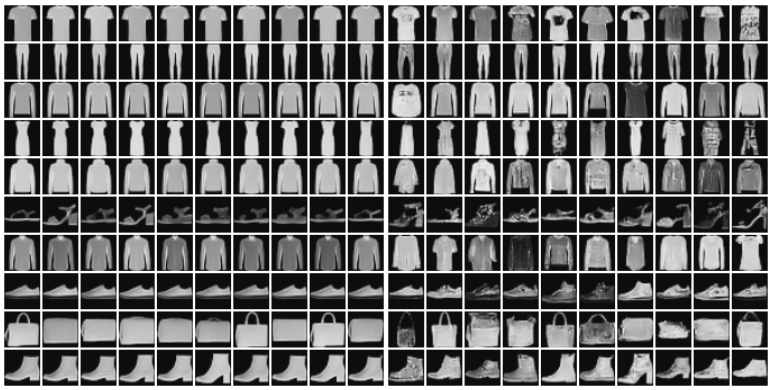
Datasets and data preparation. We train flow models[2] on a wide variety of standard benchmark datasets: MNIST [LBBH98], Extended MNIST (EMNIST) [CATvS17], Fashion MNIST [XRV17], CIFAR-10 [Kri09], CIFAR-100 [Kri09], SVHN [NWC+11], and Kuzushiji-MNIST [CBIK+18]. The EMNIST dataset has several different splits, which include splits by digits, letters, merge, class, and balanced. The images in MNIST, Fashion-MNIST, EMNIST, and Kuzushiji-MNIST are padded to 32-by-32 pixels.[3]

We remark that we observe our Bayes error estimator runs efficiently when the input is of dimension 32-by-32-by-3. However it is in general highly memory intensive to run the HDR integration routine on significantly larger datasets, e.g. when the input size grows to 64-by-64-by-3. As a consequence, in our experiments we only work on datasets of dimension no larger than 32-by-32-by-3.

Modeling and training. The normalizing flow model we use in our experiments is a pytorch implementation [Glo] of Glow [KD18]. In all our the experiments, affine coupling layers are used, the number of steps of the flow in each level $K = 16$, the number of levels $L = 3$, and number of channels in hidden layers $C = 512$.

[2] Code can be found at https://github.com/salesforce/DataHardness.

[3] Glow implementation requires the input dimension to be power of 2.



(a) τ=0.2, $\mathcal{E}_{\text{Bayes}}$ =1.11e-16 (b) τ=1.0, $\mathcal{E}_{\text{Bayes}}$ =3.36e-2

(c) τ=1.4, $\mathcal{E}_{\text{Bayes}}$ =1.07e-1 (d) τ=3.0, $\mathcal{E}_{\text{Bayes}}$ =4.06e-1

Figure 4.2: Generated Fashion-MNIST Samples with Different Temperatures

During training, we minimize the Negative Log Likelihood Loss (NLL)

$$\text{NLL}(\{\boldsymbol{x}_i, y_i\}) = -\frac{1}{N}\sum_{i=1}^{N}\left(\log p_{y_i}(\boldsymbol{x}_i;\theta) + \log \pi_{y_i}\right). \tag{4.15}$$

As suggested in [KD18], we also add a classification loss to predict the class labels from the second-to-last layer of the encoder with a weight of λ. During the experiments we traversed configurations with $\lambda = \{0.01, 0.1, 1.0, 10\}$, and report the numbers produced by the model with the smallest NLL loss on the test set. Note here even though we add the classification loss in the objective as a regularizer, the model is selected based on the smallest NLL loss in the test set instead of the classification loss or the total loss. The training and evaluation are done on a workstation with 2 NVIDIA V100 GPUs.

Evaluating SOTA models against generated datasets

In this section, we use our trained flow models to generate synthetic versions of standard benchmark datasets, for which the Bayes error is known exactly. In particular, we generate

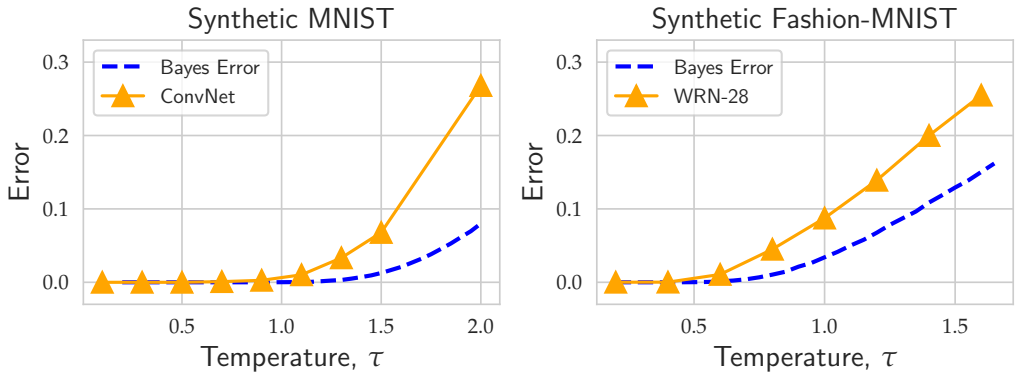


Figure 4.3: Test errors of synthetic versions of MNIST and Fashion-MNIST, generated at various temperatures, and their corresponding Bayes error. Here we used 60,000 training samples, and 10,000 testing samples, to mimic the original datasets. The model used in Fashion-MNIST was a Wide-ResNet-28-10, which attains nearly start of the accuracy on the original Fashion-MNIST dataset [ZZK+20]. The model used in MNIST is a popular ConvNet [Con].

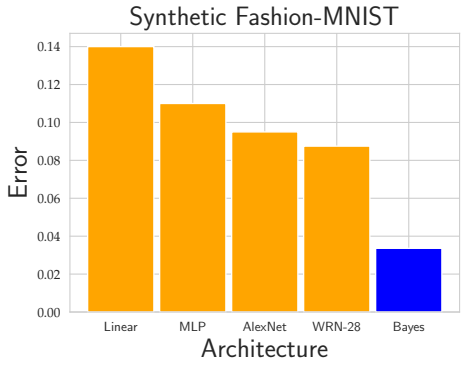


Figure 4.4: Errors of various model architectures (from old to modern) on a Synthetic Fashion-MNIST dataset ($\tau = 1$). We can see that for this task, while accuracy has improved with modern models, there is still a substantial gap between the SOTA and Bayes optimal.

synthetic versions of the MNIST and Fashion-MNIST datasets at varying temperatures. As we saw in Section 4.3, varying the temperature allows us to generate datasets with different difficulty. Here, we train a Wide-ResNet-28-10 model (i.e. a ResNet with depth 28 and width multiple 10) [ZK16, Wid] on these datasets, and compare the test error to the exact Bayes error for these problems. This Wide-ResNet model (together with appropriate data augmentation) attains nearly state-of-the-art accuracy on the original Fashion-MNIST dataset [ZZK+20], and so we expect that our results here reflect roughly the best accuracy presently attainable on these synthetic datasets as well. To make the comparison fair, we use a training set size of 60,000 to mimic the size of the original MNIST series of datasets.

The Bayes errors as well as the test errors achieved by the Wide-ResNet or ConvNet models are shown in Figure 4.3. As one would expect, the errors of trained models increase with temperature. It can be observed that Wide-ResNet and ConvNet are able to achieve close-to-optimal performance when the dataset is relatively easy, e.g., $\tau < 1$ for MNIST and $\tau < 0.5$ for Fashion-MNIST. The gap becomes more significant when the dataset is harder, e.g. $\tau > 1.5$ for MNIST and $\tau > 1$ for Fashion-MNIST.

For the Synthetic Fashion-MNIST dataset at temperature $\tau = 1$, in addition to the Wide-ResNet (WRN-28) considered above, we also trained three other architectures: a simple linear classifier (Linear), a 1-hidden layer ReLU network (MLP) with 500 hidden units, and a standard AlexNet convolutional architecture [KSH12]. The resulting test errors, as well as the Bayes error, are shown in Figure 4.4. We see that while the development of modern architectures has led to substantial improvement in the test error, there is still a reasonably large gap between the performance of the SOTA Wide-ResNet and Bayes optimality. Nonetheless, it is valuable to know that, for this task, the state-of-the-art has substantial room to be improved.

Dataset Hardness Evaluation

A important application of our Bayes error estimator is to estimate the inherent *hardness* of a given dataset, regardless of model. We run our estimator on several popular image classification corpora and rank them based on our estimated Bayes error. The results are shown in Table 4.1. As a comparison we also put the SOTA numbers in the table.

Before proceeding, we make two remarks. First, all of the Bayes errors reported here were computed using temperature $\tau = 1$. This is for two main reasons: 1) setting $\tau = 1$ reflects the flow model attaining the lowest testing NLL, and hence is in some sense the "best" approximation for the true distribution, 2) in our experiments, the ordering of the hardness of classes is unchanged by varying temperature, and so taking $\tau = 1$ is a reasonable default. Second, the reliability of the Bayes errors reported here as a measure of inherent difficulty are dependent on the quality of the approximate distribution $\hat{p}$; if this distribution is not an adequate estimate of the true distribution p, then it is possible that the Bayes errors do not

Corpus	#classes	#samples	NLL	Bayes Error	SOTA Error [Pap]
MNIST	10	60,000	8.00e2	1.07e-4	1.6e-3 [BKD20]
EMNIST (digits)	10	280,000	8.61e2	1.21e-3	5.7e-3 [PNK+20]
SVHN	10	73,257	4.65e3	7.58e-3	9.9e-3 [BKD20]
Kuzushiji-MNIST	10	60,000	1.37e3	8.03e-3	6.6e-3 [Gas17]
CIFAR-10	10	50,000	7.43e3	2.46e-2	3e-3 [FKMN21]
Fashion-MNIST	10	60,000	1.75e3	3.36e-2	3.09e-2 [TKK21]
EMNIST (letters)	26	145,600	9.15e2	4.37e-2	4.12e-2 [KAJ+20]
CIFAR-100	100	50,000	7.48e3	4.59e-2	3.92e-2 [FKMN21]
EMNIST (balanced)	47	131,600	9.45e2	9.47e-2	8.95e-2 [KAJ+20]
EMNIST (bymerge)	47	814,255	8.53e2	1.00e-1	1.90e-1 [CATvS17]
EMNIST (byclass)	62	814,255	8.76e2	1.64e-1	2.40e-1 [CATvS17]

Table 4.1: We evaluate the estimated Bayes error on image data sets and rank them by relative difficulty. Comparisons with prediction performance of state-of-the-art neural network models shows that our estimation is highly aligned with empirically observed performance.

accurately reflect the true difficulty of the original dataset. Therefore, we also report the test NLL for each model as a metric to evaluate the quality of the approximant $\hat{p}$.

First, we observe that, by and large, the estimated Bayes errors align well with SOTA. In particular, if we constrain the NLL loss to be smaller than 1000, then ranking by our estimated Bayes error aligns exactly with SOTA.

Second, the NLL loss in MNIST, Fashion MNIST, EMNIST and Kuzushiji-MNIST is relatively low, suggesting a good approximation by normalizing flow. However corpora such as CIFAR-10, CIFAR-100, and SVHN may suffer from a lack of training samples. In general large NLL loss may be due to either insufficient model capacity or lack of samples. In our experiments, we always observe the Glow model is able to attain essentially zero error on the training corpus, so it is highly possible the large NLL loss is caused by the lack of training samples.

Third, for datasets such as MNIST, EMNIST (digits, letters, balanced), SVHN, Fashion-MNIST, Kuzushiji-MNIST, CIFAR-10, and CIFAR-100 the SOTA numbers are roughly the same order of magnitude as the Bayes error. On the other hand, for EMNIST (bymerge and byclass) there is still substantial gap between the SOTA and estimated Bayes errors. This is consistent with the fact that there is little published literature about these two datasets; as a result models for them are not as well-developed.

4.5 Conclusions and Future Directions

In this chapter, we have proposed a new approach to benchmarking state-of-the-art models. Rather than comparing trained models to each other, our approach leverages normalizing flows and a key invariance result to be able to generate benchmark datasets closely mimicking

standard benchmark datasets, but with *exactly controlled* Bayes error. This allows us to evaluate the performance of trained models on an absolute, rather than relative, scale. In addition, our approach naturally gives us a method to assess the relative hardness of classification tasks, by comparing their estimated Bayes errors.

While our work has led to several interesting insights, there are also several limitations at present that may be a fruitful source of future research. For one, it is possible that the Glow models we employ here could be replaced with higher quality flow models, which would perhaps lead to better benchmarks and better estimates of the hardness of classification tasks. To this end, it is possible that the well-documented label noise in standard datasets contributes to our inability to learn higher-quality flow models [NAM21]. To the best of our knowledge, there has not been significant work using normalizing flows to accurately estimate class-conditional distributions for NLP datasets; this in itself would be an interesting direction for work. Second, a major limitation of our approach is that there isn't an immediately obvious way to assess how well the Bayes error of the approximate distribution $\mathcal{E}_{\text{Bayes}}(\hat{p})$ estimates the true Bayes error $\mathcal{E}_{\text{Bayes}}(p)$. Theoretical results which bound the distance between these two quantities, perhaps in terms of a divergence $D(p\|\hat{p})$, would be of great interest here.

As detailed in [VKS19], there may be pernicious impacts of the common task framework and the so-called Holy Grail performativity that it induces. For example, a singular focus by the community on the leaderboard performance metrics without regard for any other performance criteria such as fairness or respect for human autonomy. The work here may or may not exacerbate this problem, since trying to approach fundamental Bayes limits is psychologically different than trying to do better than SOTA. As detailed in [Var20], the shift from competing against others to a pursuit for the fundamental limits of nature may encourage a wider and more diverse group of people to participate in ML research, e.g. those with personality type that has less orientation to competition. It is still to be investigated how to do this, but the ability to generate infinite data of a given target difficulty (yet style of existing datasets) may be used to improve the robustness of classifiers and perhaps decrease spurious correlations.

Chapter 5

Average-Case Improvement Through Ensembling

The contents of this chapter are partially based on the pre-print, currently under review, "When are Ensembles Really Effective?", co-authored with Hyunsuk Kim, Yaoqing Yang, Liam Hodgkinson and Michael W. Mahoney [TKY+23].

5.1 Introduction

The fundamental ideas underlying ensemble methods can be traced back at least two centuries, with Condorcet's Jury Theorem among its earliest developments [Con85]. This result asserts that if each juror on a jury makes a correct decision independently and with the same probability $p > 1/2$, then the majority decision of the jury is more likely to be correct with each additional juror. The general principle of aggregating knowledge across imperfectly correlated sources is intuitive, and it has motivated many ensemble methods used in modern statistics and machine learning practice. Among these, tree-based methods like random forests [Bre01] and XGBoost [CG16] are some of the most effective and widely-used.

With the growing popularity of deep learning, a number of approaches have been proposed for ensembling neural networks. Perhaps the simplest of them are so-called deep ensembles, which are ensembles of neural networks trained from independent initializations [ABP+22, ABPC22, FHL19]. In some cases, it has been claimed that such deep ensembles provide significant improvement in performance [FHL19, OFR+19, ALMV20]. Such ensembles have also been used to obtain uncertainty estimates for prediction [LPB17] and to provide more robust predictions under distributional shift. However, the benefits of deep ensembling are not universally accepted. Indeed, other works have found that ensembling is less necessary for

larger models, and that in some cases a single larger model can perform as well as an ensemble of smaller models [HVD+15, BCNM06, GJS+20, ABP+22]. Similarly mixed results, where empirical performance does not conform with intuitions and popular theoretical expectations, have been reported in the Bayesian approach to deep learning [INLW21]. Furthermore, an often-cited practical issue with ensembling, especially of large neural networks, is the constraint of storing and performing inference with many distinct models.

In light of the increase in computational cost, it is of great value to understand exactly when we might expect ensembling to improve performance non-trivially. In particular, consider the following practical scenario: a practitioner has trained a single (perhaps large and expensive) model, and would like to know whether they can expect significant gains in performance from training additional models and ensembling them. This question lacks a sufficient answer, both from the theoretical and empirical perspectives, and hence motivates the main question of this chapter:

When are ensembles ***really*** effective?

The present work addresses this question, both theoretically and empirically, under very general conditions. We focus our study on the most popular ensemble classifier—the *majority vote classifier* (Definition 5.1), which we denote by h_{MV}—although our framework also covers variants such weighted majority vote methods.

Theoretical results. Our main theoretical contributions, contained in Section 5.3, are as follows. First, we formally define the *ensemble improvement rate* (EIR, Definition 5.3), which measures the decrease in error rate from ensembling, on a relative scale. We then introduce a new condition called *competence* (Assumption 5.1) that rules out pathological cases, allowing us to prove stronger bounds on the ensemble improvement rate. Specifically, 1) we prove (in Theorem 5.1) that competent ensembles can never hurt performance, and 2) we prove (in Theorem 5.2) that the EIR can be upper and lower bounded by linear functions of the *disagreement-error ratio* (DER, Definition 5.4). ***Our theoretical results predict that ensemble improvement will be high whenever disagreement is large relative to the average error rate (i.e.,*** $\text{DER} > 1$***).*** Moreover, we show (in Appendix D.1) that as Corollaries of our theoretical results, we obtain new bounds on the error rate of the majority vote classifier that significantly improve on previous results, provided the competence assumption is satisfied.

Empirical results. In light of our new theoretical understanding of ensembling, we perform a detailed empirical analysis of ensembling in practice. In Section 5.4, we evaluate the assumptions and predictions made by the theory presented in Section 5.3. In particular, we verify on a variety of tasks that the competence condition holds, we verify empirically the linear relationship between the EIR and the DER, as predicted by our bounds, and

we suggest directions through which our theoretical results might be improved. In Section 5.5, we provide significant evidence for distinct behavior arising for ensembles in and out of the "interpolating regime," i.e., when each of the constituent classifiers in an ensemble has sufficient capacity to achieve zero training error. ***We show 1) that interpolating ensembles exhibit consistently lower ensemble improvement rates, and 2) that this corresponds to ensembles transitioning (sometimes sharply) from the regime*** $\mathrm{DER} > 1$ ***to*** $\mathrm{DER} < 1$**.** Finally, we also show that tree-based ensembles represent a unique exception to this phenomenon, making them particularly well-suited to ensembling.

In addition to the results presented in the main text, we provide supplemental theoretical results (including all proofs) in Appendix D.1, as well as supplemental empirical results in Appendix D.2.

5.2 Background and preliminaries

In this section, we present some relevant background and setup.

Setup

In this work, we focus on the K-class classification setting, wherein the data $(X, Y) \in \mathcal{X} \times \mathcal{Y} \sim \mathcal{D}$ consist of features $\boldsymbol{x} \in \mathcal{X}$ and labels $y \in \mathcal{Y} = \{1, \ldots, K\}$. Classifiers are then functions $h : \mathcal{X} \to \mathcal{Y}$ that belong to some set $\mathcal{H}$. To measure the performance of a single classifier h on the data distribution $\mathcal{D}$, we use the usual error rate:

$$L_{\mathcal{D}}(h) = \mathbb{E}_{X,Y \sim \mathcal{D}}[\mathbb{1}(h(X) \neq Y)].$$

For notational convenience, we drop the explicit dependence on $\mathcal{D}$ whenever it is apparent from context.

A central object in our study is a distribution ρ over classifiers. Depending on the context, this distribution could represent a variety of different things. For example, ρ could be:

i) A discrete distribution on a finite set of classifiers $\{h_1, \ldots, h_M\}$ with weights $\rho_1, \ldots, \rho_M$, e.g., representing normalized weights in a weighted ensembling scheme;

ii) A distribution over parameters θ of a parametric family of models, h_θ, determined, e.g., by a stochastic optimization algorithm with random initialization;

iii) A Bayesian posterior distribution.

The distribution ρ induces two error rates of interest. The first is the ***average error rate of any single classifier*** under ρ, defined to be $\mathbb{E}_{h \sim \rho}[L(h)]$. The second is the ***error rate of the majority vote classifier***, h_{MV}, which is defined for a distribution ρ as follows.

Definition 5.1 (Majority vote classifier)**.** Given ρ, the ***majority vote classifier*** is the classifier which, for an input $\boldsymbol{x}$, predicts the most probable class for this input among classifiers drawn from ρ,

$$h_{\mathrm{MV}}(\boldsymbol{x}) = \arg\max_j \mathbb{E}_{h\sim\rho}[\mathbb{1}(h(\boldsymbol{x}) = j)].$$

In the Bayesian context, $\rho = \rho(h \mid X_{\mathrm{train}}, y_{\mathrm{train}})$ is a posterior distribution over classifiers. In this case, the majority vote classifier is often called the Bayes classifier, and the error rate $L(h_{\mathrm{MV}})$ is called the Bayes error rate. In such contexts, the average error rate is often referred to as the Gibbs error rate associated with ρ and $\mathcal{D}$.

Finally, we will present results in terms of the *disagreement rate* between classifiers drawn from a distribution ρ, defined as follows.

Definition 5.2 (Disagreement)**.** The ***disagreement rate*** between two classifiers h, h' is given by $D_{\mathcal{D}}(h, h') = \mathbb{E}_{X\sim\mathcal{D}}[\mathbb{1}(h(X) \neq h'(X))]$. The ***expected disagreement rate*** is $\mathbb{E}_{h,h'\sim\rho}[D_{\mathcal{D}}(h, h')]$, where $h, h' \sim \rho$ are drawn independently.

Prior work

Ensembling theory. Perhaps the simplest general relation between the majority vote error rate and the average error rate guarantees only that the majority vote classifier is no worse than twice the average error rate [LMRR17, MLIS20]. To see this, let $W_\rho \equiv W_\rho(X, Y) = \mathbb{E}_{h\sim\rho}[\mathbb{1}(h(X) \neq Y)]$ denote the proportion of erroneous classifiers in the ensemble for a randomly sampled input-output pair $(X, Y) \sim \mathcal{D}$, and note that $\mathbb{E}[W_\rho] = \mathbb{E}[L(h)]$. Then, by a "first-order" application of Markov's inequality, we have that

$$0 \leq L(h_{\mathrm{MV}}) \leq \mathbb{P}(W_\rho \geq 1/2) \leq 2\,\mathbb{E}[W_\rho] = 2\,\mathbb{E}_{h\sim\rho}[L(h)]. \tag{5.1}$$

This bound is almost always uninformative in practice. Indeed, it may seem surprising that an ensemble classifier could perform *worse* than the average of its constituent classifiers, much less a factor of two worse. Nonetheless, the first-order upper bound is, in fact, tight: there exist distributions ρ (over classifiers) and $\mathcal{D}$ (over data) such that the majority vote classifier is twice as erroneous as any one classifier, on average. As one might expect, however, this tends to happen only in pathological cases; we give examples of such ensembles in Appendix D.3.

To circumvent the shortcomings of the simple first-order bound, more recent approaches have developed bounds incorporating "second-order" information from the distribution ρ [MLIS20]. One successful example of this is given by a class of results known as C-bounds [GLL+15, LMRR17]. The most general form of these bounds states, provided $\mathbb{E}[M_\rho(X, Y)] > 0$, that

$$L(h_{\mathrm{MV}}) \leq 1 - \frac{\mathbb{E}[M_\rho(X, Y)]^2}{\mathbb{E}[M_\rho^2(X, Y)]}, \tag{5.2}$$

where $M_\rho(X,Y) = \mathbb{E}_{h\sim\rho}[\mathbb{1}(h(X)=Y)] - \max_{j\neq Y}\mathbb{E}_{h\sim\rho}[\mathbb{1}(h(X)=j)]$ is called the *margin*. In the binary classification case, the condition $\mathbb{E}[M_\rho(X,Y)] > 0$ is equivalent to the assumption $\mathbb{E}_{h\sim\rho}[L(h)] < 1/2$. Hence, it can be viewed as a requirement that individual classifiers are "weak learners." The same condition is used to derive a very similar bound for random forests in [Bre01], which is then further upper bounded to obtain a more intuitive (though weaker) bound in terms of the "c/s2" ratio. Relatedly, [MLIS20] obtains a bound on the error rate of the majority-vote classifier, in the special case of binary classification, directly in terms of the disagreement rate, taking the form $4\mathbb{E}[L(h)]-2\mathbb{E}[D(h,h')]$. We note that our theory improves this bound by factor of 2 (see Appendix D.1). Other results obtain similar expressions, but in terms of different loss functions, e.g., cross-entropy loss [ABP+22, OCnM22].

Other related studies. In addition to theoretical results, there have been a number of recent empirical studies investigating the use of ensembling. Perhaps the most closely related is [ABPC22], which shows, perhaps surprisingly, that ensembles do not benefit significantly from encouraging diversity during training. In contrast to the present work, [ABPC22] focuses on the cross entropy loss for classification (which facilitates somewhat simpler theoretical analysis), whereas we focus on the more intuitive and commonly used classification error rate. Moreover, while [ABPC22] study the ensemble improvement *gap* (i.e., the difference between the average loss of a single classifier and the ensemble loss), we focus on the gap in error rates on a relative scale. As we show, this provides *much* finer insights into ensembles improvement. To complement this, [FHL19] study ensembling from a loss landscape perspective, evaluating how different approaches to ensembling, such as deep ensembles, Bayesian ensembles, and local methods like Gaussian subspace sampling compare in function and weight space diversity. Other recent work has studied the use of ensembling to provide uncertainty estimates for prediction [LPB17], and to improve robustness to out-of-distribution data [OFR+19], although the ubiquity of these findings has recently been questioned in [ABP+22].

5.3 Ensemble improvement, competence, and the disagreement-error ratio

In this section, our goal is to characterize theoretically the rate at which ensembling improves performance. To do this, we first need to formalize a metric to quantify the benefit from ensembling. One natural way of measuring this improvement would be to compute the gap $\mathbb{E}_{h\sim\rho}[L(h)] - L(h_{\text{MV}})$. A similar gap was the focus of [ABPC22], although in terms of the cross-entropy loss, rather than the classification error rate. However, the unnormalized gap can be misleading—in particular, it will tend to be small whenever the average error rate itself is small, thus making it impractical to compare, e.g., across tasks of varying difficulty. Instead, we work with a normalized version of the average-minus-ensemble error rate gap, where the effect of the normalization is to measure this error in a relative scale. We call this the ensemble improvement rate.

Definition 5.3 (Ensemble improvement rate)**.** Given distributions ρ over classifiers and $\mathcal{D}$ over data, provided that $\mathbb{E}_{h\sim\rho}[L(h)] \neq 0$, the ***ensemble improvement rate (EIR)*** is defined as

$$\text{EIR} = \frac{\mathbb{E}_{h\sim\rho}[L(h)] - L(h_{\text{MV}})}{\mathbb{E}_{h\sim\rho}[L(h)]}.$$

In contrast to the unnormalized gap, the ensemble improvement rate can be large even for very easy tasks with a small average error rate. Recall the simple first-order bound on the majority-vote classifier: $L(h_{\text{MV}}) \leq 2\mathbb{E}[L(h)]$. Rearranging, we deduce that $\text{EIR} \geq -1$. Unfortunately, in the absence of additional information, this first-order bound is in fact tight: one can construct ensembles for this $L(h_{\text{MV}}) = 2\mathbb{E}[L(h)]$ (see Appendix D.3). However, this bound is inconsistent with how ensembles generally behave and practice, and indeed it tells us nothing about when ensembling can improve performance. In the subsequent sections, we derive improved bounds on the EIR that do.

Competent ensembles never hurt

Surprisingly, to our knowledge, there is no known characterization of the majority-vote error rate that *guarantees* it can be no worse than the error rate of any individual classifier, on average. Indeed, it turns out this is the result of strange behavior that can arise for particularly pathological ensembles rarely encountered in practice (see Appendix D.3 for a more detailed discussion of this). To eliminate these cases, we introduce a mild condition that we call *competence*.

Assumption 5.1 (Competence)**.** Let $W_\rho \equiv W_\rho(X, Y) = \mathbb{E}_{h\sim\rho}[\mathbb{1}(h(X) \neq Y)]$. The ensemble ρ is ***competent*** if for every $0 \leq t \leq 1/2$,

$$\mathbb{P}(W_\rho \in [t, 1/2)) \geq \mathbb{P}(W_\rho \in [1/2, 1-t]).$$

The competence assumption guarantees that the ensemble is not pathologically bad, and in particular it eliminates the scenarios under which the naive first-order bound (5.1) is tight. As we will show in Section 5.4, the competence condition is quite mild, and it holds broadly in practice. Our first result uses competence to improve non-trivially the naive first-order bound.

Theorem 5.1. *Competent ensembles never hurt performance, i.e.,* $\text{EIR} \geq 0$.

Translated into a bound on the majority vote classifier, Theorem 5.1 guarantees that $L(h_{\text{MV}}) \leq \mathbb{E}[L(h)]$, improving on the naive first-order bound (5.1) by a factor of two. To the best of our knowledge, the competence condition is the first of its kind, in that it guarantees what is widely observed in practice, i.e., that ensembling cannot *hurt* performance. However, it is insufficient to answer the question of *how much* ensembling can improve performance. To address this question, we turn to a "second-order" analysis involving the disagreement rate.

Quantifying ensemble improvement with the disagreement-error ratio

Our central result in this section will be to relate the EIR to the ratio of the disagreement to average error rate, which we define formally below.

Definition 5.4 (Disagreement-error ratio)**.** Given the distributions ρ over classifiers and $\mathcal{D}$ over data, provided that $\mathbb{E}_{h\sim\rho}[L(h)] \neq 0$, the ***disagreement-error ratio (DER)*** is defined as

$$\text{DER} = \frac{\mathbb{E}_{h,h'\sim\rho}[D(h,h')]}{\mathbb{E}_{h\sim\rho}[L(h)]}.$$

Our next result relates the EIR to a linear function of the DER.

Theorem 5.2. *For any competent ensemble ρ of K-class classifiers, provided $\mathbb{E}_{h\sim\rho}[L(h)] \neq 0$, the ensemble improvement rate satisfies*

$$\text{DER} \geq \text{EIR} \geq \frac{2(K-1)}{K}\text{DER} - \frac{3K-4}{K}.$$

Note that neither Theorem 5.1 nor Theorem 5.2 is uniformly stronger. In particular, if $\text{DER} < (3K-4)/(2K-2)$ then the lower bound provided in Theorem 5.1 will be superior to the one in Theorem 5.2.

Theorem 5.2 predicts that the EIR is fundamentally governed by a linear relationship with the DER — a result that we will verify empirically in Section 5.4. Importantly, we note that there are two distinct regimes in which the bounds in Theorem 5.2 provide non-trivial guarantees.

> **DER small** ($< \mathbf{1}$). In this case, by the trivial bound (5.1), $\text{EIR} \leq 1$, and thus the upper bound in Theorem 5.2 guarantees ensemble improvement cannot be too large whenever $\text{DER} < 1$, that is, whenever disagreement is small relative to the average error rate.
>
> **DER large** ($> \mathbf{1}$). In this case, the lower bound in Theorem 5.2 guarantees ensemble improvement whenever disagreement is sufficiently large relative to the average error rate, in particular when $\text{DER} \geq (3K-4)/(2K-2) \geq 1$.

In our empirical evaluations, we will see that these two regimes ($\text{DER} > 1$ and $\text{DER} < 1$) strongly distinguish between situations in which ensemble improvement is high, and when the benefits of ensembling are significantly less pronounced.

Moreover, Theorem 5.2 captures an important subtlety in the relationship between ensemble improvement and predictive diversity. In particular, while general intuition—and a significant

body of prior literature, as discussed in Section 5.2—suggests that higher disagreement leads to high ensemble improvement, this may *not* be the case if the disagreement is nominally large, but small relative to the average error rate.

Remark 5.1 (Corollaries of Theorems 5.1 and 5.2)**.** Using some basic algebra, the upper and lower bounds presented in Theorems 5.1 and 5.2 can easily be translated into upper and lower bounds on the error rate of the majority vote classifier itself. For the sake of space, we defer discussion of these Corollaries to Appendix D.1, although we note that the resulting bounds constitute significant improvements on existing bounds, which we verify both analytically (when possible) and empirically.

5.4 Evaluating the theory

In this section, we investigate the assumptions and predictions of the theory proposed in Section 5.3. In particular we will show 1) that the competence assumption holds broadly in practice, across a range of architectures, ensembling methods and datasets, and 2) that the EIR exhibits a close linear relationship with the DER, as predicted by Theorem 5.2.

Before presenting our findings, we first briefly describe the experimental settings analyzed in the remainder of the paper. Our goal is to select a sufficiently broad range of tasks and methods so as to demonstrate the generality of our conclusions.

Setup for empirical evaluations

In Table 5.1 we provide a brief description of our experimental setup; more extensive experimental details can be found in Appendix D.2.

Table 5.1: Datasets and ensembles used in empirical evaluations, where C denotes the number of classes, and M denotes the number of classifiers.

Datasets		
Dataset	**C**	**Reference**
MNIST (5K subset)	10	[Den12]
CIFAR-10	10	[KH+09]
IMDB	2	[MDP+11]
QSAR	2	[BGCT19]
Thyroid	2	[QCHL87]
GLUE (7 tasks)	2-3	[WSM+19]

Ensembles			
Base classifier	**Ensembling**	**M**	**Reference**
ResNet20-Swish	Bayesian Ens.	100	[IVHW21]
ResNet18	Deep Ens.	5	[KH+09]
CNN-LSTM	Bayesian Ens.	100	[IVHW21]
BERT (fine-tune)	Deep Ens.	25	[DCLT19, SYT+22]
Random Features	Bagging	30	N/A
Decision Trees	Random Forests	100	[PVG+11, Bre01]

Verifying competence in practice.

Our theoretical results in Section 5.3 relied on the competence condition. One might wonder whether competent ensembles exist, and if so how ubiquitous they are. Here, we test that

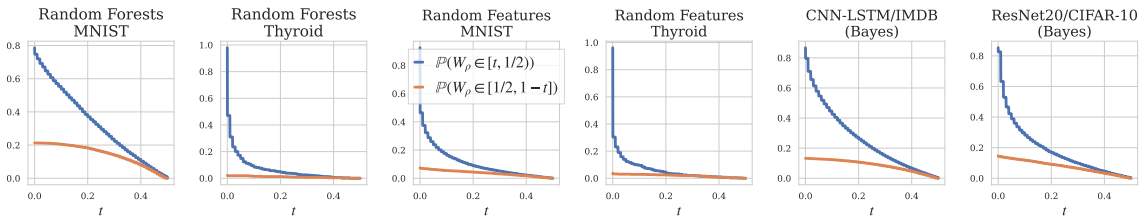


Figure 5.1: **Verifying the competence assumption in practice.** $W_\rho(X, Y)$ in Assumption 5.1 is estimated using hold-out data. Across all tasks, $\blacksquare > \blacksquare$, supporting Assumption 5.1.

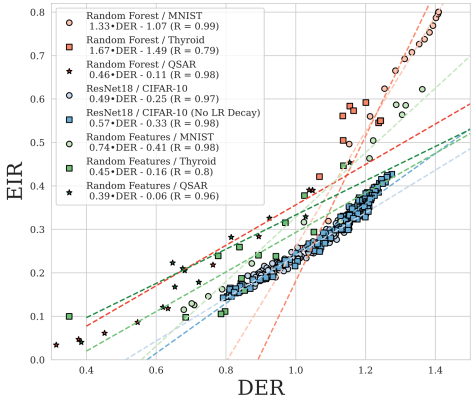


Figure 5.2: EIR **is linearly correlated with the** DER**.** We plot the EIR against the DER across a variety of experimental settings, and we observe a close linear relationship between the EIR and DER, as predicted by our theoretical results. In the legend, we also report the equation for the line of best fit within each setting, as well as the Pearson correlation.

assumption. (That is, we test not just the predictions of our theory, but also the assumptions of our theory.)

We have observed that the competence assumption is empirically very mild, and that in practice it applies very broadly. In Figure 5.1, we estimate both $\mathbb{P}(W_\rho \in [t, 1/2))$ and $\mathbb{P}(W_\rho \in [1/2, 1-t])$ on test data, validating that competence holds for various types of ensembles across a subset of tasks. To do this, given a test set of examples $\{(\boldsymbol{x}_j, y_j)\}_{j=1}^m$ and classifiers $h_1, \ldots, h_N$ drawn from ρ, we construct the estimator

$$\widehat{W}_\rho^{(j)} = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}(h_n(\boldsymbol{x}_j) \neq y_j),$$

and we calculate $\mathbb{P}(W_\rho \in [t, 1/2))$ and $\mathbb{P}(W_\rho \in [1/2, 1-t])$ from the empirical CDF of $\{\widehat{W}_\rho^{(j)}\}_{j=1}^m$. In Appendix D.2, we provide additional examples of competence plots across more experimental settings (and we observe substantially the same results).

The linear relationship between DER and EIR.

Theorem 5.2 predicts a linear relationship between the EIR and the DER; here we verify that this relationship holds empirically. In Figure 5.2, we plot the EIR against the DER across several experimental settings, varying capacity hyper-parameters (width for the ResNet18 models, number of random features for the random feature classifiers, and number of leaf nodes for the random forests), reporting the equation of the line of best fit, as well as the Pearson correlation between the two metrics. Across 6 of the 8 experimental settings evaluated, we find a very strong linear relationship – with the Pearson $R \geq 0.96$. The two exceptions are found for the Thyroid classification dataset (though there is still a strong trend between the two quantities, with $R \approx 0.8$).

Interestingly, we can compare the lines of best fit to the theoretical linear relationship predicted by Theorem 5.2. In the case of the binary classification datasets (QSAR and Thyroid), the bound predicts that $\text{EIR} \approx \text{DER} - 1$; while for the 10-class problems (MNIST and CIFAR-10), the equation is $\text{EIR} \approx 1.8\,\text{DER} - 2.6$. While for some examples (e.g., random forests on MNIST), the equations are close to those theoretically predicted, there is a clear gap between theory and the experimentally measured relationships for other tasks. In particular, we notice a significant difference in the governing equations for the same datasets between random forests and the random feature ensembles, suggesting that the relationship is to some degree modulated by the model architecture – something our theory cannot capture. We therefore see refining our theory to incorporate such information as an important direction for future work.

5.5 Ensemble improvement is low in the interpolating regime

In this section, we will show that the DER behaves qualitatively differently for interpolating versus non-interpolating ensembles, in particular exhibiting behavior associated with phase transitions. Such phase transitions are well-known in the statistical mechanics approach to learning [EdB01, MM17, TKM21, YHT+21], but they have been viewed as surprising from the more traditional approach to statistical learning theory [BHMM19a]. We will use this to understand when ensembling is and is not effective for deep ensembles, and to explain why tree-based methods seem to benefit so much from ensembling across all settings. We say that a model is *interpolating* if it achieves exactly zero training error, and we say that it is *non-interpolating* otherwise; we call an ensemble interpolating if each of its constituent classifiers is interpolating. Note that for methods that involve resampling of the training data (e.g., bagging methods), we define the training error as the "in-bag" training, i.e., the error evaluated only on the points a classifier was trained on.

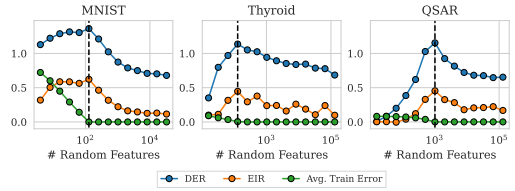


Figure 5.3: **Bagged random feature classifiers.** Blacked dashed line represents the interpolation threshold. Across all tasks, DER and EIR are maximized at this point, and then decrease thereafter.

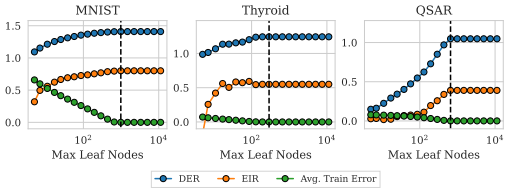
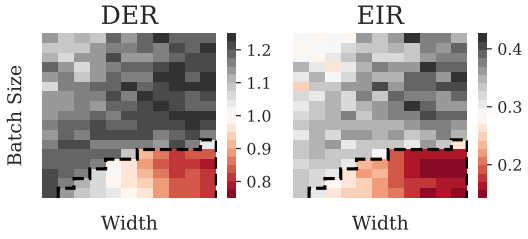
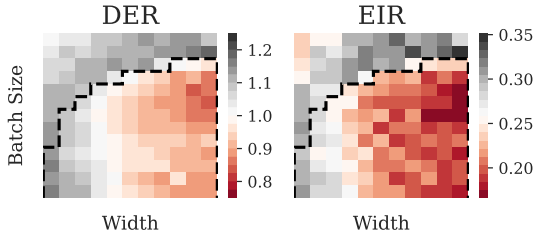


Figure 5.4: **Random forest classifiers.** Blacked dashed line represents the interpolation threshold. Across all tasks, DER and EIR are maximized at this point, and then remain constant thereafter.



(a) **Without LR decay.**



(b) **With LR decay.**

Figure 5.5: **Large scale studies of deep ensembles on ResNet18/CIFAR-10.** We plot the DER, EIR, average error and majority vote error rate across a range of hyper-parameters, for two training settings: one with learning rate decay, and one without. The black dashed line indicates the *interpolation threshold*, i.e., the curve below which individual models achieve exactly zero training error. Observe that interpolating ensembles attain distinctly lower EIR than non-interpolating ensembles, and correspondingly have low DER (< 1), compared to non-interpolating ensembles with high DER (> 1).

Interpolating random feature classifiers. We first look at the bagged random feature ensembles on the MNIST, Thyroid, and QSAR datasets. In Figure 5.3, we plot the EIR, DER and training error for each of these ensembles (recall that for ensembles which use bagging, the training error is computed as the *in-bag* training error). We observe the same phenomenon across these three tasks: *as a function of model capacity, the* EIR *and* DER *are both maximized at the interpolation threshold, before decreasing thereafter. This indicates that much higher-capacity models, those with the ability to easily interpolate the training data, benefit significantly less from ensembling. In particular, observe that for sufficiently high-capacity ensembles, the* DER *become less than 1, entering the regime in which our theory guarantees low ensemble improvement.*

Interpolating deep ensembles. We next consider the DER, EIR, average test error, and majority vote test error rates for large-scale empirical evaluations on ResNet18/CIFAR-10 models in batch size/width space, both with and without learning rate decay. See Figure 5.5 for the results. Note that the use of learning rate decay facilitates easier interpolation

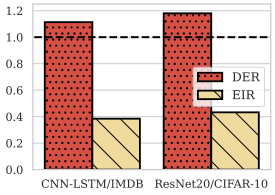


Figure 5.6: **Bayesian ensembles on IMDB and CIFAR-10.** These provide examples of ensembles which do *not* interpolate the training data, and which have high DER (> 1) and correspondingly high EIR.

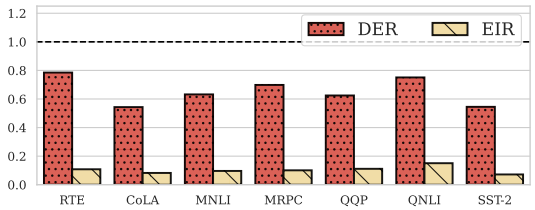


Figure 5.7: **Ensembles of fine-tuned BERT on GLUE tasks.** Here the models are large relative to the dataset size, and consequently they exhibit low DER (< 1) and EIR across all tasks.

of the training data during training, hence broadening the range of hyper-parameters for which interpolation occurs. The figures are colored so that ensembles in the regime DER < 1 are in red, while ensembles with DER > 1 are in grey. The black dashed line indicates the interpolation threshold, i.e., the curve in hyper-parameter space below which ensembles achieve zero training error (meaning every classifier in the ensemble has zero training error). *Observe in particular that, across all settings,* all *models in the regime* DER < 1 *are interpolating models; and, more generally, that interpolating models tend to exhibit* much *smaller* DER *than non-interpolating models.* Observe moreover that there can be a sharp transition between these two regimes, wherein the DER is large just at the interpolation threshold, and then it quickly decreases beyond that threshold. Correspondingly, ensemble improvement is *much* less pronounced for interpolating versus more traditional non-interpolating ensembles. This is consistent with results previously observed in the literature, e.g., [GJS+20]. We remark also that the behavior observed in Figure 5.5 exhibits the same phases identified in [YHT+21] (an example of phase transitions in learning more generally [EdB01, MM17]), although the DER itself was not considered in that previous study.

Bayesian neural networks. Next, we show that Bayesian neural networks benefit significantly from ensembling. In Figure 5.6, we plot the DER and EIR for Bayesian ensembles on the CIFAR-10 and IMDB tasks, using the ResNet20 and CNN-LSTM architectures, respectively. The samples we present here are provided by [INLW21], who use Hamiltonian Monte Carlo to sample accurately from the posterior distribution over models. For both tasks, we observe that both ensembles exhibit DER > 1 and high EIR. In light of our findings regarding ensemble improvement and interpolation, we note that the Bayesian ensembles by design do *not* interpolate the training data (when drawn at non-zero temperature), as the samples are drawn from a distribution not concentrated only on the modes of the training loss. While we do not perform additional experiments with Bayesian neural networks in the present work, evaluating the DER/EIR as a function of posterior temperature for these models is an interesting direction for future work. We hypothesize that the qualitative effective of decreasing the sampling temperature will be similar to that of increasing the batch size in

the plots in Figure 5.5.

Fine-tuned BERT ensembles. Here, we present results for ensembles of BERT models fine-tuned on the GLUE classification tasks. These provide examples of a very large model trained on small datasets, on which interpolation is easily possible. For these experiments, we use 25 BERT models pre-trained from independent initializations provided in [SYT+22]. Each of these 25 models is then fine-tuned on the 7 classification tasks in the GLUE [WSM+19] benchmark set and evaluated on relatively small test sets, ranging in size from 250 (RTE) to 40,000 (QQP) samples. In Figure 5.7, we plot the EIR and DER across these benchmark tasks and observe that, as predicted, the ensemble improvement rate is low and DER is uniformly low (< 1).

The unique case of random forests. Random forests are one of the most widely-used ensembling methods in practice. Here, we show that the effectiveness of ensembling is much greater for random forest models than for highly-parameterized models like the random feature classifiers and the deep ensembles. Note that for random forests, interpolation of the training data *is* possible, in particular whenever the number of terminal leaf nodes is sufficiently large (where here we again compute the average training error using on the in-bag training examples for each tree), but it is not possible to go "into" the interpolating regime. In Figure 5.4, we plot the DER, EIR, and training error as a function of the max number of leaf nodes (a measure of model complexity). Before the interpolation threshold, both the EIR and DER increase as a function of model capacity, in line with what is observed for the random feature and deep ensembles. However, we observe distinct behavior at the interpolation threshold: both EIR and DER become constant past this threshold. This is fundamental to tree-based methods, due to the method by which they are fit, e.g., using a standard procedure like CART [BFSO84]. As soon as a tree achieves zero training error, any impurity method used to split the nodes further is saturated at zero, and therefore the models cannot continue to grow. This indicates that trees are particularly well-suited to ensembling across all hyper-parameter values, in contrast to other parameterized types of classifiers.

5.6 Discussion and conclusion

To help answer the question of when ensembling is effective, we introduce the ensemble improvement rate (EIR), which we then study both theoretically and empirically. Theoretically, we provide a comprehensive characterization of the EIR in terms of the disagreement-error ratio DER. The results are based on a new, mild condition called *competence*, which we introduce to rule out pathological cases that have hampered previous theoretical results. Using a simple first-order analysis, we show that the competence condition is sufficient to guarantee that ensembling cannot hurt performance—something widely observed in practice, but surprisingly unexplained by existing theory. Using a second-order analysis, we are able to theoretically characterize the EIR, by upper and lower bounding it in terms of a linear

function of the DER. On the empirical side, we first verify the assumptions of our theory (namely that the competence assumption holds broadly in practice), and we show that our bounds are indeed descriptive of ensemble improvement in practice. We then demonstrate that improvement decreases precipitously for interpolating ensembles, relative to non-interpolating ones, providing a very practical guideline for when to use ensembling.

Our work leaves many directions to explore, of which we name a few promising ones. First, while our theory represents a significant improvement on previous results, there are still directions to extend our analysis. For example, Figure 5.2 suggests that the relationship between EIR and DER can be even more finely characterized. Is it possible to refine our analysis further to incorporate information about the data and/or model architecture? Second, can we formalize the connection between ensemble effectiveness and the interpolation point, and relate it to similar ideas in the literature?

Chapter 6

Final Thoughts and Outlook

In this thesis, we introduced the problem of generalization in modern machine learning, and various approaches designed to address it. In Chapter 2, we saw classical worst-case analyses that produce PAC-style bounds on the generalization gap. However, the practical utility of the worst-case approach is challenged in Chapter 3, in which we observed that the vast majority of large-scale, interpolating models behave nothing like the worst-case model. Indeed, in Chapter 4, we saw that many models appear to perform much closer to the *best possible* model than to the worst-case. In Chapter 5, we saw that we can closely characterize the behavior of ensemble classifiers in terms of the disagreement-error ratio. Perhaps surprisingly, we find that ensembles exhibit very different behavior when in or out of the interpolating regime.

As the field of machine learning has developed, and become a more important part of society, the desire to understand its workings has become increasingly important. Impressive efforts have been made on both from the theoretical and experimental perspectives, yielding significant progress. The work contained in this thesis represents but a small effort towards designing a functional theory of modern machine learning systems, and there is much work left to do.

As the field advances, we suspect that the community will rely more heavily on carefully designed, large-scale experimental studies to help guide it forward—much like any other science. The scientific approach has already seen great success in providing a practically useful understanding of how and when large machine learning systems can be expected to work. However, we see the scientific approach developing alongside theoretical analysis of simpler models, providing complementary insights. In this way, machine learning seems to be converging to a paradigm that shares many similarities with physics.

Bibliography

[ABP+22] Taiga Abe, E. Kelly Buchanan, Geoff Pleiss, Richard Zemel, and John Patrick Cunningham. Deep ensembles work, but are they necessary? In *Advances in Neural Information Processing Systems*, 2022.

[ABPC22] Taiga Abe, E Kelly Buchanan, Geoff Pleiss, and John Patrick Cunningham. The best deep ensembles sacrifice predictive diversity. In *I Can't Believe It's Not Better Workshop: Understanding Deep Learning Through Empirical Falsification*, 2022.

[Abr64] I. G. Abrahamson. Orthant Probabilities for the Quadrivariate Normal Distribution. *Annals of Mathematical Statistics*, 35(4):1685–1703, 1964.

[ADH+19] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On Exact Computation with an Infinitely Wide Neural Net. In *Advances in Neural Information Processing Systems*, 2019.

[AGNZ18] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger Generalization Bounds for Deep Nets via a Compression Approach. In *35th International Conference on Machine Learning*, 2018.

[AKL13] Dimitris Achlioptas, Zohar Karnin, and Edo Liberty. Matrix Entry-wise Sampling: Simple is Best. In *KDD*, 2013.

[ALMV20] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *International Conference on Learning Representations (ICLR)*, 2020.

[AZLS19] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 242–252. PMLR, 09–15 Jun 2019.

[Bar91] Andrew R. Barron. Complexity Regularization with Application to Artificial Neural Networks. In *Nonparametric Functional Estimation and Related Topics*, pages 561–576. Springer Netherlands, Dordrecht, 1991.

[Bar93] Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 5 1993.

[BB21] Tristan Bepler and Bonnie Berger. Learning the protein language: Evolution, structure, and function. *Cell Systems*, 12(6):654–669.e3, 2021.

[BCNM06] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[BFSO84] Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[BFT17] Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *31st Conference on Neural Information Processing Systems*, 2017.

[BGCT19] Davide Ballabio, Francesca Grisoni, Viviana Consonni, and Roberto Todeschini. Integrated qsar models to predict acute oral systemic toxicity. *Molecular Informatics*, 38(8-9):1800124, 2019.

[BHLM19] Peter L. Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

[BHMM19a] M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proc. Natl. Acad. Sci. USA*, 116:15849–15854, 2019.

[BHMM19b] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences of the United States of America*, 116(32):15849–15854, 8 2019.

[BHX19] Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. Technical Report arXiv preprint: 1903.07571, 2019.

[BK18] Andrew R. Barron and Jason M. Klusowski. Approximation and Estimation for High-Dimensional Deep Learning Networks. 2018.

[BKD20] Adam Byerly, Tatiana Kalganova, and Ian Dear. A branching and merging convolutional network with homogeneous filter capsules. arXiv:2001.09136 [cs.CV]., January 2020.

[BKM+19] Jean Barbier, Florent Krzakala, Nicolas Macris, Léo Miolane, and Lenka Zdeborová. Optimal errors and phase transitions in high-dimensional generalized linear models. *Proceedings of the National Academy of Sciences*, 116(12):5451–5460, 2019.

[BLG+19] Cenk Baykal, Lucas Liebenwein, Igor Gilitschenski, Dan Feldman, and Daniela Rus. Data-Dependent Coresets for Compressing Neural Networks with Applications to Generalization Bounds. In *International Conference on Learning Representations*, 2019.

[BLLT20] Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[BWHS16] Visar Berisha, Alan Wisler, Alfred O. Hero, III, and Andreas Spanias. Empirically estimable classification bounds based on a nonparametric divergence measure. *IEEE Transactions on Signal Processing*, 64(3):580–591, February 2016.

[CATvS17] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik. EMNIST: An extension of MNIST to handwritten letters. In *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926, May 2017.

[CBIK+18] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

[CF07] Daniel J. Costello, Jr. and G. David Forney, Jr. Channel coding: The road to channel capacity. *Proceedings of the IEEE*, 95(6):1150–1177, June 2007.

[CG16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[CHM+15a] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann Lecun. The Loss Surfaces of Multilayer Networks. In *18th International Conference on Artificial Intelligence and Statistics*, 2015.

[CHM+15b] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The Loss Surfaces of Multilayer Networks. In *18th International Conference on Artificial Intelligence and Statistics*, 2015.

[CL20] Niladri S. Chatterji and Philip M. Long. Finite-sample analysis of interpolating linear classifiers in the overparameterized regime. Technical Report arXiv preprint: 2004.12019, 4 2020.

[CLH+23] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms, 2023.

[Con] Basic mnist example. https://github.com/pytorch/examples/tree/master/mnist. Accessed: 2021-05-08.

[Con85] Marquis de Condorcet. Essay on the application of analysis to the probability of majority decisions. *Paris: Imprimerie Royale*, 1785.

[CS18] Emmanuel J. Candes and Pragya Sur. The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression. Technical Report arXiv preprint: 1804.09753, 2018.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, June 2019.

[DDN+20] Gintare Karolina Dziugaite, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M Roy. In search of robust measures of generalization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11723–11733. Curran Associates, Inc., 2020.

[Den12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[DG17] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[DH95] Persi Diaconis and Susan Holmes. Three examples of Monte-Carlo Markov chains: At the interface between statistical computing, computer science, and statistical mechanics. In David Aldous, Persi Diaconis, Joel Spencer, and J. Michael Steele, editors, *Discrete Probability and Algorithms*, pages 43–56. Springer, New York, NY, 1995.

[DKT19] Zeyu Deng, Abla Kammoun, and Christos Thrampoulidis. A Model of Double Descent for High-dimensional Binary Linear Classification. Technical report, 2019.

[DKT20] Z. Deng, A. Kammoun, and C. Thrampoulidis. A model of double descent for high-dimensional logistic regression. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4267–4271, 2020.

[DLM19] Michał Dereziński, Feynman Liang, and Michael W Mahoney. Exact expressions for double descent and implicit regularization via surrogate random design. Technical Report arXiv preprint: 1912.04533, 2019.

[Don19] David Donoho. Comments on Michael Jordan's essay "the AI revolution hasn't happened yet". *Harvard Data Science Review*, June 2019.

[DR17] Gintare Karolina Dziugaite and Daniel M Roy. Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. In *Uncertainty in Artificial Intelligence (UAI)*, 2017.

[DS55] C. W. Dunnett and M. Sobel. Approximations to the Probability Integral and Certain Percentage Points of a Multivariate Analogue of Student's t-Distribution. *Biometrika*, 42(1/2):258, 6 1955.

[Dui00] R. W. Duin. Classifiers in almost empty spaces. In *Pattern Recognition, International Conference on*, volume 2, page 2001, 2000.

[DZ11] Petros Drineas and Anastasios Zouzias. A note on element-wise matrix sparsification via a matrix-valued Bernstein inequality. *Information Processing Letters*, 111(8):385–389, 3 2011.

[EdB01] A. Engel and C. P. L. Van den Broeck. *Statistical mechanics of learning*. Cambridge University Press, 2001.

[EVdB01] A. Engel and C. Van den Broeck. *Statistical Mechanics of Learning*. Cambridge University Press, 3 2001.

[FDP+20] Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M. Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *NeurIPS*, 2020.

[Fel68] William Feller. *An introduction to probability theory and its applications*. Wiley, 1968.

[FHL19] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

[FJGZ20] Ethan Fetaya, Jörn-Henrik Jacobsen, Will Grathwohl, and Richard Zemel. Understanding the limitations of conditional generative models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, April 2020.

[FKMN21] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, May 2021.

[FS18] Thomas Fung and Eugene Seneta. Quantile Function Expansion Using Regularly Varying Functions. *Methodology and Computing in Applied Probability*, 20(4):1091–1103, 12 2018.

[Gas17] Xavier Gastaldi. Shake-shake regularization. *CoRR*, abs/1705.07485, 2017.

[GJS+20] Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d'Ascoli, Giulio Biroli, Clément Hongler, and Matthieu Wyart. Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(2):023401, 2020.

[GKH20a] Alexandra Gessner, Oindrila Kanjilal, and Philipp Hennig. Integrals over Gaussians under Linear Domain Constraints. In *Proceedings of Machine Learning Research*, 2020.

[GKH20b] Alexandra Gessner, Oindrila Kanjilal, and Philipp Hennig. Integrals over Gaussians under linear domain constraints. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2764–2774, August 2020.

[GLL+15] Pascal Germain, Alexandre Lacasse, Francois Laviolette, Mario March, and Jean-Francis Roy. Risk bounds for the majority vote: From a PAC-Bayesian analysis to a learning algorithm. *Journal of Machine Learning Research*, 16(26):787–860, 2015.

[Glo] Glow in PyTorch. https://github.com/chrischute/glow. Accessed: 2021-05-08.

[GRS18] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-Independent Sample Complexity of Neural Networks. In *31st Conference On Learning Theory*, 2018.

[GSZ20] M. Gurbuzbalaban, U. Simsekli, and L. Zhu. The heavy-tail phenomenon in SGD. Technical Report Preprint: arXiv:2006.04740, 2020.

[GWB+18] Suriya Gunasekar, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Implicit regularization in matrix factorization. In *2018*

Information Theory and Applications Workshop, ITA 2018. Institute of Electrical and Electronics Engineers Inc., 10 2018.

[GWWM23] Margalit Glasgow, Colin Wei, Mary Wootters, and Tengyu Ma. Max-margin works while large margin fails: Generalization without uniform convergence. In *The Eleventh International Conference on Learning Representations*, 2023.

[HB02] Tin Kam Ho and Mitra Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300, March 2002.

[HBM+22] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022.

[HD19] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[HDMR21] Mahdi Haghifam, Gintare Karolina Dziugaite, Shay Moran, and Dan Roy. Towards a unified information-theoretic framework for generalization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 26370–26381. Curran Associates, Inc., 2021.

[HKSST96] David Haussler, Michael Kearns, H. Sebastian Seung, and Naftali Tishby. Rigorous learning curve bounds from statistical mechanics. *Machine Learning*, 25(2-3):195–236, 1996.

[HM20] L. Hodgkinson and M. W. Mahoney. Multiplicative noise and heavy tails in stochastic optimization,. Technical Report Preprint: arXiv:2006.06293, 2020.

[HMRT19] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in High-Dimensional Ridgeless Least Squares Interpolation. Technical Report arXiv preprint: 1903.08560, 2019.

[HNK+20] Mahdi Haghifam, Jeffrey Negrea, Ashish Khisti, Daniel M Roy, and Gintare Karolina Dziugaite. Sharpened generalization bounds based on conditional mutual information and an application to noisy, iterative algorithms. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9925–9935. Curran Associates, Inc., 2020.

[HRSG21] Hrayr Harutyunyan, Maxim Raginsky, Greg Ver Steeg, and Aram Galstyan. Information-theoretic generalization bounds for black-box learning algorithms. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[HVD+15] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[INLW21] Pavel Izmailov, Patrick Nicholson, Sanae Lotfi, and Andrew G Wilson. Dangers of Bayesian model averaging under covariate shift. *Advances in Neural Information Processing Systems*, 34:3309–3322, 2021.

[IVHW21] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are Bayesian neural network posteriors really like? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4629–4640. PMLR, 18–24 Jul 2021.

[JEP+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[JFY+20] Yiding Jiang, Pierre Foret, Scott Yak, Daniel M. Roy, Hossein Mobahi, Gintare Karolina Dziugaite, Samy Bengio, Suriya Gunasekar, Isabelle Guyon, and Behnam Neyshabur. Neurips 2020 competition: Predicting generalization in deep learning, 2020.

[JGH18] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems*, 2018.

[JNM+20] Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020.

[KAJ+20] Hussain Mohammed Dipu Kabir, Moloud Abdar, Seyed Mohammad Jafar Jalali, Abbas Khosravi, Amir F. Atiya, Saeid Nahavandi, and Dipti Srinivasan. SpinalNet: Deep neural network with gradual input. arXiv:2007.03347 [cs.CV]., September 2020.

[KD14] Abhisek Kundu and Petros Drineas. A Note on Randomized Element-wise Matrix Sparsification. Technical report, 2014.

[KD18] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1×1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10215–10224. Curran Associates, Inc., 2018.

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[KKB19] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in Deep Learning. Technical report, MIT, 2019.

[KMH+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[Kri09] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, April 2009.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[LBBH98] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[LCM20] Z. Liao, R. Couillet, and M. W. Mahoney. A random matrix analysis of random Fourier features: beyond the Gaussian kernel, a precise phase transition, and the corresponding double descent. Technical report, 2020. arXiv preprint: 2006.05013.

[LFK+22] Sanae Lotfi, Marc Anton Finzi, Sanyam Kapoor, Andres Potapczynski, Micah Goldblum, and Andrew Gordon Wilson. PAC-bayes compression bounds so tight that they can explain generalization. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[LMRR17] François Laviolette, Emilie Morvant, Liva Ralaivola, and Jean-Francis Roy. Risk upper bounds for general ensemble methods with an application to multiclass classification. *Neurocomputing*, 219:15–25, 2017.

[LNR22] Mufan Bill Li, Mihai Nica, and Daniel M. Roy. The neural covariance SDE: Shaped infinite depth-and-width networks at initialization. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[LPRS19] Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-Rao Metric, Geometry, and Complexity of Neural Networks. In *22nd International Conference on Artificial Intelligence and Statistics*, 2019.

[LSC22] Fanghui Liu, Johan Suykens, and Volkan Cevher. On the double descent of random features models trained with SGD. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[LSdP+18] Jaehoon Lee, Jascha Sohl-dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

[Mah12] M. W. Mahoney. Approximate computation and implicit regularization for very large-scale data analysis. In *Proceedings of the 31st ACM Symposium on Principles of Database Systems*, pages 143–154, 2012.

[MDP+11] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, June 2011.

[MH14] Kevin R. Moon and Alfred O. Hero, III. Multivariate f-divergence estimation with confidence. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2420–2428. Curran Associates, Inc., 2014.

[MHB17] Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic Gradient Descent as Approximate Bayesian Inference. *Journal of Machine Learning Research*, 18:1–35, 2017.

[MHD15] Kevin R. Moon, Alfred O. Hero, III, and Véronique Delouille. Meta learning of bounds on the Bayes classifier error. In *Proceedings of the 2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE)*, pages 13–18, August 2015.

[MLIS20] Andres Masegosa, Stephan Lorenzen, Christian Igel, and Yevgeny Seldin. Second order PAC-Bayesian bounds for the weighted majority vote. In *Advances in Neural Information Processing Systems*, volume 33, pages 5263–5273, 2020.

[MM17] C. H. Martin and M. W. Mahoney. Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior. Technical Report Preprint: arXiv:1710.09553, 2017.

[MM18] Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning, 2018.

[MM20] Charles H. Martin and Michael W. Mahoney. Heavy-tailed universality predicts trends in test accuracies for very large pre-trained deep neural networks, 2020.

[MM22] Charles H. Martin and Michael W. Mahoney. Post-mortem on a deep learning contest: a simpson's paradox and the complementary roles of scale metrics versus shape metrics, 2022.

[MPDM10] Iain Murray, Ryan Prescott, Adams David, and J C Mackay. Elliptical slice sampling. In *13th International Conference on Artificial Intelligence and Statistics*, 2010.

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, second edition, 2018.

[MSGH18] Kevin R. Moon, Kumar Sricharan, Kristjan Greenewald, and Alfred O. Hero, III. Ensemble estimation of information divergence. *Entropy*, 20(8):560, 2018.

[MWCC20] Cong Ma, Kaizheng Wang, Yuejie Chi, and Yuxin Chen. Implicit Regularization in Nonconvex Statistical Estimation: Gradient Descent Converges Linearly for Phase Retrieval, Matrix Completion, and Blind Deconvolution. *Foundations of Computational Mathematics*, 20(3):451–632, 6 2020.

[NAM21] Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. arXiv:2103.14749 [stat.ML]., March 2021.

[NBMS17] Behnam Neyshabur, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro. Exploring generalization in deep learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances*

in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[NBS18] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-Bayesian Approach to Spectrally-Normalized Margin Bound for Neural Networks. In *International Conference on Learning Representations*, 2018.

[Nie14] Frank Nielsen. Generalized Bhattacharyya and Chernoff upper bounds on Bayes error using quasi-arithmetic means. *Pattern Recognition Letters*, 42:25–34, June 2014.

[NK19] Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In *Advances in Neural Information Processing Systems*, 2019.

[NKB+20] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020.

[NSS15] Behnam Neyshabur, Ruslan R. Salakhutdinov, and Nati Srebro. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. In *28th Conference on Neural Information Processing Systems*, 2015.

[NTS15] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-Based Capacity Control in Neural Networks. In *28th Conference on Learning Theory*, 2015.

[NWC+11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, December 2011.

[NXH19] Morteza Noshad, Li Xu, and Alfred Hero. Learning to benchmark: Determining best achievable misclassification error from training data. arXiv:1909.07192 [stat.ML]., September 2019.

[OCnM22] Luis A. Ortega, Rafael Cabañas, and Andres Masegosa. Diversity and generalization in neural network ensembles. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 11720–11743, 28–30 Mar 2022.

[OFR+19] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

[OH91] Manfred Opper and David Haussler. Calculation of the Learning Curve of Bayes Optimal Classiication Algorithm for Learning a Perceptron With Noise. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 75–87, 1991.

[Pap] Paper with code for image classification datasets. https://paperswithcode.com/task/image-classification. Accessed: 2021-05-08.

[PM80] G. Pisier and B. Maurey. Remarques sur un résultat non publié de B. Maurey. *Séminaire Analyse fonctionnelle (dit "Maurey-Schwartz")*, pages 1–12, 1980.

[PNK+20] Pedram Pad, Simon Narduzzi, Clement Kundig, Engin Turetken, Siavash A. Bigdeli, and L. Andrea Dunbar. Efficient neural vision systems based on convolutional image acquisition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12285–12294, June 2020.

[PNR+21] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[PVG+11] 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*, 12:2825–2830, 2011.

[PXS11] Barnabás Póczos, Liang Xiong, and Jeff Schneider. Nonparametric divergence estimation with applications to machine learning on distributions. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI'11)*, pages 599–608, July 2011.

[QCHL87] J. R. Quinlan, P. J. Compton, K. A. Horn, and L. Lazarus. Inductive knowledge acquisition: A case study. In *Proceedings of the Second Australian Conference on Applications of Expert Systems*, page 137–156, USA, 1987. Addison-Wesley Longman Publishing Co., Inc.

[Rit00] K. Ritter. *Average-Case Analysis of Numerical Problems*. Number no. 1733 in Average-case Analysis of Numerical Problems. Springer, 2000.

[RRSS19] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR, 09–15 Jun 2019.

[RSR+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[SGd+19] S Spigler, M Geiger, S d'Ascoli, L Sagun, G Biroli, and M Wyart. A jamming transition from under- to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47):474001, oct 2019.

[SGT19] Yitong Sun, Anna Gilbert, and Ambuj Tewari. On the Approximation Properties of Random ReLU Features. Technical Report arXiv preprint: 1810.04374, 2019.

[SHS18] Daniel Soudry, Elad Hoffer, and Nathan Srebro. The implicit bias of gradient descent on separable data. In *International Conference on Learning Representations*, 2018.

[SQDC21] Yair Schiff, Brian Quanz, Payel Das, and Pin-Yu Chen. Predicting deep neural network generalization with perturbation response curves. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21176–21188. Curran Associates, Inc., 2021.

[Sri] Karthik Sridharan. Note on Refined Dudley Integral Covering Number Bound.

[SST92] H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical Review A*, 45(8):6056–6091, 1992.

[Ste62] G.P. Steck. Orthant Probabilities for the Equicorrelated Multivariate Normal Distribution. *Biometrika*, 49(3/4):433–445, 1962.

[SYT+22] Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Raluca Turc, Jacob Eisenstein, Dipanjan Das, and Ellie Pavlick. The multiBERTs: BERT reproductions for robustness analysis. In *International Conference on Learning Representations*, 2022.

[SZ20] Thomas Steinke and Lydia Zakynthinou. Reasoning About Generalization via Conditional Mutual Information. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 3437–3452. PMLR, 09–12 Jul 2020.

[TB23] Alexander Tsigler and Peter L. Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023.

[TKK21] Muhammad Suhaib Tanveer, Muhammad Umar Karim Khan, and Chong-Min Kyung. Fine-tuning DARTS for image classification. In *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*, pages 4789–4796, January 2021.

[TKM21] Ryan Theisen, Jason Klusowski, and Michael Mahoney. Good classifiers are abundant in the interpolating regime. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3376–3384. PMLR, 13–15 Apr 2021.

[TKW+19] Ryan Theisen, Jason M. Klusowski, Huan Wang, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. Global capacity measures for deep relu networks via path sampling, 2019.

[TKY+23] Ryan Theisen, Hyunsuk Kim, Yaoqing Yang, Liam Hodgkinson, and Michael W. Mahoney. When are ensembles really effective?, 2023.

[TWV+21] Ryan Theisen, Huan Wang, Lav R Varshney, Caiming Xiong, and Richard Socher. Evaluating state-of-the-art classification models against bayes optimality. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 9367–9377. Curran Associates, Inc., 2021.

[Var20] Lav R. Varshney. Addressing difference in orientation toward competition by bringing fundamental limits to AI challenges. In *NeurIPS workshop, ML Competitions at the Grassroots (CiML 2020)*, December 2020.

[VCR89] F. Vallet, J.-G. Cailton, and Ph Refregier. Linear and nonlinear extension of the pseudo-inverse solution for learning boolean functions. *Europhysics Letters*, 9(4):315, jun 1989.

[VKS19] Lav R. Varshney, Nitish Shirish Keskar, and Richard Socher. Pretrained AI models: Performativity, mobility, and change. arXiv:1909.03290 [cs.CY]., September 2019.

[VPL20] Guillermo Valle-Pérez and Ard A. Louis. Generalization bounds for deep learning, 2020.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[VWB16] Andreas Veit, Michael J. Wilber, and Serge Belongie. Residual Networks Behave Like Ensembles of Relatively Shallow Networks. In *29th Conference on Neural Information Processing Systems*, 2016.

[Wid] WideResnet in PyTorch. https://github.com/meliketoy/wide-resnet.pytorch. Accessed: 2021-05-08.

[WM19] Colin Wei and Tengyu Ma. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[WRB93] T. L. H. Watkin, A. Rau, and M. Biehl. The statistical mechanics of learning a rule. *Rev. Mod. Phys.*, 65(2):499–556, 1993.

[WSM+19] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

[WZ17] Lei Wu and Zhanxing Zhu. Towards Understanding Generalization of Deep Learning: Perspective of Loss Landscapes. In *ICML 2017 Workshop on Principled Approaches to Deep Learning*, 2017.

[XR17] Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747 [cs.LG]., September 2017.

[XSC19] Yue Xing, Qifan Song, and Guang Cheng. Benefit of Interpolation in Nearest Neighbor Algorithms. Technical Report arXiv preprint: 1909.11720, 2019.

[YHT+21] Yaoqing Yang, Liam Hodgkinson, Ryan Theisen, Joe Zou, Joseph E Gonzalez, Kannan Ramchandran, and Michael W Mahoney. Taxonomizing local versus global structure in neural network loss landscapes. In *Advances in Neural Information Processing Systems*, volume 34, pages 18722–18733, 2021.

[YTH+23] Yaoqing Yang, Ryan Theisen, Liam Hodgkinson, Joseph E. Gonzalez, Kannan Ramchandran, Charles H. Martin, and Michael W. Mahoney. Evaluating natural language processing models with generalization metrics that do not need access to any training or testing data, 2023.

[ZBH+17] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

[Zha02] Tong Zhang. Covering Number Bounds of Certain Regularized Linear Function Classes. *Journal of Machine Learning Research*, 2:527–550, 2002.

[ZK16] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. September 2016.

[ZLX20] Shuo Zhang, Yang Liu, and Lei Xie. Molecular mechanics-driven graph neural network with multiplex graph for molecular structures, 2020.

[ZTL22] Ruida Zhou, Chao Tian, and Tie Liu. Stochastic chaining and strengthened information-theoretic generalization bounds. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 690–695, 2022.

[ZVA+19] Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz. Non-Vacuous Generalization Bounds at the ImageNet Scale: A PAC-Bayesian Compression Approach. In *International Conference on Learning Representations*, 2019.

[ZWN+20] Peiliang Zhang, Huan Wang, Nikhil Naik, Caiming Xiong, and Richard Socher. DIME: An information-theoretic difficulty measure for AI datasets. In *NeurIPS 2020 Workshop DL-IG Blind Submission*, December 2020.

[ZZK+20] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13001–13008, Apr. 2020.

Appendix A

Chapter 2 Appendices

A.1 Proofs of main results

Theorem A.1. *Let $f(x;W)$ be an L-layer ReLU network, S a dataset, and let $1 \le q \le 2$. If $\tilde{p}$ is the Markov distribution formed from M samples from $p^{(q)}_{j_0,j_1,\dots,j_L}$, then*

$$\mathbb{E}_{\tilde{p}}\Big[\frac{1}{n}\sum_{x\in S}\|f(x;\widetilde{W}) - f(x;W)\|_2^2\Big] \le \Big(\frac{\mathcal{V}_q\zeta_q L}{\sqrt{M}}\Big)^2, \tag{A.1}$$

where $f(x;\widetilde{W}) = \mathcal{V}_q f(x;\tilde{p})$.

Proof. The proofs are the same for each $p^{(q)}$, so we write p for a generic path distribution, and $\mathcal{V}$ for a generic path variation (with the understanding that in the spectral case, we are considering ℓ_2 control on the inputs).

We can decompose the difference $f(x;p) - f(x;\tilde{p})$ into a telescoping sum

$$f(x;p) - f(x;\tilde{p}) = \sum_{\ell=1}^{L}[f^{\ell+1}(x;p,\tilde{p}) - f^{\ell}(x;p,\tilde{p})] \tag{A.2}$$

in which $f^{\ell+1}(x;p,\tilde{p})$ and $f^{\ell}(x;p,\tilde{p})$ differ only on layer ℓ, the former using $p_{j_{\ell-1}|j_\ell}$ and the later using $\tilde{p}_{j_{\ell-1}|j_\ell}$. That is, for each output unit j_L we let

$$f^{\ell}(x;p,\tilde{p})_{j_L} = \sum_{j_{L-1}} \tilde{p}_{j_L}\tilde{p}_{j_{L-1}|j_L}\phi\Big(\sum_{j_{L-2}}\tilde{p}_{j_{L-2}|j_{L-1}}\phi\Big(\cdots\phi\Big(\sum_{j_{\ell-1}}\tilde{p}_{j_{\ell-1}|j_\ell}\phi\Big(\sum_{j_{\ell-2}}p_{j_{\ell-2}|j_{\ell-1}}\Big(\cdots\phi\Big(\sum_{j_0}p_{j_0|j_1}x_{j_0}\Big)\Big)\Big)\Big)\Big)\Big).$$

In other words, $f^{\ell}(x;p,\tilde{p})$ is a network with weight matrices $(P_1,\dots,P_{\ell-1},\widetilde{P}_\ell,\widetilde{P}_{\ell+1},\dots,\widetilde{P}_L)$, where $P_\ell[j_\ell,j_{\ell-1}] = p_{j_{\ell-1}|j_\ell}$ and $\widetilde{P}_\ell[j_\ell,j_{\ell-1}] = \tilde{p}_{j_{\ell-1}|j_\ell}$ are transition matrices for the Markov

distributions p and $\tilde{p}$, respectively. Now let $\mathbb{P}_S$ be the empirical distribution for the sample S. Then using the triangle inequality for the L_2 norm associated with the joint distribution of $\tilde{p}$ and $\mathbb{P}_S$, we observe

$$\mathbb{E}_{\tilde{p}}\Big[\frac{1}{n}\sum_{x\in S}\|f(x;p)-f(x;\tilde{p})\|_2^2\Big] = \mathbb{E}_{\tilde{p},\mathbb{P}_S}[\|f(x;p)-f(x;\tilde{p})\|_2^2] \le \Big(\sum_{\ell} E_\ell\Big)^2 \tag{A.3}$$

where

$$E_\ell = \Big(\mathbb{E}_{\tilde{p}}\Big[\frac{1}{n}\sum_{x\in S}\|f^{\ell+1}(x;p,\tilde{p})-f^{\ell}(x;p,\tilde{p})\|_2^2\Big]\Big)^{1/2}.$$

We are therefore interested in bounding

$$\mathbb{E}_{\tilde{p}}\Big[\frac{1}{n}\sum_{x\in S}\sum_{j_L}|f^{\ell+1}(x;p,\tilde{p})_{j_L}-f^{\ell}(x;p,\tilde{p})_{j_L}|^2\Big]$$

for each ℓ. For $\ell = L$ we have

$$\frac{1}{n}\sum_{x\in S}\sum_{j_L}|f^{L+1}(x;p,\tilde{p})_{j_L}-f^{L}(x;p,\tilde{p})_{j_L}|^2 = \frac{1}{n}\sum_{x\in S}\sum_{j_L}\Big|\sum_{j_{L-1}}(\tilde{p}_{j_L,j_{L-1}}-p_{j_L,j_{L-1}})x_{j_{L-1}}(x)\Big|^2,$$

where $x_{j_{L-1}}(x)$ is the output of the network at the j_{L-1}th node entering the last layer. Then noticing that $\tilde{p}_{j_L,j_{L-1}} = \frac{1}{M}\sum_{i=1}^{M}\mathbb{1}((\tilde{j},\tilde{j}') = (j_L,j_{L-1}))$, where $\mathbb{1}((\tilde{j},\tilde{j}') = (j_L,j_{L-1})) \sim \mathrm{Bern}(p_{j_L,j_{L-1}})$, we may calculate

$$\begin{aligned}
&\frac{1}{n}\sum_{x\in S}\sum_{j_L}\mathbb{E}\Big[\Big|\sum_{j_{L-1}}(\tilde{p}_{j_L,j_{L-1}}-p_{j_L,j_{L-1}})x_{j_{L-1}}(x)\Big|^2\Big]\\
&= \frac{1}{n}\sum_{x\in S}\sum_{j_L}\Big(\Big[\sum_{j_{L-1}}p_{j_L,j_{L-1}}(x_{j_{L-1}}(x')-z_{j_L})^2\Big]+(1-p_{j_L})z_{j_L}^2\Big)\\
&\le \frac{1}{n}\sum_{x\in S}\frac{1}{M}\sum_{j_L}\sum_{j_{L-1}}p_{j_L,j_{L-1}}x_{j_{L-1}}^2(x')
\end{aligned}$$

where the last inequality follows from the fact that the MSE is minimized at the mean (so we can upper bound this term by plugging in $z_{j_L} = 0$). Using the Lipschitz property of ϕ, we have

$$\begin{aligned}
\frac{1}{n}\sum_{x\in S}x_{j_{L-1}}^2(x') &\le \frac{1}{n}\sum_{x\in S}\Big(\sum_{j_{L-2},j_{L-3},\ldots,j_0}p_{j_{L-2},\ldots,j_0|j_{L-1}}|x'_{j_0}|\Big)^2\\
&= \frac{1}{n}\sum_{x\in S}\Big(\sum_{j_0}p_{j_0|j_{L-1}}|x'_{j_0}|\Big)^2\\
&\le \frac{1}{n}\sum_{x\in S}\Big(\sum_{j_0}p_{j_0|j_{L-1}}|x'_{j_0}|^{q^*}\Big)^{2/q^*}\\
&\le \Big(\frac{1}{n}\sum_{x\in S}\sum_{j_0}p_{j_0|j_{L-1}}|x'_{j_0}|^{q^*}\Big)^{2/q^*}
\end{aligned}$$

where the last two inequalities follow from Jensen's inequality (since for $1 \le q \le 2$, we have $q^* \ge 2$, and hence z^{q^*} is convex and z^{2/q^*} is concave).

Next, we observe

$$\frac{1}{n}\sum_{x\in S}\sum_{j_0} p_{j_0|j_{L-1}}|x'_{j_0}|^{q^*} \le \frac{1}{n}\max_{j_0}\sum_{x\in S}|x'_{j_0}|^{q^*} = 1.$$

Hence we have

$$\frac{1}{n}\sum_{x\in S}\frac{1}{M}\sum_{j_L}\sum_{j_{L-1}} p_{j_L,j_{L-1}}x^2_{j_{L-1}}(x') \le \frac{1}{n}\sum_{x\in S}\frac{1}{M}\sum_{j_L}\sum_{j_{L-1}} p_{j_L,j_{L-1}} = \frac{1}{M}.$$

For $\ell = 1, 2, \ldots, L-1$, repeated application of the Lipschitz property of ϕ permits bounding each difference $\sum_{j_L}|f^{\ell+1}_{j_L}(x;p,\tilde{p}) - f^{\ell}_{j_L}(x;p,\tilde{p})|^2$ by

$$\sum_{j_L}\Big(\sum_{j_{L-1},\ldots,j_{\ell+1}} \tilde{p}_{j_L,\ldots,j_{\ell+1}}\Big|\sum_{j_\ell}\tilde{p}_{j_\ell|j_{\ell+1}}(\phi(\tilde{z}_{j_\ell}) - \phi(z_{j_\ell}))\Big|\Big)^2$$
$$= \sum_{j_L}\Big(\sum_{j_{\ell+1}}\tilde{p}_{j_L,j_{\ell+1}}\Big|\sum_{j_\ell}\tilde{p}_{j_\ell|j_{\ell+1}}(\phi(\tilde{z}_{j_\ell}) - \phi(z_{j_\ell}))\Big|\Big)^2$$

where $z_{j_\ell} = \sum_{j_{\ell-1}} p_{j_{\ell-1}|j_\ell}x_{j_{\ell-1}}$ and $\tilde{z}_{j_\ell} = \sum_{j_{\ell-1}}\tilde{p}_{j_{\ell-1}|j_\ell}x_{j_{\ell-1}}$. Since the quantities on the inside of the square are non-negative, and the sum of squares is less than the square of the sum, we have that this is at most

$$\Big(\sum_{j_L}\sum_{j_{\ell+1}}\tilde{p}_{j_L,j_{\ell+1}}\Big|\sum_{j_\ell}\tilde{p}_{j_\ell|j_{\ell+1}}(\phi(\tilde{z}_{j_\ell}) - \phi(z_{j_\ell}))\Big|\Big)^2 = \Big(\sum_{j_{\ell+1}}\tilde{p}_{j_{\ell+1}}\Big|\sum_{j_\ell}\tilde{p}_{j_\ell|j_{\ell+1}}(\phi(\tilde{z}_{j_\ell}) - \phi(z_{j_\ell}))\Big|\Big)^2$$

Using the triangle inequality and marginalizing, we get the further upper bound of

$$\Big(\sum_{j_\ell}\tilde{p}_{j_\ell}|\phi(\tilde{z}_{j_\ell}) - \phi(z_{j_\ell})|\Big)^2$$

It is shown in [BK18] that

$$\frac{1}{n}\sum_{x\in S}\mathbb{E}_{\tilde{p}}\Big(\sum_{j_\ell}\tilde{p}_{j_\ell}|\phi(\tilde{z}_{j_\ell}) - \phi(z_{j_\ell})|\Big)^2 \le \frac{1}{M}\Big(\sum_{j_\ell}\sigma_{j_\ell}\sqrt{p_{j_\ell}}\Big)^2,$$

where

$$\sigma^2_{j_\ell} = \frac{1}{n}\sum_{x\in S}\sigma^2_{j_\ell}(x') = \frac{1}{n}\sum_{x\in S}\sum_{j_{\ell-1}} p_{j_{\ell-1}|j_\ell}(x_{j_{\ell-1}}(x') - z_{j_\ell})^2$$

and $z_{j_\ell} = \sum_{j_{\ell-1}} p_{j_{\ell-1}|j_\ell}x_{j_{\ell-1}}$ are the variance and mean, respectively, of $x_{\tilde{j}_{\ell-1}}$ resulting from a single draw $\tilde{j}_{\ell-1} \sim p_{j_{\ell-1}|j_\ell}$. Bounding the latter term further using Jensen's inequality, we get

$$\begin{aligned}
\sigma^2_{j_\ell} &\le \frac{1}{n}\sum_{x\in S} x^2_{j_{\ell-1}}(x') \\
&\le \frac{1}{n}\sum_{x\in S}\Big(\sum_{j_0} p_{j_0|j_{\ell-1}}|x'_{j_0}|\Big)^2 \\
&\le \Big(\frac{1}{n}\sum_{x\in S}\sum_{j_0} p_{j_0|j_{\ell-1}}|x'_{j_0}|^{q^*}\Big)^{2/q^*}
\end{aligned}$$

which is at most 1 by the same reasoning as in the case of $\ell = L$. Hence we obtain

$$E_\ell^2 \le \frac{1}{M}(\sum_{j_\ell} \sqrt{p_{j_\ell}})^2 = \frac{1}{M}(e^{\frac{1}{2}H_{1/2}(p_\ell)})^2$$

Substituting this (as well as the bound of $\frac{1}{M}$ in the $\ell = L$ case) into (A.3), we obtain

$$\begin{aligned}\mathbb{E}_{\tilde{p}}\Big[\frac{1}{n}\sum_{x\in S}\|f(x;p) - f(x;\tilde{p})\|_2^2\Big] &\le \Big(\sum_\ell E_\ell\Big)^2 \\ &\le \Big(\frac{1}{\sqrt{M}} + \sum_{\ell=1}^{L-1}\frac{e^{\frac{1}{2}H_{1/2}(p_\ell)}}{\sqrt{M}}\Big)^2 \\ &= \frac{L^2\zeta^2}{M}\end{aligned}$$

Hence, multiplying both sides by $\mathcal{V}^2$, we get

$$\mathbb{E}_{\tilde{p}}\Big[\frac{1}{n}\sum_{x\in S}\|f(x;W) - f(x;\widetilde{W})\|_2^2\Big] \le \frac{\mathcal{V}^2\zeta^2L^2}{M}.$$

□

Theorem A.2. *Suppose $k = 1$, $L = 2$, $x \in \{-1,+1\}^d$, and $\mathbb{P}(x_{\tilde{j}_0} = +1|j_1) = \sum_{j_0:x_{j_0}=+1} p_{j_0|j_1} = 1/2$ and $\mathbb{P}(x_{\tilde{j}_0} = -1|j_1) = \sum_{j_0:x_{j_0}=-1} p_{j_0|j_1} = 1/2$ for all j_1. Then, for sufficiently large M,*

$$\mathbb{E}_{\tilde{p}}[|f(x;p) - f(x;\tilde{p})|^2] \ge \frac{\zeta_1^2}{32M},$$

where $\xi_1 = \frac{1}{2}(1 + \sum_{j_1}\sqrt{p_{j_1}})$.

Remark A.1. Note that the assumptions are satisfied if, for example, d is even, $p_{j_0|j_1} = 1/d$, and half of the coordinates of x are +1 and the other half are −1 (there are $\binom{d}{d/2}$ ways of choosing x in this way).

Proof. By the bias-variance decomposition,

$$\mathbb{E}_{\tilde{p}}[|f(x;p) - f(x;\tilde{p})|^2] = |f(x;p) - \mathbb{E}_{\tilde{p}}[f(x;\tilde{p})]|^2 + \mathrm{VAR}_{\tilde{p}}[f(x;\tilde{p})].$$

The assumptions imply that $f(x;p) = 0$. Hence,

$$\mathbb{E}_{\tilde{p}}|f(x;p) - f(x;\tilde{p})|^2 \ge |\mathbb{E}_{\tilde{p}}[f(x;\tilde{p})]|^2$$

Using the identity $\phi(z) = (z + |z|)/2$ and unbiasedness, we have

$$\mathbb{E}_{\tilde{p}}[f(x;\tilde{p})] = \sum_{j_1}\mathbb{E}_{\tilde{p}}[\tilde{p}_{j_1}\phi\big(\sum_{j_0}\tilde{p}_{j_0|j_1}x_{j_0}\big)] = \frac{1}{2}\sum_{j_1}\mathbb{E}_{\tilde{p}}[\tilde{p}_{j_1}\big|\sum_{j_0}\tilde{p}_{j_0|j_1}x_{j_0}\big|].$$

Next, using $\mathbb{P}(x_{\tilde{j}_0} = +1|j_1) = \mathbb{P}(x_{\tilde{j}_0} = -1|j_1) = 1/2$, we have

$$\mathbb{E}[\big|\sum_{j_0}\tilde{p}_{j_0|j_1}x_{j_0}\big|K_{j_1}] = \frac{1}{K_{j_1}}\mathbb{E}[\big|\sum_{i=1}^{K_{j_1}}\epsilon_i\big|],$$

where $\epsilon_i \overset{iid}{\sim} \text{Unif}\{-1,1\}$. By Khintchine's inequality,

$$\mathbb{E}[\big|\sum_{i=1}^{K_{j_1}}\epsilon_i\big|] \geq \sqrt{K_{j_1}/2}.$$

Thus, since $\tilde{p}_{j_1} = K_{j_1}/M$, we have

$$\mathbb{E}_{\tilde{p}}[\tilde{p}_{j_1}\big|\sum_{j_0}\tilde{p}_{j_0|j_1}x_{j_0}\big|] \geq \frac{1}{\sqrt{2}M}\mathbb{E}[\sqrt{K_{j_1}}].$$

Using a Taylor expansion of $z \mapsto \sqrt{z}$, it can be shown that $\mathbb{E}[\sqrt{K_{j_1}}] \geq \sqrt{Mp_{j_1}} - \frac{1-p_{j_1}}{2\sqrt{Mp_1}}$. The lower bound on $\mathbb{E}_{\tilde{p}}[f(x;\tilde{p})]$ is then

$$\frac{1}{2}\sum_{j_1}\mathbb{E}_{\tilde{p}}[\tilde{p}_{j_1}\big|\sum_{j_0}\tilde{p}_{j_0|j_1}x_{j_0}\big|] \geq \frac{1}{2\sqrt{2M}}\bigg(\sum_{j_1}\sqrt{p_{j_1}} - \frac{1}{2M}\sum_{j_1}\frac{1-p_{j_1}}{\sqrt{p_1}}\bigg).$$

For M sufficiently large, this expression is at least $\frac{\zeta_1}{4\sqrt{2M}}$, thus proving the claim. □

Theorem A.3. *The number of networks $f(x;\tilde{p})$ that arise from the sampling scheme is at most $8^{ML}(de)^M$. Thus, the log-cardinality of the representor set is bounded by $M(\log(de)+L\log(8))$.*

Proof. The proof makes use of the following fact. Let $I(k)$ denote the number of integer partitions of integers equal to k. Then $I(k) \leq 2^k$.

We will prove the claim by induction. Let $L = 2$. In this case, $\tilde{x}_{j_2} = f(x;\tilde{p})_{j_2}$ has the form

$$\tilde{x}_{j_2} = \phi\big(\sum_{j_1}\tilde{p}_{j_2}\tilde{p}_{j_1|j_2}\tilde{x}_{j_1}\big), \tag{A.4}$$

where $\tilde{x}_{j_1} = \phi\big(\sum_{j_0}\tilde{p}_{j_0|j_1}x_{j_0}\big)$. Let us now count the number of vectors $(\tilde{x}_{j_1})$. Note that for each j_1, the number of outputs $\tilde{x}_{j_1}$ is the number of nonnegative integers K_{j_0,j_1} that sum to K_{j_1}, or $\binom{K_{j_1}+d_0-1}{K_{j_1}}$. Thus, for a fixed sequence of integers (K_{j_1}) that sum to M, there are

$$\prod_{j_1}\binom{K_{j_1}+d_0-1}{K_{j_1}}$$

vectors $(\tilde{x}_{j_1})$. Summing over all integers K_{j_1} that sum to M yields that the number of vectors $(\tilde{x}_{j_1})$ is

$$N_1 = \sum_{(K_{j_1}):\sum_{j_1}K_{j_1}=M}\prod_{j_1}\binom{K_{j_1}+d_0-1}{K_{j_1}}.$$

Next, note that each $\tilde{p}_{j_2}\tilde{p}_{j_1|j_2} = \tilde{p}_{j_1,j_2}$ is built from counts K_{j_1,j_2} that sum to K_{j_2} for each fixed j_2. By permutation invariance of the sum (A.4), for a fixed nonnegative integer K_{j_2} and vector $(\tilde{x}_{j_1})$, each output $\tilde{x}_{j_2}$ provides at most $I(K_{j_2})$ different networks. Hence, for a fixed vector $(\tilde{x}_{j_1})$, since the K_{j_2} sum to M, the number of vectors $(\tilde{x}_{j_2})$ is

$$N_2 = \sum_{(K_{j_2}):\sum_{j_2} K_{j_2}=M} \prod_{j_2} I(K_{j_2}).$$

Hence the total number of vectors $(\tilde{x}_{j_2})$ is N_1N_2.

For general L, consider $\tilde{x}_{j_L} = f(x;\tilde{p})_{j_L}$, i.e.,

$$\tilde{x}_{j_L} = \phi\Big(\sum_{j_{L-1}} \tilde{p}_{j_L}\tilde{p}_{j_{L-1}|j_L}\tilde{x}_{j_{L-1}}\Big). \tag{A.5}$$

Note that each $\tilde{p}_{j_L}\tilde{p}_{j_{L-1}|j_L} = \tilde{p}_{j_{L-1},j_L}$ is built from counts K_{j_{L-1},j_L} that sum to K_{j_L} for each fixed j_L. By permutation invariance of the sum in (A.5), for a fixed nonnegative integer K_L and vector $(\tilde{x}_{j_{L-1}})$, each output $\tilde{x}_{j_L}$ provides at most $I(K_{j_L})$ different networks. Hence, for a fixed vector $(\tilde{x}_{j_{L-1}})$, since the K_{j_L} sum to M, the number of vectors $(\tilde{x}_{j_L})$ is

$$N_L = \sum_{(K_{j_L}):\sum_{j_L} K_{j_L}=M} \prod_{j_L} I(K_{j_L}).$$

By the induction step, the number of vectors $(\tilde{x}_{j_{L-1}})$ (each vector $(\tilde{x}_{j_{L-1}})$ is a depth $L-1$ network with d_{L-1} output nodes) is $N_1N_2\cdots N_{L-1}$. Hence, the total count is $N_1N_2\cdots N_L$.

Since at most M of the $\tilde{p}_{j_\ell|j_{\ell+1}}$ are nonzero, by relabeling the indices j_ℓ in each layer $\ell = 1, 2, \ldots, L$, we can assume that $d_\ell = M$. This means that

$$N := N_2 = N_3 = \cdots = N_L = \sum_{(K_j):\sum_{j=1}^M K_j=M} \prod_j I(K_j).$$

Next, note that $I(K_j) \le 2^{K_j}$ and hence $\prod_j I(K_j) \le 2^M$. Furthermore, $\sum_{(K_j):\sum_{j=1}^M K_j=M} 1 = \binom{2M-1}{M} \le 4^M$. Thus, $N \le 8^M$. As for N_1, we note that $\binom{K_{j_1}+d_0-1}{K_{j_1}} \le (2ed_0)^{K_{j_1}}$ and hence $N_1 \le 4^M(2ed_0)^M = (8ed_0)^M$. This shows that

$$N_1N_2\cdots N_L \le 8^{M(L-1)}(8ed_0)^M = 8^{ML}(d_0e)^M.$$

□

The following Corollary, mentioned in the main text, allows us to remove dependence on d when we have ℓ_2 constraints on the data.

Corollary A.1. *Let $\mathcal{S} = span(S)$ denote the subspace spanned by S. Let $W_1' = proj_{\mathcal{S}}(W_1)$ denote the orthogonal projection of the rowspace of W_1 onto $\mathcal{S}$. Let p' denote the path distribution induced by weight matrices $(W_1', W_2, \ldots, W_L)$. The number of networks $f(x; \tilde{p}')$ evaluated at the training data S is at most $8^{ML}(ne)^M$. Thus, the log-cardinality of the representor set is bounded by $M(\log(ne) + L\log(8))$.*

Proof. The effective input dimension of $f(x; p')$, acted on n data points S, is at most n. Hence, we obtain the conclusion from the previous lemma. □

To get the metric entropy bound that removes dependence on d, we first note that $f(x; p') = f(x; p)$ for $x \in S$. Furthermore, because an orthogonal projection is a bounded operator, if $\|x\|_2 \le r$, then $\mathcal{V}_2$ defined in terms of $(W_1', W_2, \ldots, W_L)$ can be bounded by the same quantities in terms of $(W_1, W_2, \ldots, W_L)$, i.e., $\|W_L \cdots W_2 W_1' w_0\| \le r\|W_L \cdots W_2 W_1\|$. These facts imply that an empirical cover of $\mathcal{V}' f(x; p')$ is also an empirical cover of $\mathcal{V} f(x; p)$ for $x \in S$.

Corollary A.2. *Let $\epsilon, \gamma > 0$, $1 \le q \le 2$. Then*

$$\log \mathcal{N}_2(\epsilon, \mathcal{F}_\gamma(\mathcal{V}_q, \zeta_q), S) \le \frac{9\mathcal{V}_q^2 \zeta_q^2 L^2 (L + \log(de))}{\gamma^2 \epsilon^2} \tag{A.6}$$

Proof. We first observe that R_γ is $\frac{1}{\gamma}$ Lipschitz. Moreover, Lemma A.2 in [BFT17] shows that for any j, $\mathcal{M}(\cdot, j)$ is 2-Lipschitz with respect to $\|\cdot\|_\infty$. Then for any network $f(x; W) \in \mathcal{F}(\mathcal{V}_q, \zeta_q)$, by Theorem A.1, we have that there exists $f(x; \widetilde{W})$ such that $n^{-1} \sum_{x \in S} \|f(x; W) - f(x; \widetilde{W})\|_2^2 \le \left(\frac{\mathcal{V}_q \zeta_q L}{\sqrt{M}}\right)^2$. Then

$$\begin{aligned}
&\frac{1}{n} \sum_{(x,y): x \in S} |R_\gamma(-\mathcal{M}(f(x; W), y)) - R_\gamma(-\mathcal{M}(f(x; \widetilde{W}), y))|^2 \\
&\le \frac{1}{n} \sum_{(x,y): x \in S} \frac{1}{\gamma^2} |\mathcal{M}(f(x; W), y) - \mathcal{M}(f(x; \widetilde{W}), y)|^2 \\
&\le \frac{1}{n} \sum_{(x,y): x \in S} \frac{4}{\gamma^2} \|f(x; W) - f(x; \widetilde{W})\|_2^2 \\
&\le \left(\frac{2\mathcal{V}_q \zeta_q L}{\gamma \sqrt{M}}\right)^2
\end{aligned}$$

The results is thus an immediate consequence of Theorem A.3 with $M_\epsilon = \left(\frac{2\mathcal{V}_q \zeta_q L}{\gamma \epsilon}\right)^2$. Note that the factor of 9 arises from the additional factor of $\log(8)$ in the cardinality bound. □

Theorem A.4. *Let $f(x;W)$ be an L-layer positive homogeneous network and let $\delta \in (0,1)$. For any $1 \le q \le 2$ and $\gamma > 0$, with probability at least $1-\delta$ over the training set S, the generalization error $\ell(f) - \hat{\ell}_\gamma(f)$ is bounded by*

$$\widetilde{\mathcal{O}}\Big(\frac{\mathcal{V}_q \zeta_q L\sqrt{L+\log(d)}}{\gamma\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{n}}\Big), \tag{A.7}$$

where $\mathcal{V}_q, \zeta_q$ are the q- path variation and path complexity of f.

To prove this, we first prove the generalization bound for the classes $\mathcal{F}_\gamma(\mathcal{V}_q, \zeta_q)$ with *a priori* bounded path variation and path complexity, and then take a union bound to obtain a *post hoc* guarantee.

We first recall the *empirical Rademacher complexity* of a class of real-valued functions $\mathcal{G}$ with respect to a dataset $S = \{x^1, ..., x^n\}$:

$$\widehat{\mathcal{R}}_S(\mathcal{G}) = \mathbb{E}_\epsilon\Big[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^n \epsilon_i g(x^i)\Big]$$

where $\epsilon_i \overset{iid}{\sim} \mathrm{Unif}\{-1,1\}$. For our purposes, the utility of the empirical Rademacher complexity is captured by the following standard result.

Lemma 4 ([MRT18])**.** *Let $\mathcal{G}$ be a class of functions with values in $[0,1]$. Then for any $\delta > 0$, with probability at least $1-\delta$ over S, for all $g \in \mathcal{G}$ we have*

$$\mathbb{E}[g(x)] \le \frac{1}{n}\sum_{i=1}^n g(x^i) + 2\widehat{\mathcal{R}}_S(\mathcal{G}) + 3\sqrt{\frac{\log(2/\delta)}{2n}}$$

To bound to empirical Rademacher complexity, we use a standard bound via a Dudley entropy integral.

Lemma 5 (Dudley entropy integral; see e.g. note by [Sri])**.** *For a class of functions $\mathcal{G}$ with values in $[0,1]$ and a dataset S of n points, we have*

$$\widehat{\mathcal{R}}_S(\mathcal{G}) \le \inf_{\alpha\ge 0}\left[4\alpha + 12\int_\alpha^1 \sqrt{\frac{\log \mathcal{N}(\epsilon, \mathcal{G}, S)}{n}}d\epsilon\right]$$

Using these results with Lemma A.2, we may obtain the following.

Lemma 6. *Let $\delta \in (0,1)$, $\gamma > 0$, $1 \le q \le 2$. Then with probability at least $1-\delta$ over an i.i.d. draw of S , we have for all $f \in \mathcal{F}_\gamma(\mathcal{V}_q, \zeta_q)$*

$$\ell(f) \le \hat{\ell}_\gamma(f) + \frac{8}{n} + \frac{48\mathcal{V}_q\zeta_q L\sqrt{L+\log(ed)}\log(n)}{\gamma\sqrt{n}} + 3\sqrt{\frac{\log(2/\delta)}{2n}} \tag{A.8}$$

Proof. Define

$$A^2 = \frac{4\mathcal{V}_q^2\zeta_q^2 L^2(L+\log(ed))}{\gamma^2 n}$$

so that $\log \mathcal{N}_2(\epsilon, \mathcal{F}_\gamma(\mathcal{V}_q,\zeta_q), S)/n = \frac{A^2}{\epsilon^2}$. Then by Lemma 5, we have that

$$\widehat{\mathcal{R}}_S(\mathcal{F}_\gamma(\mathcal{V}_q,\zeta_q)) \le \inf_{\alpha\ge 0}\left[4\alpha + 12A\int_\alpha^1 \frac{1}{\epsilon}d\epsilon\right] = \inf_{\alpha\ge 0}\left[4\alpha + 12A\log(1/\alpha)\right]$$

It is easy to verify that the above expression is minimized at $\alpha = 3A$, though to keep the expression somewhat cleaner, we use to choice $\alpha = \frac{1}{n}$, which produces the bound

$$\widehat{\mathcal{R}}_S(\mathcal{F}_\gamma(\mathcal{V}_q,\zeta_q)) \le \frac{4}{n} + A\log(n) = \frac{4}{n} + \frac{24\mathcal{V}_q\zeta_q L\sqrt{L+\log(ed)}\log(n)}{\gamma\sqrt{n}}.$$

The result follows from Lemma 4 together with the fact that $\ell(f) \le \mathbb{E}[R_\gamma(f(x;W),y)]$ and $\frac{1}{n}\sum_{(x,y):x\in S} R_\gamma(f(x;W),y) \le \hat{\ell}_\gamma(f)$. □

The above gives a generalization guarantee for the class $\mathcal{F}_\gamma(\mathcal{V}_q,\zeta_q)$ with *a priori* bounded path variation and path complexity, and given $\gamma > 0$. Namely, it gives a statement of the form

$\forall$ *classes* $\mathcal{F}_\gamma(\mathcal{V}_q,\zeta_q)$ *we have with probability at least* $1-\delta$ *over the training set* S *that* $\forall$ $f \in \mathcal{F}_\gamma(\mathcal{V}_q,\zeta_q)$ *the bound (A.8) holds.*

However, in practice, we do not have such guarantees on the on the size of these quantities before seeing the data. In order to obtain post hoc guarantees for a network $f(x;W)$, we need to prove a statement of the form

With probability at least $1-\delta$ *over the training set* S, $\forall$ *classes* $\mathcal{F}_\gamma(\mathcal{V}_q,\zeta_q)$ *we have that* $\forall$ $f \in \mathcal{F}_\gamma(\mathcal{V}_q,\zeta_q)$ *the bound (A.8) holds.*

To prove a statement of the latter form, we must instead instantiate the above bound for many values of $\mathcal{V},\zeta,\gamma$ and take a union bound. The below approach to doing so is similar to that of [BFT17].

Proof of Theorem A.4. Given integers (j_1, j_2, j_3), define the instances

$$\mathcal{B}(j_1, j_2, j_3) = \left\{(\gamma, S, W) : 0 < \frac{1}{\gamma} < \frac{2^{j_1}}{\sqrt{n}},\ \mathcal{V}_q(W) \le j_2,\ \zeta_q(W) \le j_3\right\}$$

And for $\delta \in (0,1)$, divide δ as

$$\delta(j_1, j_2, j_3) = \frac{\delta}{2^{j_1} j_2(j_2+1) j_3(j_3+1)}$$

so that by construction $\sum_{j_1,j_2,j_3 \in \mathbb{N}} \delta(j_1, j_2, j_3) = \delta$. Then by Lemma 6, we have that for every $(j_1, j_2, j_3) \in \mathbb{N}^3$, we have with probability at least $1 - \delta(j_1, j_2, j_3)$ that for all instances $(\gamma, S, W) \in \mathcal{B}(j_1, j_2, j_3)$,

$$\ell(f) \le \hat{\ell}_\gamma(f) + \frac{8}{n} + \frac{48 \cdot 2^{j_1} \cdot j_2 \cdot j_3 \cdot L\sqrt{L + \log(ed)}\log(n)}{n} + 3\sqrt{\frac{\log(2/\delta(j_1, j_2, j_3))}{2n}} \tag{A.9}$$

$$\le \hat{\ell}_\gamma(f) + \frac{8}{n} + \frac{48 \cdot 2^{j_1} \cdot j_2 \cdot j_3 \cdot L\sqrt{L + \log(ed)}\log(n)}{n} \tag{A.10}$$

$$+ 3\sqrt{\frac{\log(2/\delta) + \log(2^{j_1}) + 2\log(j_2+1) + 2\log(j_3+1)}{2n}} \tag{A.11}$$

Then taking a union bound over the integers (j_1, j_2, j_3), we have that (A.9-A.11) holds simultaneously over all $\mathcal{B}(j_1, j_2, j_3)$ with probability at least $1 - \delta$. Then for a *given* γ, X, and $f(x; W)$ with path variation and complexity $\mathcal{V}_q, \zeta_q$, Let j_1^*, j_2^*, j_3^* be the smallest integers such that $\frac{1}{\gamma} \le \frac{2^{j_1^*}}{\sqrt{n}}, \mathcal{V}_q \le j_2^*$, and $\zeta_q \le j_3^*$. Then we have by definition that $2^{j_1^*} \le \frac{2\sqrt{n}}{\gamma}$, $j_2^* \le \mathcal{V}_q + 1$ and $j_3^* \le \zeta_q + 1$. Plugging these values in and cleaning up with the notation $\widetilde{\mathcal{O}}$ yields the stated result. □

A.2 Additional results mentioned in the main text

Bounding normalizing constants

Lemma 7 (Induced norms as normalizing constants)**.** *Let* $1 \le q \le \infty$ *and define* $w_{j_0} = (n^{-1}\sum_{x \in S} |x_{j_0}|^q)^{1/q}$. *Then*

$$\sum_{j_0, j_1, \ldots, j_L} w_{j_0} w_{j_0, j_1, \ldots, j_L} \le (\max_{x \in S} \|x\|_q) k^{1-1/q} \|W_L W_{L-1} \cdots W_1\|_q$$

where $\|\cdot\|_q$ *is the matrix norm induced by the vector* q *norm.*

Proof. We observe that this is the same as showing that

$$\|W_L W_{L-1} \cdots W_1 w_0\|_1 \le r k^{1-1/q} \|W_L W_{L-1} \cdots W_1 w_0\|_q$$

where $w_0 = [w_1, w_2, ..., w_d]^\top$ (the vector containing the values w_{j_0}). Notice that

$$W_L W_{L-1} \cdots W_1 w_0$$

is simply a vector in $\mathbb{R}^k$, and hence by an application of Hölder's inequality, we have

$$\|W_L W_{L-1} \cdots W_1 w_0\|_1 \leq k^{1-1/q} \|W_L W_{L-1} \cdots W_1 w_0\|_q$$

Since the vector q norm induces the mstrix q norm, we have that this is at most

$$k^{1-1/q} \|W_L W_{L-1} \cdots W_1\|_q \|w_0\|_q \leq r k^{1-1/q} \|W_L W_{L-1} \cdots W_1\|_q.$$

□

Lemma 8 ($(q,1)$ norms as normalizing constants)**.** *Let $1 \leq q \leq \infty$, and let q^* be its conjugate exponent. Then*

$$\mathcal{V}_q \leq (\max_{x \in S} \|x\|_{q^*}) \Big\| \prod_1^L |W_\ell| \Big\|_{q,1}.$$

Proof. We observe from Hölder's inequality,

$$\begin{aligned}
\mathcal{V}_q &= \sum_{j_L} \sum_{j_0} w_{j_0} \sum_{j_1, \ldots, j_{L-1}} w_{j_0, \ldots, j_L} \\
&\leq \sum_{j_L} \Big(\sum_{j_0} \Big(\sum_{j_1, \ldots, j_{L-1}} w_{j_0, \ldots, j_L} \Big)^q \Big)^{1/q} \Big(\sum_{j_0} w_{j_0}^{q^*} \Big)^{1/q^*} \\
&\leq (\max_{x \in S} \|x\|_{q^*}) \Big\| \prod_1^L |W_\ell| \Big\|_{q,1}.
\end{aligned}$$

□

Lemma 9. *We have*

$$\mathcal{V}_2 \leq \sum_{j_L} \Big(\sum_{j_0, j_1, \ldots, j_{L-1}} w_{j_0, j_1, \ldots, j_L}^2 \Big)^{1/2}$$

where in the single output case, the right-hand side is equal to the 2−path norm ϕ_2 from [NTS15].

Proof. This can be seen by repeated application of the Cauchy-Schwarz inequality, as follows. Assume, without loss of generality, that $r = 1$, so that $S \subseteq \mathbb{B}_2(1)$. Then we have

$$\sum_{(j_0, j_1, \ldots, j_L)} w_{j_0} w_{j_0, j_1, \ldots, j_L} = \sum_{(j_1, j_2, \ldots, j_L)} w_{j_1, j_2, \ldots, j_L} \sum_{j_0} w_{j_0} w_{j_0, j_1}.$$

Then, for each j_1, we apply the Cauchy-Schwarz inequality to the sum $\sum_{j_0} w_{j_0} w_{j_0, j_1}$, yielding the bound

$$\sum_{(j_1, j_2, \ldots, j_L)} w_{j_1, j_2, \ldots, j_L} \Big(\sum_{j_0} w_{j_1, j_0}^2 \Big)^{1/2} \Big(\sum_{j_0} w_{j_0}^2 \Big)^{1/2}.$$

By the ℓ_2 condition on the inputs, $(\sum_{j_0} w_{j_0}^2)^{1/2} \le r$. Continuing similarly, we have for each $\ell = 1, 2, \ldots, L$,

$$\sum_{(j_\ell, j_{\ell+1}, \ldots, j_L)} w_{j_\ell, j_{\ell+1}, \ldots, j_L} \Big(\sum_{(j_0, j_1, \ldots, j_{\ell-1})} w_{j_0, j_1, \ldots, j_\ell}^2 \Big)^{1/2}$$
$$= \sum_{(j_{\ell+1}, j_{\ell+2}, \ldots, j_L)} w_{j_{\ell+1}, j_{\ell+2}, \ldots, j_L} \times \sum_{j_\ell} w_{j_{\ell+1}, j_\ell} \Big(\sum_{(j_0, j_1, \ldots, j_{\ell-1})} w_{j_0, j_1, \ldots, j_\ell}^2 \Big)^{1/2}.$$

As before, for each $j_{\ell+1}$, we apply the Cauchy-Schwarz inequality to the sum

$$\sum_{j_\ell} w_{j_{\ell+1}, j_\ell} \Big(\sum_{(j_0, j_1, \ldots, j_{\ell-1})} w_{j_0, j_1, \ldots, j_\ell}^2 \Big)^{1/2},$$

yielding the bound

$$\sum_{(j_{\ell+1}, j_{\ell+2}, \ldots, j_L)} w_{j_{\ell+1}, j_{\ell+2}, \ldots, j_L} \Big(\sum_{(j_0, j_1, \ldots, j_\ell)} w_{j_0, j_1, \ldots, j_{\ell+1}}^2 \Big)^{1/2}.$$

Repeating this procedure to $\ell = L - 1$, we may obtain the stated bound. □

Details of pooling case

We begin with some basic properties of the max/average pooling operator $\mathcal{P}_{\mathcal{Z}} : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}, \mathcal{Y}$ are vector spaces with dimension $d_{\mathcal{X}}, d_{\mathcal{Y}}$ and $\mathcal{Z}$ is a collection of subsets $\{Z_1, \ldots, Z_{d_{\mathcal{Y}}}\}$ where $Z_i \subset \{1, \ldots, d_{\mathcal{X}}\}$. For a given input $X \in \mathcal{X}$, $\mathcal{P}_{\mathcal{Z}}$ computes $[\mathcal{P}_{\mathcal{Z}}(X)_1, \ldots, \mathcal{P}_{\mathcal{Z}}(X)_{d_{\mathcal{Y}}}]^T$ where

$$\mathcal{P}_{\mathcal{Z}}^{\mathsf{max}}(X)_i = \max_{j \in Z_i} X_j.$$

for max pooling and

$$\mathcal{P}_{\mathcal{Z}}^{\mathsf{avg}}(X)_i = \frac{1}{|Z_i|} \sum_{j \in Z_i} X_j.$$

for average pooling. The argument is the same for both, so we simply use $\mathcal{P}$ to denote either max or average pooling. Now given weight matrices $W_{\ell+1}, W_\ell$ with $W_{\ell+1} \in \mathbb{R}^{d_{\ell+1}, d_\ell}$ and $W_\ell \in \mathbb{R}^{d'_\ell, d_{\ell-1}}$ with positive entries, a pooling layer can be written as

$$z_{j_{\ell+1}}(x) = \sum_{j_\ell} w_{j_{\ell+1}, j_\ell} \mathcal{P}_{j_\ell} \big(\phi \big(\sum_{j_{\ell-1}} w_{j'_\ell, j_{\ell-1}} x_{j_{\ell-1}}(x) \big) \big)$$

To handle the signs of $w_{j_{\ell+1}, j_\ell}$, we may now double the number of units d_ℓ and have $\mathcal{P}_\ell(X)$ compute $[\mathcal{P}_\ell(X)_1, \ldots, \mathcal{P}_\ell(X)_{d_\ell}, -\mathcal{P}_\ell(X)_1, \ldots, -\mathcal{P}_\ell(X)_{d_\ell}]$ and adjust weights accordingly. A

typical term in the analysis of Theorem A.1 with pooling will now look like

$$\begin{aligned}
&\sum_{j_L} |f^{\ell+1}(x;p,\tilde{p})_{j_L} - f^{\ell}(x;p,\tilde{p})_{j_L}| \\
&\leq \sum_{j_L} \sum_{j_{L-1},\ldots,j_{\ell+1}} \tilde{p}_{j_L,j_{L-1},\ldots,j_{\ell+1}} \Big| \sum_{j_\ell} \tilde{p}_{j_\ell|j_{\ell+1}} (\mathcal{P}_{j_\ell}(\phi(\tilde{z}_{j_\ell}(x))) - \mathcal{P}_{j_\ell}(\phi(z_{j_\ell}(x)))) \Big| \\
&\leq \sum_{j_L} \sum_{j_\ell} \tilde{p}_{j_L,j_\ell} |\mathcal{P}_{j_\ell}(\phi(\tilde{z}_{j_\ell}(x))) - \mathcal{P}_{j_\ell}(\phi(z_{j_\ell}(x)))| \\
&= \sum_{j_\ell} \tilde{p}_{j_\ell} \Big| \mathcal{P}_{j_\ell}(\phi(\tilde{z}_{j_\ell}(x))) - \mathcal{P}_{j_\ell}(\phi(z_{j_\ell}(x))) \Big| \\
&\leq \sum_{j'_\ell} \tilde{p}_{j'_\ell} |A_{j'_\ell}(x)|
\end{aligned}$$

where now $A_{j'_\ell}(x) = \sum_{j_{\ell-1}} (\tilde{p}_{j_{\ell-1}|j'_\ell} - p_{j_{\ell-1}|j'_\ell}) z_{j_{\ell-1}}(x)$, which is the same as the term appearing in the case without pooling. [Note we just need to define $K_{j_\ell} = \sum_{j'_\ell \in Z_{j_\ell}} K_{j'_\ell}$ in our counts.]

Computational aspects of sampling

As we mention briefly in the main text, to generate samples from Multinomial(M, p) directly, we would need to store and sample from the full path distribution $p_{j_0,j_1,\ldots,j_L}$, which quickly becomes unwieldy as L grows, since it involves storing a (potentially dense) L-tensor. It turns out, however, that we can store only the conditional distributions, which are just matrices, and can be computed easily from the collection of successive matrix products $\{W_\ell W_{\ell-1} \cdots W_1 : \ell = 1, 2, \ldots, L\}$ (the collection itself can be inductively constructed), since

$$p_{j_\ell|j_{\ell+1}} = w_{j_{\ell+1},j_\ell} \frac{\|W_\ell[j_\ell,]W_{\ell-1}\cdots W_1\|_1}{\|W_{\ell+1}[j_{\ell+1},]W_\ell \cdots W_1\|_1}$$

and

$$p_{j_L} = \frac{\|W_L[j_L,]W_{L-1}\cdots W_1\|_1}{\mathcal{V}},$$

where $W_\ell[j_\ell,]$ (resp. $W_\ell[,j_{\ell-1}]$) is row (resp. column) j_ℓ (resp. $j_{\ell-1}$) of W_ℓ. Thus, the conditional probabilities are reweighted versions of the weight matrices. Given these matrices, a sample $K \sim$ Multinomial(M, p) can be generated in $\mathcal{O}(LM)$ time by repeating the following M times:

- Sample $\tilde{j}_L \sim p_{j_L}$
- Sample $\tilde{j}_{L-1} \sim p_{j_{L-1}|\tilde{j}_L}$
  $\vdots$
- Sample $\tilde{j}_0 \sim p_{j_0|\tilde{j}_1}$

Appendix B

Chapter 3 Appendices

B.1 Technical Results

Here, we provide proofs of our main results.

Proofs of Theorems 3.1 and 3.2

Proof of Theorem 3.1. The assumption (3.12) on the correlation structure of the data implies that

$$\zeta_i \overset{d}{=} \sqrt{1-\rho}Z_i + \sqrt{\rho}Z, \tag{B.1}$$

for $i = 1, 2, \ldots, n+m$, where $Z, Z_1, Z_2, \ldots, Z_{n+m}$ are i.i.d. $\mathcal{N}(0,1)$. Let $a = \sqrt{\frac{\rho}{1-\rho}}$. According to (11), $\mathbb{P}(\mathsf{VS}(S_n))$ can be expressed as

$$\begin{aligned}\mathbb{P}(\zeta_1 \geq 0, \zeta_2 \geq 0, \ldots, \zeta_n \geq 0) &= \mathbb{E}_{Z\sim N(0,1)}[(\Phi(aZ))^n] \\ &= \int_{-\infty}^{\infty} (\Phi(az))^n \phi(z) dz,\end{aligned} \tag{B.2}$$

where $\phi(\cdot)$ is the density of a $\mathcal{N}(0,1)$ distribution. Make the change of variables $u = n(1-\Phi(az))$. Then the integral (B.2) becomes

$$\frac{(2\pi)^{\frac{1}{2}(\frac{1}{a^2}-1)}}{na} \int_0^n (1-u/n)^n (\phi(\Phi^{-1}(1-u/n)))^{\frac{1}{a^2}-1} du, \tag{B.3}$$

where $\Phi^{-1}(\cdot)$ is the quantile function of $\mathcal{N}(0,1)$. Next, consider the so-called "density quantile function" $\phi(\Phi^{-1}(v))$. Using a standard asymptotic expression for Mills' ratio [Fel68], we have

$$\frac{1-\Phi(x)}{\phi(x)} = \frac{1}{x}\Big(1 + O\Big(\frac{1}{x^2}\Big)\Big), \quad x \to \infty. \tag{B.4}$$

Furthermore, the quantile function $\Phi^{-1}(v)$ has the following asymptotic expression [FS18]

$$\Phi^{-1}(v) = \sqrt{2\log(1/(1-v))}\left(1 + O\left(\frac{\log\log(1/(1-v))}{\log(1/(1-v))}\right)\right), \tag{B.5}$$

as $v \uparrow 1$. Combining these two facts ((B.4) and (B.5)) yields

$$\phi(\Phi^{-1}(v)) = (1-v)\sqrt{2\log(1/(1-v))}\times \left(1 + O\left(\frac{\log\log(1/(1-v))}{\log(1/(1-v))}\right)\right), \quad v \uparrow 1.$$

Using this asymptotic expression for $\phi(\Phi^{-1}(v))$, we find that

$$\int_0^n (1-u/n)^n (\phi(\Phi^{-1}(1-u/n)))^{\frac{1}{a^2}-1} du$$

from (B.3) is asymptotically

$$n^{1-1/a^2}(2\log(n))^{\frac{1}{2}(\frac{1}{a^2}-1)}\times \int_0^n \left(1 + \frac{\log(1/u)}{\log(n)}\right)^{\frac{1}{2}(\frac{1}{a^2}-1)} u^{1/a^2-1}e^{-u}du. \tag{B.6}$$

Finally, by the dominated convergence theorem, $\int_0^n (1 + \frac{\log(1/u)}{\log(n)})^{\frac{1}{2}(\frac{1}{a^2}-1)} u^{1/a^2-1}e^{-u}du$ is asymptotically

$$\int_0^\infty u^{1/a^2-1}e^{-u}du = \Gamma(1/a^2).$$

Therefore, (B.6) is asymptotically

$$n^{1-1/a^2}(2\log(n))^{\frac{1}{2}(\frac{1}{a^2}-1)}\Gamma(1/a^2). \tag{B.7}$$

Combining (B.3) and (B.7), we have

$$\mathbb{P}(\zeta_1 \geq 0, \zeta_2 \geq 0, \ldots, \zeta_n \geq 0) \sim \frac{\Gamma(1/a^2)}{a}(4\pi\log(n))^{\frac{1}{2}(\frac{1}{a^2}-1)}n^{-1/a^2}. \tag{B.8}$$

Having given an asymptotic expression for the orthant probabilities, we turn our attention to the ratio $\mathbb{P}(y_{n+1} = f(\boldsymbol{x}_{n+1}) \mid \mathsf{VS}(S_n))$, which equals

$$\frac{\mathbb{P}(\{y_{n+1} = f(\boldsymbol{x}_{n+1})\} \cap \mathsf{VS}(S_n))}{\mathbb{P}(\mathsf{VS}(S_n))}. \tag{B.9}$$

Next, we recognize that the set $\{y_{n+1} = f(\boldsymbol{x}_{n+1})\} \cap \mathsf{VS}(S_n)$ is another version space with $n+1$ sample points and the same correlation structure as before, per the assumption of the

theorem. Therefore, using the above asymptotic formula (B.8) for $\mathbb{P}(\zeta_1 \geq 0, \zeta_2 \geq 0, \ldots, \zeta_n \geq 0)$, we have that (B.9) is asymptotically

$$\frac{\frac{\Gamma(1/a^2)}{a}(4\pi\log(n+1))^{\frac{1}{2}(\frac{1}{a^2}-1)}(n+1)^{-1/a^2}}{\frac{\Gamma(1/a^2)}{a}(4\pi\log(n))^{\frac{1}{2}(\frac{1}{a^2}-1)}n^{-1/a^2}} = 1 - \frac{1/a^2}{n} + \frac{1/a^2-1}{2n\log(n)} + O(1/n^2),$$

thus completing the proof. □

Proof of Theorem 3.2. To begin, note that by definition,

$$R_{n,m}(\varepsilon) = \frac{\mathbb{P}(\{\mathcal{E}_m(f) \leq \varepsilon\} \cap \mathsf{VS}(S_n))}{\mathbb{P}(\mathsf{VS}(S_n))}. \tag{B.10}$$

Using the representation (B.1) of ζ_i in terms of Z and Z_i, we have $\mathcal{E}_m(f) \overset{d}{=} \frac{1}{m}\sum_{i=1}^m \mathbb{1}(Z_i < -aZ)$ and $\mathcal{E}(f) \overset{d}{=} \lim_m \frac{1}{m}\sum_{i=1}^m \mathbb{1}(Z_i < -aZ)$, where $a = \sqrt{\frac{\rho}{1-\rho}}$. Henceforth, we take these distributional equivalents as the definitions of $\mathcal{E}_m(f)$ and $\mathcal{E}(f)$. Now,

$$\mathbb{P}(\{\mathcal{E}_m(f) \leq \varepsilon\} \cap \mathsf{VS}(S_n)) = \mathbb{E}_{Z\sim N(0,1)}[\mathbb{P}(\mathcal{E}_m(f) \leq \varepsilon \mid Z)\Phi^n(aZ)].$$

By the strong law of large numbers, given Z, $\frac{1}{m}\sum_{i=1}^m \mathbb{1}(Z_i < -aZ)$ converges almost surely (with respect to the test data S_{test} and $Z_1, Z_2, \ldots$) to its mean $\Phi(-aZ) = 1 - \Phi(aZ)$. Thus, by the dominated convergence theorem, almost surely, $\lim_m \mathbb{P}(\mathcal{E}_m(f) \leq \varepsilon \mid Z) = \mathbb{1}(1 - \Phi(aZ) \leq \varepsilon)$. Therefore, it follows that, almost surely, $\mathbb{P}(\mathcal{E}(f) \leq \varepsilon \mid Z) = \mathbb{1}(1 - \Phi(aZ) \leq \varepsilon)$. Next,

$$\begin{aligned} R_n(\varepsilon) &= \frac{\mathbb{P}(\{\mathcal{E}(f) \leq \varepsilon\} \cap \mathsf{VS}(S_n))}{\mathbb{P}(\mathsf{VS}(S_n))} \\ &= \frac{\mathbb{E}_{Z\sim N(0,1)}[\mathbb{1}(1 - \Phi(aZ) \leq \varepsilon)\Phi^n(aZ)]}{\mathbb{E}_{Z\sim N(0,1)}[\Phi^n(aZ)]} \\ &= \frac{\int_0^{\varepsilon n}(1-u/n)^n(\phi(\Phi^{-1}(1-u/n)))^{\frac{1}{a^2}-1}du}{\int_0^{n}(1-u/n)^n(\phi(\Phi^{-1}(1-u/n)))^{\frac{1}{a^2}-1}du}, \end{aligned} \tag{B.11}$$

where for the final equality, we use (B.3) from the proof of Theorem 3.1. Using the same techniques as Theorem 3.1 to derive asymptotic integral expressions therein (in fact, the integrands of the integrals are identical), (B.11) is asymptotically equivalent to

$$\frac{\int_0^{\varepsilon n} u^{1/a^2-1}e^{-u}du}{\int_0^{\infty} u^{1/a^2-1}e^{-u}du} = \mathbb{P}(U \leq n\varepsilon),$$

which proves the first claim (3.14).

To prove the second claim (3.15) about the critical value, let $c > 0$ be arbitrary. Then,

$$\mathbb{P}(U \leq n(\epsilon^* + c)) \to 1,$$

provided $n\rho \to \infty$. On the other hand, for $n\rho$ large enough, $(1-\rho)/\rho - nc < 0$ and hence,

$$\mathbb{P}(U \leq n(\epsilon^* - c)) = 0. \quad \square$$

A new asymptotic formula for the orthant probability of equicorrelated Gaussians

A consequence of the proof of Theorem 3.1 is the following asymptotic expression for the orthant probability of an equicorrelated Gaussian which, to the best of our knowledge, is new. This is referred to as Lemma 3 in the main text.

Corollary B.1 (Asymptotic expression for orthant probability in equicorrelated case)**.** *Let* $\rho \in [0,1)$ *and* $(X_1, ..., X_n) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ *with* $\mathbf{\Sigma}_{ij} = \rho$ *for* $i \neq j$ *and* $\Sigma_{ii} = 1$ *for all* i*. Then as* $n\rho \to \infty$*,*

$$\mathbb{P}(X_1 \geq 0, X_2 \geq 0, \ldots, X_n \geq 0) \sim \sqrt{\frac{1-\rho}{\rho}}\Gamma\left(\frac{1-\rho}{\rho}\right)(4\pi \log(n))^{\frac{1}{2}(\frac{1-\rho}{\rho}-1)} n^{-\frac{1-\rho}{\rho}}.$$

Lower bound on test error for Gaussian model

Recall the Gaussian data model:

$$(\boldsymbol{x}, y) \sim \frac{1}{2}(N_+, 1) + \frac{1}{2}(N_-, -1) \tag{B.12}$$

where $N_+ \sim \mathcal{N}(\mu, \Sigma)$, $N_- \sim \mathcal{N}(-\mu, \Sigma)$ and $\mu \in \mathbb{R}^d$, $\Sigma \in \mathcal{S}_+^d$. For this model, we have the lower bound

$$\mathcal{E}(\boldsymbol{w}) \geq \varepsilon^* \geq \Phi(-\sqrt{\mu^\top \Sigma^{-1} \mu}) \tag{B.13}$$

To see this, define the norm $\|\boldsymbol{x}\|_\Sigma^2 = \boldsymbol{x}^\top \Sigma \boldsymbol{x}$. The value ε^* satisfies

$$\begin{aligned} -\Phi^{-1}(\varepsilon^*) &= \operatorname*{ess\,sup}_{\boldsymbol{w} \in \mathsf{VS}} \frac{\boldsymbol{w}^\top \mu}{\sqrt{\boldsymbol{w}^\top \Sigma \boldsymbol{w}}} \leq \sup_{\boldsymbol{w} \in \mathsf{VS}} \frac{\boldsymbol{w}^\top \mu}{\sqrt{\boldsymbol{w}^\top \Sigma \boldsymbol{w}}} = \sup_{\boldsymbol{w} \in \mathsf{VS}, \|\boldsymbol{w}\|_\Sigma = 1} \boldsymbol{w}^\top \mu \\ &\leq \sup_{\|\boldsymbol{w}\|_\Sigma = 1} \boldsymbol{w}^\top \mu = \|\mu\|_{\Sigma^{-1}} = \sqrt{\mu^\top \Sigma^{-1} \mu} \end{aligned}$$

where we use the fact that $\|\cdot\|_{\Sigma^{-1}}$ is the dual norm to $\|\cdot\|_\Sigma$. Hence solving for ε^*, we get the lower bound

$$\varepsilon^* \geq \Phi(-\sqrt{\mu^\top \Sigma^{-1} \mu}).$$

When $\Sigma = \sigma^2 I$, this lower bound reduces to the usual signal-to-noise ratio $\Phi(-\frac{\|\mu\|}{\sigma})$.

B.2 Review of LIN-ESS Algorithm and Additional Empirical Results

Review of LIN-ESS

In this section, we briefly review the LIN-ESS algorithm introduced in [GKH20a], which is the main computational tool we use in our empirical evaluation. LIN-ESS builds on the Elliptical

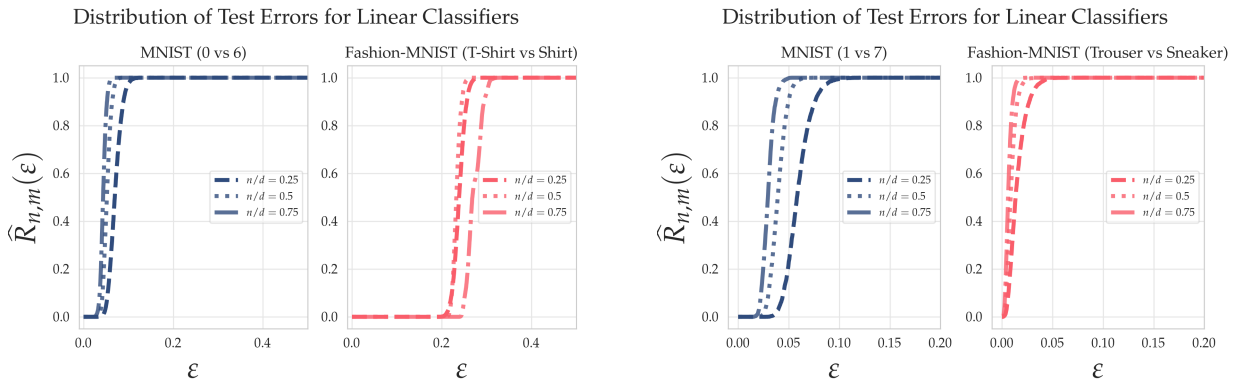


Figure B.1: Test error distributions for linear classifiers, similar to Figure 2 in the main text. Here we try four different binary problems; 0 vs 1 classification for MNIST, T-shirt vs Shirt classification for FASHION-MNIST, 1 vs 7 classification for MNIST and Trouser vs Sneaker classification for FASHION-MNIST. While we observe similar qualitative behavior as in the main text, we do note substantial differences in the location of the distributions based on the difficulty of the task. For example, note that T-shirt vs Shirt classification is significantly more difficult than 0 vs 6 classification, while Trouser vs Sneaker classification is easier that 0 vs 1 classification.

Slice Sampling algorithm [MPDM10], which can be used to sample from a posterior under a $\mathcal{N}(\mu, \Sigma)$ prior and generic likelihood L. The generic algorithm works as follows: given a starting point $\boldsymbol{x}_0$ and a new sample $\boldsymbol{x}' \sim \mathcal{N}(\mu, \Sigma)$, construct an ellipse passing through these two points:

$$\boldsymbol{x}(\theta) = \boldsymbol{x}_0 \cos(\theta) + \boldsymbol{x}' \sin(\theta).$$

We then sample an angle $\hat{\theta}$ randomly and accept $\boldsymbol{x}(\hat{\theta})$ if the likelihood at this point $L(\boldsymbol{x}(\hat{\theta}))$ is sufficiently large. Otherwise, we sample a new angle in a narrower band of feasible values. While a general and provably valid[1] algorithm, this procedure can be slow, as many samples may be rejected before finding an acceptable sample.

The key insight of [GKH20a] is that when the likelihood has the form $L(\boldsymbol{x}) = \prod_{i=1}^{n} \mathbb{1}(\mathbf{a}_i^\top \boldsymbol{x} + b_i \geq 0)$, the region of feasible angles θ can be obtained analytically, avoiding the need to reject infeasible $\hat{\theta}$. This results in significantly faster computation, even in high dimensions.

Additional empirical results

In Figures B.1-B.4 we provide additional empirical results, complementing the results presented in the main text.

[1] Meaning, it can be shown to have the true posterior as a unique stationary distribution.

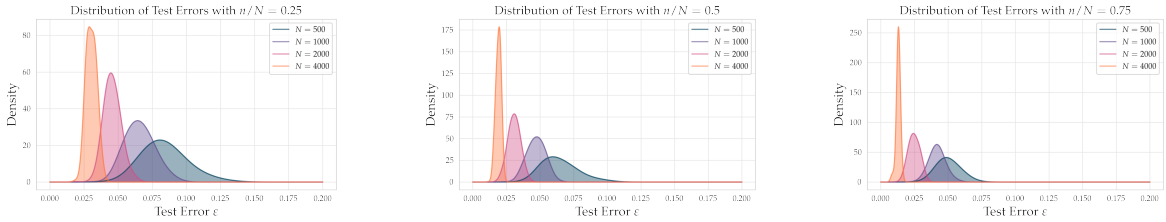


Figure B.2: Test error distributions for random ReLU feature classifiers on MNIST. Here we plot PDFs, similar to Figure 3.1, which we fit using a Gaussian kernel density estimator. These plots are not as precise as the CDF plots shown in the main text, but are more usually for seeing visually the concentration phenomenon as $N \to \infty$.

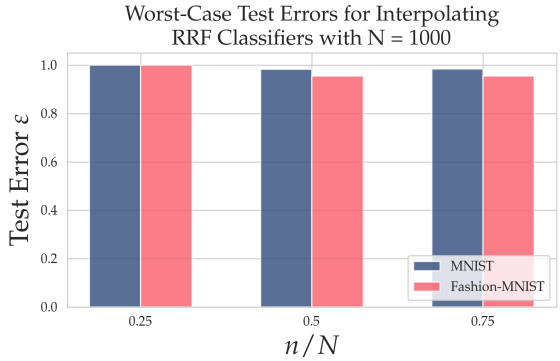


Figure B.3: Example of worst case test errors for random ReLU feature models, for $N = 1000$ random features. We observe again that bad classifiers do indeed exist, despite an abundance of classifiers with low test error.

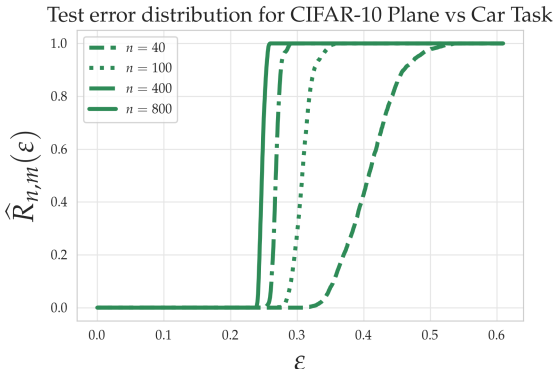


Figure B.4: As an additional example, here we plot the test error distribution for linear classification on the plane vs car task on the CIFAR10 dataset.

Appendix C

Chapter 4 Appendices

C.1 Proof of Proposition 4.2

Proposition 2. (From main text) *The Bayes error of flow models is monotonically increasing in τ. That is, for $0 < \tau \leq \tau'$, we have that $\mathcal{E}_{Bayes}(\hat{p}_\tau) \leq \mathcal{E}_{Bayes}(\hat{p}_{\tau'})$.*

Proof. Note that at temperature τ, the Bayes error is given by

$$\mathcal{E}_{\text{Bayes}}(\hat{p}_\tau) = 1 - \sum_{k=1}^{K} \pi_k \int \prod_{j \neq k} \mathbb{1}(\mathbf{a}_{jk}^\top \boldsymbol{z} + b_{jk} > 0) \mathcal{N}(d\boldsymbol{z}; \boldsymbol{\mu}_k, \tau^2 \boldsymbol{\Sigma}) \tag{C.1}$$

$$= 1 - \sum_{k=1}^{K} \pi_k \int \prod_{j \neq k} \mathbb{1}(\tilde{\mathbf{a}}_{jk}^\top \boldsymbol{z} + \frac{\tilde{b}_{jk}}{\tau} > 0) \mathcal{N}(d\boldsymbol{z}; \mathbf{0}, \mathbf{I}) \tag{C.2}$$

where $\tilde{\mathbf{a}}_{jk} = 2\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_j)$, $\tilde{b}_{jk} = (\boldsymbol{\mu}_k - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_j) \geq 0$. Then it easy to see that for $0 < \tau \leq \tau'$ and $\boldsymbol{z} \in \mathbb{R}^d$, we have that

$$\prod_{j \neq k} \mathbb{1}(\tilde{\mathbf{a}}_{jk}^\top \boldsymbol{z} + \frac{\tilde{b}_{jk}}{\tau} > 0) \geq \prod_{j \neq k} \mathbb{1}(\tilde{\mathbf{a}}_{jk}^\top \boldsymbol{z} + \frac{\tilde{b}_{jk}}{\tau'} > 0) \tag{C.3}$$

which implies that $\mathcal{E}_{\text{Bayes}}(\hat{p}_\tau) \leq \mathcal{E}_{\text{Bayes}}(\hat{p}_{\tau'})$. □

C.2 Further empirical results

Hardness of Classes

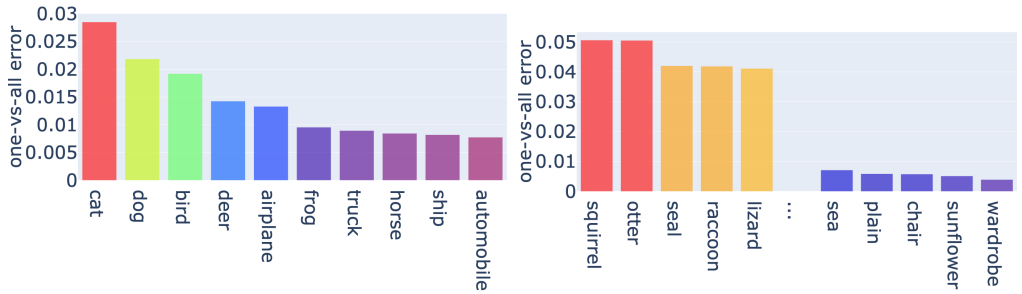
In addition to measuring the difficulty of classification tasks relative to one another, it also may be of interest to evaluate the relative difficulty of individual classes within a particular task. A natural way to do this is by looking at the error of one-vs-all classification

tasks. Specifically, for a given class $j \in \mathcal{K}$, we consider $(\boldsymbol{x}, 1)$ drawn from the distribution $p_{-j}(\boldsymbol{x}) = \frac{1}{1-\pi_j} \sum_{i \neq j} \pi_i p_i(\boldsymbol{x})$, and $(\boldsymbol{x}, 0)$ from $p_j(\boldsymbol{x})$. The optimal Bayes classifier in this task is

$$C_{\text{Bayes}}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } -\log p_j(\boldsymbol{x}) \leq -\log p_{-j}(\boldsymbol{x}), \\ 1 & \text{otherwise} \end{cases}.$$

Unfortunately, in this case, the Bayes error cannot be computed with HDR integration, since p_{-j} is now a mixture of Gaussians. However, we can get a reasonable approximation for the error (though less accurate than exact integration would be) in this case using a simple Monte Carlo estimator: $\widehat{\mathcal{E}}_{\text{Bayes}} = \frac{1}{m} \sum_{l=1}^{m} \mathbb{1}(C_{\text{Bayes}}(\boldsymbol{x}_l) \neq y_l)$, where $y_l \sim \text{Unif}\{0, 1\}$ and $\boldsymbol{x}_l \mid y_l \sim y_l p_{-j} + (1 - y_l) p_j$ as prescribed above.

The one-vs-all errors by class on CIFAR are shown in Figure C.1. It is observed that the errors between the hardest class and the easiest class is huge. On CIFAR-100 the error of the hardest class, squirrel, is almost 5 times that of the easiest class, wardrobe.

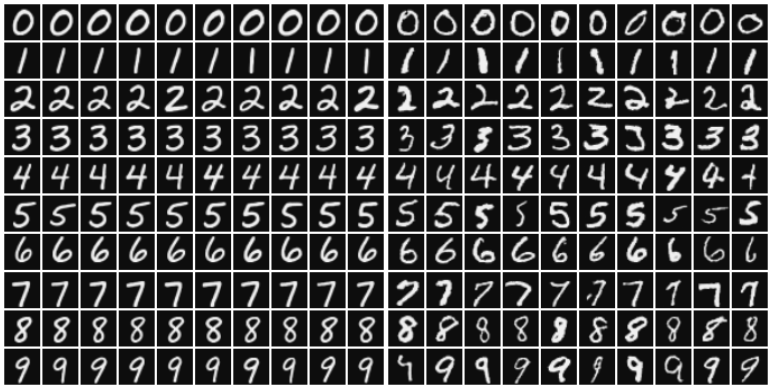


(a) CIFAR-10 (b) CIFAR-100

Figure C.1: Classes Ranked by Hardness

Additional samples and Bayes errors from flow models

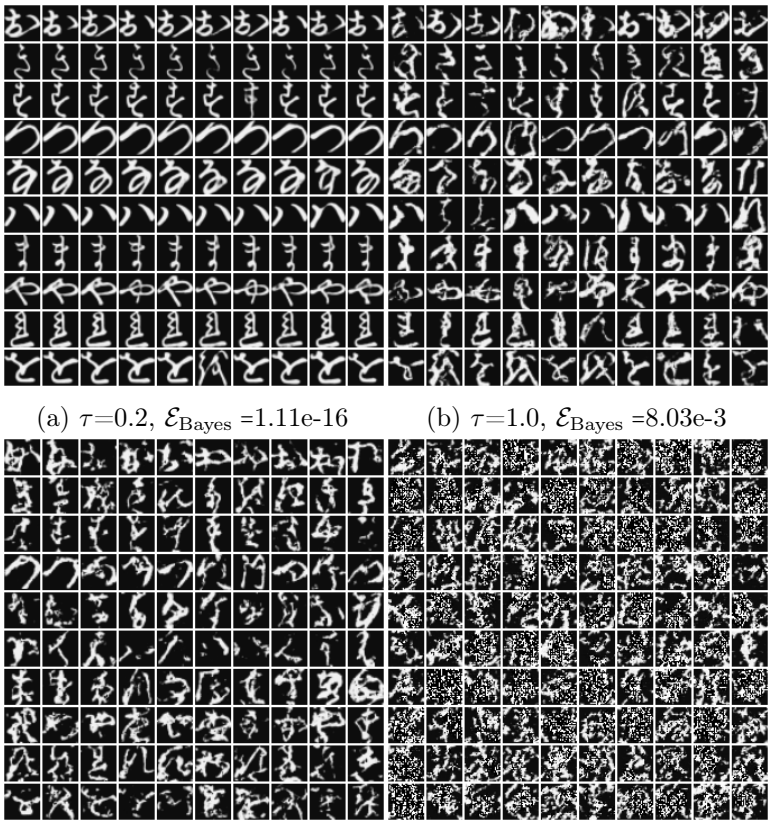
Below we include examples generating by the trained flow models, and additional datasets generated at different temperatures, and hence Bayes errors.



(a) τ=0.2, $\mathcal{E}_{\text{Bayes}}$ =1.11e-16 (b) τ=1.0, $\mathcal{E}_{\text{Bayes}}$ =1.07e-4

(c) τ=1.4, $\mathcal{E}_{\text{Bayes}}$ =7.00e-3 (d) τ=3.0, $\mathcal{E}_{\text{Bayes}}$ =2.91e-1

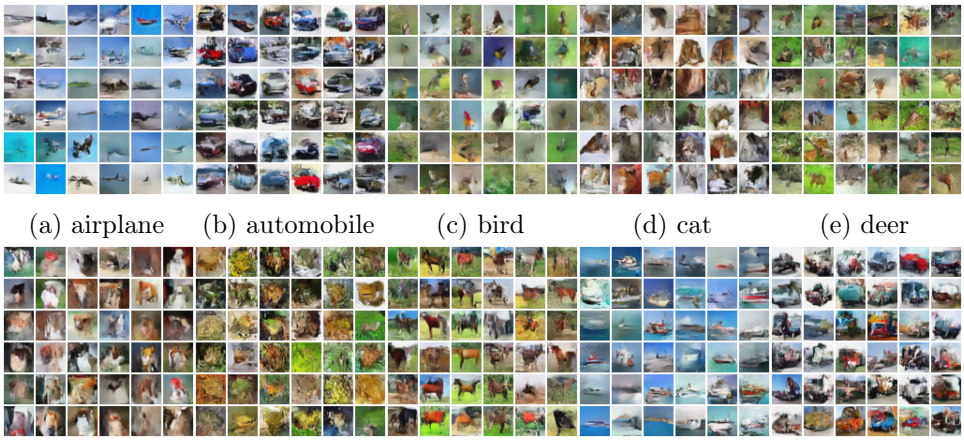
Figure C.2: Generated MNIST Samples with Different Temperatures



(a) τ=0.2, $\mathcal{E}_{\text{Bayes}}$ =1.11e-16 (b) τ=1.0, $\mathcal{E}_{\text{Bayes}}$ =8.03e-3

(c) τ=1.4, $\mathcal{E}_{\text{Bayes}}$ =7.21e-2 (d) τ=3.0, $\mathcal{E}_{\text{Bayes}}$ =4.80e-1

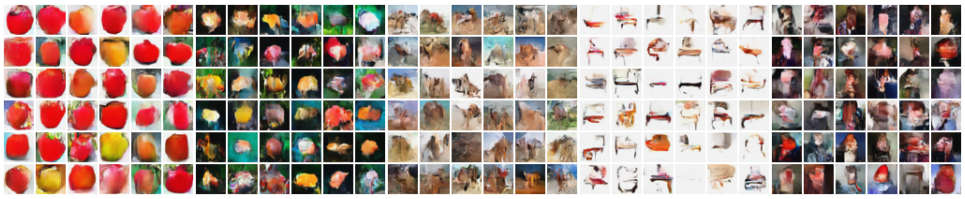
Figure C.3: Generated Kuzushiji-MNIST Samples with Different Temperatures



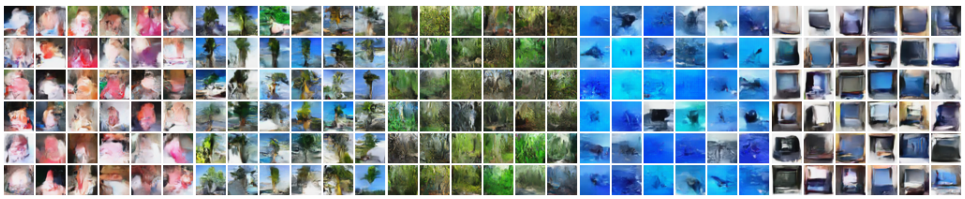
(a) airplane (b) automobile (c) bird (d) cat (e) deer

(f) dog (g) frog (h) horse (i) ship (j) truck

Figure C.4: Samples generated from conditional GLOW model trained on CIFAR-10.



(a) apple (b) aquarium fish (c) camel (d) chair (e) man



(f) baby (g) palm tree (h) forest (i) whale (j) television

Figure C.5: Samples generated from conditional GLOW model trained on CIFAR-100.

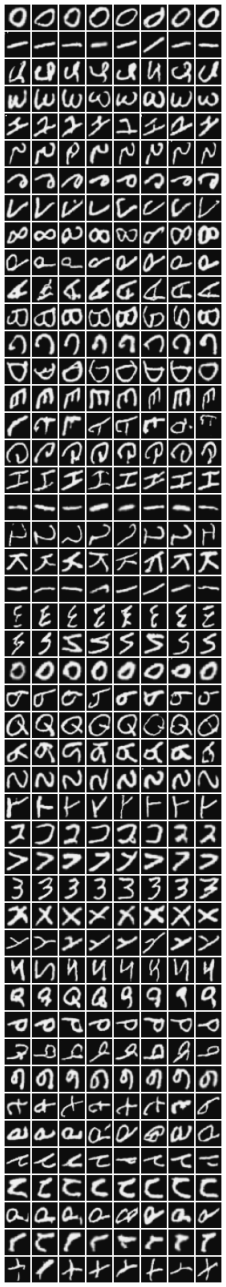


Figure C.6: Samples generated from conditional GLOW model trained on EMNIST (balanced). Estimated Bayes Error is 0.09472.

Appendix D

Chapter 5 Appendices

D.1 Proofs of our main results

In this section, we provide proofs for our main results. Throughout the section, we denote $\mathbb{P}_{h,h'\sim\rho^2}$, $\mathbb{P}_{h\sim\rho}$, $\mathbb{E}_{h\sim\rho}$ by $\mathbb{P}_{h,h'}$, $\mathbb{P}_h$, $\mathbb{E}_h$, respectively. We also denote $W_\rho(X,Y)$ simply by W_ρ. We typically omit explicit dependence on the data distribution $\mathcal{D}$ when it is apparent from context.

Proof of Theorem 5.1

We first state and prove two lemmas that will be used in the proof of Theorem 5.1. Our first lemma states that majority-vote error $L_{\mathcal{D}}[h_{\mathrm{MV}}]$ is upper bounded by probability of $W_\rho(X,Y)$ being large.

Lemma 10. *There is the inequality* $L_{\mathcal{D}}[h_{\mathrm{MV}}] \le \mathbb{P}_{\mathcal{D}}(W_\rho(X,Y) \ge 1/2)$, *where* $L_{\mathcal{D}}(h) = \mathbb{E}_{\mathcal{D}}[\mathbb{1}(h(X) \ne Y)]$, $h_{\mathrm{MV}}(\boldsymbol{x}) = \arg\max_j \mathbb{E}_h[\mathbb{1}(h(\boldsymbol{x}) = j)]$ *and* $W_\rho(X,Y) = \mathbb{E}_h[\mathbb{1}(h(X) \ne Y)]$.

Proof. For given data point x, $W_\rho < 1/2$ implies that we are predicting the true label correctly more than half of the time. Thus, the majority vote classifier will correctly predict the label on the data point. □

Our next lemma states a property of competent classifiers which plays a crucial role in the main proof.

Lemma 11. *Under Assumption 5.1 (competence), for any increasing function h satisfying $h(0) = 0$,*

$$\mathbb{E}_{\mathcal{D}}[h(W_\rho)\mathbb{1}_{W_\rho<1/2}] \ge \mathbb{E}_{\mathcal{D}}[h(\bar{W}_\rho)\mathbb{1}_{\bar{W}_\rho\le 1/2}],$$

where $\bar{W}_\rho = 1 - W_\rho$.

Proof. For every $x \in [0,1]$,

$$\mathbb{P}_{\mathcal{D}}(W_\rho \mathbb{1}_{W_\rho<1/2} \geq x) = \mathbb{P}_{\mathcal{D}}(W_\rho \in [x, 1/2))\, \mathbb{1}_{x\leq 1/2},$$
$$\mathbb{P}_{\mathcal{D}}(\bar{W}_\rho \mathbb{1}_{\bar{W}_\rho\leq 1/2} \geq x) = \mathbb{P}_{\mathcal{D}}(\bar{W}_\rho \in [x, 1/2])\, \mathbb{1}_{x\leq 1/2} = \mathbb{P}_{\mathcal{D}}(W_\rho \in [1/2, 1-x])\, \mathbb{1}_{x\leq 1/2}.$$

From Assumption 5.1, this implies that $\mathbb{P}_{\mathcal{D}}(W_\rho \mathbb{1}_{W_\rho<1/2} \geq x) \geq \mathbb{P}_{\mathcal{D}}(\bar{W}_\rho \mathbb{1}_{\bar{W}_\rho\leq 1/2} \geq x)$ for all $x \in [0,1]$. Therefore, for any increasing function h satisfying $h(0) = 0$, since $h(x\,\mathbb{1}_{x\leq c}) = h(x)\mathbb{1}_{x\leq c}$,

$$\mathbb{P}_{\mathcal{D}}(h(W_\rho)\mathbb{1}_{W_\rho<1/2} \geq x) \geq \mathbb{P}_{\mathcal{D}}(h(\bar{W}_\rho)\mathbb{1}_{\bar{W}_\rho\leq 1/2} \geq x).$$

As W_ρ is non-negative, the equality $\mathbb{E}X = \int_0^\infty \mathbb{P}(X \geq x)\mathrm{d}x$ concludes the proof. □

With these two lemmas, we now provide the proof of Theorem 5.1.

Proof of Theorem 5.1. From Lemma 10 and the relation $\mathbb{E}_h[L_{\mathcal{D}}(h)] = \mathbb{E}_{\mathcal{D}}[W_\rho]$ (Fubini's theorem), it suffices to show that $\mathbb{P}_{\mathcal{D}}(W_\rho \geq 1/2) \leq \mathbb{E}_{\mathcal{D}}[W_\rho]$. To do so, observe

$$\mathbb{E}_{\mathcal{D}}[(W_\rho - 1)\mathbb{1}_{W_\rho\geq 1/2}] + \mathbb{E}_{\mathcal{D}}[\bar{W}_\rho \mathbb{1}_{\bar{W}_\rho\leq 1/2}] = \mathbb{E}_{\mathcal{D}}[(W_\rho - 1)\mathbb{1}_{W_\rho\geq 1/2}] + \mathbb{E}_{\mathcal{D}}[(1 - W_\rho)\mathbb{1}_{W_\rho\geq 1/2}] = 0.$$

Applying Lemma 11 with $h(x) = x$,

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[W_\rho] - \mathbb{P}_{\mathcal{D}}(W_\rho \geq 1/2) &\geq \mathbb{E}_{\mathcal{D}}[(W_\rho - 1)\mathbb{1}_{W_\rho\geq 1/2}] + \mathbb{E}_{\mathcal{D}}[W_\rho \mathbb{1}_{W_\rho<1/2}] \\ &\geq \mathbb{E}_{\mathcal{D}}[(W_\rho - 1)\mathbb{1}_{W_\rho\geq 1/2}] + \mathbb{E}_{\mathcal{D}}[\bar{W}_\rho \mathbb{1}_{\bar{W}_\rho\leq 1/2}] = 0.\end{aligned}$$

which proves

$$L_{\mathcal{D}}[h_{\mathrm{MV}}] \underset{\text{Lemma10}}{\leq} \mathbb{P}_{\mathcal{D}}(W_\rho \geq 1/2) \leq \mathbb{E}_{\mathcal{D}}[W_\rho] = \mathbb{E}_h[L_{\mathcal{D}}(h)]. \tag{D.1}$$

This implies EIR ≥ 0. □

Proof of Theorem 5.2

Lower bound of EIR

To prove the lower bound, we first define the tandem loss, as used in [MLIS20].

Definition D.1 (Tandem loss)**.** Define the tandem loss to be $L(h, h') = \mathbb{E}_{\mathcal{D}}[\mathbb{1}(h(X) \neq Y)\mathbb{1}(h'(X) \neq Y)]$.

We also rely on the following lemma, which appears as Lemma 2 in [MLIS20]. It provides the connection between the average error rate for each data point, W_ρ and the tandem loss, $L(h, h')$.

Lemma 12. *The equality* $\mathbb{E}_{\mathcal{D}}[{W_\rho}^2] = \mathbb{E}_{h,h'}[L(h, h')]$ *holds.*

We first state and prove the following lemma, which provides an upper bound on the tandem loss.

Lemma 13. *For the K-class problem,*

$$\mathbb{E}_{h,h'}[L(h,h')] \le \frac{2(K-1)}{K}\left(\mathbb{E}_h[L(h)] - \frac{1}{2}\mathbb{E}_{h,h'}[D(h,h')]\right).$$

Proof. We denote $\mathbb{P}_h(h(X) \neq Y)$ by $\bar{h}_Y(X)$. Note that $\mathbb{E}_{\mathcal{D}}(1-\bar{h}_Y(X)) = \mathbb{E}_h[L(h)]$ and

$$\begin{aligned}\mathbb{E}_{h,h'}[L(h,h')] &= \mathbb{E}_{\mathcal{D}}[\mathbb{P}_h(h(X)\neq Y)\mathbb{P}_{h'}(h'(X)\neq Y)]\\ &= \mathbb{E}_{\mathcal{D}}[(1-\bar{h}_Y(X))^2].\end{aligned}$$

Then we get

$$\begin{aligned}\mathbb{E}_{h,h'}[L(h,h')] &= \mathbb{E}_{\mathcal{D}}[(1-\bar{h}_Y(X))^2]\\ &= 1 - \mathbb{E}_{\mathcal{D}}[\bar{h}_Y(X)] - \mathbb{E}_{\mathcal{D}}[\bar{h}_Y(X)(1-\bar{h}_Y(X))]\\ &= \mathbb{E}_h[L(h)] - \mathbb{E}_{\mathcal{D}}[\bar{h}_Y(X)(1-\bar{h}_Y(X))].\end{aligned}$$

Now we will derive a lower bound of the second term. Since

$$\mathbb{E}_{h,h'}[\mathbb{1}(h(X)\neq h'(X))] = \sum_j \bar{h}_j(X)(1-\bar{h}_j(X)),$$

it follows that

$$\bar{h}_Y(X)(1-\bar{h}_Y(X)) = \mathbb{E}_{h,h'}[\mathbb{1}(h(X)\neq h'(X))] - \sum_{j\neq Y}\bar{h}_j(X)(1-\bar{h}_j(X)).$$

By maximizing $\sum_{j\neq Y}\bar{h}_j(X)(1-\bar{h}_j(X))$ subject to $\sum_{j\neq Y}\bar{h}_j(X) = 1-\bar{h}_Y(X)$, we get $\bar{h}_j(X) = \frac{1-\bar{h}_Y(X)}{K-1}$, which yields the upper bound

$$\sum_{j\neq Y}\bar{h}_j(X)(1-\bar{h}_j(X)) \le \frac{K-2}{K-1}(1-\bar{h}_Y(X)) + \frac{1}{K-1}\bar{h}_Y(X)(1-\bar{h}_Y(X)).$$

It follows that

$$\mathbb{E}_{\mathcal{D}}[\bar{h}_Y(X)(1-\bar{h}_Y(X))] \ge \frac{K-1}{K}\mathbb{E}_{h,h'}[D(h,h')] - \frac{K-2}{K}\mathbb{E}_h[L(h)],$$

and thus that

$$\begin{aligned}\mathbb{E}_{h,h'}[L(h,h')] &= \mathbb{E}_h[L(h)] - \mathbb{E}_{\mathcal{D}}[\bar{h}_Y(X)(1-\bar{h}_Y(X))]\\ &\le \mathbb{E}_h[L(h)] - \left(\frac{K-1}{K}\mathbb{E}_{h,h'}[D(h,h')] - \frac{K-2}{K}\mathbb{E}_h[L(h)]\right)\\ &= \frac{2(K-1)}{K}\left(\mathbb{E}_h[L(h)] - \frac{1}{2}\mathbb{E}_{h,h'}[D(h,h')]\right).\end{aligned}$$

□

We now provide the proof for the lower bound of EIR in Theorem 5.2.

Proof. We first claim that $\mathbb{P}_{\mathcal{D}}(W_\rho \geq 1/2) \leq 2\,\mathbb{E}_{\mathcal{D}}[{W_\rho}^2]$. Then, we have

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[(2{W_\rho}^2 - 1)\mathbb{1}_{W_\rho \geq 1/2}] &= \mathbb{E}_{\mathcal{D}}[(2(1-\bar{W}_\rho)^2 - 1)\mathbb{1}_{\bar{W}_\rho \leq 1/2}] \\ &= \mathbb{E}_{\mathcal{D}}[(1 - 4\bar{W}_\rho + 2{\bar{W}_\rho}^2)\mathbb{1}_{\bar{W}_\rho \leq 1/2}],\end{aligned}$$

where $\bar{W}_\rho = 1 - W_\rho$. Therefore,

$$\mathbb{E}_{\mathcal{D}}[(2{W_\rho}^2 - 1)\mathbb{1}_{W_\rho \geq 1/2}] + \mathbb{E}_{\mathcal{D}}[2{\bar{W}_\rho}^2\mathbb{1}_{\bar{W}_\rho \leq 1/2}] = \mathbb{E}_{\mathcal{D}}[(1 - 4\bar{W}_\rho + 4{\bar{W}_\rho}^2)\mathbb{1}_{\bar{W}_\rho \leq 1/2}] \geq 0.$$

Now we apply Lemma 11 with $h(x) = 2x^2$, to obtain

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[2{W_\rho}^2] - \mathbb{P}_{\mathcal{D}}(W_\rho \geq 1/2) &\geq \mathbb{E}_{\mathcal{D}}[(2{W_\rho}^2 - 1)\mathbb{1}_{W_\rho \geq 1/2}] + \mathbb{E}_{\mathcal{D}}[2{W_\rho}^2\mathbb{1}_{W_\rho < 1/2}] \\ &\geq \mathbb{E}_{\mathcal{D}}[(2{W_\rho}^2 - 1)\mathbb{1}_{W_\rho \geq 1/2}] + \mathbb{E}_{\mathcal{D}}[2{\bar{W}_\rho}^2\mathbb{1}_{\bar{W}_\rho \leq 1/2}] \;\geq 0,\end{aligned} \tag{D.2}$$

which proves the claim, $\mathbb{P}_{\mathcal{D}}(W_\rho \geq 1/2) \leq 2\,\mathbb{E}_{\mathcal{D}}[{W_\rho}^2]$.

Now we put the claim together with Lemmas 10, 12, and 13 to conclude the proof.

$$\begin{aligned}L(h_{\mathrm{MV}}) &\underset{\text{Lemma 10}}{\leq} \mathbb{P}_{\mathcal{D}}(W_\rho \geq 1/2) \leq 2\,\mathbb{E}_{\mathcal{D}}[{W_\rho}^2] \underset{\text{Lemma 12}}{=} 2\,\mathbb{E}_{h,h'}[L(h,h')] \\ &\underset{\text{Lemma 13}}{\leq} \frac{4(K-1)}{K}\left(\mathbb{E}_h[L(h)] - \frac{1}{2}\mathbb{E}_{h,h'}[D(h,h')]\right)\end{aligned} \tag{D.3}$$

Rearranging the terms, we obtain

$$\mathbb{E}_h[L_{\mathcal{D}}(h)] - L(h_{\mathrm{MV}}) \geq \frac{2(K-1)}{K}\mathbb{E}_{h,h'}[D(h,h')] - \frac{3K-4}{K}\mathbb{E}_h[L(h)]. \tag{D.4}$$

Dividing the both terms by $\mathbb{E}_h[L(h)]$ gives the lower bound $\frac{2(K-1)}{K}\mathrm{DER} - \frac{3K-4}{K}$. □

Upper bound of EIR

We denote $\mathbb{P}_h(h(X) = Y)$ by $\bar{h}_Y(X)$. We have

$$\mathbb{E}_h[L(h)] - L(h_{\mathrm{MV}}) = \mathbb{E}_{h,\mathcal{D}}[\mathbb{1}(h(X) \neq Y) - \mathbb{1}(h_{\mathrm{MV}}(X) \neq Y)].$$

Now

$$\begin{aligned}\mathbb{1}(h(X) \neq Y) - \mathbb{1}(h_{\mathrm{MV}}(X) \neq Y) &= \mathbb{1}(h_{\mathrm{MV}}(X) = Y) - \mathbb{1}(h(X) = Y) \\ &= \mathbb{1}(h(X) \neq h_{\mathrm{MV}}(X))\,(\mathbb{1}(h_{\mathrm{MV}}(X) = Y) - \mathbb{1}(h(X) = Y)) \\ &\leq \mathbb{1}(h(X) \neq h_{\mathrm{MV}}(X)).\end{aligned}$$

Now notice $\mathbb{E}_{h,\mathcal{D}}[\mathbb{1}(h(X) \neq h_{\mathrm{MV}}(X))] = 1 - \mathbb{E}_{\mathcal{D}}[\max_k \bar{h}_k(X)]$. Moreover, by Hölder's inequality,

$$\|\bar{\boldsymbol{h}}(X)\|_2^2 \leq \max_k \bar{h}_k(X),$$

and so

$$\begin{aligned} \mathbb{E}_h[L(h)] - L(h_{\mathrm{MV}}) &\leq 1 - \mathbb{E}_{\mathcal{D}}[\max_k \bar{h}_k(X)] \\ &\leq 1 - \mathbb{E}_{\mathcal{D}}[\|\bar{\boldsymbol{h}}(X)\|^2] = \mathbb{E}_{h,h'}[D(h,h')]. \end{aligned} \tag{D.5}$$

Dividing the both terms by $\mathbb{E}_h[L(h)]$ gives the upper bound DER.

Upper and lower bounds on the error rate of the majority vote classifier

We now present upper and lower bound on the majority vote classifier that follow from the bounds in Theorem 5.1 and 5.2, and compare them with existing bounds in the literature.

Theorem D.1. *For any competent ensemble ρ of K-class classifiers, the majority vote error rate satisfies*

$$\begin{aligned} L(h_{\mathrm{MV}}) &\leq \min\left\{ \frac{4(K-1)}{K}\left(\mathbb{E}_{h\sim\rho}[L(h)] - \frac{1}{2}\mathbb{E}_{h,h'\sim\rho}[D(h,h')]\right), \mathbb{E}_{h\sim\rho}[L(h)]\right\} \\ L(h_{\mathrm{MV}}) &\geq \mathbb{E}_{h\sim\rho}[L(h)] - \mathbb{E}_{h,h'\sim\rho}[D(h,h')]. \end{aligned}$$

Proof. The upper bound follows from inequality (D.1) and (D.3). The lower bound follows from inequality (D.5). □

We have already discussed that the bound $L(h_{\mathrm{MV}}) \leq \mathbb{E}[L(h)]$ represents an improvement by a factor of 2 over the naive first-order bound (5.1). Here, we further compare the bound

$$L(h_{\mathrm{MV}}) \leq \frac{4(K-1)}{K}\left(\mathbb{E}_{h\sim\rho}[L(h)] - \frac{1}{2}\mathbb{E}_{h,h'\sim\rho}[D(h,h')]\right) \tag{D.6}$$

to other known results in the literature. The closest in form is a bound specialized to binary case from [MLIS20], which gives

$$L(h_{\mathrm{MV}}) \leq 4\mathbb{E}_{h\sim\rho}[L(h)] - 2\mathbb{E}_{h,h'\sim\rho}[D(h,h')]. \tag{D.7}$$

Note that plugging in $K = 2$ to (D.6), we obtain the bound $2\mathbb{E}_{h\sim\rho}[L(h)] - \mathbb{E}_{h,h'\sim\rho}[D(h,h')]$, immediately improving on (D.7) by a factor of 2 (interestingly, the same factor that we save on the first-order bound). Hence, provided the competence assumption holds, our bound is a direct improvement on this bound, and furthermore generalizes directly to the K-class setting.

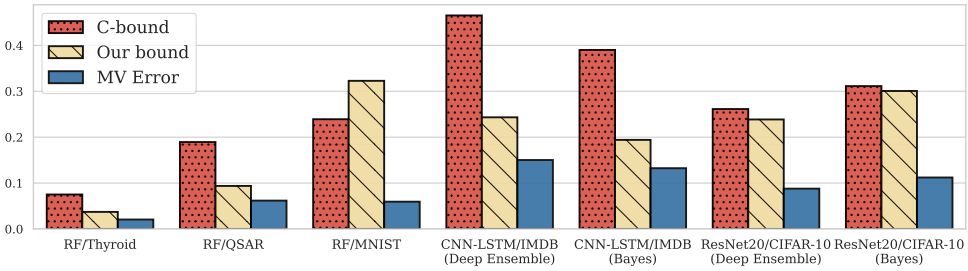


Figure D.1: **Our bound** (D.6) **versus the multi-class C-bound** (D.8)**.**

To our knowledge, the sharpest known upper bound on the majority-vote classifier is the general form of the C-bound given in [LMRR17], which states, provided $\mathbb{E}[M_\rho(X,Y)] > 0$,

$$L(h_{\mathrm{MV}}) \leq 1 - \frac{\mathbb{E}[M_\rho(X,Y)]^2}{\mathbb{E}[M_\rho^2(X,Y)]}, \tag{D.8}$$

where $M_\rho(X,Y) = \mathbb{E}_{h\sim\rho}[\mathbb{1}(h(X) = Y)] - \max_{j\neq Y} \mathbb{E}_{h\sim\rho}[\mathbb{1}(h(X) = j)]$ is called the *margin*. Unfortunately, the use of the margin function makes direct analytical comparison to our bound difficult. However, the bounds can be compared empirically, where the relevant quantities are estimated on hold-out data. In Figure D.1, we compare the value of our bound against the value of the multi-class C-bound, on tasks for which we have verified the competence assumption holds. We find that in all but one case (random forests with MNIST), our bound is superior empirically, sometimes significantly. Interestingly, we observe that our bound does particularly well on tasks with only a few classes. This behavior might be attributed to the constant $\frac{4(K-1)}{K}$ in the upper bound (D.6) which increases as the number of classes K grows.

D.2 Additional empirical results

Experimental details

Bagged random feature classifiers. We consider ensembles of random ReLU feature classifiers, constructed as follows. For each classifier, we draw a random matrix $\boldsymbol{U} \in \mathbb{R}^{N\times d}$, whose rows $\boldsymbol{u}_j$ are drawn from the uniform distribution on the sphere $\mathbb{S}^{d-1}$. For a given input $\boldsymbol{x} \in \mathbb{R}^d$, we compute the feature $\boldsymbol{z}(\boldsymbol{x}) = \sigma(\boldsymbol{U}\boldsymbol{x})$ where $\sigma(t) = \max(t, 0)$ is the ReLU function. We then fit a multi-class logistic regression model in `scikit-learn` [PVG+11] using these (random) features. To form an ensemble of these classifiers, we additionally perform bagging, by sampling a different set of size n with replacement from the training set of size n, independently for each individual classifier. Thus, each classifier is subject to two different types of randomness: the randomness from the sampling of the feature matrix $\boldsymbol{U}$; and the randomness from the bootstrapping of the training data. For the models shown in the competence plot in Figure 5.1, we use 500 random features and $M = 100$ classifiers.

Random forests. We consider random forest (RF) models as implemented in `scikit-learn` [PVG+11], each made up of 20 individual decision trees. We vary the maximum number of leaf nodes in each tree to construct models with varying performance. For the single-ensemble results presented, we use the default parameters implemented in `scikit-learn`. For the random forests, we use a small version of the MNIST dataset with 5000 randomly selected training examples (500 from each of the 10 classes). We also use two binary classification datasets retrieved from the UCI repository [DG17]: the QSAR oral toxicity dataset (7.2k train, 1.8k test examples, 1024 features) [BGCT19]; and the Thyroid disease dataset (2.5k train, 633 test examples, 21 features) [QCHL87]. For the models shown in the competence plot in Figure 5.1, we use the default settings of the random forest implementation in `scikit-learn`.

Deep ensembles. We consider four different architectures for our deep ensembles. We use a standard ResNet18 models [HZRS16] trained on the CIFAR-10 dataset [KH+09], using 100 epochs of SGD with momentum 0.9, weight decay of 5×10^{-4} and a learning rate of 0.1, while varying the batch size and width hyper-parameters. We report results from two variants of this empirical evaluation: one in which we employ learning rate decay (by dropping the learning rate to 0.01 after 75 epochs); and another in which we disable learning rate decay. For each setting, we train 5 models from independent initialization to form the respective ensembles. We also evaluate these models on two out-of-distribution databases: CIFAR-10.1 and CIFAR-10-C [RRSS19, HD19] (the latter is itself comprised of 19 different datasets employing various types of data corruption). Finally, we evaluate deep ensembles of 25 standard BERT models [DCLT19], provided with the paper [SYT+22], fine-tuned on the GLUE classification tasks [WSM+19].

Bayesian ensembles. For the Bayesian ensembles used in this paper, we consider samples provided in [IVHW21], obtained via large-scale sampling from a Bayesian posterior using Hamiltonian Monte Carlo. To our knowledge, these samples are the most precisely representative of a theoretical Bayesian neural network posterior publicly available. In particular, we use samples on the CIFAR-10 datasets with a ResNet20 architecture, and the IMDB dataset on the CNN-LSTM architecture. We defer to the original paper [IVHW21] for additional details.

More competence plots

In this section, we provide additional empirical results. To further verify that the competence assumption holds broadly in practice, here we include several more examples of competence plots for experiments presented in the main text.

ResNet18 on CIFAR-10 OOD variants. In Figures D.2 and D.3, we plot competence plots for the ResNet18 ensembles on the CIFAR-10, CIFAR-10.1 and a subset of the CIFAR-10-C datasets [RRSS19, HD19]. We find that the competence assumption holds across all examples.

Fine-tuned BERT models. In Figure D.4, we provide competence plots for the BERT/GLUE fine-tuning tasks. For the RTE, CoLA, MNLI, QQP and QNLI tasks, we find that the competence assumption holds. However, we find two examples here where it does not: the MRPC and SST-2 tasks, although the extent to which the assumption is violated in minor. Since these are particularly small datasets, this may also be a product of noise from low sample size.

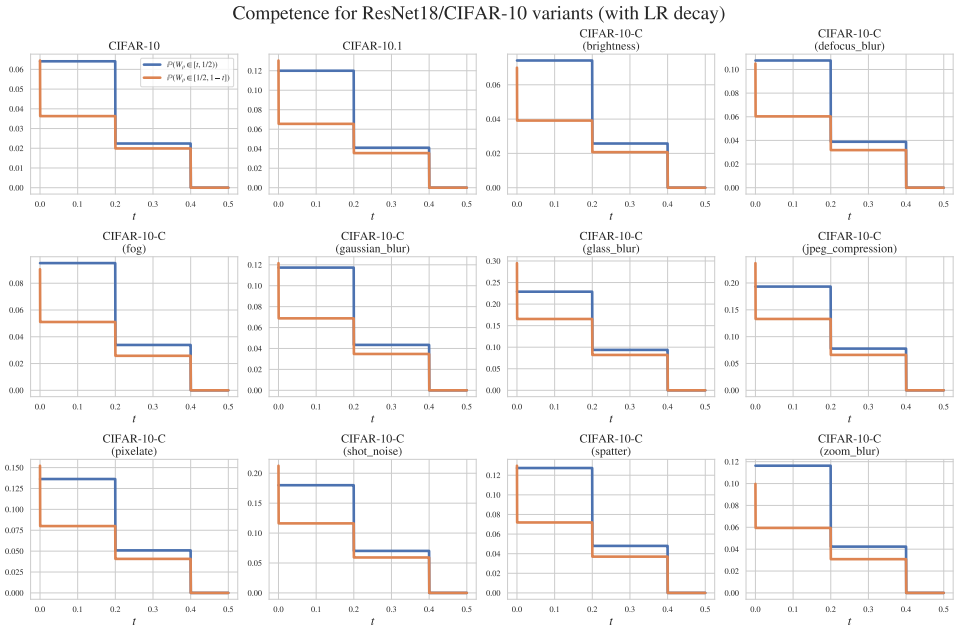


Figure D.2: **Competence for ResNet18/CIFAR-10 variants (models with learning rate decay).** We observe that the competence assumption holds across all tasks.

D.3 Pathological ensembles satisfying $L(h_{\text{MV}}) = 2\mathbb{E}[L(h)]$

In this section, we provide two pathological examples of ensembles that makes the "first-order" upper bound tight. In particular, the second example shows that positive margin condition, i.e., $\mathbb{E}[M_\rho(X,Y)] > 0$ where $M_\rho(X,Y) = \mathbb{E}_{h\sim\rho}[\mathbb{1}(h(X) = Y)] - \max_{j\neq Y} \mathbb{E}_{h\sim\rho}[\mathbb{1}(h(X) = j)]$, from existing literature is not enough to rule out pathological cases. Recall that the first-order bound introduced in section 5.2 is the following:

$$0 \leq L(h_{\text{MV}}) \leq \mathbb{P}(W_\rho \geq 1/2) \leq 2\,\mathbb{E}[W_\rho] = 2\,\mathbb{E}_{h\sim\rho}[L(h)].$$

Example D.1 (The first-order upper bound is tight)**.** Consider a classification problem with two classes. For given $\epsilon > 0$, suppose slightly less than half, $0.5 - \epsilon$, fraction of classifiers are the perfect classifier, correctly classifying test data with probability 1, and the other $0.5 + \epsilon$ fraction of classifiers are completely wrong, incorrectly predicting on test data with probability 1. With this composition of classifiers, the average error rate is $0.5 + \epsilon$ and the

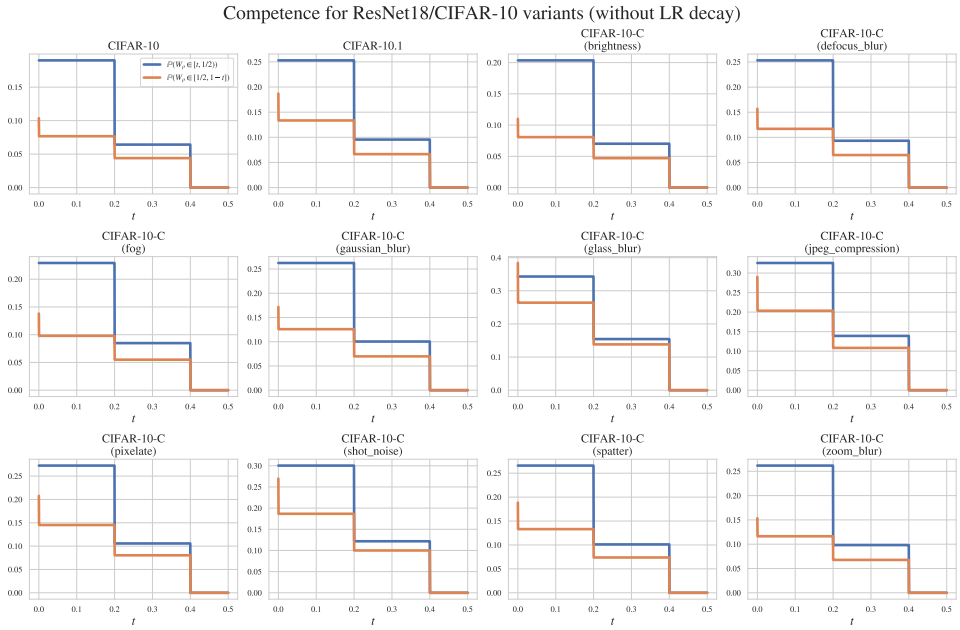


Figure D.3: **Competence for ResNet18/CIFAR-10 variants (models without learning rate decay).** We observe that the competence assumption holds across all tasks.

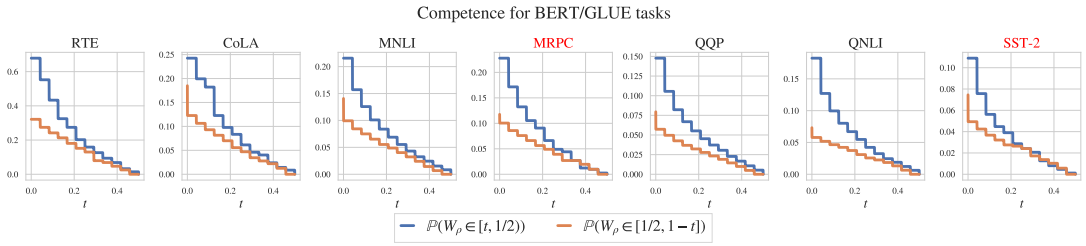
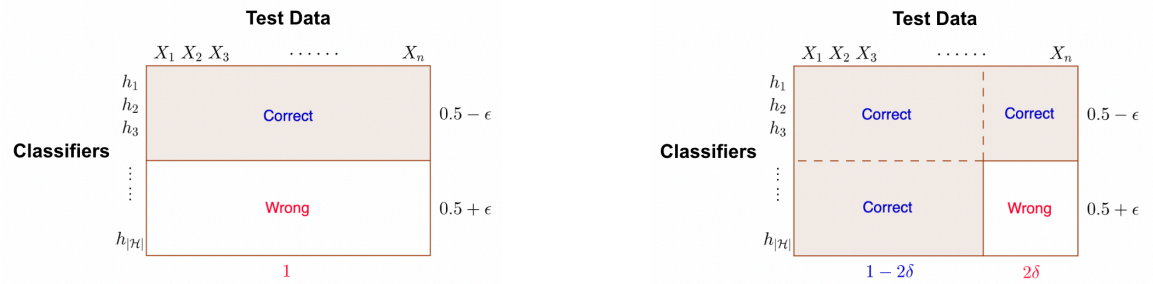


Figure D.4: **Competence for BERT/GLUE fine-tuning tasks.** The competence assumption holds for the RTE, CoLA, MNLI, QQP and QNLI tasks, though interestingly, we find that the competence assumption is (to a small degree) violated for two of the tasks: MRPC and SST-2.

majority vote error rate is 1. Taking $\epsilon \to 0$ concludes that the first-order upper bound (5.1) is tight. A visual illustration of the composition of classifiers is given in Figure D.5a.

The condition $\mathbb{E}[M_\rho(X,Y)] > 0$ rules out the ensemble described in Example D.1. Nonetheless, the first-order bound $2\mathbb{E}[L(h)]$ is tight *even when* $\mathbb{E}[M_\rho(X,Y)] > 0$ is satisfied, as we show with the following example.

Example D.2 (The first-order upper bound is tight even when the margin is large)**.** We again consider a classification problem with two classes. For given $\epsilon > 0$, as in Example D.1, slightly less than half, $0.5 - \epsilon$, fraction of classifiers are the perfect classifier. All of the other $0.5 + \epsilon$ fraction of classifiers, on the contrary, now correctly predict on the same $1 - 2\delta$ fraction



(a) Composition of classifiers in Example D.1

(b) Composition of classifiers in Example D.2

Figure D.5: **Illustration of the composition of classifiers given in Examples D.1 and D.2.** On each plot, the area of the white box equals to the average test error rate. On Figure D.5a, the majority vote error rate is 1, while the average test error rate is $0.5+\epsilon$. On Figure D.5b, the majority vote error rate is 2δ (Rightmost 2δ test data) while the average test error rate is $\delta(1+2\epsilon)$. The margin of each composition of classifiers is $2\epsilon \to 0$ and $1-2\delta(1+2\epsilon) > 0$, respectively.

of the test data and incorrectly predict on the other 2δ fraction of the test data. With this composition of classifiers, the majority vote error rate is 2δ even when the average error rate is $\delta(1+2\epsilon)$. In addition, unlike the composition of classifiers in Example D.1, the margin of which is 2ϵ, the margin of the new composition of classifiers is $1-2\delta(1+2\epsilon)$, which can be any value smaller than 1. Taking $\epsilon \to 0$ concludes that the first-order upper bound (5.1) is also tight when the margin is arbitrarily high. A visual illustration of the composition of classifiers is given in Figure D.5b.