Towards a better understanding of generalization measures in deep learning

Shuofeng Zhang

Rudolf Peierls Centre for Theoretical Physics

University of Oxford

This dissertation is submitted for the degree of

Doctor of Philosophy in Theoretical Physics

October 2025

To my beloved grandfather, Peiru Zhang

Acknowledgements

I am deeply grateful to **Professor Ard Louis**, my supervisor, for his patient guidance, steady encouragement, and incisive feedback at every stage of this thesis. His example of clarity and curiosity has shaped both the questions I ask and the way I try to answer them.

I owe a great deal to my colleagues **Guillermo Valle Pérez**, **Yoonsoo Nam**, **Chris Mingard**, **Nayara Fonseca**, **Ouns El Harzli**, **Mehrana Nejad**, **Wilber Lim**, **Hanzhi Jiang**, and **Vaibhav Mohanty** for generous conversations, careful readings, and the many small acts of help that never make it into footnotes but make the work possible.

My thanks go to the **staff of Linacre College** and to the **staff of the Department of Physics**, especially at the **Rudolf Peierls Centre for Theoretical Physics**, for creating a supportive environment in which research can thrive. I am also grateful to **Olivia Singh** at the University's Funding office and **Joy Zhang** at the University's Immigration Office for their practical assistance and kindness.

To my friends **Lawrence Wang**, **Benjamin Shi**, **Abir Patwary**, **Yun Dong**, **Siting Miao**, and **Lakmal Fonseka**: thank you for your companionship, good humour, and perspective when it was most needed.

Above all, I thank my family for their unwavering love. I am especially indebted to my grandfather, who is courageously undergoing treatment for cancer; his strength has been a constant source of inspiration. The world may be changing at an unprecedented speed, but one thing does not change: my love for them.

Any remaining errors are my own.

Abstract

Overparameterized neural networks routinely generalize despite being able to fit random labels, revealing strong algorithmic biases whose origins are not fully understood. This thesis advances a function-space perspective on those biases and couples it with invariance-aware diagnostics. First, it shows that common flatness measures can be driven to arbitrarily different values by benign parameter rescalings or optimizer choices, while the function-space prior $\log P(f)$ serves as a rescaling-invariant predictor of generalization across architectures and training algorithms. Second, it introduces *fragility audits*—targeted stress tests based on learning-curve stability, post-interpolation dynamics, and dataset difficulty—that expose qualitative mismatches in many popular generalization measures and highlight a Gaussian-process marginal-likelihood predictor that remains stable. Third, it derives closed-form scaling laws for minimum-ℓ_p interpolators and for diagonal linear networks under ℓ_p bias, identifying universal elbows and thresholds that govern the behavior of the entire family of ℓ_r norms and explaining why superficially similar proxies can diverge across regimes. Taken together, these results yield practical, theory-grounded guidance for assessing generalization: evaluate predictors in function space, insist on symmetry invariance and robustness to routine training changes, and rely on proxies whose scaling behavior is explicit and justified.

Contents

1 Introduction

Machine learning systems are judged not by how well they memorize their training sets, but by how reliably they make predictions on new data. This capacity to generalize has long been the organizing principle of statistical learning theory, and it continues to frame debates in modern deep learning. What has changed is the regime in which these debates unfold. Contemporary neural networks are routinely trained with far more parameters than training examples, often to the point of interpolation, and yet they still achieve strong out-of-sample performance on natural data. That an overparameterized model can fit arbitrary labels but still generalize when the labels carry structure is both empirically undeniable and theoretically provocative. It forces us to revisit the question of *why* certain training pipelines lead to predictors that work, and what kinds of evidence should persuade us that a proxy for generalization is capturing the right phenomenon.

Classical accounts emphasize capacity control: bounding the richness of a hypothesis class through measures such as VC dimension, Rademacher complexity, margins, compression, and algorithmic stability. These ideas remain foundational, not least because they articulate desiderata any modern explanation must respect. Yet capacity alone, treated as a static property of a parameterized function class, struggles to explain the deep-learning reality that models with effectively unbounded capacity can generalize when trained with particular algorithms, schedules, and

inductive biases. The tension shows up in several well-known observations: deep networks can interpolate noisy data without immediately collapsing test accuracy; widening or deepening a model sometimes helps *more* after interpolation; and training dynamics steer solutions toward functions with particular regularities even when no explicit regularizer is present. These observations suggest that generalization in practice is an *algorithmic* phenomenon as much as a *capacity* one, expressing the interaction between parameterization, data geometry, and the path taken by optimization.

To reason productively in this setting, we need language that connects three levels. At the level of functions, a predictor's behavior on inputs—its invariances, margins, and sensitivity to perturbations—matters more than any single coordinate system in parameter space. At the level of algorithms, stochastic optimization, data augmentation, and early stopping impose an *implicit bias* that selects among the many interpolating solutions. At the level of models, architectural choices such as depth, skip connections, and normalization layers introduce symmetries that reshape how parameter changes translate into function changes. Any measure that purports to predict generalization must therefore navigate these levels simultaneously: it must be sensitive to the structure of the learned *function*, robust to harmless reparameterizations, and reflective of how learning procedures actually traverse the landscape.

This requirement immediately clarifies why some intuitive diagnostics are more brittle than they first appear. Many popular post-hoc measures are computed from a single trained parameter vector: a sharpness or flatness score near a local minimum, a norm of the weights, or a margin computed on an internal representation. These quantities can be informative in narrow settings, but they are also vulnerable to changes that alter parameters without materially changing the predictor. Rescaling layers in a network with normalization, swapping one optimizer for another while keeping test error steady, or following learning-rate schedules that traverse equivalent function trajectories can all scramble parameter magnitudes and local curvature without affecting what the model computes. If a

diagnostic responds primarily to such superficial changes, it does not tell us much about generalization itself. The right baseline, then, is *invariance*: measures that track properties of the predictor should be indifferent to symmetries and pathologies that leave the predictor unchanged, and they should move in predictable ways when the data truly become easier or harder.

A complementary perspective comes from the function-space viewpoint. Instead of trying to read generalization off the surface geometry of the loss in parameter space, one can ask what *prior* probability a training pipeline implicitly assigns to different functions, and how that prior interacts with the data distribution. Bayesian and PAC–Bayesian analyses make this connection explicit by relating expected generalization to quantities that live in function space, such as marginal likelihood and priors over outputs. Even when exact Bayesian training is out of reach, the function-space lens provides two practical virtues. First, it encodes the invariances that parameter-space surrogates often ignore: if two parameterizations compute the same function, their function-space description coincides. Second, it naturally absorbs architectural symmetries and the effect of early layers that act as learned feature maps, aligning the measure with what matters for prediction. In modern practice, approximations to this lens—via Gaussian process limits, ensembles, or carefully constructed surrogates—offer a way to calibrate claims about difficulty across datasets and to check whether a proposed diagnostic is reacting to the data rather than to incidental details of the training path.

Equally important is a tractable theoretical setting in which we can isolate causes from consequences. Overparameterized linear models and simplified neural families such as diagonal linear networks offer such a laboratory. They capture essential ingredients of modern pipelines—interpolation, implicit regularization, and sensitivity to initialization—without the full complexity of deep architectures. In these models one can derive closed-form relationships between sample size, data anisotropy, initialization scale, and families of norms or margins that are often used as generalization proxies. These formulas do not replace full-scale experiments, but they do sharpen our intuitions: they tell us when a norm should increase

or decrease with more data, when a bulk of noisy directions should dominate a diagnostic, and when alignment with signal should take over as the effective bias changes. The resulting picture is more nuanced than "smaller is better": the same proxy can be informative in one regime and misleading in another, depending on which features of the data and algorithm are salient.

Taken together, these strands motivate a pragmatic stance on evaluation. Rather than treating any single surrogate as a universal currency for generalization, we should (i) demand invariance to symmetries that preserve the predictor, (ii) check stability under benign changes in the training pipeline, and (iii) ensure sensitivity to genuine changes in task difficulty. When such audits are built into empirical practice, they help separate measures that encode properties of the learned function from those that primarily reflect the accidents of parameterization or optimization. When theory is layered on top of these audits, it can explain *why* a measure succeeds or fails and delineate the regimes in which it should be trusted.

The chapters that follow develop this outlook from complementary angles. One thread examines geometric intuitions—such as flatness and sharpness—and asks when they align with out-of-sample behavior and when they are undone by reparameterizations and optimizer choices. Another thread advances an evaluation protocol that treats fragility as a first-class failure mode and builds simple, reproducible stress tests into the way we compare generalization surrogates across datasets and training setups. A third thread turns to simplified models where we can write down scaling laws for whole families of norms and see explicitly how data geometry and inductive bias govern their behavior. Together these threads argue for measures that live where prediction lives—in function space or in invariants of the predictor—and for empirical habits that test surrogates against the kinds of shifts that routinely occur in modern pipelines.

Roadmap. Chapter 2 surveys classical generalization frameworks and modern complexity measures that ground the rest of the thesis. Chapter 3 interrogates parameter-space notions such as flatness and contrasts them with a function-space alternative grounded in priors over network outputs. Chapter 4 proposes and applies

fragility audits that probe whether generalization measures react to optimizer swaps, scale symmetries, and dataset difficulty in ways that track test error. Chapter 5 develops closed-form scaling laws in tractable overparameterized settings to explain how norm-based proxies behave across regimes. Chapter 6 draws the lessons together and sketches directions for extending invariance-aware diagnostics and theory to richer architectures and training regimes.

2
Background

Predicting/bounding the generalization performance of supervised machine learning algorithms has a long history. The celebrated probably approximately correct (PAC) framework [Valiant, 1984], alongside with uniform convergence analysis [Vapnik, 1968, Vapnik and Chervonenkis, 1974, Vapnik, 1995] provided the first theory to bound the generalization error of a supervised model which sees the data from a training set and infers on the unseen. The high level idea is (in the agnostic regime) that if we assume an unknown probability distribution of the data and the losses can be seen as bounded random variables, then we can apply concentration inequalities (e.g. Hoeffding's) to bound the difference of empirical error and generalization error, which is the generalization gap.

Model complexity plays a central role in most of the generalization bounds. Traditional wisdom has considered uniform convergence properties such as VC dimension and Rademacher complexity, which are shown to be not satisfactory in the regime of overparameterized neural networks [Zhang et al., 2016a]. Numerous recent works have been focused on proposing better complexity measures using different approaches. In the following sections we will introduce the problem setup and some noticeable generalization bounds from different categories.

2.1 Notation for the supervised learning problem

In the context of a supervised learning problem, the standard formalization usually starts with defining the input and output domains as $\mathcal{X}$ and $\mathcal{Y}$, respectively. Assume that the underlying data distribution is $\mathcal{D}$. Our training set, denoted as S, consists of n input-output pairs that are IID sampled from $\mathcal{D}$, represented as $S = \{(x_i, y_i)\}_{i=1}^{n} \sim \mathcal{D}^n$. To gauge the quality of predictions, we employ a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, quantifying how well a prediction $\hat{y} \in \mathcal{Y}$ aligns with the actual output $y \in \mathcal{Y}$. A hypothesis h is characterized as a function mapping inputs to outputs, expressed as $h : \mathcal{X} \to \mathcal{Y}$. The *(true) risk* R associated with hypothesis h is the expected value of the loss incurred by predicted outputs on new samples, defined as $R(h) = \mathbf{E}_{(x,y)\sim\mathcal{D}}[L(h(x), y)]$. In contrast, the *empirical risk* $\widehat{R}(h, S)$ is computed as the empirical average of the loss function over the training set, given by $\widehat{R}(h, S) = \frac{1}{n}\sum_{(x,y)\in S} L(h(x), y)$. This distinction allows us to assess how well a hypothesis generalizes from the training data to new, unseen samples.

In the case of classification, $\mathcal{Y}$ is a discrete set and the typical loss to consider is the $0-1$ loss function: $L(\hat{y}, y) = \mathbb{1}[\hat{y} \neq y]$, where $\mathbb{1}$ is the indicator function. With this particular loss function, the *generalization error* ϵ and the *empirical/training error* $\hat{\epsilon}$ are defined as:

$$\epsilon(h) = \mathbf{E}_{(x,y)\sim\mathcal{D}}\mathbb{1}[h(x) \neq y] = \mathbf{P}_{(x,y)\sim\mathcal{D}}[h(x) \neq y] \tag{2.1}$$

$$\hat{\epsilon}(h, S) = \frac{1}{n}\sum_{(x,y)\in S} \mathbb{1}[h(x) \neq y]. \tag{2.2}$$

A more careful treat also includes the *learning algorithm* which is defined to be a mapping $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{Y}^{\mathcal{X}}$ from training set of any size to hypotheses. $\mathcal{A}$ can be deterministic or stochastic; a stochastic learning algorithm will map training sets to a probability distribution over hypotheses, which is the case in PAC-Bayesian bounds that we introduce in section 2.5.

2.2 The PAC learning framework

The Probably Approximately Correct (PAC) framework was originally introduced by Valiant in his seminal work [Valiant, 1984]. Alongside the analysis of uniform convergence (refer to Section 2.3), these elements constitute the dual pillars of contemporary statistical learning theory. The PAC learning framework generally concerns itself with establishing confidence bounds that articulate a connection between observed quantities (derived from the training set) and the unobserved generalization error $\epsilon(h)$—the true quantity of interest [1]. A general frequentist formulation of the PAC generalization bounds has the following form [Valle-Pérez and Louis, 2020]:

Definition 1 (Agnostic PAC generalization bound, general)**.** *For any data distribution $\mathcal{D}$, algorithm $\mathcal{A}$ and confidence level δ, the empirical error $\hat{\epsilon}$ and generalization error ϵ satisfy*

$$\mathbf{P}_{S\sim\mathcal{D}^n}\left[\epsilon(\mathcal{A}(S)) - \hat{\epsilon}(\mathcal{A}(S)) \leq \frac{\mathfrak{C}_{\mathcal{A}}(S)}{n^{\alpha}}\right] \geq 1-\delta \tag{2.3}$$

where $\mathfrak{C}_{\mathcal{A}}$ is commonly addressed as the model complexity in the literature, and $0 < \alpha \leq 1$ is the scaling component.

At the core of the bound, the term $\mathfrak{C}_{\mathcal{A}}(S)$ typically measures some notion of "complexity" of the learned hypothesis which results from the intricate interplay of data and training algorithms. The component α marks the convergence rate of learning. Some further assumptions on the data, e.g. the algorithm returns hypotheses with 0 empirical error (the realizability assumption) can help to prove a bound with a faster rate. All endeavors aiming to propose generalization bounds for deep neural networks (DNNs) within the PAC framework fundamentally seek to enhance the characterization of $\mathfrak{C}_{\mathcal{A}}(S)$. In certain instances, the reliance on the training set is entirely encapsulated within $\mathfrak{C}_{\mathcal{A}}(S)$, resulting in the omission of $\frac{1}{n^{\alpha}}$.

[1] the original PAC framework also included conditions of the computational efficiency of the learning algorithm

2.3 Uniform convergence generalization bounds

The PAC learning framework in equation 2.3 can be equivalently interpreted from the perspective of *sample complexity*, which is the number of samples needed for the learned hypothesis to have a small enough generalization gap. From this angle, PAC learnability requires that for any data distribution $\mathcal{D}$, there is a learning algorithm $\mathcal{A}$ that with sufficient training data, returns a hypothesis that is arbitrarily close to the generalization error $\hat{\epsilon}$ [Hellström et al., 2023]. When the PAC learnability is required for a hypotheses class $\mathcal{H}$, it turns out that it is equivalent to uniform convergence, defined below:

Definition 2 (Uniform convergence)**.** *The hypothesis class $\mathcal{H}$ has the uniform convergence property if for any data distribution $\mathcal{D}$, confidence level δ and scaling component α, we have*

$$\mathbf{P}_{S\sim\mathcal{D}^n}\left[\epsilon(h) - \hat{\epsilon}(h) \leq \frac{\mathfrak{C}(\mathcal{H})}{n^\alpha}\right] \geq 1 - \delta \tag{2.4}$$

hold for all $h \in \mathcal{H}$.

The nomenclature becomes clear when considering that the generalization bounds apply uniformly over both data distributions and hypotheses. In such instances, the model complexity term is determined by the characteristics of the hypothesis class $\mathcal{H}$ rather than individual hypotheses. The simplest form of uniform convergence generalization bounds can be proven using a combination of concentration inequalities plus a union bound argument:

Theorem 1 (Agnostic uniform convergence bound)**.** *Let $\mathcal{H}$ be a finite hypothesis class. Then, for any data distribution $\mathcal{D}$ and confidence level $\delta > 0$, with probability at least $1 - \delta$ over the choices of S, the following inequality holds:*

$$\forall h \in \mathcal{H}, \quad \epsilon(h) - \hat{\epsilon}(h) \leq \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta}}{2n}} \tag{2.5}$$

The square root appearing in the bound is a result of using the Hoeffding inequality which bounds the deviation of the empirical average of IID sub-Gaussian random variables from its expectation. It turns out that when operating in the realizable regime, this scaling can be improved:

Theorem 2 (Realizable uniform convergence bound)**.** *Let $\mathcal{H}$ be a finite hypothesis class. Assume for any target concept $c \in \mathcal{H}$ and i.i.d. sample $S \sim \mathcal{D}^n$ the ERM training algorithm returns a consistent hypothesis $h : \hat{\epsilon}(h) = 0$. Then, for any data distribution $\mathcal{D}$ and confidence level $\delta > 0$, with probability at least $1 - \delta$ over the choices of S, the following inequality holds:*

$$\forall h \in \mathcal{H}, \quad \epsilon(h) \leq \frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{n} \tag{2.6}$$

The appeal of uniform convergence bounds is clear: irrespective of the data or learning algorithm employed, the assurance is that the training loss serves as a reliable indicator of the generalization loss. Unfortunately, it appears that such requisites might be too stringent for many contemporary machine learning scenarios, particularly in the context of deep neural networks. For this model class, certain data distributions or hypotheses result in suboptimal generalization, even though naturally occurring data and widely used learning algorithms demonstrate satisfactory performance [Zhang et al., 2016a]. This observation underscores the motivation behind generalization bounds built through a non-uniform convergence analysis, wherein the model complexity terms are tailored to the specific data distribution and learning algorithm under consideration. Nevertheless, the concept of uniform generalization has demonstrated significant efficacy across various domains, offering a definitive characterization of learnability. The model complexity term in bound 2.5 and 2.6 - the logarithm of the cardinality of the hypothesis class - is however too crude and useless when $\mathcal{H}$ is an infinite set. In the primary scenario of binary classification, a classical notion of model complexity of the potentially infinite hypothesis class inspired by uniform convergence is the VC dimension. A stride in the direction of incorporating data dependence into the model complexity was introduced by Bartlett and Mendelson [2002] through the notion of the Rademacher complexity of a hypothesis class. We give a brief overview of them in the following.

2.3.1 VC Dimension

Focusing on binary classification, if the VC dimension of a hypothesis class $\mathcal{H}$, denoted as d_{VC}, is finite, then it is guaranteed that $\mathcal{H}$ satisfies the uniform convergence property. We proceed to define the VC dimension through the notion of growth function [Hellström et al., 2023].

Definition 3 (Growth function and VC dimension)**.** *The growth function $g_{\mathcal{H}}(n)$ is defined as the maximum number of different ways in which a dataset of size n can be classified using functions from $\mathcal{H}$, that is,*

$$g_{\mathcal{H}}(n) \overset{def}{=} \max_{S \sim \mathcal{D}^n} |\{(h(x_1), ..., h(x_n)) : h \in \mathcal{H}\}|$$

The VC dimension of $\mathcal{H}$, denoted as d_{VC} is the largest integer such that the above equality holds. That is,

$$d_{VC} \overset{def}{=} \max \{n \in \mathbb{N} : g_{\mathcal{H}}(n) = 2^n\}$$

If no such integer exists, we say that $d_{VC} = \infty$.

The intuition of why VC dimension characterizes uniform convergence: If $d_{\text{VC}} = \infty$, we can change the labels of S arbitrarily and still find a hypothesis that is consistent with S, no matter how large is n. This implies that a hypothesis with minimal or maximal training loss can be determined, irrespective of the underlying generalization loss. However, if the VC dimension is finite (and ideally $n \gg d_{\text{VC}}$), we cannot adapt arbitrarily to every sample in the training set, but only to d_{VC} of them. In some sense the remaining $n - d_{\text{VC}}$ samples provide a reasonable estimate of the generalization error. The generalization bound provided by NV dimension has the following form:

Theorem 3 (Generalization bound from VC dimension)**.** *Consider a hypothesis class $\mathcal{H}$ with VC dimension d_{VC}. With confidence δ, for all $h \in \mathcal{H}$ we have*

$$|\epsilon(h) - \hat{\epsilon}(h)| \leq \sqrt{C \frac{d_{VC} + \log \frac{1}{\delta}}{n}} \tag{2.7}$$

for some constant C. In the realizable case, this bound can be optimized with a faster convergence rate:

$$\sup_{h \in \mathcal{H}_0(S)} \epsilon(h) \leq C \frac{d_{VC} + \log \frac{1}{\delta}}{n} \tag{2.8}$$

where $\mathcal{H}_0(S)$ is the set of all $h \in \mathcal{H}$ with zero training error on S.

As one of the most iconic results from classical uniform convergence analysis, VC dimension provides a guarantee for PAC learnability in general machine learning tasks. Unfortunately, in the regime of deep learning, such a simple notion of model complexity which does not incorporate the strong inductive bias of DNNs is bound to fail. Valle-Pérez and Louis [2020] provided a nice set of desiderata that predictive theories of generalization in the deep learning regime should satisfy. We can compare the VC dimension generalization bound against some of their criteria to show why it is not a good generalization theory for deep learning:

1. The VC bound is data-independent, which means its value stays the same regardless of the actual data set. The original MNIST and its pixel-shuffled version will have the same value in VC bound, but their generalization performances are radically different.

2. d_{VC} typically grows with the number of parameters of DNNs, while in reality the generalization diminishes with overparameterization.

3. The VC dimension of contemporary DNNs is often significantly larger than the number of training examples, as noted by Zhang et al. [2016a]. Consequently, this condition results in vacuous VC bound.

2.3.2 Rademacher Complexity

Now we look at the Rademacher complexity, which still relies on uniform convergence but incorporates data dependency. It is worth noting that while the (empirical) Rademacher complexity of a hypothesis class $\mathcal{H}$ is defined with respect to a specific dataset, the commonly used Rademacher complexity takes expectation over the dataset which cancels the training-set dependence. Like VC dimension,

the Rademacher complexity also captures the "richness" of a hypothesis class by measuring the degree to which it can fit random noise. We give the formal definition below:

Definition 4 (Rademacher complexity)**.** *Let $\mathcal{H}$ be a family of functions mapping from $\mathcal{X}$ to $\mathcal{Y}$ and $S = (x_1, ..., x_n)$ a fixed sample of size n in $\mathcal{X}$. Let $\mathcal{G}$ be the family of bounded loss functions associated to $\mathcal{H}$, i.e. $\mathcal{G} = \{g : (x, y) \mapsto L(h(x), y) : h \in \mathcal{H}\}$ and $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is an arbitrary loss function. The empirical Rademacher complexity of $\mathcal{H}$ with respect to the sample S is defined as:*

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i g\left(x_i\right)\right], \tag{2.9}$$

where $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_n)^\top$, with σ_i independent uniform random variables taking values in $\{-1, +1\}$. The random variables σ_i are called Rademacher variables. Furthermore, let $\mathcal{D}$ denote the underlying data distribution. For any integer $n > 1$, the Rademacher complexity of $\mathcal{G}$ is the expectation of the empirical Rademacher complexity over all samples of size n drawn according to $\mathcal{D}$:

$$\mathfrak{R}_n(\mathcal{G}) = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^n}\left[\widehat{\mathfrak{R}}_S(\mathcal{G})\right] \tag{2.10}$$

An intuitive way to understand why the Rademacher complexity can capture generalization [Hellström et al., 2023]: imagine splitting S randomly into a training set and a test set by putting x_i into the training set if $\sigma_i = -1$ and otherwise the test set, then the empirical Rademacher complexity is exactly the worst-case loss discrepancy between the test set and the training set, with expectation taken over the randomness splitting the dataset. It is almost a generalization measure by definition. More rigorously, the following generalization bound with Rademacher complexity has been proven [Mohri et al., 2018]:

Theorem 4 (Rademacher complexity generalization bounds)**.** *Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{X} \times \mathcal{Y}$ to $[0, 1]$. Then for any $\delta > 0$, with probability at*

least $1-\delta$ *over the draw of an i.i.d. sample* S *of size* n*, each of the following holds for all* $g \in \mathcal{G}$*:*

$$\begin{aligned} &\mathbb{E}[g(x)] \leq \frac{1}{n}\sum_{i=1}^{n} g\left(x_i\right) + 2\mathfrak{R}_n(\mathcal{G}) + \sqrt{\frac{\log\frac{1}{\delta}}{2n}} \\ \textit{and} \quad &\mathbb{E}[g(x)] \leq \frac{1}{n}\sum_{i=1}^{n} g\left(x_i\right) + 2\widehat{\mathfrak{R}}_S(\mathcal{G}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2n}} \end{aligned} \tag{2.11}$$

Although being data-dependent, the Rademacher complexity is still a worst-case analysis which, given the fact that modern DNNs are more than capable of expressing functions that generalize badly, can be overly pessimistic in the deep learning regime. The calculation of Rademacher complexity can not be directly done as it takes the supreme in the hypothesis class. In practice, Rademacher complexity generalization bounds for neural networks typically rely on a bound on the norm of the parameters [Bartlett and Mendelson, 2002, Bartlett et al., 2017c, Neyshabur et al., 2015c]. We review some representative norm-based bounds in section 2.4.

2.4 Norm-based generalization bounds

Many recent theoretical works on generalization bounds have anchored themselves in the notion that the parameter norm can serve as a means to bound the Rademacher complexity of the function class encapsulated by layered neural networks. Noticeably, Neyshabur et al. [2015c] used an inductive argument to bound the Rademacher complexity of neural networks using the Rademacher complexity of linear separators with bounded l_p norm. Concretely, they proved an upper bound for the Rademacher complexity of a set of functions represented by fully connected networks parameterized by weight matrices with bounded element-wise norm, which can then be plugged into the general Rademacher complexity bound 2.11 to get generalization bounds with parameter norm.

Theorem 5 (Neyshabur et al. [2015c])**.** *Denote by* $\mathcal{N}^{(d,H)}$ *the layered fully connected network with* d *layers and* H *nodes per layer. The sublevel sets of the complexity measure* α *form a family of hypothesis classes* $\mathcal{N}_{\alpha<a}^{(d,H)} = \left\{f \in \mathcal{N}^{d,H} \mid \alpha(f) \leq a\right\}$.

Consider the measure of element-wise weight norm $\gamma_{p,q}(W) = \prod_{k=1}^{d} \|W_k\|_{p,q}$. *For any* $d, q \geq 1$, *any* $1 \leq p < \infty$ *and any dataset* $S = \{x_1, ..., x_n\} \subseteq \mathbb{R}^D$ *we have:*

$$\mathfrak{R}_n(\mathcal{N}^{(d,H)}_{\gamma_{p,q}\leq\gamma}) \leq \sqrt{\frac{\gamma^2 \left(2H^{\left[\frac{1}{p^*}-\frac{1}{q}\right]_+}\right)^{2d-2} \min\left\{p^*, 4\log(2D)\right\} \max_i \|x_i\|^2_{p^*}}{n}} \tag{2.12}$$

Furthermore, Bartlett et al. [2017c] proves generalization guarantee using uniform convergence by bounding the Rademacher complexity of the hypotheses class using an advanced covering number argument, in which the covering number of the neural network representation is bounded by the norm of parameters. (Neyshabur et al. [2017a] proved another bound with very similar dependency on the norms yet using a different non-uniform convergence PAC-Bayesian approach, in which norm is invoked to bound the output change under weight perturbation. The latter, which coincides with the concept of sharpness [Keskar et al., 2016], can be used to calculate the KL-divergence in the PAC-Bayes bound, with an assumption that the prior and perturbation are both Gaussian). Many efforts have been made to improve the applicability of these bounds to realistic scales [Golowich et al., 2018a, Wei and Ma, 2019, Li et al., 2018c, Cao and Gu, 2019, Allen-Zhu et al., 2019, Daniely and Granot, 2019].

It is worth noting that in these generalization bounds, the parameter norm serves merely as an unsolicited proxy/upper bound for the Rademacher complexity, hence they are still based on uniform convergence, i.e. the bound works for all hypotheses in the sublevel set uniformly. A recent influential paper [Nagarajan and Kolter, 2019] showed that for SGD-trained networks, uniform convergence provably cannot explain generalization in the sense that the "best"[2] double-sided bounds based on uniform convergence are almost vacuous for certain families of data distributions. In appendex J therein they extended this result to double-sided PAC-Bayesian bounds (see section 2.5) as well, but the main scenarios where they become problematic is

[2]In Nagarajan and Kolter [2019] this refers to "taking into account the implicit bias of GD to the fullest extent possible", which is specific for particular algorithm and dataset (synthetically constructed in the paper).

when the empirical error is "much" larger than the generalization error for stochastic classifiers, which causes double-sided bounds to be vacuous but is less interesting in practice. A later work [Negrea et al., 2019] showed that one can still make uniform convergence work for the distributions considered in Nagarajan and Kolter [2019] by bounding the difference in risk between the hypotheses returned by SGD and an element from a hypothesis class for which uniform convergence yields tight bounds, but here the hypothesis class on which uniform convergence bound is applied is not the learned hypothesis directly.

2.4.1 How hidden symmetries discredit norm-based bounds

Most norm-based generalization bounds are derived for feed-forward neural networks without many explicit symmetries other than the non-negative homogeneity of ReLU. A direct consequence is while some of the norm-based measures might be invariant to the re-balancing of weights norm (α-scaling), e.g. the path-norm [Neyshabur et al., 2015c], all of them start to make less to none sense when there are stronger symmetries, such as when the network has normalization layers (Batch Normalization [Ioffe and Szegedy, 2015] / Group Normalization [Wu and He, 2018] / Layer Normalization [Ba et al., 2016]), the function $f(\theta)$ simply becomes scale-invariant, i.e. for any $c \in \mathbb{R}^+$, $f(\theta) = f(c\theta)$.

2.4.2 Weight norm growth under SGD

It has been shown that when using MSE-type loss functions SGD converges to the minimal-norm solutions under certain conditions [Satpathi and Srikant, 2021, Gower et al., 2019], while unbounded loss functions without global minimizers like cross-entropy will push the weights norm to infinity[Lyu and Li, 2019, Ji and Telgarsky, 2020].

2.5 PAC-Bayesian bounds

Thus far, we have illuminated the constraints of generalization theories for deep learning grounded in classical uniform convergence analysis. This underscores the

critical significance of non-uniform generalization theories. Among the existing non-uniform generalization bounds, PAC-Bayesian bounds are the most promising ones in that they tend to capture most trends when varying hyperparameters [Jiang et al., 2019a, Dziugaite et al., 2020c] and can be non-vacuous [Dziugaite and Roy, 2017b, Pérez-Ortiz et al., 2021, Lotfi et al., 2022a, Zhou et al., 2019, Valle-Pérez and Louis, 2020]. Before delving into PAC-Bayesian bounds, let's first explore the shift in framework from uniform convergence.

The fundamental idea to introduce non-uniformity into the PAC framework can be surprisingly easy: in the union bound argument of deriving the preliminary uniform convergence generalization bounds 2.6 with the realizability assumption, if we assume a probability distribution P assigning a nonzero probability to every hypothesis in a countable hypothesis class containing the target concept c instead of just assuming they are uniform, we immediately arrive at the simplest non-uniform generalization bound:

Theorem 6 (Simplest non-uniform bound)**.** *For any probability distribution P assigning a nonzero probability to every hypothesis in a countable hypothesis class containing a target hypothesis c, and any probability distribution on instances, we have, for any $\delta > 0$, that with probability at least $1 - \delta$ over the selection of a sample of n instances, the following holds for all hypotheses h agreeing with c on that sample:*

$$\epsilon(h) \leq \frac{\log \frac{1}{P(h)} + \log \frac{1}{\delta}}{n} \tag{2.13}$$

The above non-uniform generalization bound has directly inspired the first PAC-Bayesian generalization bound proposed in McAllester [1998], in which the bound 2.13 was coincided as a preliminary theorem. Note that it reduces to the standard realizable uniform convergence bound 2.6 if P is the uniform distribution $P(h) = 1/|\mathcal{H}|$. This bound is somewhat special in the sense that the term $\log \frac{1}{P(h)}$ alone is not a measure of model complexity in common sense because one is allowed to choose the distribution P arbitrarily. It does, however, admit the idea of inductive

bias of the learning algorithm by assigning better values of the bound to those hypotheses that are more favorable than others. If instead of randomly guessing, we have an (approximate) prior understanding of what hypotheses the algorithm is likely to produce (and those hypotheses are indeed not uniformly distributed), then this basic non-uniform bound will perform much better than uniform convergence bounds. This has been done in recent work by Zhang et al. [2021a], where they showed that under the NNGP prior [Novak et al., 2018b], this $\log \frac{1}{P(h)}$ term alone is a much better indicator of generalization than many other flatness-based measures.

The bound 2.13 alongside previously introduced uniform convergence bounds work for deterministic hypotheses, i.e. they produce deterministic predictions when prompted with data. PAC-Bayesian bounds, however, apply for *stochastic* hypotheses. A stochastic hypothesis h follows a distribution Q, usually called the *posterior* in the PAC-Bayes literature, and predicts the label by drawing from Q each time with different data. The standard form of the general PAC-Bayesian bound was proven by Maurer [2004].

Theorem 7 (General PAC-Bayes, Maurer [2004])**.** *For any prior P over $\mathcal{H}$, the following bound on the generalization gap of a stochastic hypothesis h with any posterior Q holds with probability at least $1-\delta$ over the random choice of S:*

$$KL(\mathbb{E}_{h\sim Q}[\epsilon(h)], \mathbb{E}_{h\sim Q}[\hat{\epsilon}(h)]) \leq \frac{KL(Q||P) + \log\frac{1}{\delta} + \log 2n}{n-1} \tag{2.14}$$

where $KL(Q||P)$ is the KL-divergence between Q and P and for $a, b \in [0,1]$ we abuse the notation and define $KL(a,b) \equiv -a\log b - (1-a)\log(1-b)$

At the core of PAC-Bayesian bounds, the KL divergence from the posterior to the prior $\mathrm{KL}(Q||P)$ can be understood intuitively through a progressive information-theoretic way [Lotfi et al., 2022a]: in the simple uniform convergence bound 2.5 and 2.6, the complexity term $\log|\mathcal{H}|$ can be seen as counting the number of bits needed to specify any hypothesis $h \in \mathcal{H}$. If we do not believe that each hypothesis is equally likely (non-uniform), and consider a prior distribution P over $\mathcal{H}$ that

concentrates around likely hypotheses, we can construct a variable length code that uses fewer bits to specify those hypotheses. Any given hypothesis h will take $\log \frac{1}{P(h)}$ bits - we then arrive at the simplest non-uniform bound in eq. 2.13. If we further consider a distribution of "good" solutions Q, the average number of bits to code a hypothesis from Q using the prior P is the cross entropy $H(Q, P)$. This can also be seen as taking the expectation of $\log \frac{1}{P(h)}$ from the preliminary bound 2.13 over Q: $H(Q, P) = \mathbb{E}_{h\sim Q}[\log \frac{1}{P(h)}]$. We can also get $H(Q)$ bits back from being agnostic about which sample $h \sim Q$ to use (hence the bound is for the average of h over Q), yielding the KL-divergence between Q and P: $H(Q, P) - H(Q) = \mathrm{KL}(Q||P)$. It is worth noting that despite the name "Bayes" appearing in PAC-Bayesian bounds, they are still by nature frequentist. The term "prior" is a reference distribution, and what is called "posterior" is an unrestricted distribution, in the sense that there is no likelihood-type factor connecting these two distributions as in Bayesian methods. The bound itself in e.q. 2.14 applies to all possible posteriors uniformly. With that said, Germain et al. [2016] demonstrated that the optimal PAC-Bayesian posterior aligns with the Bayesian posterior when the loss function is the negative log-likelihood.

Depending on the interpretation of the KL divergence, PAC-Bayesian bounds for DNNs in literature can be roughly divided into two types. From the parameter space perspective, the prior and posterior are chosen to be the distribution of parameters, which are usually Gaussian. A typical choice is to set the prior to be a 0-mean Gaussian with its variance optimized within a predefined set [3], and the posterior to be a Gaussian centered at the trained parameters [Dziugaite and Roy, 2017b, Foret et al., 2020a, Kwon et al., 2021]. For instance, in Dziugaite and Roy [2017b] where they set the parameter-space-based PAC-Bayesian bound as a training object and optimize it to obtain a nonvacuous bound, they chose

[3]The assumption of PAC-Bayesian bounds requires that the prior must be chosen without any knowledge of the data, hence we can not directly optimize the KL term using any data-related information such as the posterior. Instead we can use this trick to make the bound hold for all discrete values of variance in this predefined set simultaneously, without violating the assumption. See [Langford and Caruana, 2001] for more details.

$P = \mathcal{N}(0, \lambda I)$ and $Q = \mathcal{N}(w, \text{diag}(s))$ where λ and s are variances and w are the weights found by the training algorithm (SGD). In this case the KL divergence between two multivariable Gaussian has a simple form:

$$\begin{aligned} &\text{KL}(\mathcal{N}(w, \text{diag}(s))||\mathcal{N}(w_0, \lambda I)) \\ =&\frac{1}{2}\left(\frac{1}{\lambda}\|s\|_1 - d + \frac{1}{\lambda}\|w - w_0\|_2^2 + d\log\lambda - 1_d \cdot \log s\right) \end{aligned} \tag{2.15}$$

where $w \in \mathbb{R}^d$. Another notable example combines the PAC-Bayesian framework with the worst-case sensitivity to parameter perturbation to get sharpness-based generalization bound:

Theorem 8 (PAC-Bayesian sharpness bound [Foret et al., 2020a])**.** *For any $\rho > 0$ and any distribution $\mathcal{D}$, with probability $1 - \delta$ over the choice of $S \sim \mathcal{D}^n$,*

$$\begin{aligned} L_{\mathcal{D}}(\boldsymbol{w}) \leq &\max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} L_S(\boldsymbol{w} + \boldsymbol{\epsilon}) + \\ &\sqrt{\frac{d\log\left(1 + \frac{\|\boldsymbol{w}\|_2^2}{\rho^2}\left(1 + \sqrt{\frac{\log(n)}{d}}\right)^2\right) + 4\log\frac{n}{\delta} + \tilde{O}(1)}{n-1}} \end{aligned} \tag{2.16}$$

where d is the number of parameters and we assume $L_{\mathcal{D}}(\boldsymbol{w}) \leq \mathbb{E}_{\epsilon_i \sim \mathcal{N}(0,\rho)}\left[L_{\mathcal{D}}(\boldsymbol{w} + \boldsymbol{\epsilon})\right]$

A commom result of these parameter-space approaches is that a norm term of the trained parameters is usually unavoidable in the final bound, which undermines their credibility. See the discussion in section 2.4.1. In section 4.6 we will introduce a method to break these parameter-space-based PAC-Bayesian bounds.

On the other hand, one can analyze the prior and posterior distribution of the functions directly. PAC-Bayesian compression bounds are typical examples of this technique [Zhou et al., 2019, Lotfi et al., 2022a]. In these bounds universal prior can be used and the posterior is often a point mass at the trained function. Consequently, the KL term scales with the number of bits needed to specify the function. Hence the more compressible the model is, the better it generalizes. See more discussion of compression bounds in section 2.6.1. Another promising function-space-based PAC-Bayesian bound is the marginal likelihood bound [Valle-Pérez and Louis, 2020]. The authors consider any prior $P(h)$ and an algorithm which

samples hypotheses according to the Bayesian posterior with $0-1$ likelihood. The following generalization bound is proven:

Theorem 9 (marginal-likelihood bound [Valle-Pérez and Louis, 2020])**.** *In binary classification, for any prior distribution P over a hypothesis class $\mathcal{H}$ and any realizable distribution $\mathcal{D}$, with confidence parameter δ and γ, we have that with probability at least $1-\delta$ over the choice of sample S of n instances and probability at least $1-\gamma$ over the choice of h:*

$$-\ln(1-\epsilon(h)) < \frac{\ln \frac{1}{P(C(S))} + \ln n + \ln \frac{1}{\delta} + \ln \frac{1}{\gamma}}{n-1} \tag{2.17}$$

where h is chosen according to the posterior $Q(h) = \frac{P(h)}{\sum_{h \in C(S)} P(h)}$ with $0-1$ likelihood. $C(S)$ is the set of h in $\mathcal{H}$ consistent with the sample S and $P(C(S)) = \sum_{h \in C(S)} P(h)$ in the marginal likelihood of data S.

The marginal-likelihood bound in e.q. 2.17 is essentially a derandomized version from Langford and Seeger [2001], where γ accounts for the extra price to pay for derandomization but is usually negligible. Valle-Pérez and Louis [2020] also proved the asymptotic optimality of marginal likelihood bound in the sense that if the generalization error decreases as a power law with training set size n (scales as $n^{-\alpha}$), the bound also decreases with the same exponential factor.

Apart from the reasons we discussed in section 2.4.1 of why we should prefer function-space-based PAC-Bayesian bounds, it is also noted in Valle-Pérez and Louis [2020] that the KL divergence in function space is provably less or equal than in the parameter space. This is essentially due to the concavity of the logarithm function and the proof is straightforward with the Jensen's inequality applied once. This means we should always aim at the function-space PAC-Bayesian bounds wherever it is possible.

2.6 Other generalization bounds and measures

2.6.1 Generalization bounds based on compression

The idea that compressible machine learning models tend to generalize better dates back to an early formalization called the minimum description length (MDL) principle [Rissanen, 1986, Hinton and van Camp, 1993]. An intuitive way to understand why it is the case: in the simple concentration bound + union bound argument, the main element in the bound is $\log |\mathcal{H}|$ where $\mathcal{H}$ is the hypotheses class. Now if these hypotheses can be represented in a quantized fashion that takes q parameters and each of them has r options, then the term becomes $q \log r$ for this compressed hypotheses class. The more compressible the model is the smaller the bound will be. More recently Arora et al. [2018a] revisited this idea and found that the compressibility of a model is linked to its noise stability. Their bound, being deterministic, is applied to the smaller compressed network. Subsequently, Zhou et al. [2019] developed a stochastic PAC-Bayes compression bound in which the KL divergence term is bounded by the compressed code length after encoding the compressed weights using some prefixed coding scheme. The (degenerate) Gaussian posterior is the same as used in [Dziugaite and Roy, 2017b], while the prior is a weighted mixture of Gaussians with the weights exponentially decaying with code length. This way they obtained a bound less than 1 on the ImageNet. Recently, Lotfi et al. [2022a] further improved the previous results by using a more aggressive setting including a faster-decaying prior, training within the intrinsic dimension [Li et al., 2018a] and stronger quantization scheme.

2.7 Previous empirical work on comparing generalization measures

In this section we review some large-scale empirical works aimed at comparing generalization measures. While it's not uncommon for theoretical work to also present empirical evaluations of their proposed measures [Arora et al., 2018a, Bartlett et al., 2017c, Dziugaite and Roy, 2017b, Neyshabur et al., 2017b, 2019a],

most of them are limited to experiment settings where only favorable results are shown. Jiang et al. [2019a] performed the first large-scale study to test the correlation of different measures with the generalization of deep CNNs with varying hyperparameters. Two metrics were used to quantify this correlation: the granulated Kendall's coefficient and the conditional mutual information (CMI). Even though the latter one is perhaps a more principled metric in that it hopes to capture any *causal* relationships between measures and generalization, the CMI is actually agnostic to the direction of correlation and it shows some conflicting results with the more intuitive granulated Kendall's coefficient. We will mainly introduce the granulated Kendall's coefficient in the following which is also the main metric in the authors' analysis.

The granulated Kendall's coefficient Kendall's correlation coefficient is an effective tool widely used to capture the relationship between 2 rankings of a set of objects. Given a set of models trained with a set of hyperparameters Θ, the Kendall's coefficient between their associated generalization gap $\{g(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\}$ and their respective values of the measure $\{\mu(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\}$ is defined through constructing a set $\mathcal{T}$ of all measure-generalization pairs:

$$\mathcal{T} \triangleq \cup_{\boldsymbol{\theta} \in \Theta} \{(\mu(\boldsymbol{\theta}), g(\boldsymbol{\theta}))\}$$

An ideal complexity measure must be such that, for any pair of trained models, if $\mu(\boldsymbol{\theta}_1) > \mu(\boldsymbol{\theta}_2)$, then so is $g(\boldsymbol{\theta}_1) > g(\boldsymbol{\theta}_2)$. The Kendall's coefficient is then defined as the degree of such consistency holds among the elements of $\mathcal{T}$.

$$\tau(\mathcal{T}) \triangleq \frac{1}{|\mathcal{T}|(|\mathcal{T}|-1)} \sum_{(\mu_1, g_1) \in \mathcal{T}} \sum_{(\mu_2, g_2) \in \mathcal{T} \setminus (\mu_1, g_1)} \operatorname{sign}(\mu_1 - \mu_2) \operatorname{sign}(g_1 - g_2) \tag{2.18}$$

i.e. this is the averaged sign error of all possible combinations of two different models. Because the set $\mathcal{T}$ contains models with varying hyperparameters of varying hyperparameter types, the ablation study becomes harder. The authors hence proposed a granulated version of Kendall's coefficient which computes τ within each hyperparameter type and then takes the average across different types.

Formally, if we partition Θ into Θ_is according to different hyperparameter types (e.g. learning rate, depth etc.) and calculate the average Kendall's coefficient of models that only differ from each other with a particular hyperparameter type i:

$$\psi_i \triangleq \frac{1}{m_i} \sum_{\theta_1 \in \Theta_1} \cdots \sum_{\theta_{i-1} \in \Theta_{i-1}} \sum_{\theta_{i+1} \in \Theta_{i+1}} \cdots \sum_{\theta_T \in \Theta_T} \tau \left(\cup_{\theta_i \in \Theta_i} \{(\mu(\boldsymbol{\theta}), g(\boldsymbol{\theta}))\} \right) \tag{2.19}$$

where $m_i \triangleq |\Theta_1 \times \cdots \times \Theta_{i-1} \times \Theta_{i+1} \times \cdots \times \Theta_T|$, the granulated Kendall's coefficient Ψ is defined as the average of ψ across all hyperparameter axes:

$$\Psi \triangleq \frac{1}{T} \sum_{i=1}^{T} \psi_i \tag{2.20}$$

where T is the number of types of hyperparameters.

The authors of Jiang et al. [2019a] provided a thought experiment to justify their choice of the granulated version of Kendall's τ coefficient over the original one: suppose there exists a measure that perfectly captures the depth of the network while producing random prediction if 2 networks have the same depth, this measure would do reasonably well in terms of $\tau(\mathcal{T})$ but much worse in terms of Ψ. This is to say that while a perfect measure can achieve a good score in both $\tau(\mathcal{T})$ and Ψ, a bad measure as described in the thought experiment will achieve only slightly worse $\tau(\mathcal{T})$ score but a much worse Ψ score.

With this methodology, they empirically found that

- Many norm-based measures not only perform poorly but negatively correlate with generalization;
- Sharpness-based measures like PAC-Bayesian bounds or the worst-case sharpness as proposed by Keskar et al. [2016] perform best under their metric;
- Some measures related to the optimization procedures can also be predictive of generalization, but the theories behind these measures tend to be elusive.

Building on Jiang et al. [2019a], Dziugaite et al. [2020c] took a slightly different approach by critiquing the use of averaging in the evaluation of predictions made by those generalization measures. Instead, they argue that a theory—realized by a generalization measure—is as strong as its weakest part. The "deceptively minor" changes they made can be summarized into the following:

- Look at the worst instead of the average sign-error with respect to hyperparameter types. Formally, their definition of *robust sign-error* is

$$\Psi_{\text{robust}} \triangleq \max_{i \in [T]} \frac{1 - \psi_i}{2} \tag{2.21}$$

- A filter is also applied on samples to account for Monte Carlo noise. Concretely, they use a weighted average in the calculation of ψ_i where the weights are proportional to the difference in generalization error, and simply discard model pairs whose generalization error difference is too small (under a given threshold).

Indeed, this worst-case analysis of generalization measures can reveal the failure of some measures that otherwise would have been obscured by the averaging analysis adopted by Jiang et al. [2019a], e.g. some bounds based on Frobenius norms can even increase with train set size, but if looked at on average with other hyperparameter axes this surprising failure may not be as noticeable. In fact, Dziugaite et al. [2020c] have concluded that no measure that they have considered is robust under this worst-case analysis as there is at least one environment in which the measure always incorrectly predicts the direction of change in generalization. That being said, they also reconfirmed some main points made in Jiang et al. [2019a] such as the comparatively better performance of the path-norm and PAC-Bayes-based measures and the poor performance of many other norm-based measures.

Apart from the aforementioned quantitative empirical work, Valle-Pérez and Louis [2020] adopted a different half-qualitative methodology by proposing a set of desiderata that a good predictive generalization theory should satisfy. These desiderata are:

1. Data complexity: A good generalization predictor should scale with data complexity correctly. This includes datasets that come with naturally different complexity, e.g. CIFAR10 is typically harder to classify than MNIST for a fixed DNN, as well as data with different noise, e.g. pixel-shuffled or label-corrupted data will likely be harder than the noise-free version.[4]

2. Architectures: The predicted error should effectively capture variations across different architectures. Moreover, a perplexing characteristic of DNNs is their weak dependence on the number of parameters, particularly when the system is sufficiently large. Given that the inquiry into why DNNs generalize effectively in the overparameterized regime is a central question in DNN theory, it becomes crucial for a predicted error to replicate this relative insensitivity to the number of parameters.

3. Training set size: The predicted error should correctly scale with training set size, not only in direction but preferably also in speed. Empirical findings suggest that the generalization error frequently exhibits a power law decay concerning the training set size [Hestness et al., 2017, Rosenfeld et al., 2019, Kaplan et al., 2020]. This is not captured by simply considering the sign-error.

4. Optimization algorithms: The theory should effectively account for variations in generalization resulting from different optimization algorithms. Distinct optimization algorithms employed in training DNNs, along with varied choices of training hyperparameters such as SGD batch size, learning rates, or diverse regularization techniques, can yield differences in generalization. The theoretical framework should strive to predict these differences.

5. Non-vacuous: For theoretically rigorous generalization bounds, the predicted error upper bound should be quantitatively close to the true error.

[4] In Jiang et al. [2019a] it was argued that artificially modifying the data complexity is not representative of the typical setting. However, this stance appears to be in contrast to their broader goal of identifying the causal mechanisms behind generalization, which are likely to manifest in scenarios beyond "natural" settings, such as those involving noisy data.

6. Computation efficiency: The predicted error should be efficiently computable. This is important for practical applications, e.g. model selection.

7. Rigorousness: We should prefer rigorous theories over measures based on intuition or unverified assumptions.

After setting up the framework for testing generalization theories, the authors listed some representative generalization bounds from different categories, mostly in an abstract form (e.g. only introducing what factors the bounds depend on without detailing the full dependency) and compared them against the desiderata in a qualitative manner. They did extensive experiments on one particular bound - the marginal likelihood bound (e.q. 2.17) - and found that this is the best one according to the desiderata. Note that this optimality of the marginal likelihood bound does not imply that it fulfills all the desiderata that they proposed, but just most of them.

3

Why Flatness Does and Does Not Correlate with Generalization for Deep Neural Networks

Overview

The intuition that local flatness of the loss landscape is correlated with better generalization for deep neural networks (DNNs) has been explored for decades, spawning many different flatness measures. Recently, this link with generalization has been called into question by a demonstration that many measures of flatness are vulnerable to parameter re-scaling which arbitrarily changes their value without changing neural network outputs. Here we show that, in addition, some popular variants of stochastic gradient descent (SGD) such as Adam and Entropy-SGD can also break the flatness–generalization correlation. As an alternative to flatness measures, we use a function-based picture and propose using the log of the Bayesian prior upon initialization, $\log P(f)$, as a predictor of the generalization when a DNN converges on function f after training to zero error. The prior is directly proportional to the Bayesian posterior for functions that give zero error on a test set. For the case of image classification, we show that $\log P(f)$ is a significantly more robust predictor of generalization than flatness measures are. Whilst local flatness measures fail under

parameter re-scaling, the prior/posterior, which is global quantity, remains invariant under re-scaling. Moreover, the correlation with generalization as a function of data complexity remains good for different variants of SGD.

3.1 Chapter introduction

Among the most important theoretical questions in the field of deep learning are: 1) What characterizes functions that exhibit good generalization?, and 2) Why do overparameterized deep neural networks (DNNs) converge to this small subset of functions that do not overfit? Perhaps the most popular hypothesis is that good generalization performance is linked to flat minima. In pioneering works [Hinton and van Camp, 1993, Hochreiter and Schmidhuber, 1997a], the minimum description length (MDL) principle [Rissanen, 1978] was invoked to argue that since flatter minima require less information to describe, they should generalize better than sharp minima. Most measures of flatness approximate the local curvature of the loss surface, typically defining flatter minima to be those with smaller values of the Hessian eigenvalues [Keskar et al., 2016, Wu et al., 2017, Zhang et al., 2018, Sagun et al., 2016, Yao et al., 2018].

Another commonly held belief is that stochastic gradient descent (SGD) is itself biased towards flatter minima, and that this inductive bias helps explain why DNNs generalize so well [Keskar et al., 2016, Jastrzebski et al., 2018, Wu et al., 2017, Zhang et al., 2018, Yao et al., 2018, Wei and Schwab, 2019, Maddox et al., 2020]. For example Keskar et al. [2016] developed a measure of flatness that they found correlated with improved generalization performance when decreasing batch size, suggesting that SGD is itself biased towards flatter minima. We note that others [Goyal et al., 2017, Hoffer et al., 2017, Smith et al., 2017a, Mingard et al., 2021] have argued that the effect of batch size can be compensated by changes in learning rate, complicating some conclusions from Keskar et al. [2016]. Nevertheless, the argument that SGD is somehow itself biased towards flat minima remains widespread in the literature.

In an important critique of local flatness measures, Dinh et al. [2017a] pointed out that DNNs with ReLU activation can be re-parameterized through a simple parameter-rescaling transformation.

$$T_\alpha : (\mathbf{w}_1, \mathbf{w}_2) \mapsto \left(\alpha \mathbf{w}_1, \alpha^{-1} \mathbf{w}_2\right) \tag{3.1}$$

where $\mathbf{w}_1$ are the weights between an input layer and a single hidden layer, and $\mathbf{w}_2$ are the weights between this hidden layer and the outputs. This transformation can be extended to any architecture having at least one single rectified network layer. The function that the DNN represents, and thus how it generalizes, is invariant under parameter-rescaling transformations, but the derivatives w.r.t. parameters, and therefore many flatness measures used in the literature, can be changed arbitrarily. *Ergo*, the correlation between flatness and generalization can be arbitrarily changed.

Several recent studies have attempted to find "scale invariant" flatness metrics [Petzka et al., 2019, Rangamani et al., 2019, Tsuzuku et al., 2019]. The main idea is to multiply layer-wise Hessian eigenvalues by a factor of $\|\mathbf{w_i}\|^2$, which renders the metric immune to layer-wise re-parameterization. While these new metrics look promising experimentally, they are only scale-invariant when the scaling is layer-wise. Other methods of rescaling (e.g. neuron-wise rescaling) can still change the metrics, so this general problem remains unsolved.

3.1.1 Main contributions

1. For a series of classic image classification tasks (MINST and CIFAR-10) we show that flatness measures change substantially as a function of epochs. Parameter re-scaling can arbitrarily change flatness, but it quickly recovers to a more typical value under further training. We also demonstrate that some variants of SGD exhibit significantly worse correlation of flatness with generalization than found for vanilla SGD. In other words popular measures of flatness sometimes do and sometimes do not correlate with generalization. This mixed performance problematizes a widely held intuition that DNNs generalize well fundamentally because SGD or its variants are themselves biased towards flat minima.

2. We next study the correlation of the Bayesian prior $P(f)$ with the generalization performance of a DNN that converges to that function f. This prior is the weighted probability of obtaining function f upon random sampling of parameters. Motivated by a theoretical argument derived from a non-uniform convergence generalization bound, we show empirically that $\log P(f)$ correlates robustly with test error, even when local flatness measures miserably fail, for example upon parameter re-scaling. For discrete input/output problems (such as classification), $P(f)$ can also be interpreted as the weighted "volume" of parameters that map to f. Intuitively, we expect local flatness measures to typically be smaller (flatter) for systems with larger volumes. Nevertheless, there may also be regions of parameter space where local derivatives and flatness measures vary substantially, even if on average they correlate with the volume. Thus flatness measures can be viewed as (imperfect) local measures of a more robust predictor of generalization, the volume/prior $P(f)$.

3.2 Definitions and notation

3.2.1 Supervised learning

For a typical supervised learning problem, the *inputs* live in an input domain $\mathcal{X}$, and the *outputs* belong to an output space $\mathcal{Y}$. For a *data distribution* $\mathcal{D}$ on the set of input-output pairs $\mathcal{X} \times \mathcal{Y}$, the *training set* S is a sample of n input-output pairs sampled i.i.d. from $\mathcal{D}$, $S = \{(x_i, y_i)\}_{i=1}^{n} \sim \mathcal{D}^n$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. The output of a DNN on an input x_i is denoted as $\hat{y}_i$. Typically a DNN is trained by minimising a *loss function* $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, which measures differences between the output $\hat{y} \in \mathcal{Y}$ and the observed output $y \in \mathcal{Y}$, by assigning a score $L(\hat{y}, y)$ which is typically zero when they match, and positive when they don't match. DNNs are typically trained by using an optimization algorithm such as SGD to minimize the loss function on a training set S. The generalization performance of the DNN, which is theoretically defined over the underlying (typically unknown) data distribution $\mathcal{D}$ but is practically measured on a *test set* $E = \{(x'_i, y'_i)\}_{i=1}^{|E|} \sim \mathcal{D}^{|E|}$. For classification problems, the *generalization error* is practically measured as

$\epsilon(E) = \frac{1}{|E|}\sum_{x'_i \in E} \mathbb{1}[\hat{y}_i \neq y'_i]$, where $\mathbb{1}$ is the standard indicator function which is one when its input is true, and zero otherwise.

3.2.2 Flatness measures

Perhaps the most natural way to measure the flatness of minima is to consider the eigenvalue distribution of the Hessian $H_{ij} = \partial^2 L(\mathbf{w})/\partial w_i \partial w_j$ once the learning process has converged (typically to a zero training error solution). Sharp minima are characterized by a significant number of large positive eigenvalues λ_i in the Hessian, while flat minima are dominated by small eigenvalues. Some care must be used in this interpretation because it is widely thought that DNNs converge to stationary points that are not true minima, leading to negative eigenvalues and complicating their use in measures of flatness. Typically, only a subset of the positive eigenvalues are used [Wu et al., 2017, Zhang et al., 2018]. Hessians are typically very expensive to calculate. For this reason, Keskar et al. [2016] introduced a computationally more tractable measure called "sharpness":

Definition 5 (Sharpness)**.** *Given parameters* $\mathbf{w}'$ *within a box in parameter space* $\mathcal{C}_\zeta$ *with sides of length* $\zeta > 0$*, centered around a minimum of interest at parameters* $\mathbf{w}$*, the sharpness of the loss* $L(\mathbf{w})$ *at* $\mathbf{w}$ *is defined as:*

$$\text{sharpness} := \frac{\max_{\mathbf{w}' \in \mathcal{C}_\zeta} \left(L(\mathbf{w}') - L(\mathbf{w})\right)}{1 + L(\mathbf{w})} \times 100.$$

In the limit of small ζ, the sharpness relates to the spectral norm of the Hessian [Dinh et al., 2017a]:

$$\text{sharpness} \approx \frac{\left|\left|\left|(\nabla^2 L(\mathbf{w}))\right|\right|\right|_2 \zeta^2}{2(1 + L(\mathbf{w}))} \times 100.$$

The general concept of flatness can be defined as $1/sharpness$, and that is how we will interpret this measure in the rest of this paper.

3.2.3 Functions and the Bayesian prior

We first clarify how we represent functions in the rest of paper using the notion of *restriction of functions*. A more detailed explanation can be found in Section B.3. Here we use binary classification as an example:

Definition 6 (Restriction of functions to C)**.** *[Shalev-Shwartz and Ben-David, 2014]*

Consider a parameterized supervised model, and let the input space be $\mathcal{X}$ and the output space be $\mathcal{Y}$, noting $\mathcal{Y} = \{0, 1\}$ in binary classification setting. The space of functions the model can express is a (potentially uncountably infinite) set $\mathcal{F} \subseteq \mathcal{Y}^{|\mathcal{X}|}$. Let $C = \{c_1, \ldots, c_n\} \subset \mathcal{X}$. The restriction of $\mathcal{F}$ to C is the set of functions from C to $\mathcal{Y}$ that can be derived from functions in $\mathcal{F}$:

$$\mathcal{F}_C = \{(f(c_1), \ldots, f(c_n)) : f \in \mathcal{F}\}$$

where we represent each function from C to $\mathcal{Y}$ as a vector in $\mathcal{Y}^{|C|}$.

For example, for binary classification, if we restrict the functions to $S + E$, then each function in $\mathcal{F}_{S+E}$ is represented as a binary string of length $|S| + |E|$. In the rest of paper, we simply refer to "functions" when we actually mean the restriction of functions to $S + E$, except for the Boolean system in Section 3.5.1 where no restriction is needed. See Section B.3 for a thorough explanation.

For discrete functions, we next define the prior probability $P(f)$ as

Definition 7 (Prior of a function)**.** *Given a prior parameter distribution $P_w(\mathbf{w})$ over the parameters, the* prior of function f *can be defined as:*

$$P(f) := \int \mathbb{1}[\mathcal{M}(\mathbf{w}) = f] P_w(\mathbf{w}) d\mathbf{w}. \tag{3.2}$$

where $\mathbb{1}$ is an indicator function:$\mathbb{1}[arg] = 1$ if its argument is true or 0 otherwise; $\mathcal{M}$ is the parameter-function map whose formal definition is in Section B.2. Note that $P(f)$ could also be interpreted as a weighted volume $V(f)$ over parameter space.

If $P_w(\mathbf{w})$ is the distribution at initialization, the $P(f)$ is the prior probability of obtaining the function at initialization. We normally use this parameter distribution when interpreting $P(f)$.

Remark. Definition 7 works in the situation where the space $\mathcal{X}$ and $\mathcal{Y}$ are discrete, where $P(f)$ has a prior probability mass interpretation. This is enough for most image classification tasks. Nevertheless, we can easily extend this definition to the continuous setting, where we can also define a *prior density* over functions upon random initialization, with the help of Gaussian Process [Rasmussen, 2003]. For the Gaussian Process prior see Section B.4. However, in this work, we focus exclusively on the classification setting, with discrete inputs and outputs.

3.2.4 Link between the prior and the Bayesian posterior

Due to their high expressivity, DNNs are typically trained to zero training error on the training set S. In this case the Bayesian picture simplifies Valle-Pérez et al. [2018], Mingard et al. [2021] because if functions are conditioned on zero error on S, this leads to a simple 0-1 *likelihood* $P(S|f)$, indicating whether the data is consistent with the function. Bayesian inference can be used to calculate a Bayesian *posterior probability* $P_B(f|S)$ for each f by conditioning on the data according to Bayes rule. Formally, if $S = \{(x_i, y_i)\}_{i=1}^n$ corresponds to the set of training pairs, then

$$P_B(f|S) = \begin{cases} P(f)/P(S) \text{ if } \forall i, \quad f(x_i) = y_i \\ 0 \text{ otherwise .} \end{cases}$$

where $P(f)$ is the Bayesian prior and $P(S)$ is called the *marginal likelihood* or *Bayesian evidence*. If we define, the training set neutral space $\mathcal{N}_S$ as all parameters that lead to functions that give zero training error on S, then $P(S) = \int_{\mathcal{N}_S} P_{\mathbf{w}}(\mathbf{w})d\mathbf{w}$. In other words, it is the total prior probability of all functions compatible with the training set S [Valle-Pérez et al., 2018, Mingard et al., 2021]. Since $P(S)$ is constant for a given S, $P_B(f|S) \propto P(f)$ for all f consistent with that S.

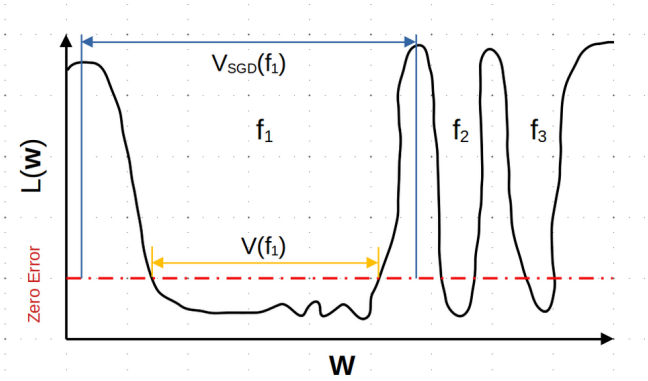


Figure 3.1: Schematic loss landscape for three functions that have zero-error on the training set. It illustrates how the relative sizes of the volumes of their basins of attraction $V_{\mathrm{SGD}}(f_i)$ correlate with the volumes $V(f_i)$ (or equivalently their priors $P(f_i)$) of the basins, and that, on average, larger $V(f_i)$ or $P(f_i)$ implies flatter functions, even if flatness can vary locally. Note that the loss $L(\mathbf{w})$ can vary within a region where the DNN achieves zero classification error on S.

3.3 The correlation between the prior and generalization

This link between the prior and the posterior is important, because it was empirically found in an extensive set of experiments by [Mingard et al., 2021] that, for popular architectures and data sets,

$$P_B(f|S) \approx P_{\mathrm{SGD}}(f|S), \tag{3.3}$$

where $P_{\mathrm{SGD}}(f|S)$ is the probability that a DNN trained with SGD converges on function f, when trained to zero error on S. In other words, to first order, SGD appears to find functions with a probability predicted by the Bayesian posterior, and thus with probabilities directly proportional to $P(f)$. The authors traced this behaviour to the geometry of the loss-landscape, as follows. Some general observations from algorithmic information theory (AIT) [Valle-Pérez et al., 2018] as well as direct calculations [Mingard et al., 2019] predict that the priors of functions should vary over many orders of magnitude. When this is the case, it is reasonable to expect that the probabilities by which an optimizer finds different functions is affected by these large differences. This is related to a mechanism identified previously in evolutionary dynamics, where it is called the arrival of the

frequent [Schaper and Louis, 2014]. We illustrate this principle in Fig. 3.1 where we intuitively use the language of "volumes". We expect that the relative sizes of the basins of attraction $V_{SGD}(f)$, defined as the set of initial parameters for which a DNN converges to a certain function f, is proportional, to first order, to those of the priors $P(f)$ (or equivalently the "volumes"). To second order there are, of course, many other features of a search method and a landscape that affect what functions a DNN converges on, but when the volumes/priors vary by so many orders of magnitude then we expect that to first order $P_{SGD}(f) \approx P_B(f|S) \propto P(f) = V(f)$.

Given that the $P(f)$ of a function helps predict how likely SGD is to converge on that function, we can next ask how $P(f)$ correlates with generalization. Perhaps the simplest argument is that if DNNs trained to zero error are known to generalize well on unseen data, then the probability of converging on functions that generalize well must be high. The $P(f)$ of these functions must be larger than the priors of functions that do not generalize well.

Can we do better than this rather simplistic argument? One way forward is empirical. Mingard et al. [2021] showed that $\log\left(P_B(f|S)\right)$ correlates quite tightly with generalization error. These authors also made a theoretical argument based on the Poisson-Binomial nature of the error distribution to explain this log-linear relationship, but this approach needs further work.

One of the best overall performing predictors in the literature for generalization performance on classification tasks is the marginal likelihood PAC-Bayes bound from [Valle-Pérez et al., 2018, Valle-Pérez and Louis, 2020]. It is non-vacuous, relatively tight, and can capture important trends in generalization performance with training set size (learning curves), data complexity, and architecture choice (see also [Liu et al., 2021]). However, the prediction uses the marginal likelihood $P(S)$ defined through a sum over all functions that produce zero error on the training set. Here we are interested in the generalization properties of single functions.

One way forward is to use a simple nonuniform bound which to the best of our knowledge was first published in [McAllester, 1998] as a preliminary theory to the full PAC-Bayes theorems. For any countable function space $\mathcal{F}$, any distribution

$\tilde{P}$, and for any selection of a training set S of size n under probability distribution $\mathcal{D}$, it can be proven that for all functions f that give zero training error:

$$\forall \mathcal{D}, \mathbf{P}_{S\sim\mathcal{D}^n}\left[\epsilon_{S,E}(f) \leq \frac{\ln \frac{1}{\tilde{P}(f)} + \ln \frac{1}{\delta}}{n}\right] \geq 1-\delta \tag{3.4}$$

for $\delta \in (0,1)$. Here we consider a space $\mathcal{F}_{S,E}$ of functions with all possible outputs on the inputs of a specific E and zero error on a specific S; $\epsilon_{S,E}(f)$ is the error measured on $E+S$, which as the error on S is 0, equals the error on the test set E. This error will converge to the true generalization error on all possible inputs as $|E|$ increases. Valle-Pérez and Louis [2020] showed this bound has an optimal average generalization error when $\tilde{P}(f)$ mimics the probability distribution over functions of the learning algorithm. If $P_{SGD}(f) \approx P_B(f|S) \propto P(f)$, then the best performance of the bound is approximately when $\tilde{P}(f)$ in Eq. (3.4) is the Bayesian prior $P(f)$. Thus this upper bound on $\epsilon_{S,E}(f)$ scales as $-\log(P(f))$.

3.4 Flatness, priors and generalization

The intuition that larger $P(f)$ correlates with greater flatness is common in the literature, see e.g. Hochreiter and Schmidhuber [1997a], Wu et al. [2017], where the intuition is also expressed in terms of volumes. If volume/$P(f)$ correlates with generalization, we expect flatness should too. Nevertheless, local flatness may still vary significantly across a volume. For example Izmailov et al. [2018a] show explicitly that even in the same basin of attraction, there can be flatter and sharper regions. We illustrate this point schematically in Fig. 3.1, where one function clearly has a larger volume and on average smaller derivatives of the loss w.r.t. the parameters than the others, and so is flatter on average. But, there are also local areas within the zero-error region where this correlation does not hold. One of the main hypotheses we will test in this paper is that the correlation between flatness and generalization can be broken even when the generalization-prior correlation remains robust.

3.5 Experimental Results

3.5.1 Prior/volume - flatness correlation for Boolean system

We first study a model system for Boolean functions of size $n = 7$, which is small enough to directly measure the prior by sampling [Valle-Pérez et al., 2018]. There are $2^7 = 128$ possible binary inputs. Since each input is mapped to a single binary output, there are $2^{128} = 3.4 \times 10^{34}$ possible functions f. It is only practically possible to sample the prior $P(f)$ because it is highly biased [Valle-Pérez et al., 2018, Mingard et al., 2019], meaning a subset of functions have priors much higher than average. For a fully connected network (FCN) with two hidden layers of 40 ReLU units each (which was found to be sufficiently expressive to represent almost all possible functions) we empirically determined $P(f)$ using 10^8 random samples of the weights $\mathbf{w}$ over an initial Gaussian parameter distribution $P_w(\mathbf{w})$ with standard deviation $\sigma_w = 1.0$ and offset $\sigma_b = 0.1$.

We also trained our network with SGD using the same initialization and recorded the top-1000 most commonly appearing output functions with zero training error on all 128 outputs, and then evaluated the sharpness/flatness using Definition 5 with an $\epsilon = 10^{-4}$. For the maximization process in calculating sharpness/flatness, we ran SGD for 10 epochs and make sure the max value ceases to change. As Fig. 3.2 demonstrates, the flatness and prior correlate relatively well; Fig. 3.7 in the appendix shows a very similar correlation for the spectral norm of the Hessian. Note that since we are studying the function on the complete input space, it is not meaningful to speak of correlation with generalization. However, since for this system the prior $P(f)$ is known to correlate with generalization [Mingard et al., 2021], the correlation in Fig. 3.2 also implies that these flatness measures will correlate with generalization, at least for these high $P(f)$ functions.

3.5.2 Priors, flatness and generalization for MNIST and CIFAR-10

We next study the correlation between generalization, flatness and $\log P(f)$ on the real world datasets MNIST [LeCun et al., 1998] and CIFAR-10 [Krizhevsky et al.,

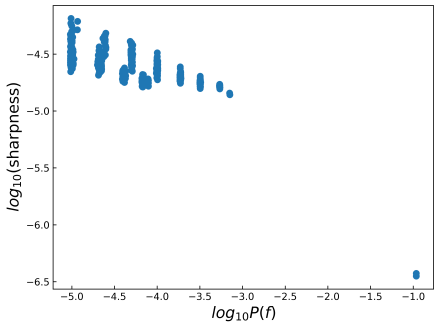


Figure 3.2: The correlation between flatness and the Bayesian prior for the n = 7 Boolean system. The functions are defined on the full space of 128 possible inputs. The priors $P(f)$ are shown for the 1000 most frequently found functions by SGD from random initialization for a two hidden layer FCN, and correlate well with log(flatness). The function the largest prior, which is the most "flat" is the trivial one of all 0s or all 1s. An additional feature is two offset bands caused by a discontinuity of Boolean functions. Most functions shown are mainly 0s or mainly 1s, and the two bands correspond to an even or odd number of outliers (e.g. 1's when the majority is 0s).

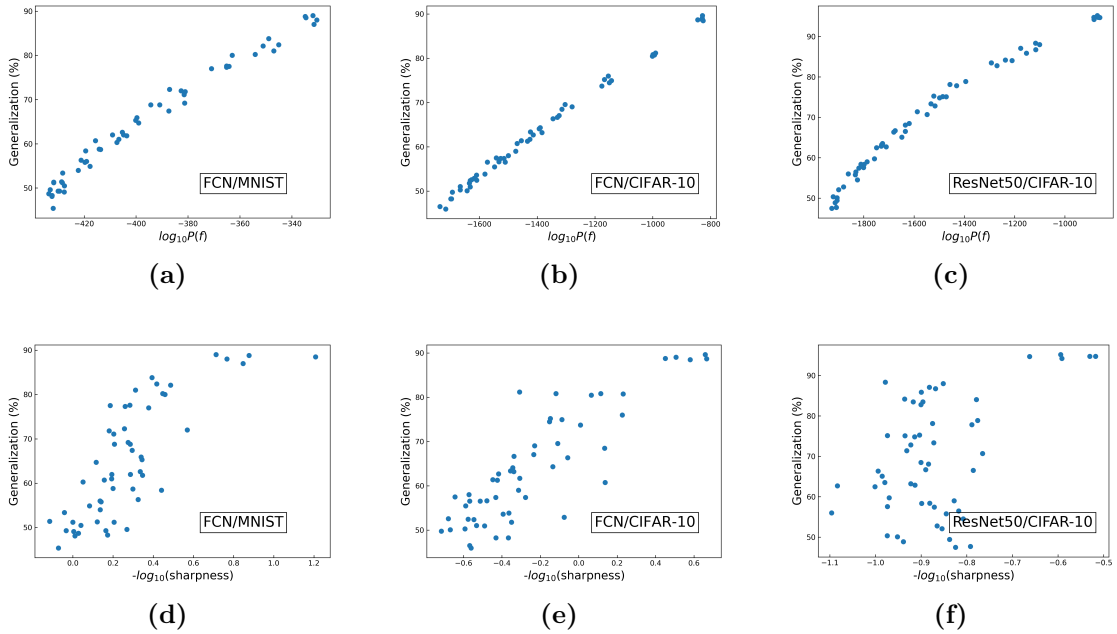


Figure 3.3: The correlation between $\log P(f)$, sharpness and generalization accuracy on MNIST and CIFAR-10. For MNIST $|S|$=500, $|E|$=1000; for CIFAR-10 $|S|$=5000, $|E|$=2000. The attack set size $|A|$ varies from 0 to $|S|$ and generates functions with different generalization performance. (a)-(c) depicts the correlation between generalization and $\log P(f)$ for FCN on MNIST, FCN on CIFAR-10 and Resnet-50 on CIFAR-10, respectively. (d)-(f) show the correlation between generalization and flatness for FCN on MNIST, FCN on CIFAR-10, and Resnet50 on CIFAR-10, respectively. In this experiment, all DNNs are trained with vanilla SGD.

2009].

Because we need to run many different experiments, and measurements of the prior and flatness are computationally expensive, we simplify the problem by binarizing MINST (one class is 0-4, the other is 5-9) and CIFAR-10 (we only study two categories out of ten: cars and cats). Also, our training sets are relatively small (500/5000 for MNIST/CIFAR-10, respectively) but we have checked that our overall results are not affected by these more computationally convenient choices. In Appendix Fig. 3.22 we show results for MNIST with $|S| = 10000$.

We use two DNN architectures: a relatively small vanilla two hidden-layer FCN with 784 inputs and 40 ReLU units in each hidden layer each, and also Resnet-50 [He et al., 2016], a 50-layer deep convolutional neural network, which is much closer to a state of the art (SOTA) system.

We measure the flatness on cross-entropy (CE) loss at the epoch where SGD first obtains zero training error. Because the Hessian is so expensive to calculate, we mainly use the sharpness/flatness measure (Definition 5) which is proportional to the Frobenius norm of the Hessian. The final error is measured in the standard way, after applying a sigmoid to the last layer to binarize the outputs.

To measure the prior, we use the Gaussian processes (GPs) to which these networks reduce in the limit of infinite width [Lee et al., 2017, Matthews et al., 2018, Novak et al., 2018b]. As demonstrated in Mingard et al. [2021], GPs can be used to approximate the Bayesian posteriors $P_B(f|S)$ for finite width networks. For further details, we refer to the original papers above and to Section B.4.

In order to generate functions f with zero error on the training set S, but with diverse generalization performance, we use the attack-set trick from Wu et al. [2017]. In addition to training on S, we add an attack set A made up of incorrectly labelled data. We train on both S and A, so that the error on S is zero but the generalization performance on a test set E is reduced. The larger A is w.r.t. S, the worse the generalization performance. As can be seen in Fig. 3.3(a)-(c), this process allows us to significantly vary the generalization performance. The

correlation between $\log P(f)$ and generalization error is excellent over this range, as expected from our arguments in Section 3.3.

Figs.3.3(d)-(f) show that the correlation between flatness and generalization is much more scattered than for $\log P(f)$. In Section 3.8 we also show the direct correlation between $\log P(f)$ an flatness which closely resembles Fig. 3.3(d)-(f) because $V(f)$ and ϵ correlate so tightly.

3.5.3 The effect of optimizer choice on flatness

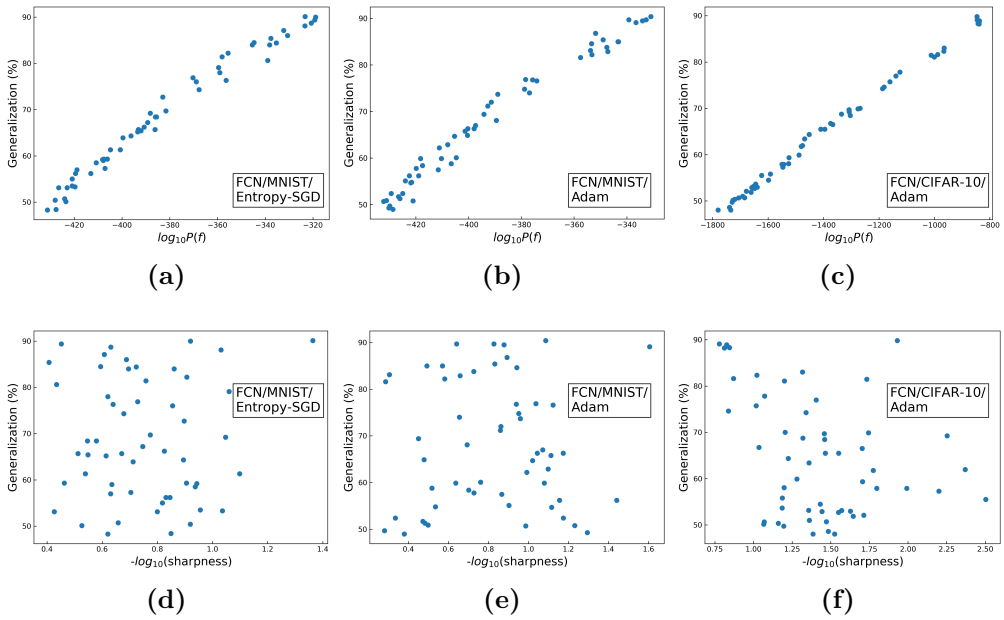


Figure 3.4: SGD-variants can break the flatness-generalization correlation, but not the $\log P(f)$-generalization correlation. The figures show generalization v.s. $\log P(f)$ or flatness for the FCN trained on (a) and (d) – MNIST with Entropy-SGD; (b) and (e) – MNIST with Adam; (c) and (f) – CIFAR-10 with Adam. for the same S and E as in Fig. 3.3. Note that the correlation with the prior is virtually identical to vanilla SGD, but that the correlation with flatness measures changes significantly.

Given that we test the effect of changing the optimizer from the vanilla SGD we used in Fig. 3.3. We use Adam [Kingma and Ba, 2014], and entropy-SGD [Chaudhari et al., 2019] which includes an explicit term to maximize the flatness. Both SGD variants show good optimization performance for the standard default Tensorflow hyperparameters we use. Their generalization performance, however, does not significantly vary from plain SGD, and this is reflected in the

priors of the functions that they find. More importantly, fig. 3.4 shows that the generalization-flatness correlation can be broken by using these optimizers, whereas the $\log P(f)$-generalization correlation remains intact. A similar breakdown of the correlation persists upon overtraining and can also be seen for flatness measures that use Hessian eigenvalues (Fig. 3.14 to Fig. 3.19).

Changing optimizers or changing hyperparameters can, of course, alter the generalization performance by small amounts, which may be critically important in practical applications. Nevertheless, as demonstrated in Mingard et al. [2021], the overall effect of hyperparameter or optimizer changes is usually quite small on these scales. The large differences in flatness generated simply by changing the optimizer suggests that flatness measures may not always reliably capture the effects of hyperparameter or optimizer changes. Note that we find less deterioration when comparing SGD to Adam for Resnet50 on CIFAR-10, (Fig. 3.20). The exact nature of these effects remains subtle.

3.5.4 Temporal behavior of sharpness and $\log P(f)$

In the experiments above, the flatness and $\log P(f)$ metrics are calculated at the epoch where the system first reaches 100% training accuracy. In Fig. 3.5, we measure the prior and the flatness for each epoch for our FCN, trained on MNIST (with no attack set). Zero training error is reached at epoch 140, and we overtrain for a further 1000 epochs. From initialization, both the sharpness measure from Definition 2.1, and $\log P(f)$ reduce until zero-training error is reached. Subsequently, $\log P(f)$ stays constant, but the cross-entropy loss continues to decrease, as expected for such classification problems. This leads to a reduction in the sharpness measure (greater flatness) even though the function, its prior, and the training error don't change. This demonstrates that flatness is a relative concept that depends, for example, on the duration of training. In Figs. 3.14 and 3.15 we show for an FCN on MNIST that the quality of flatness-generalization correlations are largely unaffected by overtraining, for both SGD and Adam respectively, even though the absolute values of the sharpness change substantially.

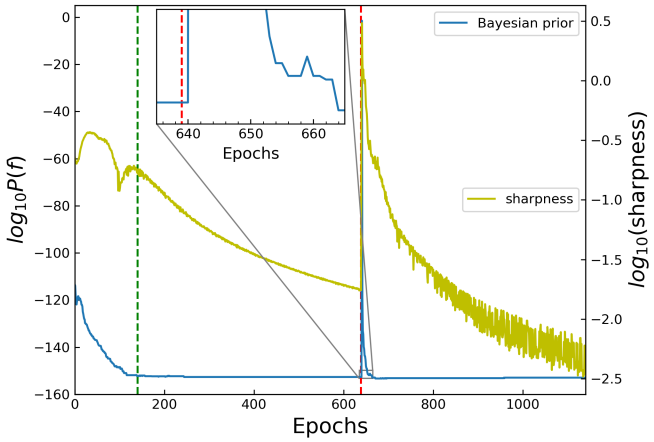


Figure 3.5: How flatness evolves with epochs. At each epoch we calculate the sharpness measure from Definition 2.1 (sharpness is the inverse of flatness) and the prior for our FCN on MNIST with $|S| = 500$. The green dashed line denotes epoch 140 where zero-training error is reached and post-training starts. The red dashed line denotes epoch 639 where α-scaling takes place with $\alpha = 5.9$. Upon parameter-rescaling, the sharpness increases markedly, but then quickly decreases again. The inset shows that the prior is initially unchanged after parameter-rescaling. However, large gradients mean that in subsequent SGD steps, the function (and its prior) changes, before recovering to (nearly) the same function and $\log P(f)$.

One of the strong critiques of flatness is that re-parameterisations such as the parameter-rescaling transformation defined in Eq. (3.1) can arbitrarily change local flatness measures [Dinh et al., 2017a]. Fig. 3.5 shows that parameter-rescaling indeed leads to a spike in the sharpness measure (a strong reduction in flatness). As demonstrated in the inset, the prior is initially invariant upon parameter-rescaling because $f(\mathbf{w})$ is unchanged. However, parameter-rescaling can drive the system to unusual parts of the volume with steep gradients in the loss function, which mean that SGD falls off the zero training error manifold. $\log P(f)$ goes up because it is more likely to randomly fall onto large $V(f)$ functions. However, the system soon relaxes to essentially the same function and $\log P(f)$. In Fig. B.2, we show that it is possible to obtain a spike in the sharpness measure without the prior changing. In each case, the sharpness measure rapidly decays after the spike, suggesting that parameter-rescaling brings the system into a parameter region that is "unnatural".

3.6 Discussion and future work

The notion that flatness correlates with generalization is widely believed in the community, but the evidential basis for this hypothesis has always been mixed. Here we performed extensive empirical work showing that flatness can indeed correlate with generalization. However, this correlation is not always tight, and can be easily broken by changing the optimizer, or by parameter-rescaling. By contrast, the $P(f)$ which is directly proportional to the Bayesian posterior $P_B(f|S)$ for functions that give zero error on the training set, is a much more robust predictor of generalization.

While the generalization performance of a DNN can be successfully predicted by the marginal likelihood PAC-Bayes bound [Valle-Pérez et al., 2018, Valle-Pérez and Louis, 2020], no such tight bound exists (to our knowledge) linking generalization and the Bayesian prior or posterior at the level of individual functions. Further theoretical work in this direction is needed.Moreover, it is natural to further extend current work towards linking flatness and the prior to other quantities which correlate with generalization such as frequency [Rahaman et al., 2018, Xu et al., 2019], or the sensitivity to changes in the inputs [Arpit et al., 2017, Novak et al., 2018a]. Improvements to the GP approximations we use are an important technical goal. $P(f)$ can be expensive to calculate, so finding reliable local approximations related to flatness may still be a worthy endeavour. Finally, our main result – that $\log P(f)$ correlates so well with generalization – still requires a proper theoretical underpinning, notwithstanding the bound in Eq.(4). Such explanations will need to include not just the networks and the algorithms, but also the data [Zdeborová, 2020]. We refer readers to Section B.1 for more discusion on related works.

Supplementary Material for Chapter 3

3.7 Comparing flatness metrics

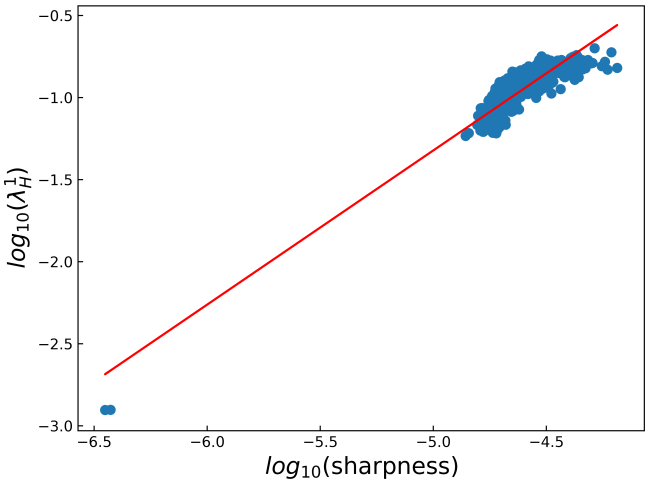


Figure 3.6: The direct correlation between sharpness and spectral norm of Hessian for the 1000 most frequently found functions found after SGD runs for a two hidden layer FCN, in the $\mathbf{n} = \mathbf{7}$ Boolean system (Same system as in Fig. 3.2) .

As mentioned in Section 3.2.2 of the main text, the sharpness metric in Definition 5 can be directly linked to spectral norm of the Hessian by considering the second order Taylor expansion of $L(\mathbf{w})$ around a critical point in powers of ζ [Dinh et al., 2017a]. We empirically confirm this relationship by showing in Fig. 3.6 the direct correlation between sharpness and spectral norm of Hessian, as well as in Fig. 3.7 the correlation between Hessian spectral norm and prior in Boolean system described in Section 3.5.1.

In addition to the spectral norm, another widely used flatness measure is the product of a subset of the positive Hessian eigenvalues, typically say the product of the top-50 largest eigenvalues [Wu et al., 2017, Zhang et al., 2018]. We measured the correlation of these Hessian-based flatness metrics with sharpness as well as

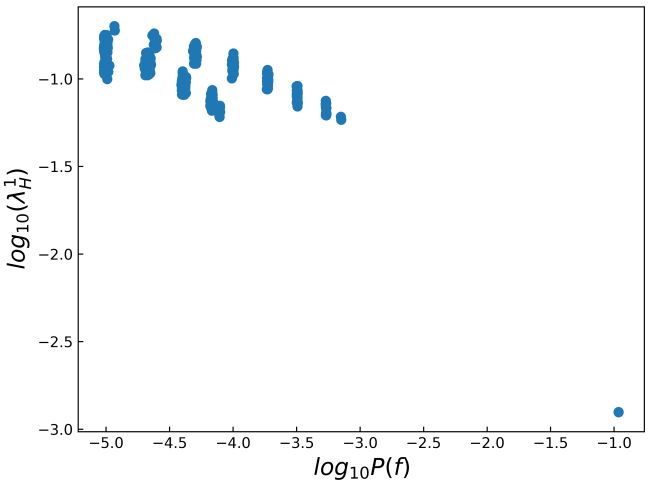


Figure 3.7: The correlation between prior and flatness in Boolean system where the flatness is measured by spectral norm of Hessian, for the 1000 most frequently occurring functions found by SGD runs with a two hidden layer FCN. The system is the same $\mathbf{n} = \mathbf{7}$ Boolean system as in Fig. 3.2 except that we use a different metric of flatness.

with generalization for the FCN/MNIST system in Fig. 3.8. Since they correlate well with the sharpness, these flatness measures show very similar correlations with generalization as sharpness does in Fig. 3.3 and Fig. 3.4. In other words, the Hessian-based flatness metrics also capture the loose correlation with generalization when the neural network is trained by SGD and the deterioration of this correlation when we change the optimizer to Adam.

Another detail worth noting is that Keskar et al. [2016] used the L-BFGS-B algorithm [Byrd et al., 1995] to perform the maximization of $L(\mathbf{w})$ in $\mathcal{C}_\zeta$, which is the box boundary around the minimum of interest:

$$\mathcal{C}_\zeta = \{\Delta\mathbf{w} \in \mathbb{R}^n : -\zeta\,(|w_i| + 1) \leq \Delta w_i \leq \zeta\,(|w_i| + 1) \quad \forall i \in \{1, 2, \cdots, n\}\} \tag{3.5}$$

However, as a quasi-Newton method, L-BFGS-B is not scalable when there are tens of millions of parameters in modern DNNs. To make Keskar-sharpness applicable for large DNNs (e.g. ResNet50), we use vanilla SGD for the maximization instead. The hyperparameters for the sharpness calculation are listed in Table 3.1. Note that the entries batch size, learning rate and number of epochs all refer to the SGD optimizer which does the maximization in the sharpness calculation process. The number of epochs is chosen such that the max value of loss function found at

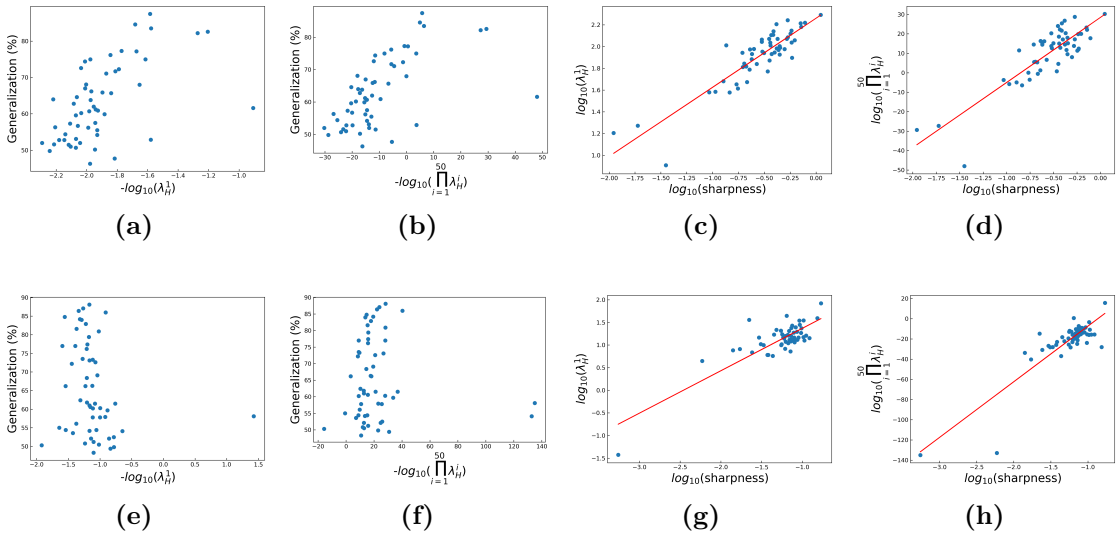


Figure 3.8: Two Hessian-based flatness metrics show analogous behavior to sharpness defined in (Definition 5). The architecture and dataset are FCN/MNIST, with training set size $|S| = 500$, and test set size $|E| = 1000$; which are the same settings as Fig. 3.3 (d) and Fig. 3.4 (e). **Optimizer: SGD** (a) - (b): The correlation between Hessian-based flatness metrics and generalization. (c) - (d): Sharpness and Hessian-based flatness metrics correlate well with one another. **Optimizer: Adam** (e) - (f): The correlation between Hessian-based flatness metrics and generalization breaks down, just as it does for sharpness in Fig. 3.4. (g) - (h): Sharpness and Hessian-based flatness metrics correlate well with one another, even though they don't correlate well with generalization.

Table 3.1: Hyperparameters for sharpness calculation

Data set	Architecture	Box size (ζ)	Batch size	Learning rate	Number of epochs
BOOLEAN	FCN	10^{-4}	16	10^{-3}	10
MNIST	FCN	10^{-4}	32	10^{-3}	100
CIFAR10	FCN	10^{-5}	128	5×10^{-5}	100
CIFAR10	ResNet50	10^{-5}	128	10^{0}	100

each maximization step converges. An example of the convergence of sharpness is shown in Fig. 3.9. As a check, we also compared our SGD-sharpness with the original L-BFGS-B-sharpness, finding similar results.

3.8 Flatness and prior correlation

In the main text, we showed the correlation of the Bayesian prior and of sharpness with generalization in Fig. 3.3 and Fig. 3.4. Here, in Fig. 3.10, we show the direct correlation of the prior and sharpness. As expected from the figures in the main text,

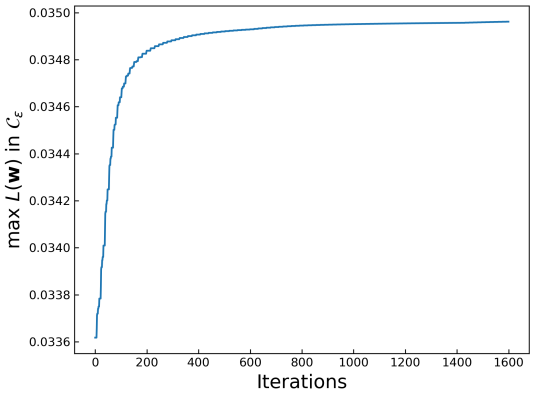


Figure 3.9: The max value of loss function $L(\mathbf{w})$ at each iteration in the process of maximization, when calculating the sharpness using SGD instead of L-BFGS-B. The plot shows the max loss value found by SGD in the box limit $\mathcal{C}_\zeta$ will converge after given number of epochs. For this plot the hyperparameters are listed in the second line of Table 3.1 (MNIST).

sharpness correlates with prior roughly as it does with generalization - i.e. reasonably for vanilla SGD but badly for entropy-SGD [Chaudhari et al., 2019] or Adam [Kingma and Ba, 2014]. We note that, as shown in Fig. 3.8, sharpness also correlates relatively well with the spectral norm of the Hessian and log product of its 50 largest eigenvalues for all the optimizers. So the correlation of flatness with prior/generalization does not depend much on which particular flatness measure is used.

Overall, it is perhaps unsurprising that a local measure such as flatness varies in how well it approximates the global prior. What is unexpected (at least to us) is that Adam and Entropy-SGD break the correlation for this data set. In Section 3.11.2, we show that this correlation also breaks down for other more complex optimizers, but, interestingly, not for full-batch SGD. Further empirical and theoretical work is needed to understand this phenomenon. For example, is the optimizer dependence of the correlation between flatness and prior a general property of the optimizer, or is it specific to certain architectures and datasets? One hint that these results may have complex dependencies on architecture and dataset comes from our observation that for ResNet50 on Cifar10, we see less difference between SGD and Adam than we see for the FCN on MNIST. More work is needed here.

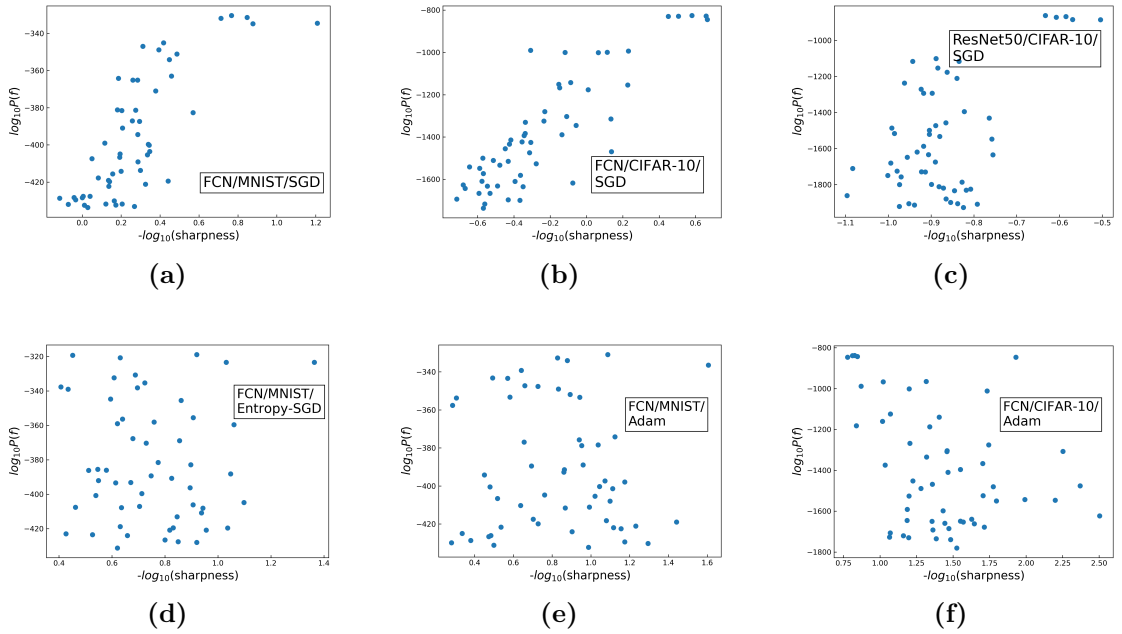


Figure 3.10: The direct correlation between prior $P(f)$ and sharpness over different datasets and optimizers. The correlation between prior and sharpness closely resembles the correlation between sharpness and generalization, mainly because prior and generalization are very closely correlated, as seen in our experiments (Fig. 3.3, Fig. 3.4).

3.9 Temporal behavior of sharpness

When using sharpness in Definition 5 as the metric of flatness, there are several caveats. First is the hyperparameters (see Table 3.1): the value of sharpness is only meaningful under specified hyperparameters, and in different experiments the sharpnesses are only comparable when the hyperparameters are the same. This renders sharpness less convenient to use (but still much more efficient than Hessian calculation). Second is the time evolving behavior of sharpness: For the classification problems we study, and for cross-entropy loss, it can continue to change even when the function (and hence generalization) is unchanged.

Before reaching zero training error, gradients can be large, and the behavior of sharpness (Definition 5) can be unstable under changes of box size ζ. This effect is likely the cause of some unusual fluctuations in the sharpness that can be observed in Fig. 3.5 and Fig. B.2 around epoch 100. In Fig. 3.11 we show that this artefact disappears for larger ζ. Similarly, when the gradients are big (typical in training),

sharpness may no longer link to spectral norm of Hessian very well.

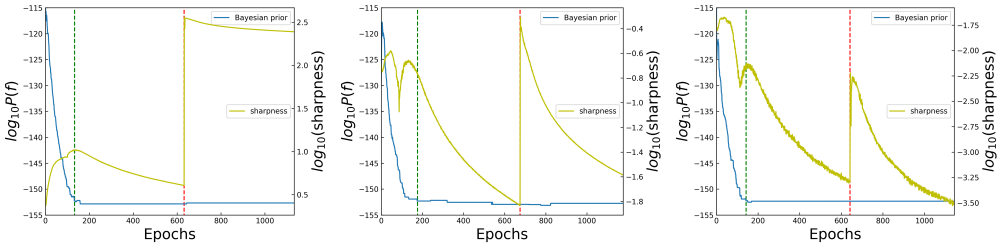


Figure 3.11: Different temporal behavior of sharpness, prior and accuracy when using different box size ζ. The dataset is MNIST with $|S| = 500$ and $|E| = 100$. The architecture is FCN. SGD optimizer is used. Scaling parameter $\alpha = 5.0$. Green and red dashed line denote reaching zero training error and alpha scaling, respectively. (a) $\zeta = 10^{-3}$, (b) $\zeta = 10^{-4}$, (c) $\zeta = 10^{-5}$. While there are quantitative differences between the values of ζ used, qualitatively we observe similar behaviour.

In Fig. 3.12, we first train the FCN to zero error, then "alpha scale" after 500 epochs, and then keep post-training for another 5000 epochs, much longer than in Fig. 3.5. The behaviour of sharpness and prior upon "alpha scaling" (not surprisingly) follows our discussion in Section 3.5.4. What is interesting to see here is that after enough overtraining, the effect of the alpha scaling spike appears to disappear, and the overall curve looks like a continuation of the curve prior to alpha scaling. What this suggests is that alpha-scaling brings the system to an area of parameter space that is somehow "unnatural". Again, this is a topic that deserves further investigation in the future.

Finally, we show the temporal behavior of a Hessian-based flatness measure in Fig. 3.13. Because of the large memory cost when calculating the Hessian, we use a smaller FCN on MNIST, with the first hidden layer having 10 units. We find that the Hessian based flatness exhibit similar temporal behavior to sharpness.

3.10 The correlation between generalization, prior, and sharpness upon overtraining

As shown in Fig. 3.5 of the main text, and further discussed in Section 3.9, flatness measures keep decreasing upon overtraining even when the function itself

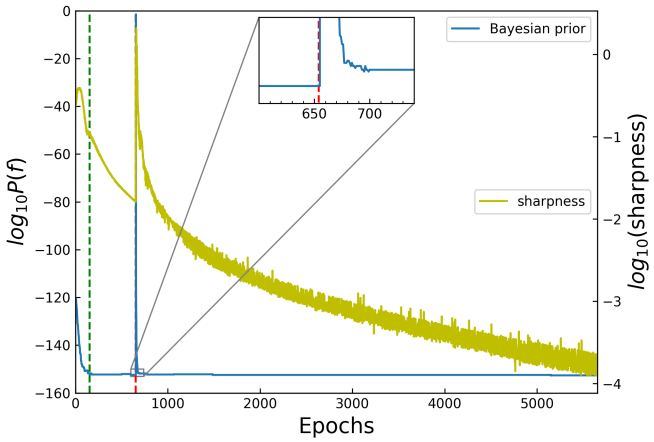


Figure 3.12: The temporal behavior of sharpness and prior after 5000 epochs of reaching zero training error. The dataset is MNIST with $|S| = 500$ and $|E| = 100$. The architecture is FCN. SGD optimizer is used. The magnitude of scaling $\alpha = 6.0$.

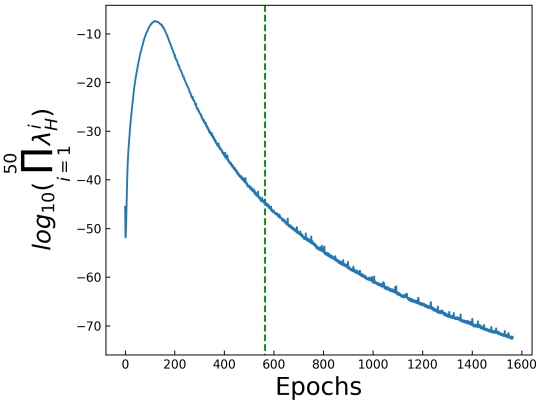


Figure 3.13: The temporal behavior of one Hessian based flatness metric. The dataset is MNIST with $|S| = 500$ and $|E| = 100$. The architecture is a smaller FCN (784-10-40-1), the optimizer is SGD. The green dashed line denotes the epoch where the system reaches zero training error. No alpha scaling is applied here. The Hessian based flatness metric shows similar temporal behaviour to the sharpness measure.

does not change. In this section, we revisit the correlation between prior, flatness and generalization at different numbers of overtraining epochs, i.e. *after* reaching zero training error.As can be seen in Fig. 3.14 to Fig. 3.19, overtraining does not meaningfully affect the correlation between sharpness, prior, and generalization we observed at the epoch where zero error is first reached in Fig. 3.3 and Fig. 3.4. When the optimizer is SGD, the flatness, no matter if it is measured by sharpness or Hessian based metrics, correlates well with prior and (hence) generalization across difference overtraining epochs; whereas when using Adam, the poor correlation

also persist in overtraining.

3.11 Further experiments

3.11.1 ResNet50 trained with Adam

When training ResNet50 on CIFAR-10, we use training set size $|S| = 5000$, attack set size $|A| = 5000$, test set size $|E| = 2000$. In each experiment, we mix the whole training set with different size of subset of attack set. The size of $|A|$ ranges as $(0, 500, 1000, 1500, ..., 5000)$. For each subset of attack set we sample 5 times. When training ResNet50 with Adam, we empirically found it is hard to train the neural net to zero training error with attack set size $|A| > 2500$. So we only show the results for those functions found with $|A| \leq 2500$. In Fig. 3.20 we show the results of correlation between sharpness and prior with generalization with limited data. The prior, as usual, correlates tightly with generalization, while the flatness-generalization correlation is much more scattered, although it is slightly better than the correlation seen for the FCN on MNIST, and closer to the behaviour we observed for SGD in the main text.

3.11.2 More SGD-variant optimizers

In Fig. 3.21 we provide further empirical results for the impact of choice of optimizer on the sharpness-generalization correlation by studying three common used SGD variants: Adagrad [Duchi et al., 2011], Momentum [Rumelhart et al., 1986] (momentum=0.9) and RMSProp [Tieleman and Hinton, 2012], as well as full batch gradient descent. Interestingly, full batch gradient descent (or simply gradient descent) shows behaviour that is quite similar to vanilla SGD. By contrast, for the other three optimizers, the correlation between sharpness and generalization breaks down, whereas the correlation between prior and generalization remains intact, much as was observed in the main text for Adam and Entropy-SGD.

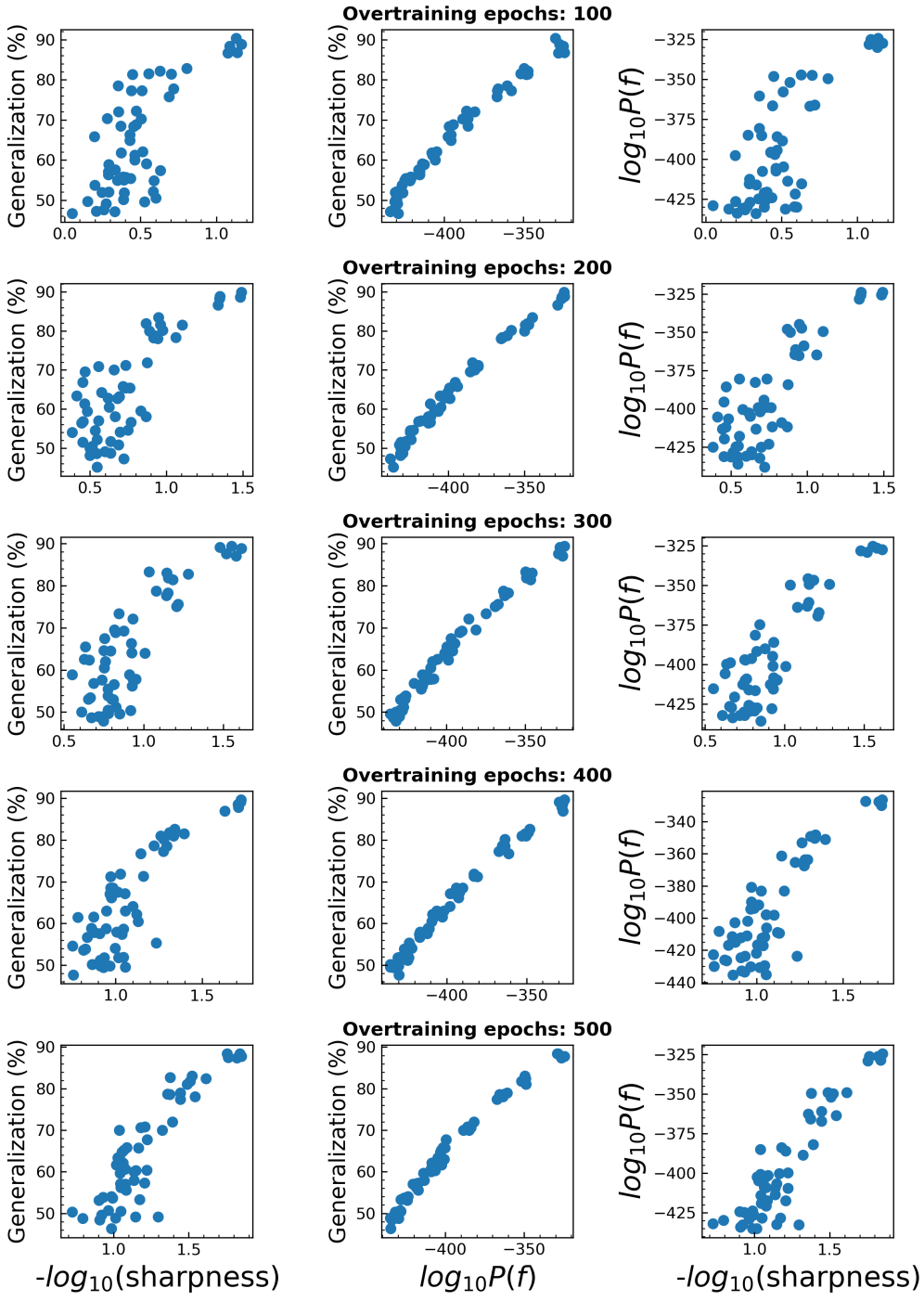


Figure 3.14: The correlation between sharpness, prior and generalization upon overtraining. The dataset is MNIST ($|S| = 500, |E| = 1000$), the optimizer is SGD. For the range of (100-500) overtraining epoch tested here, the overall values of sharpness drop with overtraining. By contrast, the priors remain largely the same. For each quantity, the correlations remain remarkably similar with overtraining.

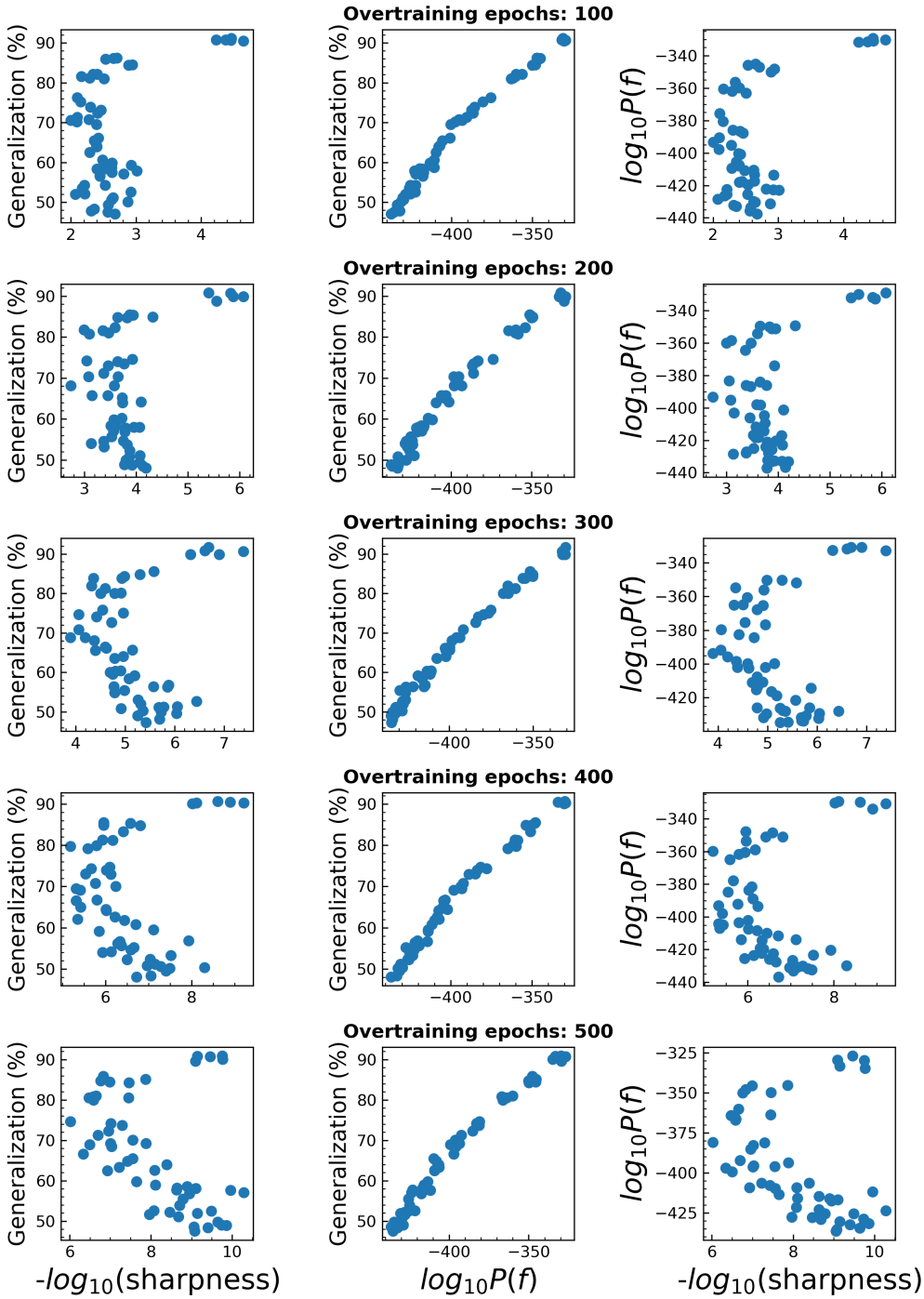


Figure 3.15: The correlation between sharpness, prior and generalization when overtrained (keep training after reaching zero training error). The dataset is MNIST ($|S| = 500, |E| = 1000$), the optimizer is Adam. The correlations are similar across different overtraining epochs.

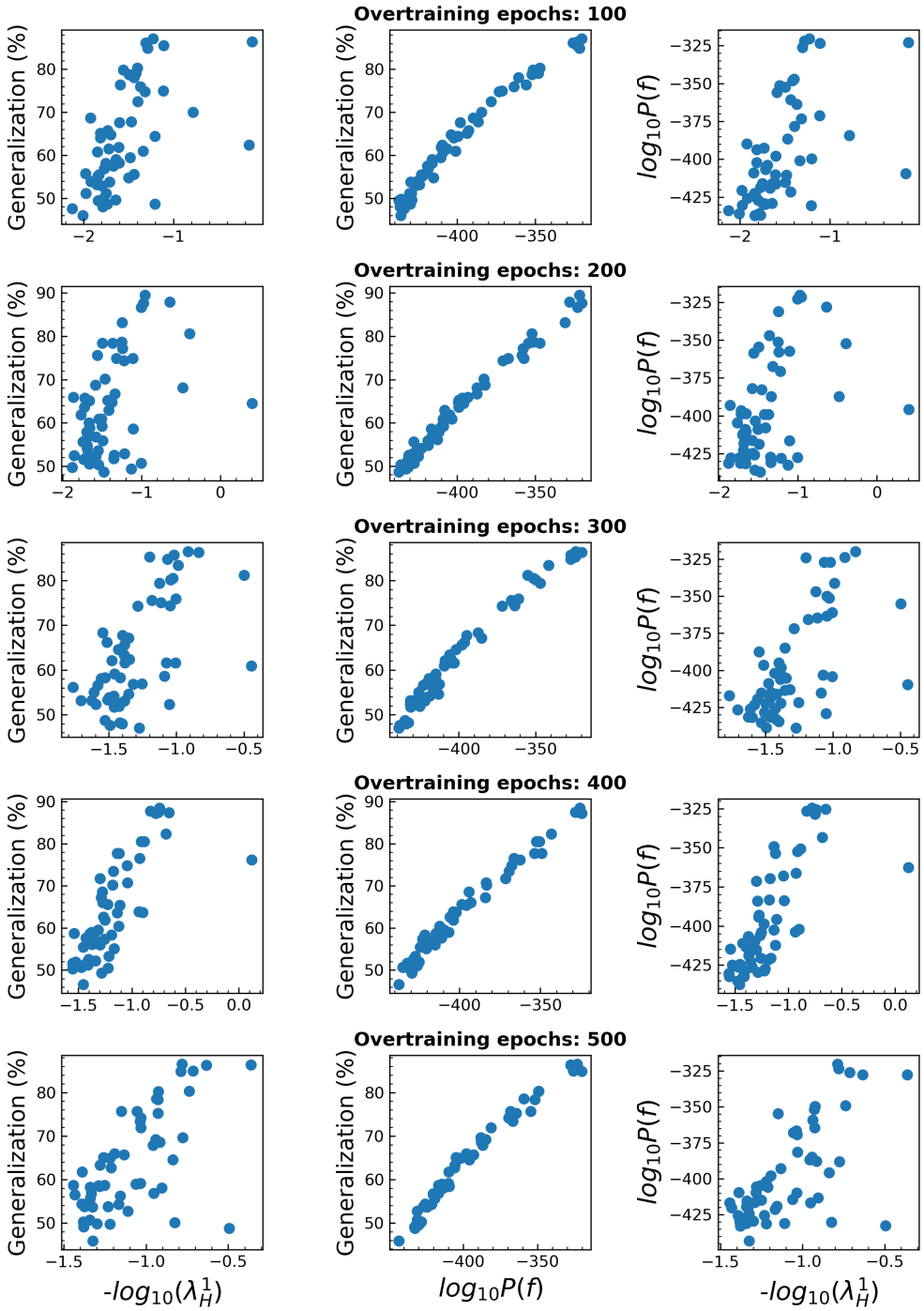


Figure 3.16: The correlation between Hessian spectral norm, prior and generalization when over-trained (keep training after reaching zero training error). The dataset is MNIST ($|S| = 500, |E| = 1000$), the optimizer is SGD. The correlations are similar across different overtraining epochs.

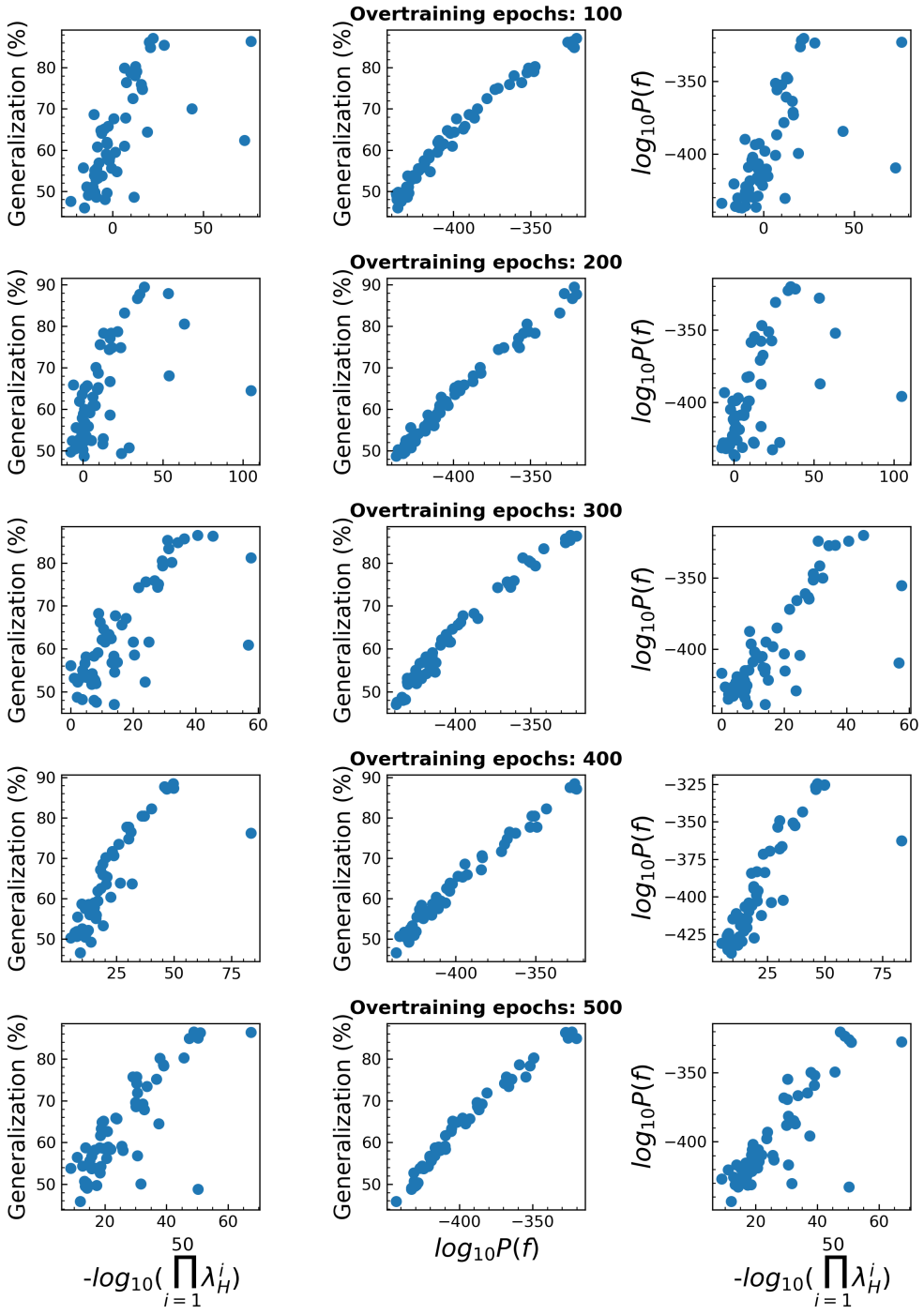


Figure 3.17: The correlation between Hessian based flatness (product of the top 50 largest Hessian eigenvalues), prior and generalization when over-trained (keep training after reaching zero training error). The dataset is MNIST ($|S| = 500, |E| = 1000$), the optimizer is SGD. The correlations are similar across different overtraining epochs.

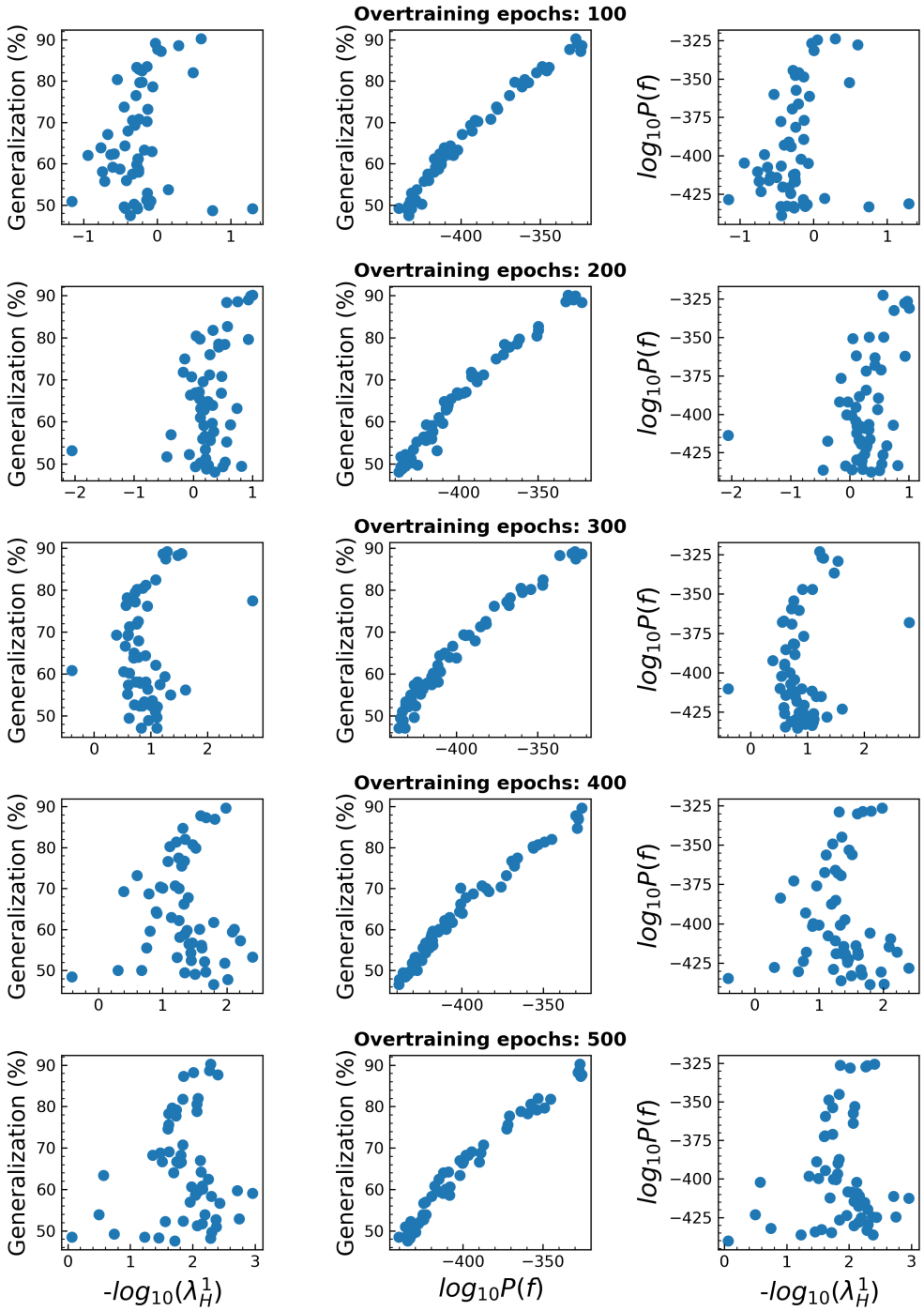


Figure 3.18: The correlation between Hessian spectral norm, prior and generalization when over-trained (keep training after reaching zero training error). The dataset is MNIST ($|S| = 500, |E| = 1000$), the optimizer is Adam. The correlations are similar across different overtraining epochs.

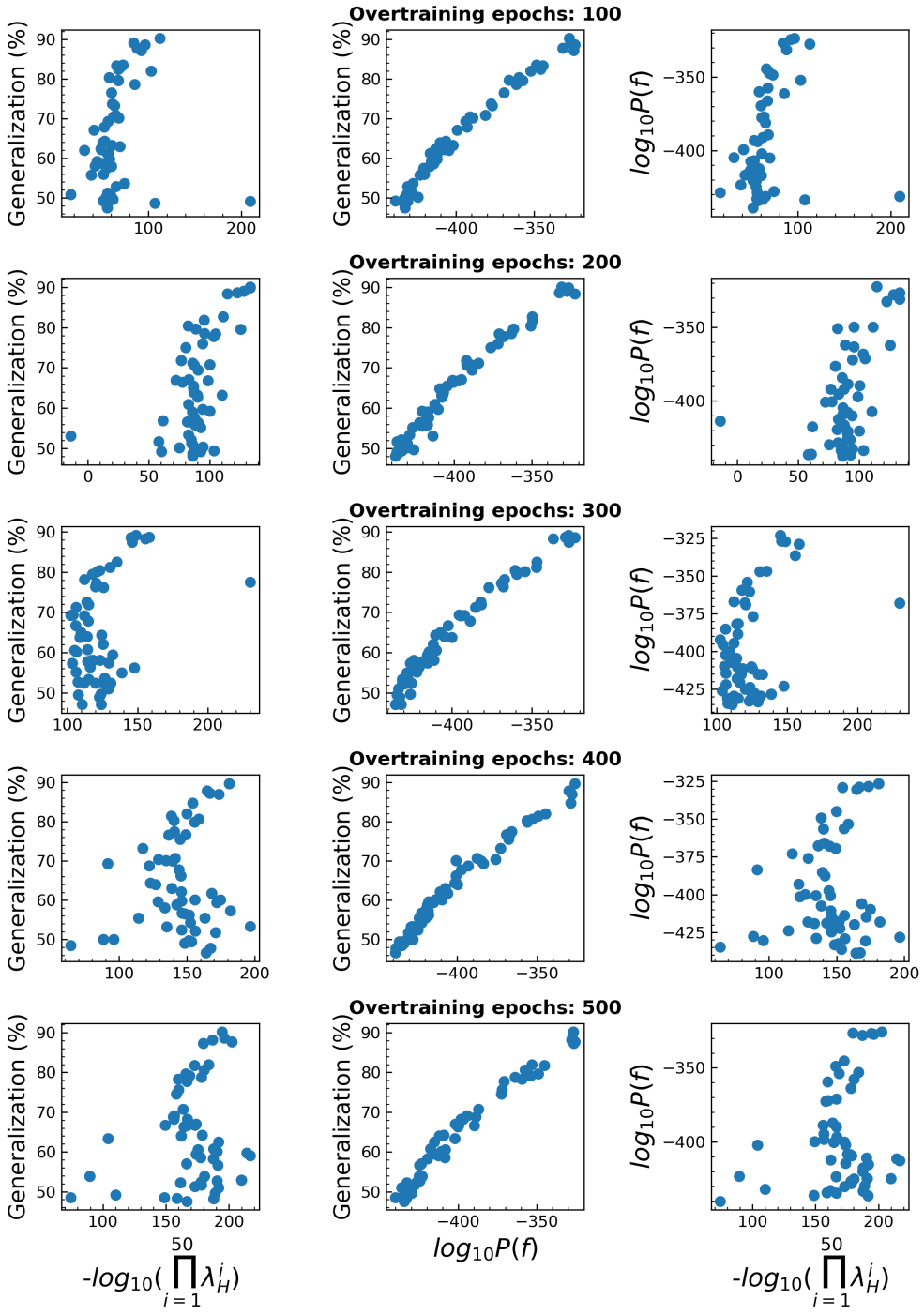
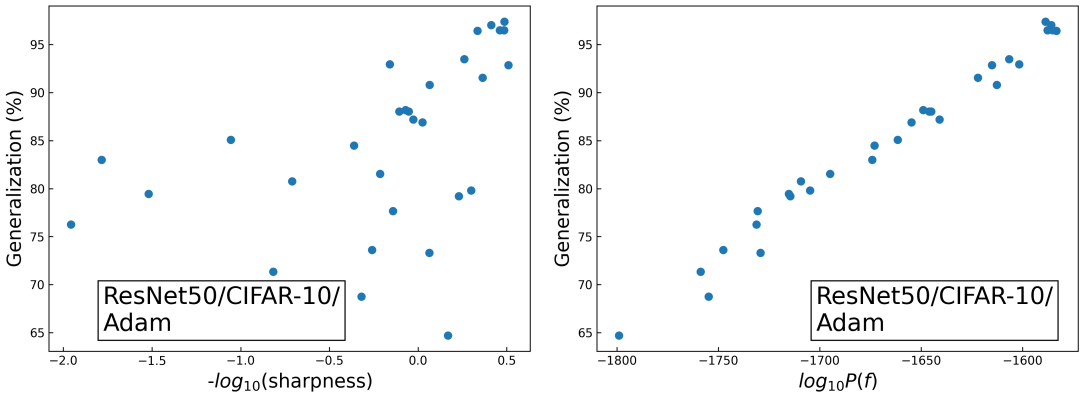


Figure 3.19: The correlation between Hessian based flatness (product of the top 50 largest Hessian eigenvalues), prior and generalization when over-trained (keep training after reaching zero training error). The dataset is MNIST ($|S| = 500, |E| = 1000$), the optimizer is Adam. The correlations are similar across different overtraining epochs.



(a) **(b)**

Figure 3.20: The correlation between generalization and (a) sharpness (b) prior for ResNet50 with $|S| = 5000$, $|E| = 2000$, and $|A|$ ranging from 0 to 2500, all on CIFAR-10.

3.11.3 Larger training set

In order to rule out any potential training size effect on our main argument of the flatness, prior and generalization relationship, we further performed the experiments on MNIST with 10k training examples. Larger training sets are hard because of the GP-EP calculation of the prior scales badly with size. The results are shown in Fig. 3.22. It is clear that the correlations between sharpness, prior and generalization follow the same pattern as we see in Fig. 3.3, in which there are only $|S| = 500, |E| = 1000$ images. If anything, the correlation with prior is tighter.

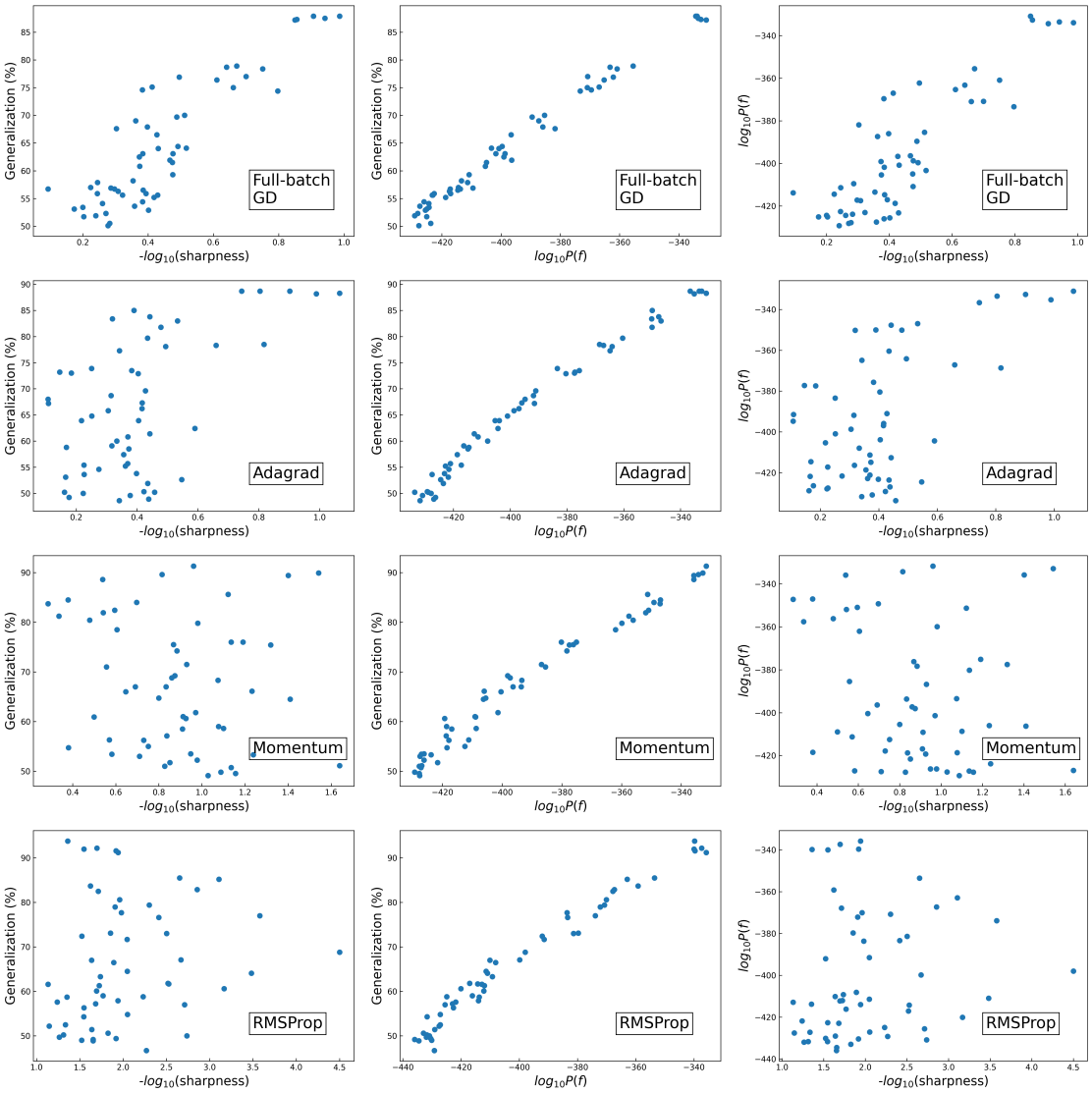


Figure 3.21: More results on the correlation between sharpness, prior and generalization when using other SGD-variant optimizers. The dataset is MNIST, $|S| = 500, |E| = 1000$. The architecture is FCN. The optimizers are full-batch gradient descent, Adagrad, Momentum (momentum=0.9) and RMSProp. All correlations are measured upon reaching zero training error.

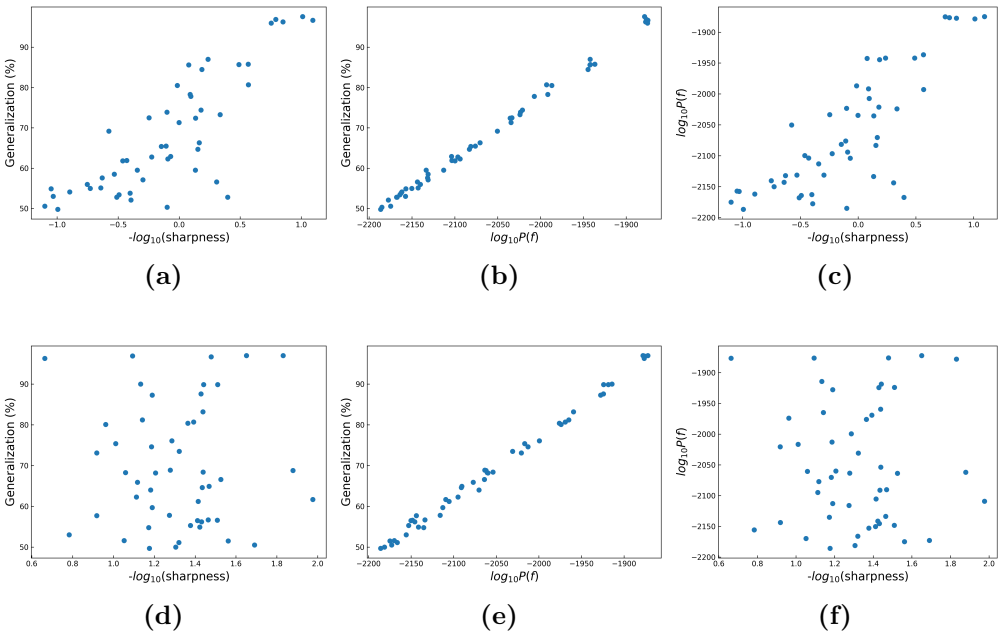


Figure 3.22: The correlation between sharpness, prior and generalization on MNIST with $|S| = 10000, |E| = 1000$. The attack set size ranges from 1000 to 9000. The architecture is FCN. (a)-(c): The FCN is trained with SGD; (d)-(f): The FCN is trained with Adam.

4

Position: Many Generalization measures for deep learning are fragile

Overview

Generalization measures have been widely applied to deep neural networks (DNNs). Although obtaining tight bounds remains difficult, these measures are often thought to at least reproduce qualitative trends. In this work, we show that most existing post-mortem generalization measures—those computed on trained networks—fail even this basic criterion. Specifically, small hyperparameter changes, such as minor learning rate adjustments or switching from SGD with momentum to Adam, can reverse qualitative trends (e.g., the slope of a learning curve) in widely used measures such as the path norm, even when the underlying DNN remains stable. We call such measures **fragile**: they exhibit qualitatively different behaviour across settings where the DNNs themselves are robust. We also identify subtler forms of fragility. For instance, while the PAC-Bayes origin measure is regarded as one of the most reliable, we show that it fails to capture differences in data complexity or scaling-laws with sample size for learning curves. In contrast, the function-based marginal-likelihood PAC-Bayes bound, which is by construction robust to hyperparameter changes, does capture differences in data-complexity and scaling behaviour in learning curves. Fragility may provide insights as well. We show this in

parameter-based post-mortem bounds by exploiting scale-invariance. Nevertheless, the general sources of these fragility failures need further investigation. In this light, we also suggest that function-based approaches may offer a more robust foundation for generalization measures. This position paper argues that many current generalization bounds are fragile, and developers of new measures should explicitly audit how they behave under hyperparameter and data variations.

4.1 Chapter introduction

Classical generalization theories—based on VC dimension, Rademacher complexity and related tools—have a long history in machine learning. They were developed with two intertwined aims: to furnish tight mathematical guarantees and to explain generalization. For deep networks the same aspiration has driven a profusion of bounds. While tight bounds have proven elusive, with some exceptions [Dziugaite and Roy, 2018, Pérez-Ortiz et al., 2020, Lotfi et al., 2022b], there has been an expectation that these measures can generate qualitative insight into the generalization behaviour of deep neural networks (DNNs). These bounds have typically focused on weights after training, which is why they are sometimes called post-mortem bounds. With these questions in mind, some large-scale comparisons have compared the behaviour of these measures, albeit often only at the sign level, e.g. does changing a setting produce the same a change in the underlying DNN and a generalization measure with the same sign [Jiang et al., 2019a, Dziugaite et al., 2020a].

A notable property of DNNs is that their generalization performance tends to be robust to modest changes in network size, hyperparameters, optimisers, or stopping criteria. While careful tuning can yield small performance gains, the overall effects are usually limited. This raises a natural question: are generalization measures themselves equally robust to such changes? If a generalization measure captures basic underlying factors that drive DNN performance, one would expect it to exhibit a similar degree of robustness.

The main finding of this paper is that many of the post-mortem bounds studied in the literature are *fragile*: small, ostensibly irrelevant training tweaks that barely affect the underlying DNN, can flip a measure's shape, trend, or scale. To test these fragilities systematically, we evaluate measures under controlled perturbations that hold data, architecture, and code fixed while nudging a single knob. Three stressors structure the audit in the main text: learning-curve behavior as sample size increases (§4.3); temporal behavior once the model has interpolated (§4.4); and responses to data complexity via label noise and dataset swaps (§4.5). In the supplementary materials we show many more tests of fragility. This pattern suggests that many measures miss something essential and should always be audited by the kinds of tests that we suggest in this paper. If only a limited set of hyperparameters is tested, or if evaluations only check for sign errors, one might incorrectly conclude that the measure captures essential behaviour of DNN generalization better than it actually does.

In contrast to the post-mortem bounds, we show that a function-based marginal-likelihood PAC-Bayes bound successfully tracks dataset complexity and learning-curve scaling [Valle-Pérez and Louis, 2020]. Because this pre-mortem bound operates in function space and does not depend explicitly on the network's weights or on training dynamics, it is inherently insensitive to the choice of optimiser or most hyperparameters. This insensitivity is both a strength, making the bound robust rather than fragile, and a limitation, as it prevents the bound from reflecting how training procedures influence generalization. Nonetheless, the strong performance of this relatively simple approach suggests that developing more sophisticated function-space generalization measures could be a fruitful direction for future research.

Finally, we ask whether we can learn something by investigating these fragility based failures further. We show that some of the qualitative failures of norm-based bounds appear to track a recent prediction for the norms in linear regression problems Zhang and Louis [2025]. We also exploit a symmetry in scale-invariant networks and prove a non-asymptotic equivalence between fixed learning-rate with fixed weight decay and a matched run with exponentially increasing learning rate

and time-varying weight decay; the two training procedures compute the same predictor at every iterate (§4.6; cf. Li and Arora, 2019). Under this invariance lever, magnitude-sensitive post-mortem measures can inflate by orders of magnitude while test error remains flat. This suggests that generalization measures should carefully track key invariances present in the underlying DNNs.

Our contribution:

- We formulate a compact fragility audit for generalization measures that targets training-hyperparameter stability, post-interpolation temporal behavior, and data-complexity response.
- We provide systematic evidence that popular post-mortem measures—path, spectral and Frobenius norms, flatness proxies, and deterministic PAC–Bayes surrogates—change their qualitative story under mild optimizer or step-size tweaks even when the underlying DNN accuracy is stable.
- We prove an equivalence for scale-invariant networks that matches fixed schedules to exponentially increasing learning-rate schedules with time-varying weight decay, yielding a controlled invariance lever that isolates function from parameter scale and exposes magnitude sensitivity.
- We offer a positive baseline: a function-space marginal-likelihood PAC–Bayes predictor at the GP limit that passes the same stress tests, and we distill guidelines for reporting and designing more robust diagnostics.

4.2 Related work

In this chapter we use *generalization bound* and *generalization measure* interchangeably: both seek to predict out-of-sample performance, differing mainly in whether they arrive with formal guarantees. In modern deep networks these quantities are best read as *measures*—diagnostics to compare across training/data regimes—rather than practically tight certificates.

From classical capacity to modern practice. Early theory framed capacity via VC and Rademacher analyses [Mendelson, 2002, Anthony and Bartlett, 2002, Bartlett and Mendelson, 2001, Koltchinskii, 2001]. As overparameterization became the norm, evidence accumulated that uniform-convergence explanations often miss the mark: networks can fit random labels yet generalize on natural data [Zhang et al., 2016b], some bounds fail to track risk [Nagarajan and Kolter, 2019], and benign overfitting can arise even at interpolation [Bartlett et al., 2020]. The field responded with diagnostics that foreground invariances, algorithmic dependence, and data interactions.

4.2.1 Capacity-oriented diagnostics: norms, margins, distance from initialization

The first work in this subfield translated classical notions into deep settings. Spectrally-normalized margin bounds tied test error to Lipschitz-like control and margins, making scale explicit [Bartlett et al., 2017b]. Path- and norm-based views clarified how depth and weight scales shape effective complexity [Neyshabur et al., 2015c, 2019b]. Empirically, modeling the *distribution* of margins—not just the minimum—improves predictiveness across trained families [Jiang et al., 2018, Novak et al., 2018a], and sample complexity can depend on norms rather than width [Golowich et al., 2018b]. In practice, enforcing Lipschitz continuity often improves out-of-sample performance [Gouk et al., 2020, Yoshida and Miyato, 2017]. Distance-from-initialization and movement-from-pretraining offer reference-aware, width-robust complexity surrogates with both empirical support and bounds [Li et al., 2018b, Zhou et al., 2021, Neyshabur et al., 2017b], while optimization dynamics link large margins to implicit bias in separable regimes [Soudry et al., 2018b].

4.2.2 Geometry-oriented diagnostics: flatness and sharpness

A parallel line approached generalization through loss-landscape geometry. The "flat minima generalize better" intuition predates deep learning [Hochreiter and Schmidhuber, 1997b] and resurfaced when large-batch training was observed to

converge to sharper minima with worse test error [Keskar et al., 2017]. Because raw sharpness is parameterization-dependent, normalized and magnitude-aware definitions were proposed and shown to correlate more robustly with test error [Dinh et al., 2017b, Tsuzuku et al., 2020, Kim et al., 2022, Jang et al., 2022]. These ideas also shaped algorithms: SAM explicitly optimizes a local worst-case neighborhood and typically improves accuracy; adaptive variants and ablations probe when and why it helps [Foret et al., 2020b, Kwon et al., 2021, Andriushchenko and Flammarion, 2022]. Independent probes—weight averaging, landscape visualizations, and training-dynamics analyses—add evidence that broader valleys often accompany better generalization [Izmailov et al., 2018b, Li et al., 2017, Cohen et al., 2021]. Stochastic optimization theory offers a mechanism: noise scale and heavy-tailed gradient noise can bias learning toward flatter regions [Smith et al., 2017b, Jastrzebski et al., 2017b, Simsekli et al., 2019, McCandlish et al., 2018].

4.2.3 Algorithm-aware certificates: PAC-Bayes as operational measures

Partly in response to limits of uniform convergence, PAC-Bayesian analysis made algorithm dependence explicit by trading empirical risk against a posterior–prior divergence [McAllester, 1999, Seeger, 2002, Catoni, 2007, Langford and Shawe-Taylor, 2002]. The framework became operational when non-vacuous deep-network certificates were obtained by optimizing stochastic posteriors [Dziugaite and Roy, 2017b], or simply working with the GP prior [Valle-Pérez et al., 2018, Zhang et al., 2021b]. Subsequent work tightened certificates with data-aware priors [Dziugaite and Roy, 2018, Pérez-Ortiz et al., 2020], and with compression-flavored posteriors that reduce effective description length [Arora et al., 2018b, Lotfi et al., 2022b]. Treating PAC-Bayes itself as a training objective yields models that are both accurate and tightly certified, while extensions broaden the scope to adversarial risk and fast/mixed-rate regimes [Rodríguez-Gálvez et al., 2024].

Other perspectives. Complementary lenses provide boundary conditions any credible measure should respect. Algorithmic stability ties generalization to insensitivity under data perturbations [Bousquet and Elisseeff, 2002, Hardt et al., 2016]; information-theoretic analyses bound excess risk by the information a learning rule extracts from the sample [Russo and Zou, 2016, Xu and Raginsky, 2017]. Description-length and compression viewpoints link compressibility to generalization [Blier and Ollivier, 2018, Taiji Suzuki, 2020]. Linearized regimes (NTK/GP) delineate when very wide nets behave like their kernel limits [Jacot et al., 2018, Lee et al., 2020b]. Finally, double-descent phenomena and scaling laws offer external checks on diagnostics [Belkin et al., 2019, Kaplan et al., 2020].

Meta-evaluations and synthesis. Large-scale comparisons underscore that no single candidate explains generalization under all interventions [Jiang et al., 2019b, Dziugaite et al., 2020b], motivating a measures-as-diagnostics mindset. A recent critique shifts the lens from *predictiveness* to *tightness*, showing that uniformly tight bounds are out of reach in overparameterized settings and clarifying what such tightness could mean [Gastpar et al., 2023]. Our focus is complementary: rather than tightness per se, we study *fragility*—how otherwise informative measures can fail under innocuous hyperparameter changes that leave underlying DNN performance essentially unchanged.

4.3 Training-hyperparameter fragility

The path norm (definition in App. 4.10) is often regarded as the strongest norm-based proxy for modern ReLU networks: it respects layerwise rescalings, composes cleanly with depth, and has supported some of the sharpest norm-style capacity statements and practical diagnostics [Neyshabur et al., 2015c, 2017c, Gonon et al., 2023]. Precisely because it is a strong candidate, it is a natural place to look for fragility.

We keep the data, model, and pipeline fixed and vary only innocuous training choices. In Fig. 4.1 we show a triptych for the *same* ResNet-50 on *FashionMNIST*, nudging exactly one hyperparameter at a time—either the optimizer or the learning

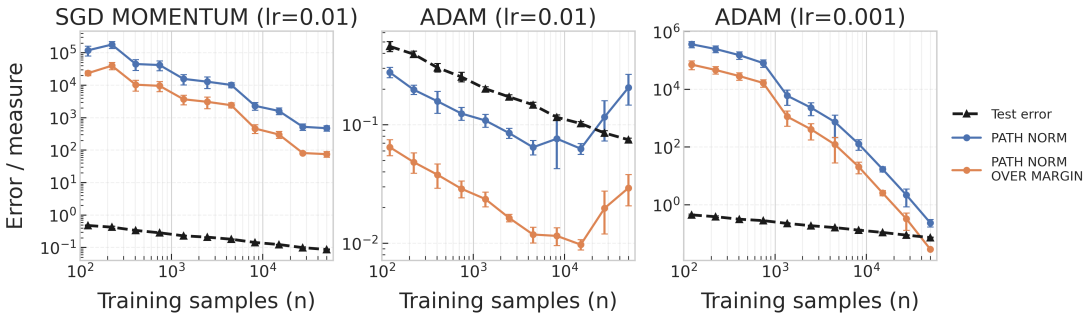


Figure 4.1: Path-norm learning curves under small training-pipeline nudges (no early stopping). *Left:* SGD with momentum, LR = 0.01. *Middle:* Adam, LR = 0.01. *Right:* Adam, LR = 0.001. Curves show test error (black), path norm (blue), and path norm over margin (orange); axes are logarithmic and error bars denote seed variability. Training set sizes n range from 10^2 to the full FashionMNIST training set (5×10^4 samples).

rate (no early stopping). Each curve shows test error (black), the raw path norm (blue), and the path norm divided by the empirical margin (orange); axes are logarithmic and error bars reflect seed variability.

Across the three panels we are looking at the same network/dataset, while nudging just one hyperparameter and watching the path-norm metrics (blue/orange) alongside test error (black). In the **left** plot, we run SGD with momentum at LR = 0.01: both path-norm curves start enormous ($\sim 10^5$) and slide down roughly log-linearly as we add data, with the margin-normalized curve tracking about a decade below; test error steadily drops without surprises. Slide the optimizer to Adam but keep LR = 0.01 (**middle** panel) and the picture flips qualitatively—path norms now live near 10^{-1}, fall smoothly at small n, then rebound once we hit $\sim 2 \times 10^4$ examples, while test error keeps its gentle downward drift. Change just one more innocuous knob—reduce Adam's learning rate to 0.001 (**right** panel)—and the metrics jump back up to $\sim 10^5$–10^6 yet collapse by four orders of magnitude as data grow; the margin-scaled path norm closely shadows the raw path norm, and the test error retraces the familiar slow decline. These starkly different trajectories arise purely from modest hyperparameter tweaks.

This contrast mirrors a simple analytic case in linear regression: fixing *which* norm you measure does not fix its learning-curve scaling, because the solution

selected by optimization—through implicit (or explicit) bias, e.g., favoring different ℓ_p-minimizers—can induce different sample-size scalings of that same norm [Zhang and Louis, 2025]. Our reading is that optimizer and learning-rate choices analogously shift the deep model's implicit bias, toggling the path-norm scaling between monotone and U-shaped. Changing only the stopping criterion did not yield a significant qualitative difference in these experiments, so we focus on the learning-rate and optimizer perturbations. See App. 4.10 for precise definitions and further experiments, including the same stress test across Frobenius, margin, spectral, PAC-Bayes, and VC-style proxies.

4.4 Temporal behavior fragility

Temporal traces ask a simple question that post-mortem snapshots cannot: once the classifier stops changing, does the measure stop moving? Let T_{int} denote the first epoch at which training accuracy reaches 100%. A stable diagnostic should largely settle for $t > T_{\text{int}}$; continued motion there indicates optimizer-driven drift rather than functional change. We examine this regime on a logarithmic epoch axis for readability.

A paired experiment on FashionMNIST makes the point. We fix the architecture (ResNet-50), learning rate (0.01), data, and stopping rule, and change only the optimizer. With Adam (left panel of Figure 4.2), a representative weight-norm measure—the path norm—climbs from ≈ 1 to $> 10^2$, and the ratio-over-margin rises in step. Interpolation occurs late ($T_{\text{int}} \approx 114$), yet both traces keep increasing beyond that point. With SGD with momentum (right), interpolation arrives much earlier ($T_{\text{int}} \approx 27$); the path norm sits near 3×10^2 and then slowly declines, and the ratio-over-margin trends downward. In both runs, the generalization curve remains on a comparable scale. The optimizer swap thus toggles the post-interpolation regime from monotone growth to stabilization/decay without a commensurate change in test error.

A plausible mechanism underlies this split. After interpolation under cross-entropy, gradients can continue to increase logit scale along near-flat directions. Adam tends

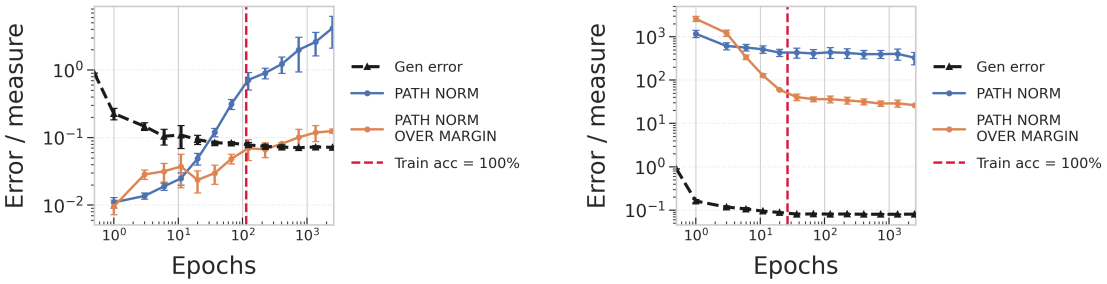


Figure 4.2: ResNet-50 on FashionMNIST at learning rate 0.01 with identical stopping; epoch axis is logarithmic. Left (ADAM): path norm grows from ≈ 1 to $> 10^2$ and continues past the 100% train-accuracy point (≈ 114 epochs); the ratio-over-margin rises accordingly. Right (SGD+momentum): path norm stays near 3×10^2 then declines after an earlier 100% crossing (≈ 27 epochs); the ratio trends downward; generalization is similar in both.

to amplify this scale drift, while SGD+momentum tempers it, yielding opposite signs for the post-interpolation slope. The lesson is that weight–norm–based measures—and margin-normalized variants that co-move with them—are not monotone indicators of optimization progress: an upward trend can be optimizer-driven scale inflation; a flat or gently declining trend need not signal stagnation.

For practice, report T_{int} and the *post-interpolation slope* $s := \frac{d \log \text{Measure}}{d \log t}\Big|_{t>T_{\text{int}}}$ alongside any temporal plot; overlay optimizers on the same axes; and compare snapshots at matched milestones (e.g., the first 100%-accuracy crossing) or with early-stopped checkpoints. A useful stress test is *hysteresis*: resume a checkpoint taken just after T_{int} with a different optimizer and check whether the measure's drift changes sign while test error holds steady. These checks make temporal fragility visible and keep optimizer-induced motion from being mistaken for learning.

4.5 Data-Complexity Fragility: Label Noise & Dataset Difficulty

What should a good "generalization measure" do as the data become harder? As we randomize a fraction of labels, the task grows noisier; as we move from MNIST to FASHIONMNIST to CIFAR10, the natural difficulty increases. A faithful measure ought to move in a predictable way (typically one–directional, with broadly similar shape and scale across reasonable training setups). Figure 4.3 juxtaposes

two post-mortem diagnostics (path norm and path norm over margin) with a function-space predictor (the marginal-likelihood PAC–Bayes bound). Merely changing *Adam*'s learning rate flips the qualitative trend of the path-norm curves (left vs. middle), whereas the marginal-likelihood bound (right) remains stable in its response to label corruption. We observe a similar path-norm effect under SGD with momentum; see Appendix C.3, Fig. C.8. This is precisely the kind of fragility our framework urges authors to reveal when treating bounds as *measures* rather than guarantees.

4.5.1 Path norms: the same setup, different stories

In Figure 4.3 (left and middle), the *only* change is ADAM's learning rate. At LR=10^{-3} the path norm begins *towering*, *collapses* by roughly two orders of magnitude as soon as we inject noise, and then *levels off*; the margin-normalized variant echoes this shape. At LR=10^{-2}, both curves instead sit *low* and *climb* gently with corruption. Same optimizer, same model, same dataset—yet the story told by the post-mortem norms flips with a ten-fold LR change.

By contrast, the **right** panel plots the marginal-likelihood PAC–Bayes (ML-PACBayes) bound (purple; definition and discussion in §4.7). Unlike post-mortem norms, ML-PACBayes is a *function-space* quantity: it depends on the architecture's prior over functions and the data, not on the particular path taken through parameter space. Consistent with this invariance, it increases smoothly with label corruption and is *agnostic* to the LR change that derails the norm-based diagnostics. (See §4.7 for why this invariance is expected and how it extends to dataset difficulty.)

Remark. Several PAC–Bayes *parameter-space* surrogates are often *insensitive* to label corruption in standard setups—another kind of fragility. Here we emphasize that a function-space view (ML-PACBayes) moves correctly with data difficulty and avoids entanglement with training hyperparameters; see §4.7 and Appendix C.3.

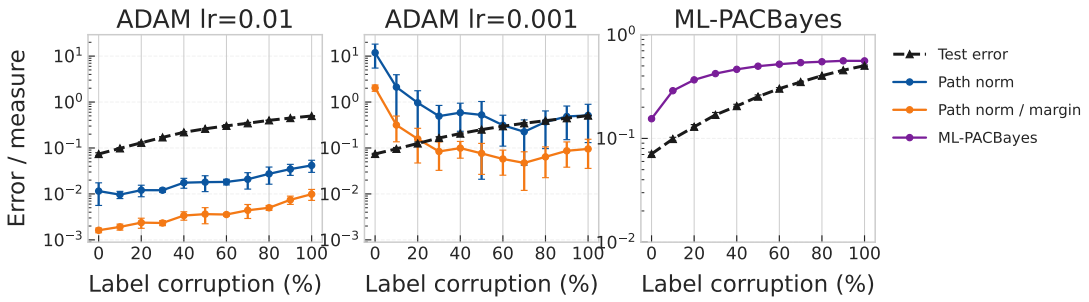


Figure 4.3: Label-corruption sweeps with post-mortem norms vs. function-space ML-PACBayes on RESNET-50. *Left:* ADAM, LR=0.01. *Middle:* ADAM, LR=0.001. *Right:* marginal-likelihood PAC–Bayes (ML-PACBayes; purple). Dashed black: test error; blue: path norm; orange: path norm over margin. A ten-fold LR change reverses the path-norm trend (left vs. middle), while ML-PACBayes remains a smooth, monotone function of corruption (right) and is, by construction, agnostic to optimizer/LR. *All panels use 10,000 training samples.*

4.5.2 Fix architecture, vary dataset: a subtler PAC–Bayes fragility

The literature often treats `PACBAYES_ORIG` as among the most informative PAC–Bayes proxies. Here we hold the *architecture and hyperparameters fixed* and vary the dataset—overlaying MNIST, FashionMNIST, and CIFAR10. As expected, the dashed generalization–error curves separate cleanly and their *slopes* differ with dataset difficulty (MNIST steepest decline, CIFAR-10 flattest). However, the first two panels of Figure 4.4 show that, *besides* giving almost identical *numerical values* for different datasets, `PACBAYES_ORIG` also *fails to pick up the correct dataset difficulty*: its learning-curve slopes are nearly indistinguishable across datasets and do not echo the test-error slopes. By contrast, the **third panel** (ML-PACBayes) is both *tight* (close to the corresponding error curves) and *data-aware*: it preserves the MNIST < FashionMNIST < CIFAR-10 ordering and reflects the different slopes. See §4.7 for the bound's definition and a fuller discussion of why this function-space quantity captures dataset difficulty while parameter-space surrogates often do not.

Beyond label corruption and dataset swaps, we also probe symmetry-preserving vs. signal-destroying data transforms via pixel permutations; the function-space predictor tracks the expected invariances while several post-mortem diagnostics do not (App. C.4).

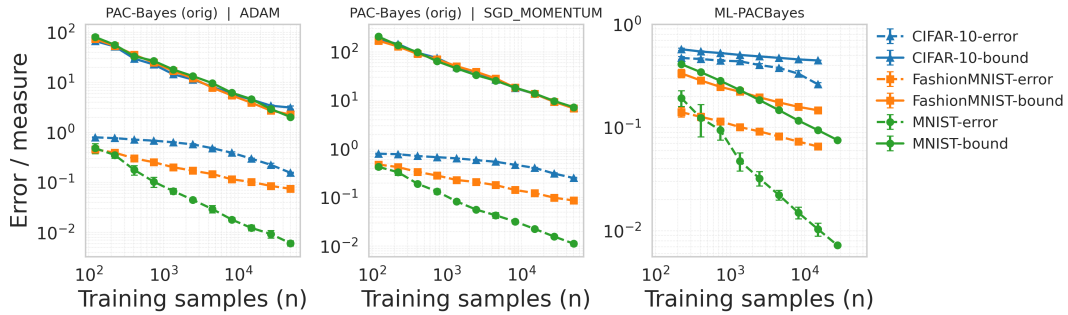


Figure 4.4: Fixed architecture, varying dataset.. The model is RESNET-50. *Left:* `PACBAYES_ORIG` with ADAM. *Middle:* `PACBAYES_ORIG` with SGD (momentum). In both, the `PACBAYES_ORIG` curves take almost identical values across MNIST, FASHIONMNIST, and CIFAR10 *and* show nearly the same slope, missing the data-complexity ordering visible in test-error slopes. *Right:* ML-PACBayes (marginal-likelihood PAC–Bayes) is tight and preserves the correct dataset ordering and slope; this panel is computed on the *binarized* versions of the three datasets.

4.6 Scale-invariant network and exponential learning rate schedule

Normalization layers make many modern networks effectively scale-invariant, and this symmetry underlies the "exponentially increasing learning-rate" idea of Li and Arora [2019]: in such models, training with fixed learning rate (LR), weight decay (WD), and momentum can be matched—in function space—by a run with an exponentially increasing LR and *no* WD (plus a small momentum correction across phases). Their analysis shows how LR and WD can be "folded into" an exponential schedule and documents successful experiments under this equivalence. We refer to this multiplicative control parameter α as the *Exp++ factor*.

We use the same symmetry but a different equivalence. Instead of removing WD, our theorem pairs an exponentially *increasing* LR with a *time-varying* WD so that every iterate computes the same predictor as a fixed-LR/fixed-WD baseline. The mapping is explicit and non-asymptotic: it gives closed-form schedules for $\tilde{\eta}_t$ and $\tilde{\lambda}_t$, and an admissible range for the multiplicative factor α via the interval $\mathcal{I}$. Conceptually, this creates a single "Exp++" knob that can drive large changes in parameter norms while leaving the learned function essentially unchanged—exactly

the benign intervention we later use to probe the fragility of magnitude-sensitive measures. Empirical details and results are deferred to Appendix 4.11.

Definition 8 (Scale invariant neural networks)**.** *Consider a parameterized neural network $f(\boldsymbol{\theta})$. We say f is scale invariant if*

$$\forall c \in \mathbb{R}^+, f(\boldsymbol{\theta}) = f(c\boldsymbol{\theta}) \tag{4.1}$$

Theorem 10 (Equivalence of schedules in scale-invariant nets)**.** (GD+WD+LR fixed $\Leftrightarrow$ GD+WD$\searrow$ + LR$\nearrow$). *Let $f(\theta_t)$ be scale-invariant and let training use SGD with momentum γ. Introduce the shorthands*

$$\Delta_\lambda := (1-\gamma)^2 - 2(1+\gamma)\lambda\eta_0 + (\lambda\eta_0)^2, \tag{4.2}$$

$$\Xi(\alpha) := \frac{\alpha^2 - \alpha(1-\lambda\eta_0+\gamma) + \gamma}{\eta_0}. \tag{4.3}$$

Define the interval endpoints

$$\alpha_L := \frac{\gamma}{1-\lambda\eta_0+\gamma}, \tag{4.4}$$

$$\alpha_- := \frac{1+\gamma-\lambda\eta_0-\sqrt{\Delta_\lambda}}{2}, \tag{4.5}$$

$$\alpha_+ := \frac{1+\gamma-\lambda\eta_0+\sqrt{\Delta_\lambda}}{2}, \tag{4.6}$$

and set

$$\mathcal{I} := (\alpha_L,\ \alpha_-]\ \cup\ [\alpha_+,\ 1). \tag{4.7}$$

Consider the two updates (with $\theta_{-1} = \theta_0$, $\tilde{\theta}_0 = \theta_0$, $\tilde{\theta}_{-1} = \alpha\,\theta_{-1}$):

$$\frac{\theta_t - \theta_{t-1}}{\eta_0} = \frac{\gamma(\theta_{t-1}-\theta_{t-2})}{\eta_0} - \nabla_\theta\Big(L(\theta_{t-1}) + \tfrac{\lambda}{2}\|\theta_{t-1}\|_2^2\Big), \tag{A}$$

$$\frac{\tilde{\theta}_t - \tilde{\theta}_{t-1}}{\tilde{\eta}_t} = \frac{\gamma(\tilde{\theta}_{t-1}-\tilde{\theta}_{t-2})}{\tilde{\eta}_{t-1}} - \nabla_\theta\Big(L(\tilde{\theta}_{t-1}) + \tfrac{\tilde{\lambda}_t}{2}\|\tilde{\theta}_{t-1}\|_2^2\Big). \tag{B}$$

If $\alpha \in \mathcal{I}$, then the schedules

$$\tilde{\eta}_t = \eta_0\,\alpha^{-2t-1}, \tag{4.8}$$

$$\tilde{\lambda}_t = \Xi(\alpha)\,\alpha^{2t-1} + \frac{\gamma(1-\alpha)}{\eta_0\,\alpha}\,\mathbb{1}\{t=0\} \tag{4.9}$$

ensure $\tilde{\theta}_t = \alpha^{-t}\theta_t$ for all $t \geq 0$; hence (A) and (B) generate identical functions.

Remark (Parameter range and nonnegativity). Assume

$$\lambda\eta_0 \le (1-\sqrt{\gamma})^2, \tag{4.10}$$

$$\alpha_L < \alpha_- \quad \text{(equivalently, } \Delta_\lambda \ge 0\text{)}. \tag{4.11}$$

Then the interval $\mathcal{I}$ above is nonempty and the square-root term is real; both conditions hold for common hyperparameters in practice.

Consequences. The proof of Theorem 10 is in Appendix A.1. In the scale-invariant setting, this equivalence lets us pump parameter norms without changing $f(\theta)$, a lever we use to stress-test generalization measures. Applying this dial, we find that even the measure that looked most stable in our earlier probes—the standard parameter-space PAC–Bayes proxy `PACBAYES_orig`—*fails catastrophically* here: as the Exp++ factor grows, the proxy inflates by orders of magnitude while test error is essentially unchanged; see Appendix 4.11.

4.7 Post-mortem vs. ML-PACBayes

We contrast post-training ("post-mortem") generalization measures with a function-space approach based on the marginal-likelihood PAC–Bayes bound of Valle-Pérez and Louis [2020]. The bound controls the test error of a hypothesis sampled from a Bayesian posterior over *functions* (not parameters); the key capacity term is the marginal likelihood (Bayesian evidence) of the data under the architecture's Gaussian-process (GP) limit. **All figure numbers below refer to Valle-Pérez and Louis [2020].**

Definition 9 (Marginal-likelihood PAC–Bayes bound)**.** *Consider binary classification with data distribution $\mathcal{D}$ over $\mathcal{X}\times\{0,1\}$ and a hypothesis space $\mathcal{H}$ of functions $h:\mathcal{X}\to\{0,1\}$. Let $S\sim\mathcal{D}^n$ be a training set of size $n\ge 2$, and let P be a prior on $\mathcal{H}$. Define the consistency set*

$$C(S) \;:=\; \{\, h\in\mathcal{H} \;:\; \hat{\varepsilon}(h,S)=0 \,\},$$

where $\hat{\varepsilon}(h,S) = \frac{1}{n}\sum_{(x,y)\in S} \mathbf{1}\{h(x) \neq y\}$ *is the empirical 0–1 error. The (realizable) Bayesian posterior supported on* $C(S)$ *is*

$$Q(h) \;=\; \begin{cases} \dfrac{P(h)}{P(C(S))}, & h \in C(S), \\ 0, & \textit{otherwise}, \end{cases}$$

where $P(C(S)) = \sum_{h\in C(S)} P(h)$ *is the* marginal likelihood *(Bayesian evidence) of* S*.*

For any confidence levels $\delta, \gamma \in (0,1]$*, with probability at least* $1-\delta$ *over* $S \sim \mathcal{D}^n$ *and, conditional on* S*, with probability at least* $1-\gamma$ *over* $h \sim Q$*, the generalization error* $\varepsilon(h) = \Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq y]$ *satisfies*

$$-\ln\Big(1-\varepsilon(h)\Big) \;<\; \frac{\ln\frac{1}{P(C(S))} + \ln n + \ln\frac{1}{\delta} + \ln\frac{1}{\gamma}}{n-1}. \tag{4.12}$$

Remark (binary vs. multiclass). We state the bound for binary classification for clarity; the function-space marginal-likelihood perspective extends to multiclass via standard reductions (e.g., one-vs-rest) or multiclass likelihoods.

What our fragility checks reveal—and what the ML-PACBayes bound gets right. We evaluated each measure against consequential perturbations: dataset difficulty, data-size scaling, training-pipeline changes, and temporal behavior (overtraining). Post-mortem measures (sharpness/flatness, norm/margin proxies, compression-style, and deterministic PAC–Bayes surrogates) often fail at least one stressor; in our runs, even strong performers such as `PACBAYES_orig` failed key tests. By contrast, the GP-based bound tracks the *function-level* regularities we care about:

- **Data complexity.** The marginal-likelihood bound *tracks dataset difficulty*: it increases with label corruption and preserves the canonical ordering MNIST < Fashion-MNIST < CIFAR-10; see Figure 1 in [Valle-Pérez and Louis, 2020]. In our audits, several post-mortem measures—including `PACBAYES_orig`—either flattened under corruption or were not able to catch the ordering, even when the test error did.

- **Learning-curve scaling.** The marginal-likelihood bound *tracks learning-curve scaling*: it mirrors the empirical power-law in n and clusters exponents primarily by *dataset*; see Figures 2–5 in [Valle-Pérez and Louis, 2020]. By contrast, some post-mortem measures barely moved with n; others bent the wrong way or mixed optimizer effects with data effects (cf. [Nagarajan and Kolter, 2019]).

- **Temporal behavior (overtraining invariance).** The marginal-likelihood bound *tracks temporal invariance*: once S is fixed and training has interpolated ($\hat{\varepsilon} = 0$), the bound depends only on $P(C(S))$ and n—it is agnostic to how long or by which path the parameters were trained. In our temporal-behavior experiments, several norm-based post-mortem measures (e.g., ℓ_2/spectral/path norms) *kept growing* during overtraining while the generalization error stayed roughly unchanged; the GP bound remained stable. (This echoes broader observations that longer training can leave test error flat or improved [Hoffer et al., 2017], and that norms can diverge under separable losses without hurting classification error [Soudry et al., 2018b].)

- **Training-pipeline changes.** Many post-mortem measures swing with optimizer, batch size, explicit/implicit regularization, and early stopping—sometimes tracking curvature artifacts rather than out-of-sample error [e.g., Keskar et al., 2017, Hochreiter and Schmidhuber, 1997b, Jiang et al., 2019b, Dziugaite et al., 2020b]. The GP marginal-likelihood bound is, by construction, insensitive to these knobs: it depends on the architecture's function prior and the data, not on the path taken through parameter space. We view this *invariance* as a virtue for a predictor; the bound is not designed to explain *differences caused by* optimizer choice and should not be judged on that criterion.

These properties are consistent with our optimizer-swap and pixel-permutation stress tests (Apps. C.2 and C.4) and with our scale-symmetry Exp++ dial, which only alters parameter scale (App. 4.11).

Why does the function-space route outperform post-mortems? Two ingredients stand out. First, the bound evaluates an Occam factor in *function space*: architectures that place high prior mass on data-compatible functions earn better evidence, naturally capturing dataset ordering and learning-curve slopes. Second, there is a credible external yardstick: infinite-width GP predictors (NNGP/NTK) often *quantitatively* predict finite-width DNN generalization trends [Lee et al., 2020a]. Together, these explain why Figures 1–5 show greater stability for the GP-based approach than we observe for post-hoc, parameter-space measures.

So why do post-mortems underperform? The live hypothesis is that they *measure the wrong object*. Post-mortem scores probe properties of a single trained parameter vector (curvature, norms, compressibility), entangling optimization details and reparameterization choices with generalization. That enterprise is important—and the community has argued forcefully that such post-training diagnostics deserve attention [e.g., compression and robustness perspectives in Arora et al., 2018b, 2019b, Jiang et al., 2019b, Dziugaite et al., 2020b]—but our evidence suggests their predictions are fragile under routine perturbations. By contrast, the GP bound targets the distribution over *functions* implied by architecture and data.

Open question. Even if the GP-based predictor/bound wins these stress tests, finite-width networks can outperform their GP limits; yet the GP still predicts a striking fraction of performance trends (Figures 2–5). Closing this gap—by designing reparameterization-invariant, data-aware post-mortem diagnostics that inherit the Occam flavour of marginal likelihood—remains open.

4.8 Conclusion and discussion

We take a pragmatic view of generalization bounds: in modern deep learning they function best as measures to be judged by how they behave across data and training regimes rather than by worst-case tightness. Read this way, fragility becomes the key property to audit. With the task fixed and accuracy essentially steady, small

and reasonable pipeline tweaks—changing the learning rate by a factor of ten, swapping ADAM for momentum SGD, toggling early stopping—can bend surrogate curves, invert trends, or inflate their scale. When the task truly becomes harder, via label corruption or by moving from MNIST to FashionMNIST to CIFAR-10, several surrogates fail to preserve the expected ordering or slope. Our scale-invariance construction makes this failure mode explicit: one can inflate parameter magnitudes without changing the predictor, and magnitude-sensitive post-mortems then balloon while test error does not. This gap underscores that properties of a single parameter vector need not reflect properties of the learned function.

The comparison between post-mortem diagnostics and the marginal-likelihood PAC–Bayes route in function space sharpens the lesson. The GP-based predictor respects dataset difficulty, mirrors learning-curve slopes chiefly as a property of the data distribution, and remains largely indifferent to optimizer path once the training set is interpolated; post-mortems, by contrast, tend to entangle reparameterization and optimizer-driven scale with generalization. The function-space approach is not a panacea, but it points to the right invariances for any useful measure and provides a concrete foil against which parameter-space surrogates can be refined.

Looking forward, several concrete directions could strengthen this position:

- design reparameterization-invariant, data-aware post-training diagnostics that borrow the Occam flavour of marginal likelihood while remaining practical at scale;
- build a public fragility benchmark and automated audit harness covering optimizer/schedule swaps, scale-symmetry probes, and data-complexity sweeps;
- develop approximate function-space surrogates (finite-width corrections, amortized evidence estimators, ensembles as posteriors over functions) to make invariance-friendly predictors usable in routine training;
- tie diagnostics to training dynamics by discouraging scale-only drift after interpolation or by incorporating function-space objectives during training;

- extend evaluation beyond vision classification to sequence models, generative modelling, reinforcement learning, and explicit distribution shift, testing whether the same invariance principles hold.

4.9 Alternative Views

Reasonable researchers can disagree with our emphasis on parameter-space fragility and our recommendation to favor function–space diagnostics. We highlight several viable counterpositions and address them in turn:

- **Optimizer sensitivity is signal, not noise.** Because implicit bias *drives* which predictor is learned, a useful measure should move with the training pipeline, not be invariant to it.
- **Post-mortem measures can be stabilized.** With the right reparameterization-aware definitions (e.g., normalized sharpness, margin distributions, distance from initialization), much of the apparent fragility disappears.
- **Function-space GP baselines are mis-specified for modern, finite-width practice.** GP priors, absence of data augmentation in the prior, and finite-width effects limit how decisively marginal-likelihood predictors should be used as a reference.

Optimizer sensitivity as a feature. A common objection is that optimizer, schedule, and batch size are first-order determinants of generalization; therefore, measures *should* reflect them. This view points to well-documented implicit-bias phenomena—e.g., margin growth under separable losses and optimizer-dependent convergence paths—that influence which classifier is ultimately selected [Soudry et al., 2018b, Smith et al., 2017b, Simsekli et al., 2019]. From this perspective, the very invariances we prize risk washing out real, practically actionable differences between training recipes.

Response. We agree that optimizer choice can change the learned function and that a diagnostic should detect *functionally meaningful* changes. Our claim is narrower: diagnostics that swing under *pure scale drift* or minor hyperparameter nudges while test error and predictions remain essentially fixed can mislead day-to-day comparisons. This is why our audit always pairs any optimizer comparison with either (i) matched-prediction checkpoints (e.g., at the first 100%-accuracy crossing) or (ii) a symmetry lever that alters parameter scale without changing the predictor (§4.6). When pipeline changes *do* alter the predictor, a stable measure should move in tandem with test error; when they do not, a robust measure should be indifferent.

Stabilizing post-mortem diagnostics. A second objection holds that many post-training measures already address scale and parameterization issues. Normalized or reparameterization-aware sharpness correlates more reliably with generalization than raw curvature [Dinh et al., 2017b, Tsuzuku et al., 2020, Kim et al., 2022, Jang et al., 2022]; modeling the *distribution* of margins rather than the minimum improves predictiveness across runs [Jiang et al., 2018]; reference-aware surrogates such as distance from initialization or movement from pretraining reduce width and scale artifacts and come with supporting analyses [Li et al., 2018b, Neyshabur et al., 2017b, Zhou et al., 2021]. On this view, fragility largely reflects naive implementations, not intrinsic flaws.

Response. We see these developments as complementary and encouraging. Our results target precisely such "best-effort" variants (normalized, margin-aware, reference-aware) and still uncover qualitative flips under mild pipeline changes when predictions are stable. We do not argue that post-mortems are useless; rather, we argue they remain *fragile enough* that authors should routinely report their stability under the stressors we outline (§4.3–4.5), and prefer versions that (i) are explicitly invariant to layerwise rescalings, (ii) condition on matched prediction milestones, and (iii) are benchmarked against data-difficulty shifts and temporal drift after interpolation. In short, our audit is a bar to clear, not a dismissal of the enterprise.

Limits of function–space GP references. A third objection is that using a GP-based marginal-likelihood PAC–Bayes predictor as a baseline (§4.7) overstates its practical authority. Modern models are finite-width, heavily augmented, and often trained far from the GP regime; priors that ignore augmentation, architectural quirks, or fine-tuned tokenization may be badly mis-specified. In addition, the bound controls the error of a stochastic Gibbs classifier drawn from a posterior over functions, not necessarily the deterministic network one actually deploys. Therefore, the GP route's stability could stem from *omitting* factors that matter in practice [Lee et al., 2020b, Valle-Pérez and Louis, 2020].

Response. We agree the GP prior is not a panacea and that finite-width networks can outperform their GP limits. We use the GP marginal-likelihood bound as a *calibration* tool, not an oracle: it encodes desirable invariances (insensitivity to parameter scale and optimizer path once the dataset is fixed) and consistently tracks data difficulty and learning-curve scaling across families. Those properties make it a valuable foil for stress-testing post-mortems. Where the GP prior is clearly mis-specified (e.g., heavy augmentation or domains far from image classification), our recommendation is empirical: rerun the same fragility audit. If the GP predictor ceases to track error while a reparameterization-invariant post-mortem does, that is evidence *for* the post-mortem in that regime.

Synthesis. These counterarguments motivate a middle path. Rather than elevating any single diagnostic, we advocate (i) reporting multiple invariance-aware post-mortems, (ii) auditing them with our stressors (learning-curve shape, temporal drift after interpolation, and data-complexity response), and (iii) anchoring interpretation with at least one function–space reference when feasible. This recognizes optimizer-dependent mechanisms while discouraging conclusions drawn from features (e.g., raw scale) that can be changed without affecting predictions.

Supplementary Material for Chapter 4

4.10 Training-hyperparameter fragility for all measures

In this section we give concise definitions of the measures we study (aligning notation with prior large-scale evaluations) and provide additional experimental evidence beyond the main text [Dziugaite et al., 2020b, Jiang et al., 2019b]. We then present figure-backed comparisons where small learning-rate or optimizer tweaks trigger qualitatively different curve shapes for the surrogate—plateaus, rebounds, late spikes, and crossings—while the accompanying accuracy curves remain comparatively calm.

4.10.1 Frobenius distance

Two qualitatively different shapes appear when only the learning rate or optimizer changes, and they are visible in Fig. 4.5 and Fig. 4.6. On ResNet-50/CIFAR-10 with Adam, η=10^{-3} produces a broad plateau followed by a late collapse, whereas η=10^{-2} decays smoothly throughout (compare left vs. right in Fig. 4.5). On DenseNet-121/FashionMNIST at fixed η=10^{-2}, Adam's curve is U-shaped (drop then rebound), while momentum SGD traces a near log-linear decline (left vs. right in Fig. 4.6); the accuracy curves remain closely aligned in both pairs.

4.10.2 Inverse margin

Figure 4.7 (left vs. right) shows that for ResNet-50/CIFAR-10 with Adam, η=10^{-3} produces a smooth, power-law-like decline in inverse margin, whereas η=10^{-2} stalls mid-training before resuming its drop; this kink has no analogue in the accuracy curve. At fixed η=10^{-2} on FCN/FashionMNIST, Adam develops a clear bump

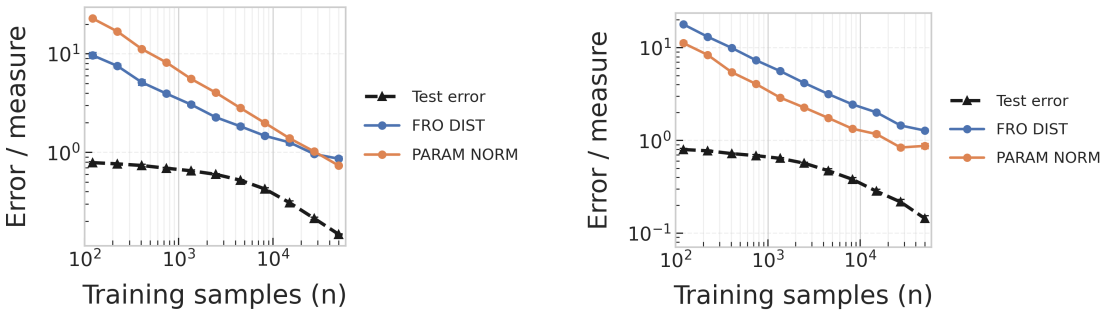


Figure 4.5: Frobenius distance, ResNet-50 on CIFAR-10 (Adam). Left: η=10^{-3} shows a plateau and late collapse; right: η=10^{-2} decays smoothly while accuracy tracks similarly.

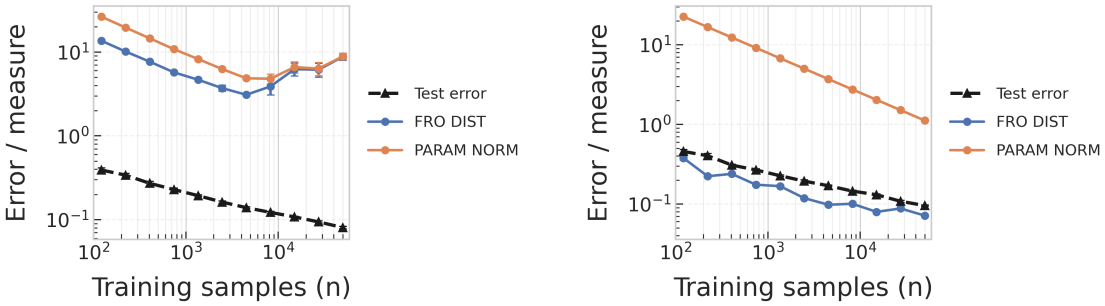


Figure 4.6: Frobenius distance, DenseNet-121 on FashionMNIST (η=10^{-2}). Left: Adam yields a U-shape; right: SGD+mom declines nearly log-linearly; in both cases test accuracy evolves similarly.

after roughly 10^3 samples while SGD's curve decreases monotonically (Fig. 4.8); again, both reach similar generalization.

4.10.3 Spectral metrics

Spectral surrogates display late spikes and order reversals under the same minimal tweaks. With DenseNet-121/FashionMNIST and Adam, η=10^{-3} sends the distance-from-initialization in spectral norm down and then sharply up late in training, crossing the *FRO-OVER-SPEC* curve, whereas η=10^{-2} keeps both traces monotone but flips their ordering (compare panels in Fig. 4.9). On ResNet-50/FashionMNIST at η=10^{-2}, Adam shows a valley then rise, while SGD declines steadily (Fig. 4.10); accuracy curves overlap in both pairs.

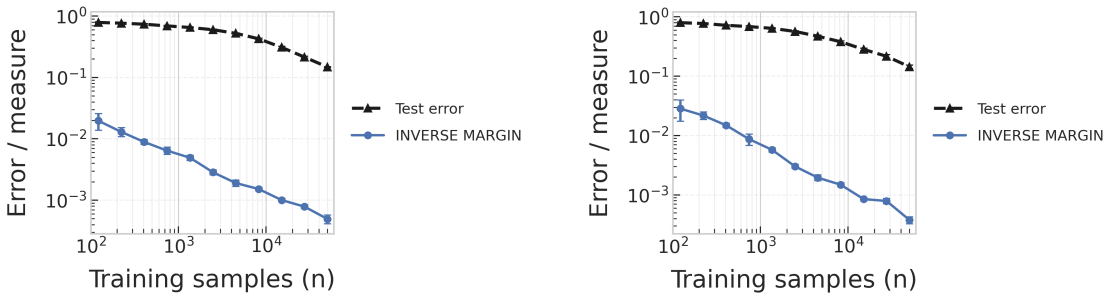


Figure 4.7: Inverse margin, ResNet-50 on CIFAR-10 (Adam). Left: η=10^{-3} decays smoothly; right: η=10^{-2} stalls then resumes, a kink absent from the accuracy curve.

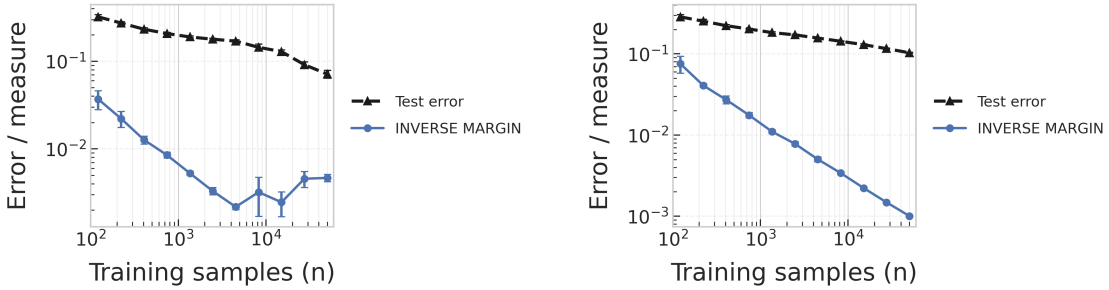


Figure 4.8: Inverse margin, FCN on FashionMNIST (η=10^{-2}). Left: Adam develops a mid-course bump; right: SGD+mom is monotone; both generalize similarly.

4.10.4 PAC-Bayes bounds

Optimizer swaps and moderate learning-rate changes also produce curved versus straight "bound profiles" and late order crossings. For ResNet-50/FashionMNIST at η=10^{-2}, Adam yields kinked "banana" trajectories across variants, whereas SGD renders near-straight lines (Fig. 4.11). Holding Adam fixed and raising η from 10^{-3} to 10^{-2} reorders the variants late in training—for example, PACBAYES MAG and PACBAYES ORIG swap rank—even though accuracy shows no such crossing (Fig. 4.12).

4.10.5 VC-dimension proxy (robust baseline)

As a control, Fig. 4.13 shows that the parameter-count proxy for ResNet-50/CIFAR-10 with Adam is strikingly shape-stable across η=10^{-3} and η=10^{-2}: both traces are near-identical monotone decays, tracking one another while accuracy also aligns. This pair serves as a rare counterexample resistant to qualitative shifts.

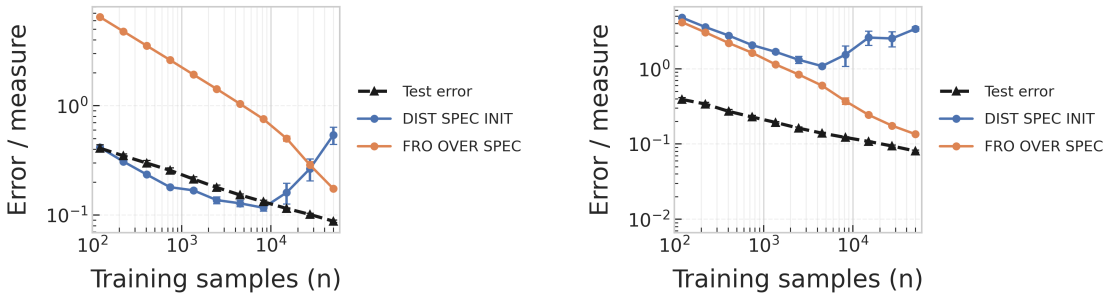


Figure 4.9: Spectral metrics, DenseNet-121 on FashionMNIST (Adam). Left: η=10^{-3} drops then spikes, crossing FRO-OVER-SPEC; right: η=10^{-2} stays monotone but reverses ordering.

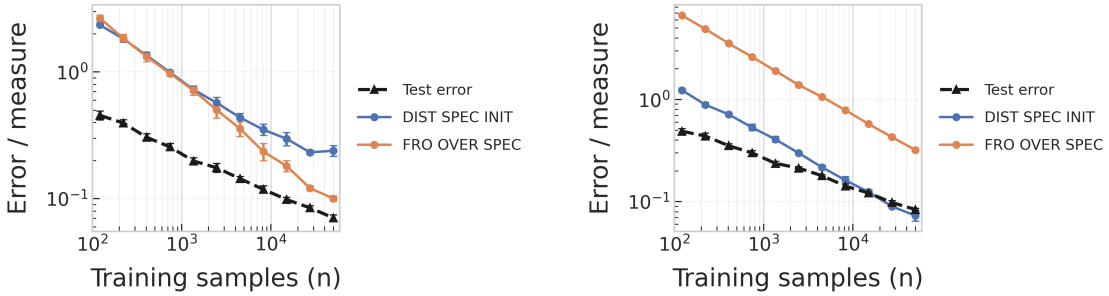


Figure 4.10: Spectral metrics, ResNet-50 on FashionMNIST (η=10^{-2}). Left: ADAM exhibits a valley then rise; right: SGD+mom declines steadily; accuracy is similar.

Across all families above, the figures make a consistent point: changing only the learning rate or swapping the optimizer can reshape the surrogate's learning curve—introducing plateaus, rebounds, late spikes, or crossings—without a commensurate shift in test accuracy. This shape-level fragility extends far beyond Path norms and cautions against reading causal explanations of generalization from any single surrogate without dedicated stress tests [Dziugaite et al., 2020b, Jiang et al., 2019b].

4.11 Exp++ in scale-invariant nets: protocol and results

We instantiate the symmetry in a fully connected, *scale-invariant* network (normalization after each hidden linear layer; the final linear readout is frozen), trained on MNIST with SGD + momentum. We sweep a single control, the *Exp++ multiplicative factor* α from Theorem 10, which makes the learning rate grow exponentially across steps, and repeat the sweep both without and with weight decay.

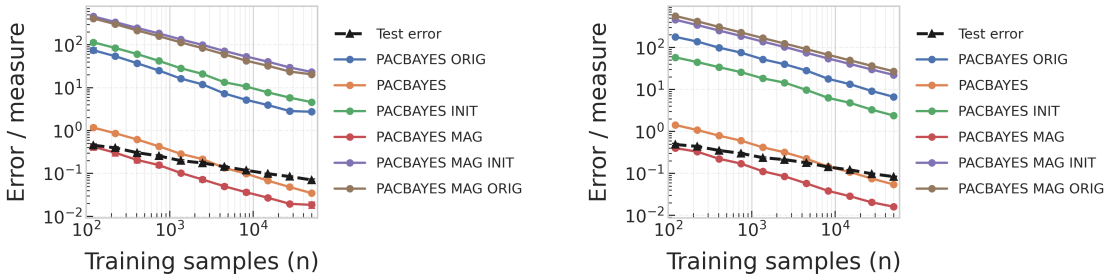


Figure 4.11: PAC-Bayes, ResNet-50 on FashionMNIST (η=10^{-2}). Left: ADAM produces curved, kinked trajectories; right: SGD+mom is nearly straight; test error changes little.

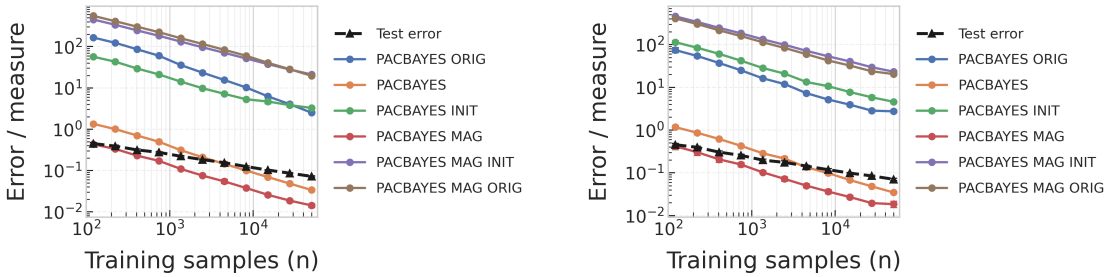


Figure 4.12: PAC-Bayes, ResNet-50 on FashionMNIST (Adam). Left: η=10^{-3}; right: η=10^{-2}. Increasing η induces late-training order swaps (e.g., MAG vs. ORIG) absent from the accuracy curves.

For each run we log parameter-space PAC–Bayes proxies (including magnitude-aware variants), path-norm proxies (with/without margin normalization), and test error. The stopping rule is cross-entropy unless stated otherwise.

Findings. Across both regimes (no WD and with WD) the test-error trace barely moves as α grows, yet the magnitude-sensitive diagnostics explode on a log scale. In particular, the PAC–Bayes family and the path-norm proxies rise by many orders of magnitude even though the error curve hugs a dashed 10% reference. This is exactly what the equivalence predicts in a scale-invariant net: Exp++ changes parameter *scale*, not the predictor—exposing the fragility of magnitude-dependent measures (notably `PACBAYES_orig`) in this setting.

Reading the panels. In both Figure 4.14 (no WD) and Figure 4.15 (with WD), left panels aggregate PAC–Bayes proxies; right panels show path-norm proxies. Axes are logarithmic for the measures and linear for the test error (right axis); the dashed horizontal line marks 10% test error for visual reference. Together,

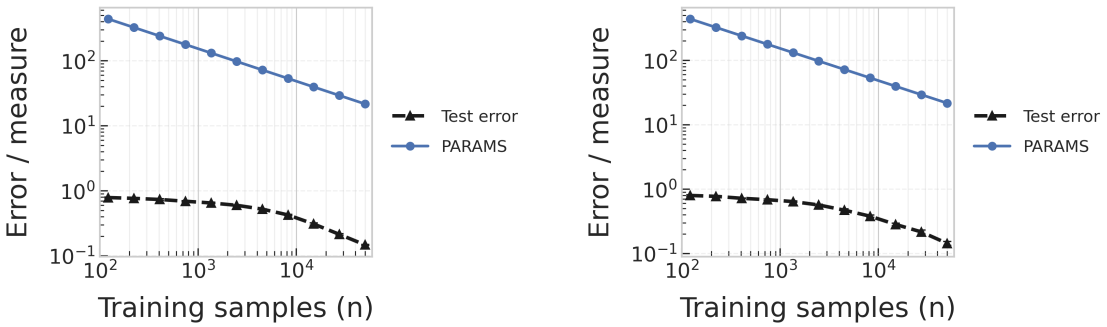


Figure 4.13: VC-dimension proxy, ResNet-50 on CIFAR-10 (Adam). Left: η=10^{-3}; right: η=10^{-2}. Nearly identical monotone decays; a useful control.

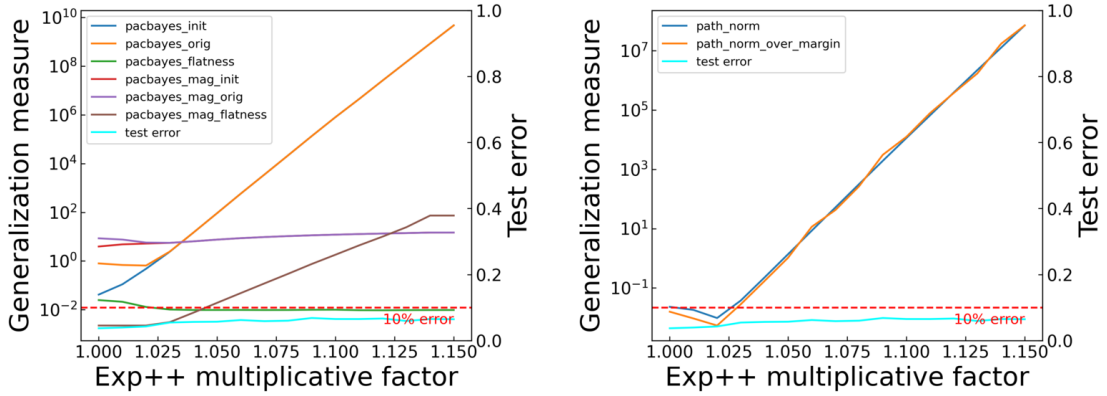


Figure 4.14: Exp++ in a scale-invariant FCN (no WD; CE stopping). As the Exp++ factor increases, PAC–Bayes proxies (left) and path-norm proxies (right) swell by orders of magnitude, while the test-error curve (right axis; dashed 10% line) remains essentially flat.

the panels illustrate a clean separation between *parameter scale* (which Exp++ manipulates) and *predictive behavior* (which stays essentially fixed).

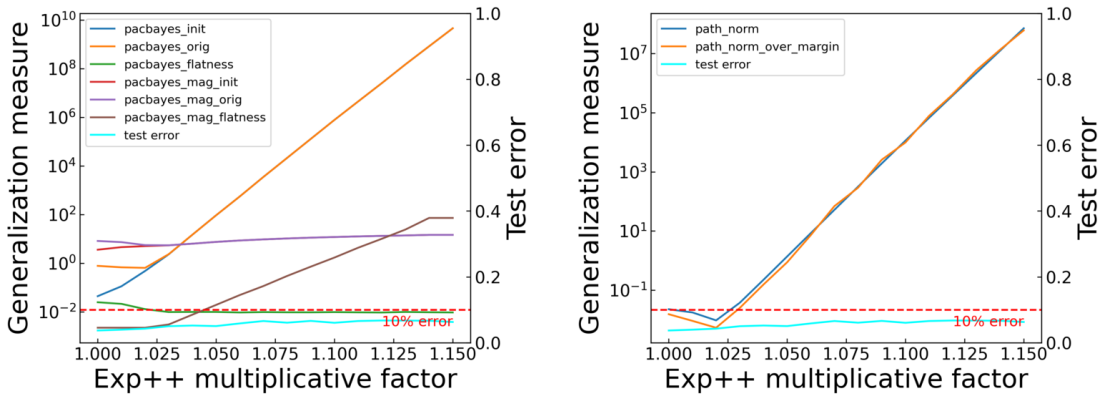


Figure 4.15: Exp++ with weight decay (CE stopping). The qualitative picture persists with WD: magnitude-sensitive PAC–Bayes and path-norm diagnostics climb sharply with the Exp++ factor, but the predictor's test error barely changes.

5

Closed-form ℓ_r Norm Scaling with Data for Overparameterized Linear Regression and Diagonal Linear Networks under ℓ_p Bias

Overview

For overparameterized linear regression with isotropic Gaussian design and minimum-ℓ_p interpolator with $p \in (1, 2]$, we give a unified, high-probability characterization for the scaling of the family of parameter norms $\{\|\widehat{w_p}\|_r\}_{r\in[1,p]}$ with sample size. We solve this basic, but unresolved question through a simple dual-ray analysis, which reveals a competition between a signal *spike* and a *bulk* of null coordinates in $X^\top Y$, yielding closed-form predictions for (i) a data-dependent transition $n_\star$ (the "elbow"), and (ii) a universal threshold $r_\star = 2(p-1)$ that separates $\|\widehat{w_p}\|_r$'s which plateau from those that continue to grow with an explicit exponent. This unified solution resolves the scaling of *all* ℓ_r norms within the family $r \in [1, p]$ under ℓ_p-biased interpolation, and explains in one picture which norms saturate and which increase as n grows. We then study diagonal linear networks (DLNs) trained by gradient descent. By calibrating the initialization scale α to an effective $p_{\text{eff}}(\alpha)$ via the DLN separable potential, we show empirically that DLNs inherit the same elbow/threshold laws, providing a predictive bridge between explicit and implicit

bias. Given that many generalization proxies depend on $\|\widehat{w}_p\|_r$, our results suggest that their predictive power will be highly sensitive to which ℓ_r norm is used.

5.1 Chapter introduction

Many modern generalization measures for machine learning tasks are anchored on the parameter norm instead of parameter count [Neyshabur et al., 2015c,a, Yoshida and Miyato, 2017, Miyato et al., 2018, Cisse et al., 2017]. Yet, most analyses of overparameterized regression still treat "the norm" monolithically—typically defaulting to ℓ_2. If one is going to use a parameter norm, *which* ℓ_r should be used, and how does that choice interact with the inductive bias that selects the interpolator (e.g., minimum-ℓ_p)? This question has been comparatively less studied. We address this question first in a simpler but core setting—linear regression—and then connect the picture to diagonal linear networks (DLNs). Our experiments reveal that sweeping (r, p) produces non-trivial behavior: even for the *same* interpolating predictor, some ℓ_r norms plateau while others keep growing with distinct slopes; in mixed cases, the elbow's location shifts with p, and *which* r's plateau depends on the setting.

In linear regression it is well understood that the value of p *shapes* the inductive bias (sparser as $p\downarrow 1$, denser as $p\uparrow 2$), making the r–p interaction concrete. Beyond explicit ℓ_p penalties, first-order optimization can *implicitly* select a geometry: in overparameterized linear regression, gradient methods recover the minimum-ℓ_2 interpolant; in separable classification, gradient descent converges to a max-margin solution; and in diagonal/deep linear parameterizations, the separable potentials governing the dynamics interpolate between sparse- and dense-leaning behaviors depending on initialization and parameterization [Tibshirani, 1996, Frank and Friedman, 1993, Hoerl and Kennard, 1970, Chen et al., 2001, Zou and Hastie, 2005, Hastie et al., 2015, 2022a, Soudry et al., 2018a, Gunasekar et al., 2018a, Ji and Telgarsky, 2019b, Chizat et al., 2019, Woodworth et al., 2020]. This variety

Code for this work can be found at `https://github.com/sofuncheung/minlp_codebase`

of explicit/implicit pathways for p-like biases motivates our unified treatment of the *family* $\{\|\widehat{w}\|_r\}$ and explains why different ℓ_r proxies can exhibit qualitatively different n-dependence under a fixed training pipeline.

Concretely, we study the minimum-ℓ_p interpolator in high-dimensional linear regression with isotropic Gaussian design ($d > n$, $p \in (1, 2]$), and we characterize—*in closed form and with high probability*—how the entire family $\{\|\widehat{w}_p\|_r\}_{r\in[1,p]}$ scales with n. A one-dimensional *dual–ray* analysis exposes a competition between a signal *spike* and a high-dimensional *bulk* in $X^\top Y$, yielding: (i) a data-dependent transition size $n_\star$ (an elbow in n), and (ii) a universal threshold $r_\star = 2(p-1)$ that separates norms that ultimately plateau ($r > r_\star$) from those that continue to grow with explicit exponents ($r \le r_\star$). We also extend the picture to DLNs trained by gradient descent: calibrating the initialization scale α via the network's separable potential that gives an *effective* exponent $p_{\text{eff}}(\alpha)$, and with this calibration the observed ℓ_r–vs–n curves inherit the same elbow/threshold structure as explicit minimum-ℓ_p interpolation. *When the inductive bias is unknown a priori*—e.g., the operative p of the training pipeline is unclear—our results imply that choosing the "right" r for norm-based generalization measures can be delicate, since different (r, p) pairs can produce opposite scaling behaviors (plateau vs. growth) as n increases.

Our contributions:

1. **Strong sensitivity of the parameter norm as a function of the pair (r, p)** We find a strong *qualitative* effect for the scaling of the parameter norm with data: for fixed p, certain ℓ_r norms plateau while others grow with different slopes; varying p moves the elbow and reassigns which r's plateau.

2. **Closed-form scaling laws for parameter norms.** We derive the first unified closed-form scaling laws for this problem. For $p \in (1, 2]$ and all $r \in [1, p]$, we identify the universal threshold $r_\star = 2(p-1)$, give an explicit expression for the transition size $n_\star$, and provide plateau levels and growth exponents in both spike- and bulk-dominated regimes via a compact dual–ray argument.

3. **Extension of our theory to DLNs.** We map the DLN initialization scale to geometry: $\alpha \mapsto p_{\text{eff}}(\alpha)$. Using this map, we transfer the theory to DLNs and verify the predicted elbow/threshold behavior of the parameter norm empirically.

Implications for practice. Because many norm-based generalization measures and diagnostics depend on $\|\widehat{w}\|_r$, our results imply that practitioners using norm-based bounds or proxies—especially in more complex models such as DNNs—should be cautious: conclusions can be *highly sensitive* to the choice of r, and the sensitivity depends on the underlying ℓ_p bias that selects the interpolator.

5.2 Related work

The focus of this paper is a basic question: for overparameterized linear regression and related diagonal linear networks (DLNs), how do the *parameter norms* $\{\|\widehat{w}\|_r\}_{r\in[1,p]}$ scale with sample size when the interpolator is selected by an ℓ_p bias? The links to generalization are therefore indirect: norm quantities often appear as proxies in modern generalization measures [Neyshabur et al., 2015b, Bartlett et al., 2017a, Dziugaite and Roy, 2017a], so understanding their n–scaling is informative, but we do not develop new generalization bounds here. Relatedly, recent analyses derive explicit norm upper bounds as intermediate steps toward generalization—often via Gaussian min–max techniques—for interpolators and max-margin procedures [Koehler et al., 2021, Donhauser et al., 2022].

The ℓ_r family of linear–regression interpolators. A large body of work characterizes how explicit ℓ_p penalties shape linear estimators: ridge/Tikhonov (ℓ_2) [Hoerl and Kennard, 1970], lasso (ℓ_1) [Tibshirani, 1996, Efron et al., 2004, Knight and Fu, 2000, Zou, 2006], elastic net (mixtures of ℓ_1 and ℓ_2) [Zou and Hastie, 2005], and the bridge family (ℓ_p for $0 < p \leq 2$) [Frank and Friedman, 1993]; basis pursuit gives the sparse interpolating extreme under equality constraints [Chen et al., 2001, Candès and Tao, 2007, Donoho, 2006, Bickel et al., 2009]. High–dimensional convex-geometric analyses explain when these programs select structured solutions and

how their solutions move with the data geometry [Chandrasekaran et al., 2012, Amelunxen et al., 2014, Bühlmann and van de Geer, 2011, Wainwright, 2019], and recent developments give precise characterizations for ridgeless (minimum-ℓ_2) interpolation and its risk [Hastie et al., 2022a,b]. Our contribution complements this literature by treating the *entire* norm family $\{\|\widehat{w}_p\|_r\}_{r\in[1,p]}$ for minimum–ℓ_p *interpolators* (with $p \in (1, 2]$) and deriving closed-form, high-probability scaling laws in n across r. In this sense we move from "which p shapes which estimator?" to "given p, how do all ℓ_r diagnostics behave as data grow?"

Overparameterization in regression and deep networks. The deep-learning era stimulated a re-examination of overparameterized regression, revealing phenomena such as double descent [Belkin et al., 2019, Nakkiran et al., 2021, Zhang et al., 2017, Nakkiran et al., 2020a, Adlam and Pennington, 2020] and benign overfitting for minimum-norm interpolators [Bartlett et al., 2020, Hastie et al., 2022b, Muthukumar et al., 2020]. These results show that linear regression can capture qualitative behaviors seen in deep learning models and that the *selected* interpolator's norm matters for risk. Our work leverages this bridge as motivation only: by explaining, in closed form, which ℓ_r norms plateau and which grow (and at what rates) under an ℓ_p bias, we clarify what one should expect from norm-based proxies commonly used in deep-net analyses. Because practical pipelines for deep models rarely specify the effective p, our finding that $\|\widehat{w}_p\|_r$ depends sensitively on the *pair* (r, p) suggests caution when interpreting norm-based generalization diagnostics.

Explicit/implicit regularization and DLNs. Beyond explicit penalties, optimization can select solutions with an *implicit* geometry [Soudry et al., 2018a, Lyu and Li, 2020, Gunasekar et al., 2018b, 2017a]. In overparameterized linear regression, gradient methods recover the minimum-ℓ_2 interpolant; in factorized or deep-linear parameterizations, the training dynamics induce separable potentials that interpolate between sparse- and dense-leaning behaviors depending on initialization and parameterization [Saxe et al., 2014b, Gunasekar et al., 2018a, Ji and Telgarsky, 2019b, Chizat et al., 2019, Woodworth et al., 2020]. We build on

this perspective for DLNs: by calibrating the initialization scale to an effective p_{eff}, we show empirically that DLNs inherit the same elbow/threshold laws for $\{\|\widehat{w}\|_r\}$ as explicit minimum–ℓ_p interpolation.

Proof techniques. Our analysis borrows standard high-dimensional tools used throughout the modern regression literature—Gaussian concentration, blockwise (signal-vs-bulk) decompositions, and dual certificates in convex programs [Vershynin, 2018, Tropp, 2015]—and combines them with a one-dimensional "dual–ray" reduction tailored to the ℓ_p penalty. Two closely related works derive norm *upper* bounds as an intermediate step toward generalization, using the Gaussian Min–Max Theorem (GMT) and its convex analogue (CGMT): Koehler et al. [2021], Donhauser et al. [2022]. Their GMT/CGMT-based proofs are conceptually similar in spirit; by contrast, our argument proceeds from first principles via a simple dual–ray balance and yields closed-form n-scaling laws without invoking GMT/CGMT (see also Gordon [1985], Thrampoulidis et al. [2015] for the GMT and CGMT statements).

5.3 Family of norm measures of minimum ℓ_p-norm interpolator in linear models

We now formalize the object introduced in the overview: for $p \in (1, 2]$ in overparameterized linear regression, we study the family $\{\|\widehat{w}_p\|_r\}_{r\in[1,p]}$ where $\widehat{w}_p$ is the minimum-ℓ_p interpolator. Our goal is to characterize how these norms scale with sample size n. Our results identify (i) a data-dependent elbow $n_\star$ and (ii) a universal threshold $r_\star = 2(p-1)$ that separates plateauing from growing ℓ_r's.

Data and settings. We consider overparameterized linear models with $X \in \mathbb{R}^{n\times d}$, $d > n$, rows i.i.d. $\mathcal{N}(0, I_d)$, and

$$Y = Xw^\star + \xi, \qquad \xi \sim \mathcal{N}(0, \sigma^2 I_n).$$

The minimum-ℓ_p interpolator is

$$\widehat{w}_p \in \arg\min_{w\in\mathbb{R}^d} \|w\|_p \quad \text{s.t.} \quad Xw = Y, \qquad p \in (1, 2].$$

Let $s = \|w^\star\|_0$ denote the support size and $\tau_s^2 := \|w^\star\|_2^2 + \sigma^2$. In contrast to interesting recent work by Donhauser et al. [2022], our theory is *not* restricted to the $w^* = e_1$ limit of extreme sparse regression.

5.3.1 Main theorem

Theorem 11 (ℓ_r scaling of minimum-ℓ_p interpolators)**.** *Fix $p \in (1, 2]$, set $q = \frac{p}{p-1}$, and take $r \in [1, p]$. Assume*

$$\frac{d}{n} \to \kappa \in (1, \infty) \quad \textit{and} \quad \liminf_{n\to\infty} \frac{d-s}{n} = \kappa_{\text{bulk}} > 0.$$

Let $w^\star$ have support S with $|S| = s$, and let

$$\widehat{w}_p \in \arg\min_{w\in\mathbb{R}^d} \|w\|_p \quad s.t. \quad Xw = Y.$$

Write $W_q := \|w^\star\|_q^q$ and $m_t := \mathbb{E}|Z|^t$ for $Z \sim \mathcal{N}(0, 1)$. Define the ray scale $t_\star$ *via*

$$t_\star^{q-1} \asymp \frac{\|Y\|_2^2}{\|X^\top Y\|_q^q} \asymp \frac{\tau_s^2 n}{\underbrace{n^q W_q}_{spike} + \underbrace{(d-s)\, m_q\, \tau_s^q\, n^{q/2}}_{bulk} + \underbrace{O\Big(\tau_s^q\,(s\, n^{q/2} + s^{1+q/2})\Big)}_{remainder}} \quad w.h.p.. \tag{5.1}$$

Then, w.h.p. (see Remark A.2),

$$\|\widehat{w}_p\|_r \asymp \max\Big\{ t_\star^{q-1}\, n^{q-1}\, \|w^\star\|_{(q-1)r}^{q-1},\ (d-s)^{1/r} \left(t_\star\, \tau_s \sqrt{n}\right)^{q-1},\ s^{\max\{1/r,\ (q-1)/2\}} \left(t_\star\, \tau_s \sqrt{n}\right)^{q-1} \Big\}. \tag{5.2}$$

Introduce the transition scale

$$n_\star \asymp \left(\kappa_{\text{bulk}} \frac{\tau_s^q}{W_q}\right)^{\frac{2}{q-2}}. \tag{5.3}$$

In the two extremes, we obtain:

Spike-dominated** ($n \gg n_\star$)**:

$$\|\widehat{w}_p\|_r \asymp \begin{cases} \dfrac{\tau_s^{q+1}}{W_q}\, n^{\frac{1}{r} - \frac{1}{2(p-1)}}, & r \le 2(p-1), \\ \dfrac{\tau_s^2}{W_q}\, \|w^\star\|_{(q-1)r}^{q-1}, & r > 2(p-1). \end{cases} \tag{5.4}$$

Bulk-dominated *($n \ll n_\star$):*

$$\|\widehat{w}_p\|_r \asymp \max\Big\{ \kappa_{\text{bulk}}^{\frac{1}{r}-1}\,\tau_s\,n^{\frac{1}{r}-\frac{1}{2}},\ \kappa_{\text{bulk}}^{-1}\,\tau_s^{2-q}\,\|w^\star\|_{(q-1)r}^{q-1}\,n^{\frac{q}{2}-1}, \\ \kappa_{\text{bulk}}^{-1}\,\tau_s\,s^{\max\{1/r,\ (q-1)/2\}}\,n^{-1/2}\Big\}. \tag{5.5}$$

Since $d - s \asymp \kappa_{\text{bulk}} n$, the last term equals $\frac{\tau_s}{d-s}\, s^{\max\{1/r,\,(q-1)/2\}}\sqrt{n}$. All $\asymp$ hide absolute constants depending only on $(p, \kappa_{\text{bulk}}, r)$.

Remark (Dual viewpoint). The constrained problem $\min_w \frac{1}{p}\|w\|_p^p$ s.t. $Xw = Y$ has unconstrained dual $\max_\lambda\ \lambda^\top Y - \frac{1}{q}\|X^\top\lambda\|_q^q$, with KKT conditions $Xw = Y$ and $X^\top\lambda = \nabla f(w)$. Restricting to the ray $\lambda = tY$ yields $t_\star^{q-1} = \|Y\|_2^2/\|X^\top Y\|_q^q$. The "spike" vs. "bulk" terminology refers to which part of $\|X^\top Y\|_q$ controls $t_\star$.

Proof sketch. The behavior of the minimum-ℓ_p interpolator can be read through a simple dual lens: rather than track the optimizer directly, we examine a dual certificate that both fits the labels and respects a norm budget after passing through the design; pushing the dual along the label direction (a one-dimensional "ray") reveals a single diagnostic scale where the budget tightens, and this scale is controlled by two competing sources in the correlations $X^\top Y$: a "spike" part (true signal coordinates) that coherently accumulates with n, and a "bulk" part (many null coordinates) that aggregates small, mostly noisy effects. Balancing these two contributions defines a data-dependent transition sample size $n_\star$: for $n \ll n_\star$ the bulk dominates, the solution's mass is effectively spread over many coordinates, and the family $\{\|\widehat{w}_p\|_r\}$ grows with n in the way our bulk formulas predict (including the usual cross-r ordering and an $n^{1/2}$-type trend visible in the plots); for $n \gg n_\star$ the spike dominates, mass concentrates on the support, and a clean threshold—determined by p at $r = 2(p-1)$—splits the outcomes: ℓ_r plateaus for r above the threshold and grows with a gentler, explicit slope for r below it. Standard concentration for Gaussian designs justifies the spike/bulk decomposition and the stability of the ray scale, and the KKT linkage between the dual certificate and the primal coordinates turns these ingredients into the unified bound, the expression for $n_\star$, and the two regime descriptions stated in the theorem. Full details are deferred to Appendix A.2. □

5.4 Corollaries for canonical targets

To make the unified laws in Theorem 11 concrete, we specialize them to two canonical targets: (i) a single spike $w^\star = e_1$, and (ii) a flat s-sparse vector with equal magnitude a on its support. Substituting the problem-specific scales $W_q = \|w^\star\|_q^q$ and $\tau_s^2 = \|w^\star\|_2^2 + \sigma^2$ into the elbow formula (5.3) and the spike-/bulk-dominated expressions (5.4)–(5.5) yields closed-form, high-probability predictions for $\|\widehat{w}_p\|_r$ and the transition size $n_\star$. We record these specializations below as Corollaries 5.4.1 and 5.4.2, and use them as reference overlays in our experiments.

5.4.1 Single spike

Corollary 5.4.1 (Single spike). *Under Theorem 11 with $w^\star = e_1$ and $\tau^2 = 1 + \sigma^2$, for any $r \in [1, p]$:*

$$\text{Bulk-dominated } (n \ll n_\star): \quad \|\widehat{w}_p\|_r \asymp \tau\,(d-1)^{\frac{1}{r}-1}\, n^{1/2},$$

$$\text{Spike-dominated } (n \gg n_\star): \quad \|\widehat{w}_p\|_r \asymp \begin{cases} \tau^{\,q+1}\, n^{\frac{1}{r}-\frac{1}{2(p-1)}} & \text{if } r \le 2(p-1), \\ \tau^{\,2} & \text{if } r > 2(p-1). \end{cases}$$

Interpretation. Here $W_q{=}1$ and $n_\star \asymp \left(\kappa_{\text{bulk}}\tau^{\,q}\right)^{2/(q-2)}$ from (5.3). For $r > 2(p{-}1)$ the ℓ_r curves *plateau* at level $\asymp \tau^2$ once $n \gg n_\star$; for $r \le 2(p{-}1)$ they continue to grow with slope $\frac{1}{r} - \frac{1}{2(p-1)}$.

5.4.2 Flat support

Corollary 5.4.2 (Flat support). *Under Theorem 11 and a flat $w^\star$ on S with $|S| = s$ and $w_j^\star = a\, s_j$ for $j \in S$ ($|s_j| = 1$), for any $r \in [1, p]$, w.h.p.:*

$$\text{Spike-dominated } (n \ge C n_\star): \quad \|\widehat{w}_p\|_r \asymp \begin{cases} \dfrac{(sa^2+\sigma^2)^{\frac{q+1}{2}}}{s|a|^q}\, n^{\frac{1}{r}-\frac{1}{2(p-1)}} & r \le 2(p-1), \\ s^{\frac{1}{r}-1}\,\dfrac{sa^2+\sigma^2}{|a|} & 2(p-1) < r \le p, \end{cases}$$

$$\text{Bulk-dominated } (n \le c n_\star): \quad \|\widehat{w}_p\|_r \asymp \max\Big\{ \kappa_{\text{bulk}}^{\frac{1}{r}-1}\,\tau_s\, n^{\frac{1}{r}-\frac{1}{2}},\ \kappa_{\text{bulk}}^{-1}\,\tau_s^{\,2-q}\, s^{1/r}|a|^{q-1}\, n^{\frac{q}{2}-1}, \\ \kappa_{\text{bulk}}^{-1}\,\tau_s\, s^{\max\{1/r,\,(q-1)/2\}}\, n^{-1/2} \Big\}.$$

Interpretation. Here $W_q{=}s|a|^q$ and $\tau_s^2{=}sa^2{+}\sigma^2$, so (5.3) gives $n_\star \asymp \left(\kappa_{\text{bulk}}\tau_s^q/(s|a|^q)\right)^{2/(q-2)}$, which grows with s (the elbow shifts to larger n). In the spike-dominated plateau branch ($r > 2(p{-}1)$) the level scales as $s^{\frac{1}{r}-1}\,(sa^2{+}\sigma^2)/|a|$, which is typically of the same order as the single-spike plateau for moderate s.

Comparison across targets. The threshold $r = 2(p-1)$ and the n-exponents in both regimes are *unchanged* between Corollaries 5.4.1 and 5.4.2. The differences lie in the *scales*: (i) the transition size moves from $n_\star \asymp (\kappa_{\text{bulk}}\tau^{\,q}/W_q)^{2/(q-2)}$ with $W_q{=}1$, $\tau^2{=}1{+}\sigma^2$ (single spike) to $n_\star \asymp (\kappa_{\text{bulk}}\tau_s^q/W_q)^{2/(q-2)}$ with $W_q{=}s|a|^q$, $\tau_s^2{=}sa^2{+}\sigma^2$ (flat), which scales roughly linearly in s (cf. (5.3)). Hence the elbow for regime change shifts to *larger* n when we move from e_1 to a flat $w^\star$ with $s{=}50$. (ii) In the spike-dominated plateau branch ($r > 2(p-1)$), the level changes from $\asymp \tau^2$ (single spike) to $\asymp s^{\frac{1}{r}-1}\,(sa^2{+}\sigma^2)/|a|$ (flat) [cf. (5.4) and Corollary 5.4.2]; for moderate s this produces *comparable* numerical magnitudes, which is why the vertical ranges in our figures are similar. The regime labels (bulk vs. spike) and their slopes/plateaus therefore provide the informative contrast.

5.4.3 Linear regression with explicit minimum-ℓ_p bias

Here the inductive bias is explicit: for a chosen p, the interpolator is the minimum-ℓ_p element among all w with $Xw = Y$. Sweeping p slides the solution from a more sparse-leaning geometry as $p \downarrow 1$ toward a more dense-leaning geometry as $p \uparrow 2$, revealing how the objective itself shapes the family $\{\|\widehat{w}_p\|_r\}_r$.

Experimental protocol. We set $\sigma = 0.1$, sweep $p \in \{1.1, 1.5, 1.9\}$, and vary n. Each plot overlays test MSE (left axis) and representative ℓ_r curves (right axis). For flat w^* experiments, we kept $\|w^*\|_2 = 1$, i.e. $a = \frac{1}{\sqrt{s}}$. Additional noise sweeps are reported in Appendix 5.7.

What the figures show and why. In Fig. 5.1 (single spike), the left/middle/right panels follow the corollary's regime rules. In the left panel, for $r > 2(p{-}1)$ the curves flatten after the transition, while for smaller r they retain the predicted growth; thin reference overlays (where present) trace these slopes. The middle

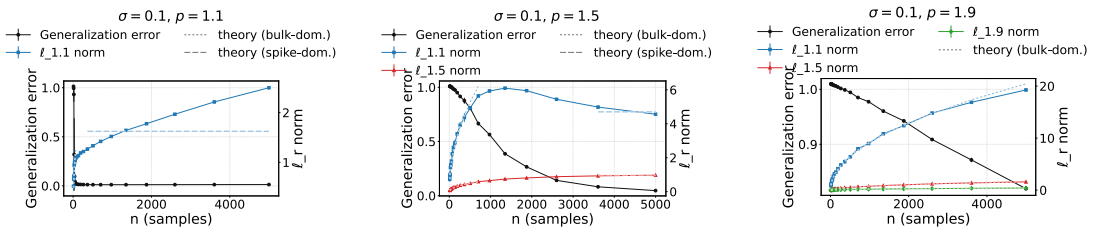
panel exhibits a clear elbow near the predicted $n_\star$; beyond it, the $r > 2(p-1)$ curves plateau in line with (5.4), while the others keep their slope. The right panel stays bulk-dominated across the range, with the ℓ_r traces growing approximately as $n^{1/2}$ and ordered across r as the bulk formula prescribes.

In Fig. 5.2 (flat $w^\star$ with s=50), the *same* slope/plateau rules apply, but the transition scale is larger: the elbow for p=1.5 appears at a later n (or just off-range), consistent with $n_\star$ increasing roughly linearly with s in (5.3). Across panels, the absolute ℓ_r values are numerically similar to Fig. 5.1; this matches the flat-support plateau level in Corollary 5.4.2, which for moderate s is close to the single-spike level. The informative distinction is thus *where* the curves switch from bulk growth to spike plateaus and the persistence of the $n^{1/2}$ slope in regimes that remain bulk-dominated.

Experiments with larger sparsity. We repeat the explicit minimum-ℓ_p runs at larger supports, $s \in \{500, 5000\}$, with the same $\|w^\star\|_2$=1 and noise level ($\sigma = 0.1$); see Appendix 5.9, Figs. 5.17-5.18. The qualitative picture from s=50 reappears but shifts to larger n, consistent with the transition size $n_\star$ in (5.3) growing with s. For small p (p=1.1), the prolonged bulk-dominated window makes the *double-descent* pattern visible—generalization error first *increases* and then drops (most clearly at s=5000)—while the blue $\ell_{1.1}$ curve keeps rising along the bulk guide across the plotted range [Belkin et al., 2019, Nakkiran et al., 2020b, Hastie et al., 2022a]. For larger p (p=1.5, 1.9), the curves remain monotonically decreasing; the minimized ℓ_p traces drift only mildly upward (no flattening within the range), reflecting the rounder geometry that avoids early over-reliance on noisy bulk directions. In all panels, the dashed overlays track the bulk/spike trends and the expected r-ordering of the ℓ_r diagnostics, matching the regime structure highlighted in the theory.

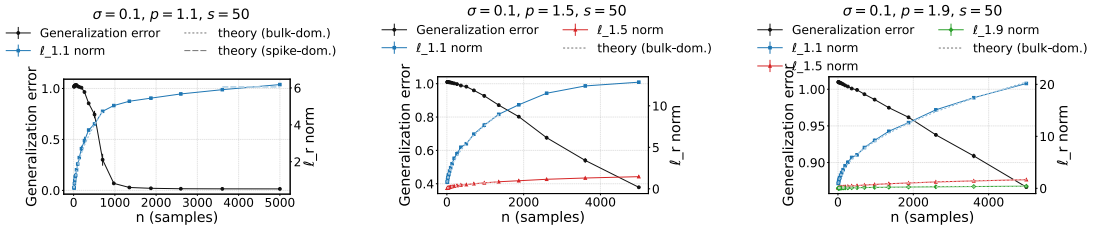
5.4.4 Diagonal linear network with implicit bias

Diagonal linear networks (DLNs) - deep linear models whose weight matrices are diagonal so that the effective predictor is the coordinatewise product of layer parameters—provide a tractable testbed for understanding optimization-induced



(a) $p = 1.1$ (sparsity-leaning) **(b)** $p = 1.5$ **(c)** $p = 1.9$ (dense-leaning)

Figure 5.1: Single spike $w^\star = e_1$; explicit minimum-ℓ_p interpolation. Ordering across r and the presence/absence of elbows follow Corollary 5.4.1; the bulk panels rise like $n^{1/2}$ and the spike-side panels plateau for $r > 2(p-1)$, consistent with (5.4)-(5.5).



(a) $p = 1.1$ (sparsity-leaning) **(b)** $p = 1.5$ **(c)** $p = 1.9$ (dense-leaning)

Figure 5.2: Flat $w^\star$ ($s = 50$); explicit minimum-ℓ_p interpolation. The scaling rules mirror the flat-support corollary: bulk growth persists until a larger transition scale, while spike-side r values plateau; absolute levels are comparable to the single-spike case, as predicted by the plateau formulas.

geometry and implicit bias in overparameterized systems. They connect classical analyses of linear nets and factorized parameterizations [Saxe et al., 2014a, Ji and Telgarsky, 2019a, Arora et al., 2019a, Gunasekar et al., 2017b] with recent perspectives on how initialization and parameterization interpolate between "rich" and "kernel" behaviors [Chizat et al., 2019, Woodworth et al., 2020]. A particularly useful feature—formalized for DLNs via a separable gradient-flow potential—is that the *scale of the initialization*, denoted α, *continuously tunes* the implicit bias: small α yields a sparse-leaning geometry (an ℓ_1-like penalty up to logarithmic factors), while large α approaches an ℓ_2^2-type geometry; see the potential Q_α and its limits (Theorem 1 in Woodworth et al. [2020]) and related characterizations in Gunasekar et al. [2017b], Arora et al. [2019a].

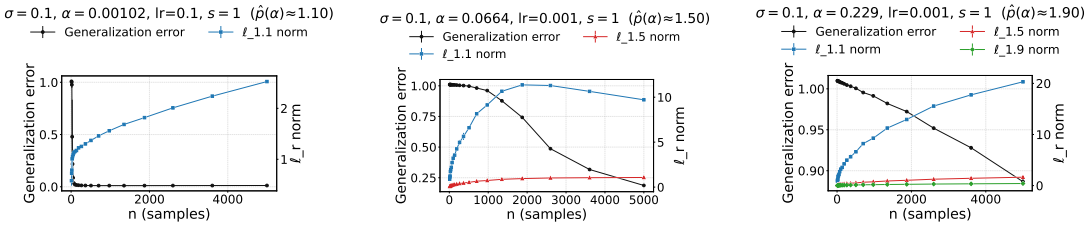
Calibrating α via an effective p. To compare DLN runs with our explicit minimum-ℓ_p experiments, we convert α into an *effective* p by a data-free calibration.

Following the separable potential view, we evaluate Q_α on k-sparse, unit-ℓ_2 probes and fit the log-log slope of its k-dependence; matching that slope to the exact $k^{1-p/2}$ law of $\|\cdot\|_p^p$ yields a monotone map $\alpha \mapsto p_{\text{eff}}(\alpha)$ with limits $p_{\text{eff}}(\alpha) \to 1$ as $\alpha \to 0$ and $p_{\text{eff}}(\alpha) \to 2$ as $\alpha \to \infty$. This calibration is independent of (n, σ) and lets us select α values that span a sparse-to-dense range comparable to $p \in \{1.1, 1.5, 1.9\}$. A full derivation and a visualization of $\alpha \mapsto p_{\text{eff}}(\alpha)$ are provided in Appendix 5.6.

Finite learning rate. With a single-spike target ($w^\star = e_1$, sparsity s=1) and small initialization ($\alpha = 0.00102$, so $p_{\text{eff}} \approx 1.10$), we find that the learning rate lr can materially change the ℓ_r-vs-n scaling once label noise is present. When σ=0, the $\ell_{1.1}$ curve rapidly plateaus and is essentially insensitive to lr (see Appendix 5.8 for more details). In contrast, for $\sigma \in \{0.1, 0.5\}$ increasing lr produces a steadily rising $\ell_{1.1}$ and shifts the elbow to larger n; at the highest noise the effect is strongest-lr=10^{-1} yields monotone growth across our range, whereas lr=10^{-3} exhibits a transient rise followed by relaxation toward a plateau, indicating a rightward-moving elbow. We observe qualitatively similar trends for larger sparsity (s=50). A natural explanation is that finite step size together with noisy gradients turns (stochastic) gradient descent into a noisy dynamical system with an *effective temperature* that scales with lr and the noise level. The resulting diffusion broadens the stationary distribution and biases the predictor toward rounder (less sparse) geometries-effectively increasing p_{eff}-so mass leaks into bulk coordinates, delaying spike dominance and inflating ℓ_r before the eventual plateau [Mandt et al., 2017, Smith et al., 2018, Yaida, 2018, Jastrzebski et al., 2017a].

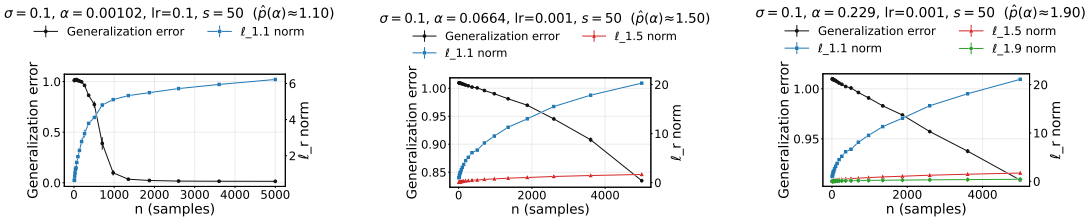
Experimental protocol. We set $\sigma = 0.1$, sweep $\alpha \in \{0.00102, 0.0664, 0.229\}$ (which according to our α to p calibration $\approx p \in \{1.1, 1.5, 1.9\}$), and vary n. Each plot overlays test MSE (left axis) and representative ℓ_r curves (right axis). For flat w^* again $a = \frac{1}{\sqrt{s}}$. Additional noise sweeps are reported in Appendix 5.7.

Because α has been empirically calibrated to $p_{\text{eff}}(\alpha)$, the DLN panels mirror the *scaling* behavior seen with explicit minimum-ℓ_p: for $w^\star = e_1$ (Fig. 5.3), smaller α (smaller p_{eff}) enters the spike-dominated regime earlier so that, for $r > 2(p-1)$,



(a) $\alpha = 0.00102$, lr $= 0.1$, s=1 **(b)** $\alpha = 0.0664$, lr $= 0.001$, s=1 **(c)** $\alpha = 0.229$, lr $= 0.001$, s=1

Figure 5.3: Single spike $w^\star = e_1$; diagonal linear network (DLN). After calibrating α to $p_{\text{eff}}(\alpha)$, the regime structure matches the explicit p case: smaller α exhibits earlier spike dominance and plateaus for $r > 2(p-1)$; larger α remains bulk-dominated with $n^{1/2}$-like growth.



(a) $\alpha = 0.00102$, lr $= 0.1$, s=50 **(b)** $\alpha = 0.0664$, lr $= 0.001$, s=50 **(c)** $\alpha = 0.229$, lr $= 0.001$, s=50

Figure 5.4: Flat $w^\star$ ($s = 50$); diagonal linear network (DLN). The same scaling rules hold, but the elbow appears at larger n—in line with the flat-support transition scale—while absolute ℓ_r magnitudes remain comparable to the single-spike case.

the ℓ_r curves flatten after the transition; larger α remains bulk-dominated longer and the traces grow with the characteristic $n^{1/2}$ trend. For the flat target with s=50 (Fig. 5.4), the same rules apply but the elbow shifts to larger n, consistent with the s-dependent transition scale in the flat-support corollary. The absolute magnitudes of $\|\widehat{w}\|_r$ are similar across the two targets, as predicted by the plateau formulas, so the informative contrast again lies in the *location* of the elbow and the presence/absence of plateaus vs. bulk growth. We do not overlay theory on the DLN plots: our guarantees are stated in terms of the explicit parameter p, and deriving a closed-form α-indexed analogue (especially under finite learning rates) is outside the scope of this work; the $\alpha \mapsto p_{\text{eff}}$ calibration serves precisely to make the scaling correspondence visible. In Appendix D.1 we discuss how can we extend our main theorem to DLNs with explicit α.

5.5 Conclusion and discussion

We provided the first unified, closed-form characterization of how the entire family of norms $\{\|\widehat{w}_p\|_r\}_{r\in[1,p]}$ scales with sample size in overparameterized linear regression under minimum-ℓ_p interpolation ($p \in (1, 2]$). A one-dimensional dual-ray argument exposes a competition between a signal *spike* and a *bulk* of null coordinates in $X^\top Y$ and yields, with high probability: (i) a data-dependent elbow $n_\star$ at which bulk and spike balance [Eq. 5.3], and (ii) a universal threshold

$$r_\star \;=\; 2(p-1),$$

which separates ℓ_r's that ultimately plateau ($r > r_\star$) from those that continue to grow with an explicit exponent ($r \le r_\star$) in the spike-dominated regime (Theorem 11). The formulas give plateau levels and slopes in both bulk- and spike-dominated regimes, and specialize cleanly for canonical targets (single spike and flat support). Empirically, diagonal linear networks (DLNs) trained by gradient descent inherit the same elbow/threshold laws once the initialization scale α is calibrated to an effective $p_{\text{eff}}(\alpha)$ via the separable potential. Together, these results show that which ℓ_r one tracks matters: for a fixed ℓ_p bias, different ℓ_r's can exhibit qualitatively different n-laws.

Intuition behind the regime transition. The dual-ray lens reduces the interpolation geometry to a single scale $t_\star$ controlled by $\|X^\top Y\|_q$ ($q = p/(p-1)$). The *bulk* contributes $\asymp (d-s)\, m_q\, \tau_s^q n^{q/2}$ while the *spike* contributes $\asymp n^q W_q$, and their balance sets the elbow $n_\star$. Above the elbow, the KKT map raises correlations to the $(q-1)$ power; the sign of $\frac{1}{r} - \frac{1}{2(p-1)}$ dictates whether the bulk-type term decays (plateau) or grows (slope). This is the origin of the sharp threshold $r_\star = 2(p-1)$. Geometrically, smaller p (sparser inductive bias) lowers $r_\star$, so more ℓ_r's plateau once the spike dominates; as $p \uparrow 2$, $r_\star$ approaches 2 and spike-side plateaus recede, consistent with the special role of $p = 2$ where there is no n-driven transition in the proportional limit.

Implications for generalization proxies. Many diagnostics and bounds in modern learning scale with a parameter norm (or a reparameterization-aware

surrogate). Our results indicate that the predictive power of such proxies is *norm-choice sensitive*. For a given ℓ_p bias, ℓ_r's above $r_\star$ stabilize (after $n_\star$) and can serve as geometry-aligned capacity proxies, while ℓ_r's below $r_\star$ continue to reflect data growth through explicit exponents. In practice, the pair $(n_\star, r_\star)$ acts as a *norm-scaling signature*. Reporting only one norm—often ℓ_2—risks conflating bulk vs. spike effects and can obscure regime changes that are visible in the ℓ_r family.

From explicit to implicit bias. By calibrating DLN initialization via a simple slope-matching map $\alpha \mapsto p_{\text{eff}}(\alpha)$, the empirical DLN curves line up with the explicit minimum-ℓ_p predictions under $p \leftarrow p_{\text{eff}}(\alpha)$. This provides a quantitative bridge between explicit and implicit bias: initialization steers the effective geometry, and the $(n_\star, r_\star)$ structure is inherited. Finite learning rates in the presence of label noise act like an effective temperature, increasing p_{eff} and shifting elbows rightward—consistent with recent views of SGD as a noisy dynamical system.

Relation to double descent and benign overfitting. The bulk-side growth ($\propto n^{1/2}$ in prominent terms) and its eventual handoff to spike control rationalize when increasing n first harms and then helps: early fits draw from many noisy bulk directions (large norms and higher variance), while beyond $n_\star$ the spike dominates and the relevant ℓ_r's plateau. Our explicit exponents and thresholds sharpen this picture and make precise which ℓ_r will display which trend at a given (p, n).

Scope and limitations. Our guarantees assume isotropic Gaussian design, $p \in (1, 2]$, squared loss, and exact interpolation. At $p = 2$ the proportional regime admits no n-driven elbow. The DLN extension uses a data-free calibration to $p_{\text{eff}}(\alpha)$ rather than a fully rigorous, learning-rate-aware theory. Finally, classification losses and non-linear features (beyond DLNs) are outside our formal scope.

Actionable guidance. (i) When using norm-based capacity control, *choose the norm with the geometry*: if training is ℓ_p-biased (explicitly or implicitly), track ℓ_r with $r > 2(p-1)$ to obtain a stable, post-elbow proxy; use $r \leq 2(p-1)$ when one *wants* a readout that continues to reflect data growth. (ii) Empirically estimate $(n_\star, r_\star)$ by

fitting the predicted slopes to a small ℓ_r grid; this gives a compact fingerprint of model-data geometry and a practical meter for bulk vs. spike dominance.

Future directions:

- **Beyond isotropy and Gaussianity.** Extend the dual-ray analysis to anisotropic/sub-Gaussian designs (via whitening) and to heavy-tailed covariates; characterize how $n_\star$ and possibly $r_\star$ deform with the spectrum and tails of X.
- **From DLNs to nonlinear nets.** Replace the power link by depth-dependent implicit links in deep (nonlinear) architectures (e.g., path-norm or neural tangent/feature-learning regimes) and test whether an $r_\star$-type threshold persists.
- **Algorithmic knobs as geometry.** Develop a theory of p_{eff} that accounts for step size, batch size, momentum, and label noise (Langevin/SGD limits), turning these knobs into quantitative geometric parameters with predictions for $(n_\star, r_\star)$.
- **Classification and margins.** Generalize the scaling laws to separable classification with cross-entropy/hinge losses, relating $r_\star$ to margin exponents and the growth/saturation of norm families along max-margin flows.
- **Tighter, r-aware bounds.** Build generalization bounds that track the *family* $\{\|\widehat{w}\|_r\}$ and explicitly incorporate the elbow/threshold structure, connecting to PAC-Bayes and margin-based analyses.
- **Practical diagnostics.** On modern deep models, measure several ℓ_r-style surrogates (e.g., path norms) across data scale to estimate $(n_\star, r_\star)$ and evaluate which norms are reliable predictors of test error across regimes.

Overall, our results advocate replacing the monolithic notion of "the norm" by a *family* view. The elbow $n_\star$ and the threshold $r_\star$ provide simple, interpretable invariants that tie together explicit/implicit bias, data growth, and norm-based generalization measures, and they offer a compact vocabulary for describing—and ultimately controlling—interpolation in high dimensions.

Supplementary Material for Chapter 5

5.6 From initialization scale to an effective ℓ_p: a slope-matching view

Figure 5.5 visualizes the mapping $\alpha \mapsto p_{\text{eff}}(\alpha)$ we use throughout. The construction is data-free (independent of n and σ) and relies only on the gradient-flow potential that characterizes the two-layer DLN implicit bias. Pseudocode can be found in Algorithm 1.

We start from the separable potential

$$Q_\alpha(\beta) = \alpha^2 \sum_{i=1}^{d} q\left(\frac{\beta_i}{\alpha^2}\right), \tag{5.6}$$

$$q(z) = \int_0^z \operatorname{arcsinh}\left(\frac{u}{2}\right) du \;=\; 2 - \sqrt{4+z^2} \;+\; z \operatorname{arcsinh}\left(\frac{z}{2}\right). \tag{5.7}$$

At the coordinate level, letting $\psi_\alpha(t) \equiv \alpha^2\, q(t/\alpha^2)$ gives

$$\psi'_\alpha(t) = \operatorname{arcsinh}\left(\frac{t}{2\alpha^2}\right), \tag{5.8}$$

$$\psi''_\alpha(t) = \frac{1}{\alpha^2\sqrt{4+(t/\alpha^2)^2}} \;=\; \frac{1}{\sqrt{4\alpha^4+t^2}}. \tag{5.9}$$

Asymptotics for q control the limiting geometry (all logs are natural):

$$q(z) = \frac{z^2}{4} - \frac{z^4}{192} + O(z^6), \qquad z \to 0, \tag{5.10}$$

$$q(z) = z(\log z - 1) + 2 - \frac{1}{z} + O\left(\frac{1}{z^3}\right), \qquad z \to \infty. \tag{5.11}$$

Hence Q_α behaves like ℓ_2^2 as $\alpha \to \infty$ and like an ℓ_1-type penalty (up to a log) as $\alpha \to 0$.

To turn this into a quantitative $\alpha \mapsto p$ mapping, we evaluate Q_α on the k-sparse, unit-ℓ_2 probes

$$\beta^{(k)} \in \mathbb{R}^d, \qquad \beta_i^{(k)} \in \{0, k^{-1/2}\}, \qquad \|\beta^{(k)}\|_2 = 1, \qquad \#\{i : \beta_i^{(k)} \neq 0\} = k. \tag{5.12}$$

For this family,

$$Q_\alpha\Big(\beta^{(k)}\Big) = \alpha^2\, k\, q\left(\frac{1}{\alpha^2\sqrt{k}}\right), \tag{5.13}$$

while ℓ_p (calibrated via $\|\beta\|_p^p$) has the exact scaling

$$\|\beta^{(k)}\|_p^p = k\left(\frac{1}{\sqrt{k}}\right)^p \;=\; k^{\,1-\frac{p}{2}}. \tag{5.14}$$

We now fit a log-log slope to the k-dependence of Q_α and match exponents. Fix $\alpha > 0$, choose a logarithmic grid $\mathcal{K} \subset \{1, 2, \ldots, d\}$ (e.g., up to 10^4), and solve

$$\log Q_\alpha\Big(\beta^{(k)}\Big) \approx c(\alpha) + s(\alpha)\, \log k, \qquad k \in \mathcal{K}. \tag{5.15}$$

Comparing with (5.14) (which grows as $k^{\,1-p/2}$) yields

$$s(\alpha) = 1 - \frac{p_{\text{eff}}(\alpha)}{2} \quad\Longrightarrow\quad p_{\text{eff}}(\alpha) = 2(1 - s(\alpha))\,. \tag{5.16}$$

The limits in (5.10)–(5.11) imply

$$\alpha \to \infty: \quad Q_\alpha\Big(\beta^{(k)}\Big) = \frac{1}{4\alpha^2} + O\Big(\frac{1}{\alpha^6 k}\Big), \quad s(\alpha) \to 0, \quad p_{\text{eff}}(\alpha) \to 2, \tag{5.17}$$

$$\begin{aligned} \alpha \to 0: \quad Q_\alpha\Big(\beta^{(k)}\Big) &= \sqrt{k}\left(\log\Big(\tfrac{1}{\alpha^2\sqrt{k}}\Big) - 1\right) + 2\alpha^2 k - \;\alpha^4 k\sqrt{k} \;+\; O\Big(\alpha^8 k^2\sqrt{k}\Big), \\ s(\alpha) &\to \tfrac{1}{2}, \\ p_{\text{eff}}(\alpha) &\to 1. \end{aligned} \tag{5.18}$$

Thus $p_{\text{eff}}(\alpha)$ increases smoothly and monotonically from 1 to 2 as α grows, exactly as depicted in Figure 5.5. The inverse problem—choosing α for a target $p^\star \in [1, 2]$—is the scalar root

$$p_{\text{eff}}(\alpha) = p^\star, \tag{5.19}$$

which we solve by bisection using the monotonicity in α (Algorithm 2).

Algorithm 1 Slope-matching map $\alpha \mapsto p_{\text{eff}}(\alpha)$

Require: Log-grid $\mathcal{A}$ of α values; log-grid $\mathcal{K} \subset \{1, \dots, d\}$ of k values
Ensure: $\{(\alpha, p_{\text{eff}}(\alpha)) : \alpha \in \mathcal{A}\}$
for all $\alpha \in \mathcal{A}$ **do**
  Initialize lists $X \leftarrow [\,]$, $Y \leftarrow [\,]$ ▷ $X = \{\log k\}$, $Y = \{\log Q_\alpha(\beta^{(k)})\}$
  for all $k \in \mathcal{K}$ **do**
    $z_k \leftarrow 1/(\alpha^2\sqrt{k})$
    Compute $q(z_k)$ using the closed form in (5.7); if $|z_k|$ is small, use the series $q(z) = z^2/4 - z^4/192 + z^6/2560 + \cdots$ for stability
    $Q_k \leftarrow \alpha^2\, k\, q(z_k)$
    Append $\log k$ to X; append $\log Q_k$ to Y
  end for
  Fit $Y \approx c(\alpha) + s(\alpha)\, X$ by least squares
  $p_{\text{eff}}(\alpha) \leftarrow 2\,(1 - s(\alpha))$ ▷ by (5.16)
end for
return $\{(\alpha, p_{\text{eff}}(\alpha)) : \alpha \in \mathcal{A}\}$

Algorithm 2 Inverse map $p^\star \mapsto \alpha^\star$ by bisection in $\log \alpha$

Require: Target $p^\star \in [1, 2]$; grid $\mathcal{K}$; bracket $0 < \alpha_{\min} < \alpha_{\max}$ with $p_{\text{eff}}(\alpha_{\min}) \le p^\star \le p_{\text{eff}}(\alpha_{\max})$; tolerance $\varepsilon > 0$
Ensure: $\alpha^\star$ with $\left|p_{\text{eff}}(\alpha^\star) - p^\star\right| \le \varepsilon$
$u_{\min} \leftarrow \log \alpha_{\min}$, $u_{\max} \leftarrow \log \alpha_{\max}$
while $u_{\max} - u_{\min} > \varepsilon$ **do**
  $u_{\text{mid}} \leftarrow \frac{1}{2}(u_{\min} + u_{\max})$, $\alpha_{\text{mid}} \leftarrow e^{u_{\text{mid}}}$
  Compute $p_{\text{eff}}(\alpha_{\text{mid}})$ via Algorithm 1 restricted to this single α
  if $p_{\text{eff}}(\alpha_{\text{mid}}) < p^\star$ **then**
    $u_{\min} \leftarrow u_{\text{mid}}$
  else
    $u_{\max} \leftarrow u_{\text{mid}}$
  end if
end while
return $\alpha^\star \leftarrow e^{(u_{\min} + u_{\max})/2}$

5.7 Additional noise sweeps: $\sigma \in \{0, 0.5\}$

Experimental protocol. We replicate the experiments of §5.4.3 and §5.4.4 at two additional noise levels, $\sigma = 0$ and $\sigma = 0.5$, keeping everything else fixed (same $p \in \{1.1, 1.5, 1.9\}$ for explicit minimum-ℓ_p runs; same $\alpha \in \{0.00102, 0.0664, 0.229\}$ for DLNs with the same $\alpha \mapsto p_{\text{eff}}$ calibration as in Appendix 5.6; same seeds and learning rates as indicated in the panel captions). Each plot overlays test MSE (left axis) and representative ℓ_r curves (right axis).

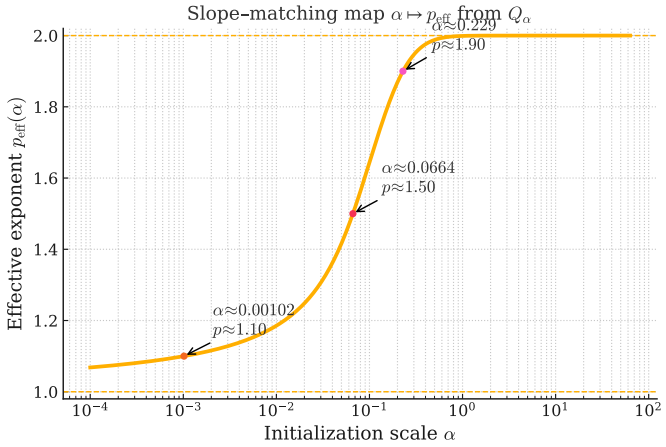
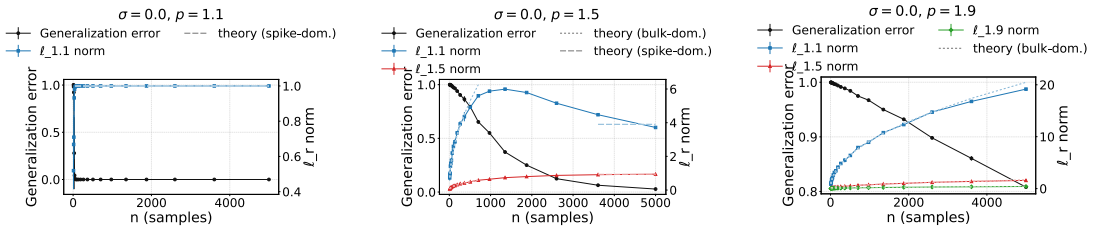


Figure 5.5: Slope-matching map $\alpha \mapsto p_{\text{eff}}(\alpha)$ (Algorithm 1), obtained by fitting the k-sparse scaling of $Q_\alpha(\beta^{(k)})$ against the exact $k^{1-p/2}$ scaling of $\|\beta^{(k)}\|_p^p$. Target points ($p \in \{1.1, 1.5, 1.9\}$) are annotated; their corresponding α are solved by Algorithm 2.



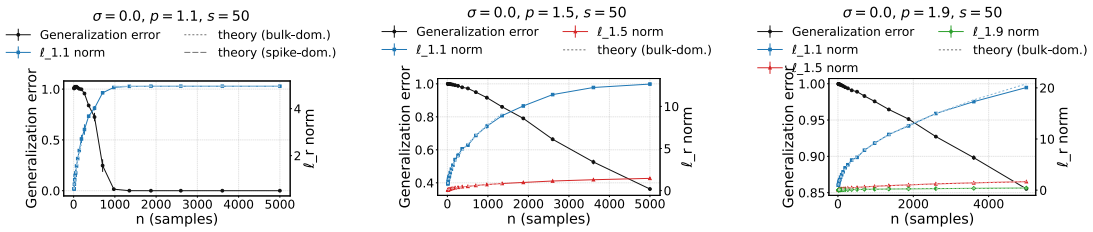
(a) $p = 1.1$ (sparsity-leaning) **(b)** $p = 1.5$ **(c)** $p = 1.9$ (dense-leaning)

Figure 5.6: Single spike $w^\star = e_1$; explicit minimum-ℓ_p interpolation ($\sigma = 0$). Earlier elbows and lower spike-side plateaus than at σ=0.1; bulk-side traces keep the $n^{1/2}$ slope, consistent with (5.4)-(5.5).

What the figures show and why. In Fig. 5.6-Fig. 5.13, the slopes and regime rules from Theorem 11 and Corollaries 5.4.1-5.4.2 are unchanged across σ; noise only rescales τ_s and thereby shifts the transition size $n_\star \asymp (\kappa_{\text{bulk}} \tau_s^q / W_q)^{2/(q-2)}$ [(5.3)] and the spike-side plateau levels [(5.4)]. Thus, compared to σ=0.1 in the main text: (i) at σ=0 elbows appear earlier and plateaus (for $r > 2(p-1)$) occur sooner and at lower levels; (ii) at σ=0.5 elbows are delayed and spike-side plateaus are higher. Bulk-dominated panels retain the $n^{1/2}$ growth and the r-ordering in (5.5).

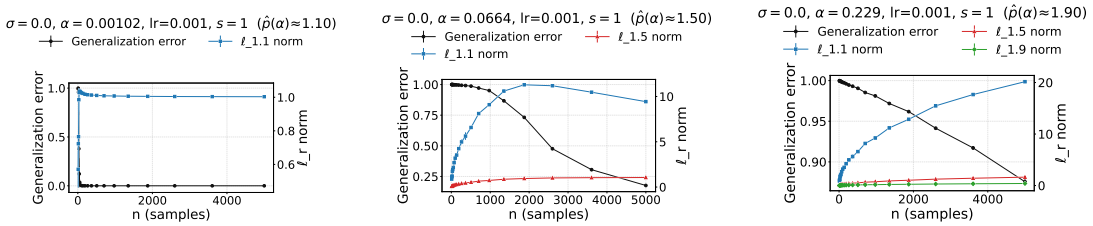
5.8 Finite learning rate effects

We consider the single-spike case $w^\star = e_1$ and a small shape parameter $\alpha = 0.00102$ (so the calibrated $p_{\text{eff}}(\alpha) \approx 1.10$). We vary the learning rate $\text{lr} \in \{10^{-1}, 10^{-2}, 10^{-3}\}$



(a) $p = 1.1$ (sparsity-leaning) **(b)** $p = 1.5$ **(c)** $p = 1.9$ (dense-leaning)

Figure 5.7: Flat $w^\star$ ($s = 50$); explicit minimum-ℓ_p interpolation ($\sigma = 0$). Same slope/plateau rules as Corollary 5.4.2, with a reduced transition scale and lower absolute ℓ_r levels compared to σ=0.1.



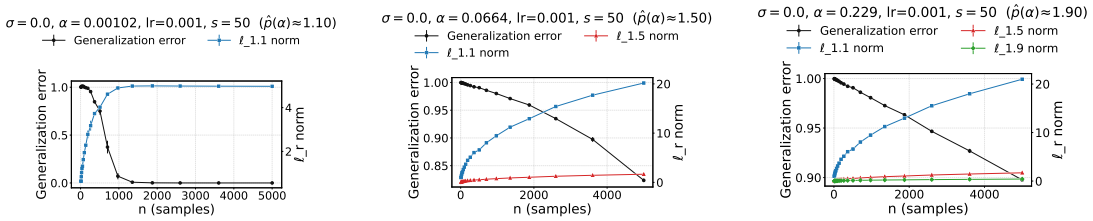
(a) $\alpha = 0.00102$, lr $= 0.001$, s=1 **(b)** $\alpha = 0.0664$, lr $= 0.001$, s=1 **(c)** $\alpha = 0.229$, lr $= 0.001$, s=1

Figure 5.8: Single spike $w^\star = e_1$; DLN ($\sigma = 0$). With α calibrated to $p_{\text{eff}}(\alpha)$, the regime structure mirrors the explicit p case: smaller p_{eff} exhibits earlier spike dominance and plateaus for $r > 2(p-1)$; larger p_{eff} stays bulk-dominated longer.

and the label-noise level $\sigma \in \{0, 0.1, 0.5\}$. All panels plot generalization error (left axis) and $\ell_{1.1}$ norm (right axis) versus sample size n.

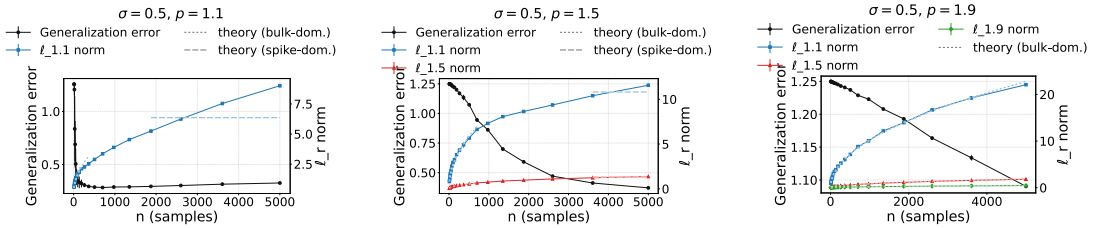
Observed effect. With **clean labels** ($\sigma = 0$), the $\ell_{1.1}$ norm is essentially flat across n and insensitive to lr (Fig. 5.14), consistent with a low-p_{eff} (sparse) implicit bias at small α. When **label noise is present** ($\sigma \in \{0.1, 0.5\}$), increasing the learning rate makes $\ell_{1.1}$ *increase with* n (Figs. 5.15, 5.16); the transition point (the "elbow") beyond which the norm would plateau shifts to larger n as lr grows. Within the accessible sample sizes this rightward shift makes the curve look bulk-dominated and rising—*as if* the effective exponent p_{eff} were larger.

Why this happens. Finite step size together with label/gradient noise injects additional stochasticity into the discrete dynamics. A useful approximation views (stochastic) gradient descent as a Langevin-type process with an *effective temperature* controlled by the learning rate and the noise level; this broadens the stationary



(a) $\alpha = 0.00102$, lr $= 0.001$,**(b)** $\alpha = 0.0664$, lr $= 0.001$, s=50 **(c)** $\alpha = 0.229$, lr $= 0.001$, s=50 s=50

Figure 5.9: Flat $w^\star$ ($s = 50$); DLN ($\sigma = 0$). The elbow shifts with support size as in the flat-support corollary; plateaus for $r > 2(p-1)$ occur earlier and at lower levels than at σ=0.1, while bulk-side $n^{1/2}$ growth persists where predicted.



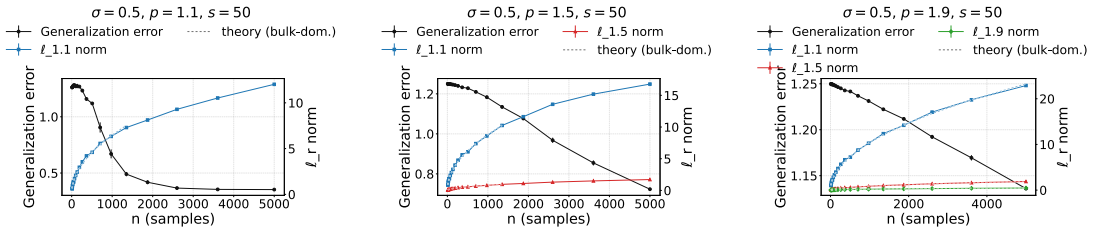
(a) $p = 1.1$ (sparsity-leaning) **(b)** $p = 1.5$ **(c)** $p = 1.9$ (dense-leaning)

Figure 5.10: Single spike $w^\star = e_1$; explicit minimum-ℓ_p interpolation ($\sigma = 0.5$). Larger τ increases both $n_\star$ and plateau heights relative to σ=0.1. Bulk-dominated panels retain the $n^{1/2}$ trend; $r > 2(p-1)$ traces flatten only after the later transition, in line with (5.4)-(5.5).

distribution and leads to wider, less sparse solutions [Mandt et al., 2017, Smith et al., 2018, Yaida, 2018, Jastrzebski et al., 2017a]. For a single-spike target, that diffusion leaks mass into off-signal coordinates during early training, nudging the geometry away from "ℓ_1-like" toward a higher-p regime and delaying when the spike dominates—hence the elbow shifts right. With **clean labels**, the gradient remains aligned with the spike and the small-step implicit bias toward path/diagonal-norm solutions is recovered [Neyshabur et al., 2015a, Gunasekar et al., 2018a]. The same qualitative phenomenon also appears for the denser case s=50 with a smaller magnitude.

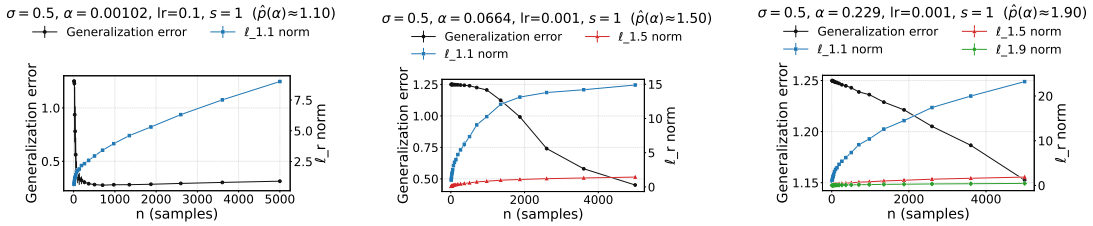
5.9 Larger sparsity s for explicit $\min \|w\|_p$ linear regression

We revisit the explicit $\min \|w\|_p$ experiments at larger sparsities $s \in \{500, 5000\}$ for $p \in \{1.1, 1.5, 1.9\}$ under the same Gaussian design and noise $\sigma = 0.1$ as in



(a) $p = 1.1$ (sparsity-leaning) **(b)** $p = 1.5$ **(c)** $p = 1.9$ (dense-leaning)

Figure 5.11: Flat $w^\star$ ($s = 50$); explicit minimum-ℓ_p interpolation ($\sigma = 0.5$). The same slope/plateau rules apply, but both the elbow and plateau heights shift upward with σ via τ_s and (5.3).

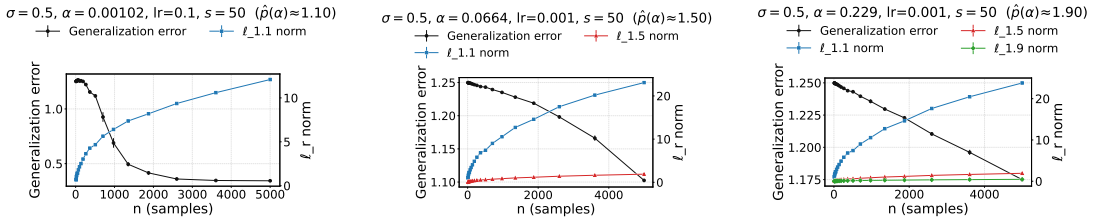


(a) $\alpha = 0.00102$, lr $= 0.1$, s=1 **(b)** $\alpha = 0.0664$, lr $= 0.001$, s=1 **(c)** $\alpha = 0.229$, lr $= 0.001$, s=1

Figure 5.12: Single spike $w^\star = e_1$; DLN ($\sigma = 0.5$). After calibrating $\alpha \mapsto p_{\text{eff}}$, bulk growth persists to larger n (larger $n_\star$), and spike-side plateaus for $r > 2(p-1)$ emerge later and at higher levels.

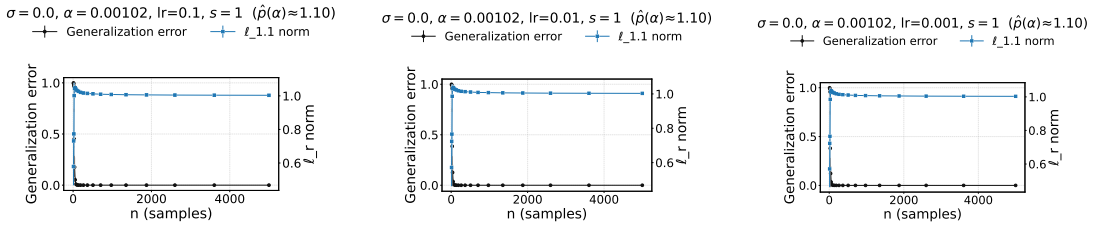
the main text. Each panel reports generalization error (left axis) and several ℓ_r-norms of the *same* interpolating w (right axis); gray dashed curves are the bulk/spike theory overlays used earlier.

Comparison to s=50. Across all three p values, the larger-s experiments reprise the main-text regime structure at larger sample sizes. For $p \approx 1$, lengthening the bulk-dominated segment makes the initial *increase* in generalization error clearly visible (especially at s=5000), after which the curve turns downward as alignment improves. For $p \in \{1.5, 1.9\}$, the same right-shift occurs yet the curves remain monotone; the rounder objectives keep the estimator from over-relying on noisy directions early on. In every panel, the blue $\ell_{1.1}$ curve remains a useful "regime meter": rapid growth signals bulk influence, and gradual approach toward the spike guide signals improving alignment—even though none of the ℓ_r curves truly flatten within our plotted range.



(a) $\alpha = 0.00102$, lr $= 0.1$, s=50 **(b)** $\alpha = 0.0664$, lr $= 0.001$, s=50 **(c)** $\alpha = 0.229$, lr $= 0.001$, s=50

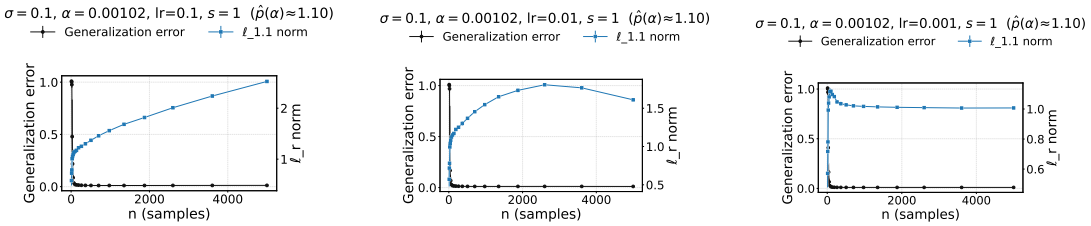
Figure 5.13: Flat $w^\star$ ($s = 50$); DLN ($\sigma = 0.5$). The σ-driven increase in τ_s shifts $n_\star$ to larger n; otherwise the bulk vs. spike regime behavior matches the theory and the explicit p experiments.



(a) $\sigma = 0$, $\alpha = 0.00102$, lr $= 0.1$, s=1 **(b)** $\sigma = 0$, $\alpha = 0.00102$, lr $= 0.01$, s=1 **(c)** $\sigma = 0$, $\alpha = 0.00102$, lr $= 0.001$, s=1

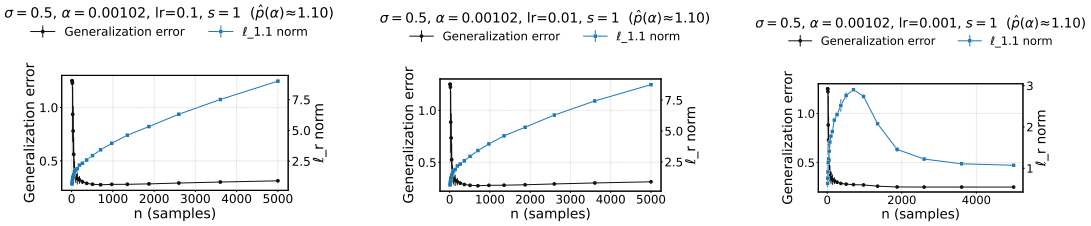
Figure 5.14: $w^\star = e_1$ (sparsity s=1), clean labels. $\ell_{1.1}$ rapidly plateaus and is insensitive to learning rate, consistent with a low-p_{eff} implicit bias at small α.

Small p (here p=1.1). Relative to the s=50 panels in the main text, both larger-s slices preserve the same two-phase story but the handoff happens later in n. At s=500 (Fig. 5.17a), generalization error is flat-to-slightly higher at small n while $\|w\|_{1.1}$ rises rapidly; as n grows, generalization error begins to fall and the blue curve bends toward (but, in our range, does not meet) the spike overlay. At s=5000 (Fig. 5.18a), the shape is unmistakable: generalization error *first increases* to a visible peak at intermediate n and then drops. The $\ell_{1.1}$ curve keeps climbing throughout the displayed range, tracking the bulk-dominated guide before gradually approaching the spike prediction (without flattening). This "up-then-down" with more samples matches the double-descent picture for interpolating estimators—early fits lean on high-variance bulk directions; only later does the solution align with signal—well documented in linear and deep settings [Belkin et al., 2019, Nakkiran et al., 2020b, Hastie et al., 2022a].



(a) $\sigma = 0.1$, $\alpha = 0.00102$, lr = 0.1, s=1 **(b)** $\sigma = 0.1$, $\alpha = 0.00102$, lr = 0.01, s=1 **(c)** $\sigma = 0.1$, $\alpha = 0.00102$, lr = 0.001, s=1

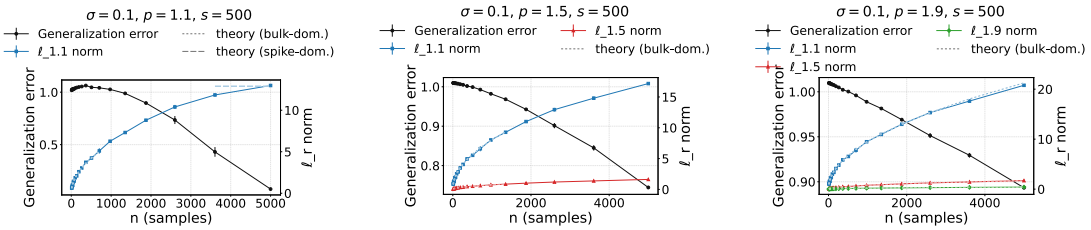
Figure 5.15: $w^\star = e_1$ **(sparsity** s=1**), moderate noise.** Larger learning rates produce a steadily rising $\ell_{1.1}$ and shift the elbow to larger n; decreasing lr suppresses the rise and restores a near-plateau.



(a) $\sigma = 0.5$, $\alpha = 0.00102$, lr = 0.1, s=1 **(b)** $\sigma = 0.5$, $\alpha = 0.00102$, lr = 0.01, s=1 **(c)** $\sigma = 0.5$, $\alpha = 0.00102$, lr = 0.001, s=1

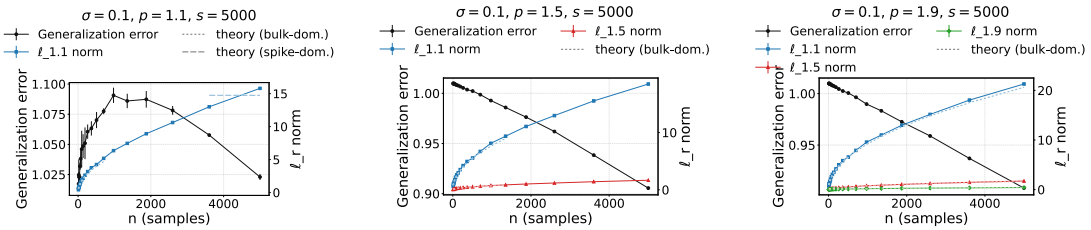
Figure 5.16: $w^\star = e_1$ **(sparsity** s=1**), heavy noise.** The learning-rate-induced increase in $\ell_{1.1}$ is strongest at high noise: lr=0.1 (and to a lesser extent 0.01) yields monotone growth with n, whereas lr=0.001 shows a transient bump and then relaxes toward a plateau—evidence that the elbow shifts right under larger lr.

Larger p **(here** p=1.5 **and** p=1.9**).** Compared to s=50, the curves again shift rightward in n, but the qualitative picture is unchanged: generalization error decreases *monotonically* over the whole range for both sparsities (Figs. 5.17b-c and 5.18b-c). The minimized ℓ_p-norms (red for p=1.5, green for p=1.9) drift only slightly upward rather than plateauing, while the auxiliary $\ell_{1.1}$ diagnostic continues its steady growth along the bulk guide. The absence of an initial increase in generalization error is consistent with the rounder geometry of larger-p balls: the interpolating solution spreads weight more evenly and avoids the brittle, variance-heavy fits that create the small-p bump, echoing analyses of benign overfitting/ridgeless least squares and convex-geometric shrinkage of descent cones [Bartlett et al., 2020, Hastie et al., 2022a, Chandrasekaran et al., 2012, Amelunxen et al., 2014].



(a) p=1.1, s=500 **(b)** p=1.5, s=500 **(c)** p=1.9, s=500

Figure 5.17: Large sparsity, s=500. Black—generalization error; colored—ℓ_r-norms of the same interpolator (blue: $\ell_{1.1}$, red: $\ell_{1.5}$, green: $\ell_{1.9}$); gray dashed—bulk/spike overlays.



(a) p=1.1, s=5000 **(b)** p=1.5, s=5000 **(c)** p=1.9, s=5000

Figure 5.18: Even larger sparsity, s=5000. Same conventions as Fig. 5.17. Increasing s shifts the bulk→spike crossover to larger n.

6
Conclusion

Modern deep learning systems succeed because they often generalize even when heavily overparameterized. This thesis has argued that to understand and reliably measure that generalization, it is more fruitful to reason in function space and to insist on diagnostics that are invariant to benign reparameterizations and stable under routine training changes. The three studies assembled here move along a single arc: from interrogating geometric intuitions in parameter space, through auditing the fragility of popular surrogates, to deriving explicit scaling laws that explain when and why norm-based proxies can help or mislead.

The first study showed that flatness, while often correlated with test performance, cannot be treated as a universal yardstick. Simple rescalings and optimizer choices can push flatness measures to arbitrarily different values without changing what the network computes, whereas a function-space quantity—the prior over functions, or $\log P(f)$—tracks generalization robustly across architectures and training algorithms. This contrast highlights a central theme of the thesis: good diagnostics should respect the symmetries of the predictor and live where prediction lives, namely in function space rather than in a particular parameterization. Read this way, $\log P(f)$ is valuable not because it is fashionable, but because it encodes invariances that flatness lacks.

The second study widened the lens and asked a pragmatic question: how do popular generalization measures behave under small, reasonable changes to the training pipeline or the task? The answer, made concrete by fragility audits, is that many magnitude-sensitive, post-mortem surrogates bend, invert, or balloon even when accuracy hardly moves, and they often fail to reflect genuine increases in task difficulty. By contrast, a marginal-likelihood route in function space mirrors learning-curve trends as properties of the data, and remains largely indifferent to optimizer path once the training set is interpolated. This suggests a practical stance: before trusting a surrogate, we should stress-test it for reparameterization invariance, optimizer/schedule stability, and sensitivity to data difficulty. When such audits are routine, we learn quickly which diagnostics capture properties of the learned function and which ones mostly measure accidents of the training path.

The third study provided theory that explains why norm-based proxies can behave so differently across regimes. In high-dimensional linear regression with explicit minimum-ℓ_p bias and in diagonal linear networks with implicit bias, we derived closed-form scaling laws for the entire family of parameter norms $\{\|w\|_r\}$ as the sample size grows. Two structural features emerged: a data-dependent transition size (an elbow in n) and a universal threshold $r^\star = 2(p-1)$ that separates norms that eventually plateau from those that continue to grow with explicit exponents. Calibrating initialization in diagonal linear networks reveals the same elbow/threshold pattern through an effective p, clarifying why seemingly similar norm diagnostics can diverge sharply in practice. These results advise care when choosing a norm on which to hang generalization claims: different (r, p) pairs can produce opposite scaling behaviors under the same training pipeline, so the right proxy depends on the operative bias and data geometry.

Taken together, the studies support a coherent picture. If our goal is to anticipate out-of-sample performance, we should prefer diagnostics that are invariant under transformations that preserve the predictor, we should verify stability under benign pipeline changes, and we should ground proxy quantities in explicit, data-aware

scaling behavior. The function-space prior provides one such anchor; fragility-first audits supply a lightweight methodology for vetting alternatives; and scaling laws in simplified but revealing models show how inductive bias, sample size, and data anisotropy shape whole families of proxies. None of these elements is sufficient on its own, but together they offer a robust path from geometric intuition to reliable practice.

Looking ahead, several avenues seem most promising and impactful. First, it would be valuable to build a tighter theoretical bridge from function-space priors (and $\log P(f)$) to generalization at the level of individual predictors, extending beyond existing bounds and clarifying how data, architecture, and algorithm interact. Second, approximate function-space surrogates that retain the right invariances—finite-width corrections to Gaussian-process approximations, amortized evidence estimators, or ensembles interpreted as posteriors over functions—could make invariance-friendly predictors usable in routine training and model selection. Third, fragility audits should be integrated into automated evaluation pipelines and broadened beyond image classification to sequence models, generative modeling, reinforcement learning, and distribution shift, so that stability becomes a first-class criterion across modalities. Finally, it would be instructive to connect the norm-scaling framework to modern deep architectures whose effective p varies across layers and time, testing whether elbow/threshold phenomena can be detected and exploited for better diagnostics during training.

In summary, this thesis invites a shift in emphasis: from surrogate measures that happen to correlate with accuracy in narrow settings, to principled diagnostics grounded in invariance, stability, and explicit scaling. By moving the conversation into function space and by auditing what our measures truly respond to, we can make generalization assessment both more interpretable and more reliable.

Bibliography

Ben Adlam and Jeffrey Pennington. Understanding double descent requires a fine-grained bias-variance decomposition. *Advances in neural information processing systems*, 33: 11022–11032, 2020.

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019.

Mauricio A Alvarez, Lorenzo Rosasco, and Neil D Lawrence. Kernels for vector-valued functions: A review. *arXiv preprint arXiv:1106.6251*, 2011.

Dennis Amelunxen, Miroslav Lotz, Michael B. McCoy, and Joel A. Tropp. Living on the edge: Phase transitions in convex programs with random data. *Information and Inference: A Journal of the IMA*, 3(3):224–294, 2014. URL https://doi.org/10.1093/imaiai/iau005.

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, pages 639–668, 2022. URL https://proceedings.mlr.press/v162/andriushchenko22a/andriushchenko22a.pdf.

Martin Anthony and Peter L. Bartlett. *Neural Network Learning - Theoretical Foundations*. 2002. URL http://www.cambridge.org/gb/knowledge/isbn/item1154061/?site_locale=en_GB.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 254–263. PMLR, 10–15 Jul 2018a. URL http://proceedings.mlr.press/v80/arora18b.html.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 10–15 Jul 2018b. doi: 10.48550/arxiv.1802.05296. URL https://doi.org/10.48550/arxiv.1802.05296.

Sanjeev Arora, Zhiyuan Li, and Kaifeng Lyu. Theoretical analysis of auto rate-tuning by batch normalization. *arXiv preprint arXiv:1812.03981*, 2018c. URL https://arxiv.org/abs/1812.03981.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019a. URL https://arxiv.org/abs/1905.13655.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019b. URL https://arxiv.org/abs/1901.08584.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. *arXiv preprint arXiv:1706.05394*, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 2001. doi: 10.1007/3-540-44581-1_15. URL https://doi.org/10.1007/3-540-44581-1_15.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017a. URL https://arxiv.org/abs/1706.08498.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017b. doi: 10.48550/arxiv.1706.08498. URL https://doi.org/10.48550/arxiv.1706.08498.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017c.

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. URL https://doi.org/10.1073/pnas.1907378117.

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*, 116(32):15849–15854, 2019.

Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009. URL https://doi.org/10.1214/08-AOS620.

Léonard Blier and Yann Ollivier. The description length of deep learning models. *arXiv preprint arXiv:1802.07044v5*, 2018. URL https://arxiv.org/abs/1802.07044v5.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

Leo Breiman. Reflections after refereeing papers for nips. In *The Mathematics of Generalization*, pages 11–15. Addison-Wesley, 1995.

Peter Bühlmann and Sara van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, 2011. URL https://doi.org/10.1007/978-3-642-20192-9.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

Emmanuel J. Candès and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 10836–10846, 2019.

Olivier Catoni. *PAC-Bayes Bounds for Supervised Classification*. 2007. URL https://doi.org/10.1007/978-3-319-21852-6_20.

Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012. URL https://doi.org/10.1007/s10208-012-9130-3.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001. URL https://doi.org/10.1137/S003614450037906X.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 2933–2943, 2019. URL https://arxiv.org/abs/1812.07956. arXiv:1812.07956.

Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. *Advances in neural information processing systems*, 22:342–350, 2009.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*. PMLR, 2017. URL https://proceedings.mlr.press/v70/cisse17a.html.

Jeremy Cohen, John Duchi, Ramin Hasani, et al. The edge of stability: Test loss can be lower than training loss and nonconvexity. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2103.06886.

Amit Daniely and Elad Granot. Generalization bounds for neural networks via approximate description length. *Advances in Neural Information Processing Systems*, 32, 2019.

Giacomo De Palma, Bobak Toussi Kiani, and Seth Lloyd. Random deep neural networks are biased towards simple functions. *arXiv preprint arXiv:1812.10156*, 2018.

Kamaludin Dingle, Steffen Schaper, and Ard A Louis. The structure of the genotype–phenotype map strongly constrains the evolution of non-coding rna. *Interface focus*, 5(6):20150053, 2015.

Kamaludin Dingle, Chico Q Camargo, and Ard A Louis. Input–output maps are strongly biased towards simple outputs. *Nature Communications*, 9(1):1–7, 2018.

Kamaludin Dingle, Guillermo Valle Pérez, and Ard A Louis. Generic predictions of output probability based on complexities of inputs and outputs. *Scientific Reports*, 10 (1):1–9, 2020.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1019–1028. JMLR. org, 2017a.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*, 2017b. URL https://arxiv.org/abs/1703.04933.

Konstantin Donhauser, Nicolo Ruggeri, Stefan Stojanovic, and Fanny Yang. Fast rates for noisy interpolation require rethinking the effect of inductive bias. In *International Conference on Machine Learning*, pages 5397–5428. PMLR, 2022. URL https://proceedings.mlr.press/v162/donhauser22a.html.

David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52 (4):1289–1306, 2006. URL https://doi.org/10.1109/TIT.2006.871582.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul): 2121–2159, 2011.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *arXiv preprint arXiv:1703.11008*, 2017a.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*, 2017b. URL http://auai.org/uai2017/proceedings/papers/173.pdf.

Gintare Karolina Dziugaite and Daniel M Roy. Data-dependent pac-bayes priors via differential privacy. In *Advances in Neural Information Processing Systems*, pages 8430–8441, 2018.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a. doi: 10.48550/arxiv.2010.11924. URL https://doi.org/10.48550/arxiv.2010.11924.

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. *arXiv preprint arXiv:2010.11924*, 2020b. URL https://arxiv.org/abs/2010.11924.

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. *arXiv preprint arXiv:2010.11924*, 2020c.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004. URL https://doi.org/10.1214/009053604000000067.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020a.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020b. URL https://arxiv.org/abs/2010.01412.

IE Frank and JH Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993. URL https://doi.org/10.1080/00401706.1993.10485033.

Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep convolutional networks as shallow gaussian processes. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bklfsi0cKm.

Michael Gastpar, Ido Nachum, Jonathan Shafer, and Thomas Weinberger. Fantastic generalization measures are nowhere to be found. In *International Conference on Learning Representations (ICLR)*, 2023. doi: 10.48550/arxiv.2309.13658. URL https://doi.org/10.48550/arxiv.2309.13658.

Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien. Pac-bayesian theory meets bayesian inference. In *Advances in Neural Information Processing Systems*, pages 1884–1892, 2016.

Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modelling the influence of data structure on learning in neural networks. *arXiv preprint arXiv:1909.11500*, 2019.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 297–299. PMLR, 06–09 Jul 2018a. URL http://proceedings.mlr.press/v75/golowich18a.html.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299, 2018b. URL http://proceedings.mlr.press/v75/golowich18a/golowich18a.pdf.

Antoine Gonon, Nicolas Brisebarre, Elisa Riccietti, and Rémi Gribonval. A path-norm toolkit for modern networks: consequences, promises and challenges. *arXiv preprint arXiv:2310.01225*, 2023. URL https://arxiv.org/abs/2310.01225. ICLR 2024 Spotlight.

Yehoram Gordon. Some inequalities for gaussian processes and applications. *Israel Journal of Mathematics*, 50(4):265–289, 1985. doi: 10.1007/BF02764726.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J. Cree. Regularisation of neural networks by enforcing lipschitz continuity. In *International Conference on Learning Representations (ICLR)*, 2020. doi: 10.1007/s10994-020-05929-w. URL https://doi.org/10.1007/s10994-020-05929-w.

Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. Sgd: General analysis and improved rates. In *International conference on machine learning*, pages 5200–5209. PMLR, 2019.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Suriya Gunasekar, Prateek Jain, Daniel Soudry, Sham M. Kakade, and Nathan Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017a. URL https://arxiv.org/abs/1705.09280.

Suriya Gunasekar, Blake E. Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, 2017b. URL https://arxiv.org/abs/1705.09280.

Suriya Gunasekar, Daniel Soudry, Mor Nacson, and Nathan Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018a. URL https://arxiv.org/abs/1806.00468.

Suriya Gunasekar, Blake E. Woodworth, Sham Kakade, and Nathan Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018b. URL https://arxiv.org/abs/1806.00468.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234, 2016.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015. URL https://doi.org/10.1201/b18401.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2): 949–986, 2022a. URL https://doi.org/10.1214/21-AOS2108. Earlier version: arXiv:1903.08560 (2019).

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Proceedings of the National Academy of Sciences*, 119(28):e2101426119, 2022b. URL https://doi.org/10.1073/pnas.2101426119.

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.

Fredrik Hellström, Giuseppe Durisi, Benjamin Guedj, and Maxim Raginsky. Generalization bounds: Perspectives from information theory and pac-bayes. *arXiv preprint arXiv:2309.04381*, 2023.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Geoffrey E Hinton and Drew van Camp. Keeping neural networks simple. In *International Conference on Artificial Neural Networks*, pages 11–18. Springer, 1993.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997a.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997b. doi: 10.1162/neco.1997.9.1.1. URL https://doi.org/10.1162/neco.1997.9.1.1.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634. URL https://doi.org/10.1080/00401706.1970.10488634.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in neural information processing systems*, pages 1731–1741, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018a.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018b. URL https://arxiv.org/abs/1803.05407.

Arthur Paul Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, 2018. doi: 10.48550/arxiv.1806.07572. URL https://doi.org/10.48550/arxiv.1806.07572.

Cheongjae Jang, Sungyoon Lee, Frank Park, and Yung-Kyun Noh. A reparametrization-invariant sharpness measure based on information geometry. *Advances in neural information processing systems*, 35:27893–27905, 2022.

Stanisław Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017a. URL https://arxiv.org/abs/1711.04623.

Stanislaw Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos J Storkey. Finding flatter minima with sgd. In *ICLR (Workshop)*, 2018.

Stanisław Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. In *International Conference on Artificial Intelligence and Statistics (AISTATS) Workshop*, 2017b. doi: 10.48550/arxiv.1711.04623. URL https://doi.org/10.48550/arxiv.1711.04623.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations (ICLR)*, 2019a. URL https://openreview.net/forum?id=H1gBviAqYQ. arXiv:1810.02032.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations (ICLR)*, 2019b. URL https://arxiv.org/abs/1810.02032.

Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 33:17176–17186, 2020.

Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. *arXiv preprint arXiv:1810.00113*, 2018.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019a.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019b. URL https://arxiv.org/abs/1912.02178.

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. *arXiv preprint arXiv:2001.08361*, 2020.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017. URL https://openreview.net/forum?id=H1oyRlYgg.

SungYub Kim, Sihwan Park, Kyungsu Kim, and Eunho Yang. Scale-invariant bayesian neural networks with connectivity tangent kernel. *arXiv preprint arXiv:2209.15208*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Keith Knight and Wenjiang Fu. Asymptotics for lasso-type estimators. *The Annals of Statistics*, 28(5):1356–1378, 2000. URL https://doi.org/10.1214/aos/1015957397.

Frederic Koehler, Lijia Zhou, Danica J. Sutherland, and Nathan Srebro. Uniform convergence of interpolators: Gaussian width, norm bounds and benign overfitting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2106.09276. arXiv:2106.09276.

V. Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001. doi: 10.1109/18.930926. URL https://doi.org/10.1109/18.930926.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Jung-Min Kwon, Jeong-Seop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning (ICML)*, 2021. doi: 10.48550/arxiv.2102.11600. URL https://doi.org/10.48550/arxiv.2102.11600.

John Langford and Rich Caruana. (not) bounding the true error. *Advances in Neural Information Processing Systems*, 14, 2001.

John Langford and Matthias Seeger. Bounds for averaging classifiers. 2001.

John Langford and John Shawe-Taylor. Pac-bayes & margins. In *NIPS*, 2002. URL https://proceedings.neurips.cc/paper/2002/hash/68d309812548887400e375eaa036d2f1-Abstract.html.

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

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Jaehoon Lee, Samuel Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems*, 33, 2020a.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent [*]. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020b. doi: 10.1088/1742-5468/abc62b. URL https://doi.org/10.1088/1742-5468/abc62b.

L.A. Levin. Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Problemy Peredachi Informatsii*, 10(3):30–35, 1974.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018a.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. doi: 10.48550/arxiv.1712.09913. URL https://doi.org/10.48550/arxiv.1712.09913.

M. Li and P.M.B. Vitanyi. *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag New York Inc, 2008.

Xingguo Li, Junwei Lu, Zhaoran Wang, Jarvis Haupt, and Tuo Zhao. On tighter generalization bound for deep neural networks: Cnns, resnets, and beyond. *arXiv preprint arXiv:1806.05159*, 2018b. URL https://arxiv.org/abs/1806.05159.

Xingguo Li, Junwei Lu, Zhaoran Wang, Jarvis Haupt, and Tuo Zhao. On tighter generalization bound for deep neural networks: Cnns, resnets, and beyond. *arXiv preprint arXiv:1806.05159*, 2018c.

Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. *arXiv preprint arXiv:1910.07454*, 2019. URL https://arxiv.org/abs/1910.07454.

Qianli Liao, Brando Miranda, Andrzej Banburski, Jack Hidary, and Tomaso Poggio. A surprising linear relationship predicts test performance in deep networks. *arXiv preprint arXiv:1807.09659*, 2018.

Henry W Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, 2017.

Yang Liu, Jeremy Bernstein, Markus Meister, and Yisong Yue. Learning by turning: Neural architecture aware optimisation. *arXiv preprint arXiv:2102.07227*, 2021.

Sanae Lotfi, Marc Finzi, Sanyam Kapoor, Andres Potapczynski, Micah Goldblum, and Andrew G Wilson. Pac-bayes compression bounds so tight that they can explain generalization. *Advances in Neural Information Processing Systems*, 35:31459–31473, 2022a.

Sanae Lotfi, Marc Finzi, Sanyam Kapoor, Andres Potapczynski, Micah Goldblum, and Andrew G Wilson. Pac-bayes compression bounds so tight that they can explain generalization. *Advances in Neural Information Processing Systems*, 35:31459–31473, 2022b. URL https://doi.org/10.48550/arxiv.2211.13609.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1906.05890. arXiv:1906.05890.

David JC Mackay. Introduction to gaussian processes. *NATO ASI series. Series F: computer and system sciences*, pages 133–165, 1998.

Wesley J Maddox, Gregory Benton, and Andrew Gordon Wilson. Rethinking parameter counting in deep models: Effective dimensionality revisited. *arXiv preprint arXiv:2003.02139*, 2020.

Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017. URL https://www.jmlr.org/papers/volume18/17-214/17-214.pdf.

Susanna Manrubia, José A Cuesta, Jacobo Aguirre, Sebastian E Ahnert, Lee Altenberg, Alejandro V Cano, Pablo Catalán, Ramon Diaz-Uriarte, Santiago F Elena, Juan Antonio García-Martín, et al. From genotypes to organisms: State-of-the-art and perspectives of a cornerstone in evolutionary dynamics. *arXiv preprint arXiv:2002.00363*, 2020.

Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

Andreas Maurer. A note on the pac bayesian theorem. *arXiv preprint cs/0411099*, 2004.

David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234. ACM, 1998.

David A. McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999. doi: 10.1023/a:1007618624809. URL https://doi.org/10.1023/a:1007618624809.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018. URL https://arxiv.org/abs/1812.06162.

Shahar Mendelson. A few notes on statistical learning theory. In *Machine Learning Summer School*. Wiley, 2002. doi: 10.1007/3-540-36434-X_1. URL https://doi.org/10.1007/3-540-36434-X_1.

Chris Mingard, Joar Skalse, Guillermo Valle-Pérez, David Martínez-Rubio, Vladimir Mikulik, and Ard A Louis. Neural networks are a priori biased towards boolean functions with low entropy. *arXiv preprint arXiv:1909.11522*, 2019.

Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, and Ard A Louis. Is sgd a bayesian sampler? well, almost. *Journal of Machine Learning Research*, 22(79):1–64, 2021.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://arxiv.org/abs/1802.05957.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai. Harmless interpolation of noisy data in regression. *IEEE Journal on Selected Areas in Information Theory*, 1(1):67–83, 2020.

Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 11611–11622, 2019.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2020a. URL https://arxiv.org/abs/1912.02292.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2020b. URL https://openreview.net/forum?id=B1N3b9VYw. ICLR 2020 version available.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data still hurt. *Journal of Statistical Mechanics: Theory and Experiment*, (12):124003, 2021. URL https://doi.org/10.1088/1742-5468/ac2a48.

Radford M Neal. Priors for infinite networks (tech. rep. no. crg-tr-94-1). *University of Toronto*, 1994.

Jeffrey Negrea, Gintare Karolina Dziugaite, and Daniel M Roy. In defense of uniform convergence: Generalization via derandomization with an application to interpolating predictors. *arXiv preprint arXiv:1912.04265*, 2019.

Behnam Neyshabur, Ruslan Salakhutdinov, and Nathan Srebro. Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015a. URL https://arxiv.org/abs/1506.02617.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Proceedings of the 28th Annual Conference on Learning Theory (COLT)*, 2015b. URL https://proceedings.mlr.press/v40/neyshabur15.html.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015c.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017a.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5949–5958, 2017b.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017c. URL https://arxiv.org/abs/1707.09564.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=BygfghAcYX.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=BygfghAcYX.

Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018a.

Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018b.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Z. Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. URL https://openreview.net/pdf?id=BJJsrmfCZ.

María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári. Tighter risk certificates for neural networks. *The Journal of Machine Learning Research*, 22(1): 10326–10365, 2021.

Henning Petzka, Linara Adilova, Michael Kamp, and Cristian Sminchisescu. A reparameterization-invariant flatness measure for deep neural networks. *arXiv preprint arXiv:1912.00058*, 2019.

María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári. Tighter risk certificates for neural networks. *The Journal of Machine Learning Research*, 2020. doi: 10.48550/arxiv.2007.12911. URL https://doi.org/10.48550/arxiv.2007.12911.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. *arXiv preprint arXiv:1806.08734*, 2018.

Akshay Rangamani, Nam H Nguyen, Abhishek Kumar, Dzung Phan, Sang H Chin, and Trac D Tran. A scale invariant flatness measure for deep network minima. *arXiv preprint arXiv:1902.02434*, 2019.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Jorma Rissanen. Stochastic complexity and modeling. *The annals of statistics*, pages 1080–1100, 1986.

Borja Rodríguez-Gálvez, Ragnar Thobaben, and Mikael Skoglund. More pac-bayes bounds: From bounded losses, to losses with general tail behaviors, to anytime validity. *Journal of Machine Learning Research*, 25(110):1–43, 2024. URL http://jmlr.org/papers/v25/23-1360.html.

Jonathan S Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. *arXiv preprint arXiv:1909.12673*, 2019.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1232–1240, Cadiz, Spain, 09–11 May 2016. PMLR. URL https://proceedings.mlr.press/v51/russo16.html.

Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016.

Siddhartha Satpathi and Rayadurgam Srikant. The dynamics of gradient descent for overparametrized neural networks. In *Learning for Dynamics and Control*, pages 373–384. PMLR, 2021.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations (ICLR)*, 2014a. URL https://arxiv.org/abs/1312.6120. arXiv:1312.6120.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations (ICLR)*, 2014b. URL https://arxiv.org/abs/1312.6120.

Steffen Schaper and Ard A Louis. The arrival of the frequent: how bias in genotype-phenotype maps can steer populations to local optima. *PloS one*, 9(2): e86635, 2014.

Matthias Seeger. Pac-bayesian generalization error bounds for gaussian process classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2002.

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

Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pages 5827–5837. PMLR, 2019. URL https://proceedings.mlr.press/v97/simsekli19a.html.

Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017a.

Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017b. URL https://arxiv.org/abs/1711.00489.

Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. Don't decay the learning rate, increase the batch size. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=B1Yy1BxCZ.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57, 2018a. URL https://www.jmlr.org/papers/volume19/18-188/18-188.pdf.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 2018b. doi: 10.5555/3291125.3309632. URL https://doi.org/10.5555/3291125.3309632.

Stefano Spigler, Mario Geiger, and Matthieu Wyart. Asymptotic learning curves of kernel methods: empirical data vs teacher-student paradigm. *arXiv preprint arXiv:1905.10843*, 2019.

Tomoaki Nishimura Taiji Suzuki, Hiroshi Abe. Compression based bound for non-compressed network: unified generalization error analysis of large compressible deep neural network. volume 8, 2020. URL https://arxiv.org/abs/1909.11274.

Mingyue Tan. Expectation propagation of gaussian process classification and its application to gene expression analysis. 01 2008.

Kyrill Thrampoulidis, Emre Oymak, and Babak Hassibi. The convex gaussian min–max theorem, 2015. URL https://arxiv.org/abs/1506.07868.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996. URL https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015. URL https://doi.org/10.1561/2200000046.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis. *arXiv preprint arXiv:1901.04653*, 2019.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis. In *International Conference on Machine Learning*, pages 9636–9647. PMLR, 2020. URL https://arxiv.org/abs/1901.04653.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11): 1134–1142, 1984.

Guillermo Valle-Pérez and Ard A Louis. Generalization bounds for deep learning. *arXiv preprint arXiv:2012.04115*, 2020.

Guillermo Valle-Pérez, Chico Q Camargo, and Ard A Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*, 2018.

Guillermo Valle-Pérez and Ard A. Louis. Generalization bounds for deep learning. *arXiv preprint arXiv:2012.04115*, 2020. URL https://arxiv.org/abs/2012.04115.

Guillermo Valle-Pérez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*, 2018. URL https://arxiv.org/abs/1805.08522.

Vladimir Vapnik. On the uniform convergence of relative frequencies of events to their probabilities. In *Doklady Akademii Nauk USSR*, volume 181, pages 781–787, 1968.

Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition, 1974.

Vladimir N Vapnik. The nature of statistical learning theory. 1995.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018. URL https://www.cambridge.org/core/books/highdimensional-probability/0689FD9F6DB7874DF6F899B02B3B2FA0.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019. URL https://doi.org/10.1017/9781108627771.

Colin Wei and Tengyu Ma. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. *arXiv preprint arXiv:1905.03684*, 2019.

Mingwei Wei and David J Schwab. How noise affects the hessian spectrum in overparameterized neural networks. *arXiv preprint arXiv:1910.00195*, 2019.

Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Proceedings of the 33rd Conference on Learning Theory (COLT)*, volume 125 of *Proceedings of Machine Learning Research*, pages 3635–3673. PMLR, 2020. URL https://proceedings.mlr.press/v125/woodworth20a.html.

Lei Wu, Zhanxing Zhu, et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. *Advances in Neural Information Processing Systems*, 30, 2017. URL https://arxiv.org/abs/1705.07809.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

Sho Yaida. Fluctuation–dissipation relations for stochastic gradient descent. *arXiv preprint arXiv:1810.00004*, 2018. URL https://arxiv.org/abs/1810.00004.

Greg Yang. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. 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. URL https://proceedings.neurips.cc/paper/2019/file/5e69fda38cda2060819766569fd93aa5-Paper.pdf.

Greg Yang and Hadi Salman. A fine-grained spectral perspective on neural networks. *arXiv preprint arXiv:1907.10599*, 2019.

Zhewei Yao, Amir Gholami, Qi Lei, Kurt Keutzer, and Michael W Mahoney. Hessian-based analysis of large batch training and robustness to adversaries. In *Advances in Neural Information Processing Systems*, pages 4949–4959, 2018.

Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017. URL https://arxiv.org/abs/1705.10941.

Lenka Zdeborová. Understanding deep learning is also a job for physicists. *Nature Physics*, 16(6):602–604, 2020.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016a.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016b. URL https://arxiv.org/abs/1611.03530.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Shuofeng Zhang and Ard Louis. Closed-form ℓ_r norm scaling with data for overparameterized linear regression and diagonal linear networks under ℓ_p bias. *arXiv preprint arXiv:2509.21181*, 2025. URL https://arxiv.org/abs/2509.21181.

Shuofeng Zhang, Isaac Reid, Guillermo Valle Pérez, and Ard Louis. Why flatness does and does not correlate with generalization for deep neural networks. *arXiv preprint arXiv:2103.06219*, 2021a.

Shuofeng Zhang, Isaac Reid, Guillermo Valle Pérez, and Ard Louis. Why flatness does and does not correlate with generalization for deep neural networks. *arXiv preprint arXiv:2103.06219*, 2021b. URL https://arxiv.org/abs/2103.06219.

Yao Zhang, Andrew M Saxe, Madhu S Advani, and Alpha A Lee. Energy–entropy competition and the effectiveness of stochastic gradient descent in machine learning. *Molecular Physics*, 116(21-22):3214–3223, 2018.

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. URL https://openreview.net/forum?id=BJgqqsAct7.

Yi Zhou, Yingbin Liang, and Huishuai Zhang. Understanding generalization error of sgd in nonconvex optimization. *stat*, 2021. doi: 10.1007/s10994-021-06056-w. URL https://doi.org/10.1007/s10994-021-06056-w.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006. URL https://doi.org/10.1198/016214506000000735.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005. URL https://doi.org/10.1111/j.1467-9868.2005.00503.x.

Appendices

A Proof of theorems

A.1 Proof for theorem 10

In this section we prove Theorem 10. We first reintroduce the scale invariance lemma from Arora et al. [2018c], Li and Arora [2019] which is the key source of intuition about scale-invariant networks.

Lemma 1 (Scale-invariant networks)**.** *If* $\forall c \in \mathbb{R}^+$, $L(\boldsymbol{\theta}) = L(c\boldsymbol{\theta})$, *then*

1. $\langle \nabla_{\boldsymbol{\theta}} L, \boldsymbol{\theta} \rangle = 0$
2. $\nabla_{\boldsymbol{\theta}} L|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} = c \nabla_{\boldsymbol{\theta}} L|_{\boldsymbol{\theta}=c\boldsymbol{\theta}_0}$, *for any* $c > 0$

Proof. Treat c as a differentiable variable. Clearly the derivative of L w.r.t. c is 0.

1. $0 = \frac{\partial L}{\partial c} = \langle \nabla_{\boldsymbol{\theta}} L, \boldsymbol{\theta} \rangle$
2. Take gradient of the both sides of the equation $L(\boldsymbol{\theta}) = L(c\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ and set $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, we immediately arrive at the result.

□

We give the technical definition of the commonly used training algorithm SGD with momentum and weight decay (WD) (with respect to the L2 norm of the parameters) using a convenient form given in Li and Arora [2019], which is equivalent to the default implementation in Pytorch [Paszke et al., 2017].

Definition 10 (SGD with momentum and WD)**.** *At iteration t, given the current parameters and learning rate $(\boldsymbol{\theta}_{t-1}, \eta_{t-1})$, the buffered parameters and learning rate $(\boldsymbol{\theta}_{t-2}, \eta_{t-2})$, momentum factor γ, current loss $L(\boldsymbol{\theta}_{t-1})$ and WD factor λ_{t-1}, update the parameters $\boldsymbol{\theta}_t$ as the following:*

$$\frac{\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}}{\eta_{t-1}} = \gamma \frac{\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_{t-2}}{\eta_{t-2}} - \nabla_{\boldsymbol{\theta}} \left(L\left(\boldsymbol{\theta}_{t-1}\right) + \frac{\lambda_{t-1}}{2} \left\|\boldsymbol{\theta}_{t-1}\right\|_2^2 \right) \tag{A.1}$$

For the boundary conditions, it is a common practice to set $\boldsymbol{\theta}_{-1} = \boldsymbol{\theta}_0$ and η_{-1} can be arbitrary.

From the above definition, it is easy to see that we can represent the state of the algorithm using a four-tuple $(\boldsymbol{\theta}, \eta, \boldsymbol{\theta'}, \eta')$, which stand for the parameters/learning rate at the current step and their buffers from the last step, respectively. A gradient descent step at time t with momentum factor γ and WD factor λ can be seen as a mapping between two states:

- A GD step with momentum and WD: $\mathrm{GD}_t^{\rho}\left(\boldsymbol{\theta}, \eta, \boldsymbol{\theta'}, \eta'\right) = \left(\rho\boldsymbol{\theta} + \eta\left(\gamma\frac{\boldsymbol{\theta}-\boldsymbol{\theta'}}{\eta'} - \nabla L(\boldsymbol{\theta})\right), \eta, \boldsymbol{\theta}, \eta\right)$

Here with WD factor being λ, ρ is set to be $1 - \lambda\eta$. Furthermore, we define some extra basic mappings that can be composed together to represent the temporal behavior of the algorithm.

- Scale current parameters $\boldsymbol{\theta}$:
$$\Pi_1^c\left(\boldsymbol{\theta}, \eta, \boldsymbol{\theta'}, \eta'\right) = \left(c\boldsymbol{\theta}, \eta, \boldsymbol{\theta'}, \eta'\right);$$
- Scale current LR η: $\Pi_2^c\left(\boldsymbol{\theta}, \eta, \boldsymbol{\theta'}, \eta'\right) = \left(\boldsymbol{\theta}, c\eta, \boldsymbol{\theta'}, \eta'\right);$
- Scale buffered parameter $\boldsymbol{\theta'}$:
$$\Pi_3^c\left(\boldsymbol{\theta}, \eta, \boldsymbol{\theta'}, \eta'\right) = \left(\boldsymbol{\theta}, \eta, c\boldsymbol{\theta'}, \eta'\right);$$
- Scale buffered LR η': $\Pi_4^c\left(\boldsymbol{\theta}, \eta, \boldsymbol{\theta'}, \eta'\right) = \left(\boldsymbol{\theta}, \eta, \boldsymbol{\theta'}, c\eta'\right).$

We know that in scale-invariant neural nets, two networks $f(\boldsymbol{\theta})$ and $f(\widetilde{\boldsymbol{\theta}})$ are equivalent if $\exists c > 0$ such that $\widetilde{\boldsymbol{\theta}} = c\boldsymbol{\theta}$; Here we extend the equivalence between weights to the equivalence between states of algorithms:

Definition 11 (Equivalent states)**.** *Two states* $(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta')$ *and* $(\widetilde{\boldsymbol{\theta}}, \widetilde{\eta}, \widetilde{\boldsymbol{\theta}}', \widetilde{\eta}')$ *are equivalent iff* $\exists c > 0$ *such that* $(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta') = \left[\Pi_1^c \circ \Pi_2^{c^2} \circ \Pi_3^c \circ \Pi_4^{c^2}\right]\left(\widetilde{\boldsymbol{\theta}}, \widetilde{\eta}, \widetilde{\boldsymbol{\theta}}', \widetilde{\eta}'\right) = \left(c\widetilde{\boldsymbol{\theta}}, c^2\widetilde{\eta}, c\widetilde{\boldsymbol{\theta}}', c^2\widetilde{\eta}'\right)$*, which is also noted as* $(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta') \overset{c}{\sim} (\widetilde{\boldsymbol{\theta}}, \widetilde{\eta}, \widetilde{\boldsymbol{\theta}}', \widetilde{\eta}')$*. We call* $\left[\Pi_1^c \circ \Pi_2^{c^2} \circ \Pi_3^c \circ \Pi_4^{c^2}\right]$ *as equivalent scaling for all* $c > 0$*.*

Here we provide an intuitive explanation of why the equivalent scaling takes the form above. If we rearrange the first term of the R.H.S. of the GD update, and assume we are operating in a regime where $\eta' = \eta$ [1], we have

$$\boldsymbol{\theta}_{\text{update}} = (\rho + \gamma)\boldsymbol{\theta} - \eta\nabla L(\boldsymbol{\theta}) - \gamma\boldsymbol{\theta}' \tag{A.2}$$

In order to keep the updated parameters in the same direction, the three terms in equation A.2 need to have the same scaling. From lemma 1 we know that when $\boldsymbol{\theta}$ is scaled by c, the gradient $\nabla L(\boldsymbol{\theta})$ will actually be scaled by $\frac{1}{c}$. Hence for the second term $\eta\nabla L(\boldsymbol{\theta})$ to have the same amount of scaling as the first and third terms, η has to be scaled by c^2.

The following lemma tells us that equivalent scaling commutes with GD update with momentum and WD, implying that equivalence is preserved under GD updates. Hence we are free to stack GD updates and insert equivalent scaling anywhere in a sequence of basic maps without changing the network function.

Lemma 2 (Equivalent scaling commutes with GD)**.** $\forall c, \rho > 0$ *and* $t \geq 0$*,*

$$GD_t^\rho \circ \left[\Pi_1^c \circ \Pi_2^{c^2} \circ \Pi_3^c \circ \Pi_4^{c^2}\right] = \left[\Pi_1^c \circ \Pi_2^{c^2} \circ \Pi_3^c \circ \Pi_4^{c^2}\right] \circ GD_t^\rho.$$

In other words, if $(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta') \overset{c}{\sim} (\widetilde{\boldsymbol{\theta}}, \widetilde{\eta}, \widetilde{\boldsymbol{\theta}}', \widetilde{\eta}')$ *then* $GD_t^\rho(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta') \overset{c}{\sim} GD_t^\rho(\widetilde{\boldsymbol{\theta}}, \widetilde{\eta}, \widetilde{\boldsymbol{\theta}}', \widetilde{\eta}')$*.*

Proof. For any given state $(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta')$, the L.H.S. of the equation is:

$$\begin{aligned}
\mathrm{GD}_t^\rho \circ \left[\Pi_1^c \circ \Pi_2^{c^2} \circ \Pi_3^c \circ \Pi_4^{c^2}\right](\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta') &= \mathrm{GD}_t^\rho(c\boldsymbol{\theta}, c^2\eta, c\boldsymbol{\theta}', c^2\eta') \\
&= \left(\rho c\boldsymbol{\theta} + c^2\eta\left(\gamma\frac{c\boldsymbol{\theta} - c\boldsymbol{\theta}'}{c^2\eta'} - \nabla L(c\boldsymbol{\theta})\right), c^2\eta, c\boldsymbol{\theta}, c^2\eta\right) \\
&= \left(c\left(\rho\boldsymbol{\theta} + \eta\left(\gamma\frac{\boldsymbol{\theta} - \boldsymbol{\theta}'}{\eta'} - \nabla L(\boldsymbol{\theta})\right)\right), c^2\eta, c\boldsymbol{\theta}, c^2\eta\right) \\
&= \left[\Pi_1^c \circ \Pi_2^{c^2} \circ \Pi_3^c \circ \Pi_4^{c^2}\right] \circ \mathrm{GD}_t^\rho(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta')
\end{aligned}$$

[1] This just means in the eyes of the GD algorithm, the buffered LR and the current LR are the same. It does not exclude the possibility that we can still scale the current LR between GD updates.

□

Now comes the important step: in order to show the equivalence between two series of parameters with (fixed WD + fixed LR)/(exponentially decreasing WD + exponentially increasing LR), respectively, we need to rewrite GD_t^ρ as a composition of itself with varying WD factor and upscaling LR, conjugated with other scaling terms that cancel with each other eventually. Here again we choose to work in the regime where the current and buffered LR are the same in the input of GD_t^ρ.

Lemma 3 (Conjugated GD updates)**.** *For any input with equal current and buffered LR* $(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta)$ *and* $\forall \alpha \in (Z_0, Z_1] \cup [Z_2, 1)$ [2] *, we have*

$$GD_t^\rho\left(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta\right) = \left[\Pi_4^\alpha \circ \Pi_2^\alpha \circ \Pi_1^\alpha \circ GD_t^\beta \circ \Pi_2^{\alpha^{-1}} \circ \Pi_3^\alpha \circ \Pi_4^\alpha\right]\left(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta\right)$$

which can be written in the form of equivalent states:

$$GD_t^\rho\left(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta\right) \overset{\alpha}{\sim} \left[\Pi_3^{\alpha^{-1}} \circ \Pi_4^{\alpha^{-1}} \circ \Pi_2^{\alpha^{-1}} \circ GD_t^\beta \circ \Pi_2^{\alpha^{-1}} \circ \Pi_3^\alpha \circ \Pi_4^\alpha\right]\left(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta\right) \quad \text{(A.3)}$$

where

- $\beta = \frac{(\rho+\gamma)}{\alpha} - \frac{\gamma}{\alpha^2}$
- $Z_0 = \frac{\gamma}{1-\lambda\eta_0+\gamma}$
- $Z_1 = \frac{1+\gamma-\lambda\eta_0-\sqrt{(1-\gamma)^2-2(1+\gamma)\lambda\eta_0+\lambda^2\eta_0^2}}{2}$
- $Z_2 = \frac{1+\gamma-\lambda\eta_0+\sqrt{(1-\gamma)^2-2(1+\gamma)\lambda\eta_0+\lambda^2\eta_0^2}}{2}$ [3]

[2] Technically α can be larger than 1, but in that case we will be shrinking the LR between GD steps which is not what we mainly care about here.

[3] It's easy to verify that Z_2 is always smaller than 1.

Proof. We directly verify the equivalence. The R.H.S. is:

$$\begin{aligned}
&\left[\Pi_4^\alpha \circ \Pi_2^\alpha \circ \Pi_1^\alpha \circ \mathrm{GD}_t^\beta \circ \Pi_2^{\alpha^{-1}} \circ \Pi_3^\alpha \circ \Pi_4^\alpha\right](\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta) \\
=&\left[\Pi_4^\alpha \circ \Pi_2^\alpha \circ \Pi_1^\alpha\right] \circ \mathrm{GD}_t^\beta \left(\boldsymbol{\theta}, \alpha^{-1}\eta, \alpha\boldsymbol{\theta}', \alpha\eta\right) \\
=&\left[\Pi_4^\alpha \circ \Pi_2^\alpha \circ \Pi_1^\alpha\right] \left(\beta\boldsymbol{\theta} + \alpha^{-1}\eta\left(\gamma\frac{\boldsymbol{\theta} - \alpha\boldsymbol{\theta}'}{\alpha\eta} - \nabla L(\boldsymbol{\theta})\right), \alpha^{-1}\eta, \boldsymbol{\theta}, \alpha^{-1}\eta\right) \\
=&\left(\alpha\beta\boldsymbol{\theta} + \eta\left(\gamma\frac{\boldsymbol{\theta} - \alpha\boldsymbol{\theta}'}{\alpha\eta} - \nabla L(\boldsymbol{\theta})\right), \eta, \boldsymbol{\theta}, \eta\right) \\
=&\left((\rho+\gamma)\,\boldsymbol{\theta} - \gamma\boldsymbol{\theta}' - \eta\nabla L(\boldsymbol{\theta}), \eta, \boldsymbol{\theta}, \eta\right) \\
=&\left(\rho\boldsymbol{\theta} + \eta\left(\gamma\frac{\boldsymbol{\theta} - \boldsymbol{\theta}'}{\eta} - \nabla L(\boldsymbol{\theta})\right), \eta, \boldsymbol{\theta}, \eta\right) \\
=&\mathrm{GD}_t^\rho\left(\boldsymbol{\theta}, \eta, \boldsymbol{\theta}', \eta\right)
\end{aligned}$$

The range of α can be easily shown by combining the following two constraints and assuming remark 4.6 is true:

- $\alpha \in (0, 1]$;
- $\frac{(\rho+\gamma)}{\alpha} - \frac{\gamma}{\alpha^2} \in (0, 1]$

□

Now we are ready to prove theorem 10.

Proof of Theorem 10. From the assumption we have the following equivalence between the boundary conditions of the two series of states:

$$\begin{aligned}
(\boldsymbol{\theta}_0, \eta_0, \boldsymbol{\theta}_{-1}, \eta_{-1}) &= (\boldsymbol{\theta}_0, \eta_0, \boldsymbol{\theta}_0, \eta_0) \\
\left(\widetilde{\boldsymbol{\theta}}_0, \widetilde{\eta}_0, \widetilde{\boldsymbol{\theta}}_{-1}, \widetilde{\eta}_{-1}\right) &= \left[\Pi_2^{\alpha^{-1}} \circ \Pi_3^\alpha \circ \Pi_4^\alpha\right](\boldsymbol{\theta}_0, \eta_0, \boldsymbol{\theta}_{-1}, \eta_{-1})
\end{aligned}$$

Lemma 2 tells us that equivalent states are still equivalent after both being transformed by a GD step. Hence we can stack up on both sides of equation A.3 for a finite number of times. i.e. for $\forall t \geq 0$, we have

$$\begin{aligned}
&\mathrm{GD}_{t-1}^\rho \circ \mathrm{GD}_{t-2}^\rho \circ \cdots \circ \mathrm{GD}_0^\rho\left(\boldsymbol{\theta}_0, \eta_0, \boldsymbol{\theta}_{-1}, \eta_{-1}\right) \\
\overset{\alpha^t}{\sim}&\left[\Pi_3^{\alpha^{-1}} \circ \Pi_4^{\alpha^{-1}} \circ \Pi_2^{\alpha^{-1}} \circ \mathrm{GD}_{t-1}^\beta \circ \Pi_2^{\alpha^{-1}} \circ \Pi_3^\alpha \circ \Pi_4^\alpha\right] \\
&\circ \cdots \circ \left[\Pi_3^{\alpha^{-1}} \circ \Pi_4^{\alpha^{-1}} \circ \Pi_2^{\alpha^{-1}}\right] \circ \mathrm{GD}_0^\beta\left(\widetilde{\boldsymbol{\theta}}_0, \widetilde{\eta}_0, \widetilde{\boldsymbol{\theta}}_{-1}, \widetilde{\eta}_{-1}\right) \\
\overset{\alpha^t}{\sim}&\left[\Pi_3^{\alpha^{-1}} \circ \Pi_4^{\alpha^{-1}} \circ \Pi_2^{\alpha^{-1}} \circ \mathrm{GD}_{t-1}^\beta \circ \Pi_2^{\alpha^{-2}} \circ \mathrm{GD}_{t-2}^\beta \circ \cdots \circ \Pi_2^{\alpha^{-2}} \circ \mathrm{GD}_0^\beta\right]\left(\widetilde{\boldsymbol{\theta}}_0, \widetilde{\eta}_0, \widetilde{\boldsymbol{\theta}}_{-1}, \widetilde{\eta}_{-1}\right)
\end{aligned}$$

which implies that

- $\boldsymbol{\theta}_t = \alpha^t \widetilde{\boldsymbol{\theta}}_t$
- $\widetilde{\eta}_t = \alpha^{-2t}\widetilde{\eta}_0 = \alpha^{-2t-1}\eta_0$
- $\widetilde{\lambda}_t = \frac{1-\beta}{\widetilde{\eta}_t}$, except for $t = 0$, in which case $\widetilde{\lambda}_0 = \frac{1-\beta+\frac{\gamma}{\alpha^2}-\frac{\gamma}{\alpha}}{\widetilde{\eta}_0}$

We note the special boundary condition for $\widetilde{\lambda}_0$ due to the fact that $\widetilde{\boldsymbol{\theta}}_{-1} \neq \widetilde{\boldsymbol{\theta}}_0$. □

A.2 Minimum-ℓ_p interpolator with s-sparse ground truth

For completeness, we first introduce again the mathematical settings and restate our main theorem. We study $p \in (1, 2]$, set $q = \frac{p}{p-1} \in [2, \infty)$, and let $r \in [1, p]$. Dimensions $n, d \in \mathbb{N}$ with $d \geq n$. All $X \in \mathbb{R}^{n\times d}$ have i.i.d. $\mathcal{N}(0, 1)$ entries; columns are $X_{:,j}$. Noise $\xi \sim \mathcal{N}(0, \sigma^2 I_n)$, independent of X. The signal $w^\star \in \mathbb{R}^d$ is s-sparse with support $S \subset [d]$, $|S| = s$; we write $w^\star_S$ for its nonzeros. The response is $Y := Xw^\star + \xi$. The min-ℓ_p interpolator

$$\widehat{w}_p \in \arg\min\{\|w\|_p : Xw = Y\} \qquad (p > 1 \text{ ensures uniqueness})$$

is our object of interest. Shorthands:

$$\tau_s^2 := \|w^\star\|_2^2 + \sigma^2, \qquad W_q := \|w^\star\|_q^q = \sum_{j\in S} |w_j^\star|^q.$$

Remark (Standing assumptions and probability shorthand). We work in the proportional regime

$$\frac{d}{n} \to \kappa \in (1, \infty), \qquad \kappa_{\text{bulk}} := \liminf_{n\to\infty} \frac{d-s}{n} \in (0, \infty),$$

so $d - s = \Theta(n)$ and $s = O(n)$ (we do not require $s \leq n$). Unless stated otherwise, all hidden constants depend only on $(p, \kappa_{\text{bulk}})$ (and on r when relevant), and "w.h.p." means probability at least $1 - Ce^{-cn} - 2d^{-\gamma}$. When we simplify remainders using $s \leq n$ (e.g., $\sqrt{sn} + s \rightsquigarrow \sqrt{sn}$), the corresponding $s > n$ form is always available and does not affect any $\asymp$ conclusions in Theorem 12.

On proportionality. The assumption $d/n \to \kappa$ is only for cleanliness of exposition and to keep constants tidy; it is not essential to the argument. All places where it

enters (e.g., the bulk ℓ_t embedding and the uniform column–norm control) can be run under the weaker—and often more realistic—conditions

$$\liminf_{n\to\infty} \frac{d-s}{n} = \kappa_{\text{bulk}} > 0, \qquad \log d = o(n), \qquad s = O(n).$$

In particular, our proofs and conclusions (same exponents in n, the threshold $r_\star = 2(p-1)$, and the high-probability events) remain valid even in "larger" aspect-ratio regimes (including $d/n \to \infty$) as long as $\log d = o(n)$ and the bulk density is bounded below. Under these weaker assumptions the hidden constants are uniform in (n, d, s) and depend only on $(p, r, \kappa_{\text{bulk}})$ (and on a fixed upper bound for s/n if desired), so no changes to the proofs are needed.

A.2.1 Main theorem

Theorem 12 (Theorem 11 restated)**.** *Fix $p \in (1, 2]$, $q = \frac{p}{p-1}$, $r \in [1, p]$, and suppose $\liminf (d-s)/n = \kappa_{\text{bulk}} > 0$ while $d/n \to \kappa \in (1, \infty)$. Then, w.h.p.,*

$$\|\widehat{w}_p\|_r \asymp \max\Big\{ \underbrace{t_\star^{q-1}\, n^{\,q-1}\, \|w^\star\|_{(q-1)r}^{q-1}}_{\text{spike main } (S)},\ \underbrace{(d-s)^{1/r} \left(t_\star\, \tau_s \sqrt{n}\right)^{q-1}}_{\text{bulk } (S^c)},\ \underbrace{s^{\max\{1/r,\ (q-1)/2\}} \left(t_\star\, \tau_s \sqrt{n}\right)^{q-1}}_{\text{spike remainder}} \Big\}. \tag{A.4}$$

where the ray scale $t_\star$ *satisfies*

$$t_\star^{q-1} \asymp \frac{\|Y\|_2^2}{\|X^\top Y\|_q^q} \asymp \frac{\tau_s^2 n}{n^q W_q + (d-s)\, m_q\, \tau_s^q\, n^{q/2} \ +\ O\Big(\tau_s^q (s\, n^{q/2} + s^{1+q/2})\Big)} \qquad \text{w.h.p.} \tag{A.5}$$

with $m_t := \mathbb{E}|Z|^t$ and $Z \sim \mathcal{N}(0,1)$. Define the dual-transition scale

$$n_\star \asymp \left(\kappa_{\text{bulk}} \frac{\tau_s^q}{W_q}\right)^{\frac{2}{q-2}}. \tag{A.6}$$

Then, w.h.p., the following asymptotic simplifications hold:

***Dual spike-dominated** $n \gg n_\star$.*

$$\|\widehat{w}_p\|_r \asymp \begin{cases} \dfrac{\tau_s^{q+1}}{W_q}\, n^{\frac{1}{r} - \frac{1}{2(p-1)}}, & r \le 2(p-1), \\ \dfrac{\tau_s^2}{W_q}\, \|w^\star\|_{(q-1)r}^{q-1}, & r > 2(p-1). \end{cases} \tag{A.7}$$

Dual bulk-dominated $n \ll n_\star$**.**

$$\|\widehat{w}_p\|_r \asymp \max\left\{ \kappa_{\text{bulk}}^{\frac{1}{r}-1} \tau_s \, n^{\frac{1}{r}-\frac{1}{2}}, \ \kappa_{\text{bulk}}^{-1} \tau_s^{2-q} \, \|w^\star\|_{(q-1)r}^{q-1} \, n^{\frac{q}{2}-1}, \ \kappa_{\text{bulk}}^{-1} \tau_s \, s^{\max\{1/r,\ (q-1)/2\}} \, n^{-1/2} \right\}. \tag{A.8}$$

(Equivalently, using $d-s \asymp \kappa_{\text{bulk}} n$, the third term can be written as $\frac{\tau_s}{d-s} \, s^{\max\{1/r,\,(q-1)/2\}} \sqrt{n}$.)

Remark (When the third term is absorbed). If $r \le 2(p-1)$ and $s \le C\,(d-s)$ for an absolute constant C, then the third term in (5.5) is dominated by the first term (their ratio is $\lesssim (s/(d-s))^{1/r}$). In that case, (5.5) reduces to the two-term maximum

$$\|\widehat{w}_p\|_r \asymp \max\left\{ \kappa_{\text{bulk}}^{\frac{1}{r}-1} \tau_s \, n^{\frac{1}{r}-\frac{1}{2}}, \ \kappa_{\text{bulk}}^{-1} \tau_s^{2-q} \, \|w^\star\|_{(q-1)r}^{q-1} \, n^{\frac{q}{2}-1} \right\}.$$

For $r > 2(p-1)$, no uniform absorption holds in general; the third term can dominate when $\|w^\star\|_{(q-1)r}$ is small relative to τ_s.

Remark (Boundary $p = 2$). At $p = 2$ (so $q = 2$) the exponent in (5.3) diverges. In the proportional-d regime ($d/n \to \kappa$) there is no n-driven transition; the relative sizes of the spike and bulk are constant-level. In the finite-d regime (below) a concrete n-threshold does exist because $(d-s)$ does not scale with n.

A.2.2 Key lemmas and proof outline

Roadmap. We prove Theorem 11 by (i) reducing the min-ℓ_p interpolator to a dual maximization and restricting the dual to the one-dimensional ray $\lambda = tY$, (ii) decomposing $\|X^\top Y\|_q^q$ into a spike term ($j \in S$) and a bulk term ($j \notin S$), and (iii) converting back to the primal via the KKT map, which raises correlations to the power $(q-1)$ and produces the three-term maximum in (5.2). The elbow at $r = 2(p-1)$ comes from the sign of $1/r - 1/(2(p-1))$, i.e., exactly whether the bulk-type contribution grows or plateaus in the spike-dominated regime. We work on a single high-probability event $\mathcal{E}$ (defined below) on which all concentration facts hold simultaneously.

Global event. Let $\mathcal{E}$ be the intersection of the column-norm, spectral, and bulk ℓ_t events from Lemmas 4, 7, and 8. Then $\mathbb{P}(\mathcal{E}) \geq 1 - Ce^{-cn} - 2d^{-\gamma}$. All bounds below hold on $\mathcal{E}$.

Dual problem and KKT

We briefly review Lagrangian duality for convex programs with equality constraints and then apply it to the minimum-ℓ_p interpolator.

Primal problem and feasibility. We consider

$$\min_{w\in\mathbb{R}^d} f(w) \quad \text{subject to} \quad Xw = Y, \qquad \text{with} \quad f(w) := \frac{1}{p}\|w\|_p^p,$$

where $p \in (1, 2]$. Since $X \in \mathbb{R}^{n\times d}$ has full row rank n a.s. (for $d \geq n$ with i.i.d. $\mathcal{N}(0, 1)$ entries), the affine constraint set $\{w : Xw = Y\}$ is nonempty for every $Y \in \mathbb{R}^n$. The objective f is proper, closed, and *strictly convex* for $p > 1$ (indeed uniformly convex). Therefore, the primal minimizer $\widehat{w}_p$ exists and is unique. Introduce a Lagrange multiplier $\lambda \in \mathbb{R}^n$ for the equality constraint, and form the Lagrangian

$$\mathcal{L}(w, \lambda) \;:=\; f(w) + \langle \lambda,\, Y - Xw\rangle.$$

The *dual function* is obtained by minimizing the Lagrangian over w:

$$g(\lambda) \;:=\; \inf_{w\in\mathbb{R}^d} \Big\{ f(w) - \langle X^\top\lambda,\, w\rangle \Big\} + \langle Y, \lambda\rangle \;=\; -f^\star(X^\top\lambda) + \langle Y, \lambda\rangle,$$

where $f^\star$ is the convex conjugate of f:

$$f^\star(z) \;:=\; \sup_{w\in\mathbb{R}^d} \Big\{\langle z, w\rangle - f(w)\Big\}.$$

Since $f(w) = \sum_{i=1}^d |w_i|^p/p$ is separable, its conjugate is $f^\star(z) = \sum_{i=1}^d |z_i|^q/q = (1/q)\|z\|_q^q$, where $q = p/(p-1)$ is the Hölder conjugate of p. Indeed, for each coordinate

$$\sup_{t\in\mathbb{R}} \{z\, t - |t|^p/p\}$$

is achieved at $t = \operatorname{sgn}(z)|z|^{q-1}$, with optimal value $|z|^q/q$. Therefore the dual function is

$$g(\lambda) \;=\; \langle Y, \lambda\rangle - \frac{1}{q}\|X^\top\lambda\|_q^q.$$

Dual problem and strong duality. The *dual problem* is $\max_{\lambda\in\mathbb{R}^n} g(\lambda)$, i.e.

$$\max_{\lambda\in\mathbb{R}^n} \; D(\lambda), \qquad D(\lambda) := \langle Y, \lambda\rangle - \tfrac{1}{q}\|X^\top\lambda\|_q^q.$$

This is a concave maximization problem (a smooth concave objective with no constraints). Strong duality holds in our setting by standard convex duality: the primal is convex, the constraint is affine, and feasibility holds (Slater's condition for equalities reduces to existence of a feasible point). Hence

$$\min_{w:\, Xw=Y} f(w) \;=\; \max_{\lambda\in\mathbb{R}^n} D(\lambda).$$

For convex programs with equality constraints, the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for optimality under strong duality. They read:

$$\text{(primal feasibility)} \quad Xw = Y, \qquad \text{(stationarity)} \quad 0 \in \partial f(w) - X^\top\lambda.$$

Because $p > 1$, f is differentiable on $\mathbb{R}^d$ with gradient

$$\nabla f(w) = |w|^{p-2} \odot w \;=\; \operatorname{sgn}(w) \odot |w|^{p-1},$$

so the subdifferential collapses to the singleton $\{\nabla f(w)\}$ and stationarity is

$$\nabla f(w) = X^\top\lambda.$$

At any primal-dual optimum $(\widehat{w}_p, \lambda^\star)$ we therefore have

$$X\widehat{w}_p = Y, \qquad X^\top\lambda^\star = \nabla f(\widehat{w}_p) = |\widehat{w}_p|^{\,p-2} \odot \widehat{w}_p. \tag{A.9}$$

The conjugate $f^\star$ is differentiable with $\nabla f^\star(z) = |z|^{q-2} \odot z = \operatorname{sgn}(z) \odot |z|^{q-1}$, and the gradients are mutual inverses: $\nabla f^\star = (\nabla f)^{-1}$. Applying $\nabla f^\star$ to both sides of $X^\top\lambda^\star = \nabla f(\widehat{w}_p)$ gives the *coordinatewise KKT map*:

$$\widehat{w}_{p,i} = \Big(\nabla f^\star(X^\top\lambda^\star)\Big)_i = \operatorname{sgn}\Big((X^\top\lambda^\star)_i\Big)\,\Big|(X^\top\lambda^\star)_i\Big|^{q-1}. \tag{A.10}$$

Equivalently, $\widehat{w}_p = \nabla f^\star\Big(X^\top\lambda^\star\Big)$ and $X^\top\lambda^\star = \nabla f(\widehat{w}_p)$.

At optimality, Fenchel–Young gives $f(\widehat{w}_p) + f^\star(X^\top\lambda^\star) = \langle \widehat{w}_p, X^\top\lambda^\star\rangle$. Using $X\widehat{w}_p = Y$ and the expressions for f and $f^\star$ yields the identities

$$\|X^\top\lambda^\star\|_q^q = \|\widehat{w}_p\|_p^p = \langle Y, \lambda^\star\rangle. \tag{A.11}$$

These will be used repeatedly to pass between the primal and dual scales.

The affine set $\{w : Xw = Y\}$ is a translate of $\ker(X)$, and minimizing $\|w\|_p$ over it finds the point where a scaled ℓ_p ball first touches this affine subspace. The *outer normal* to the ℓ_p ball at the touching point is $\nabla f(\widehat{w}_p) = |\widehat{w}_p|^{p-2} \odot \widehat{w}_p$, and the KKT condition $X^\top\lambda^\star = \nabla f(\widehat{w}_p)$ says that this normal lies in the row space of X. In coordinates, (A.10) shows that each coefficient of $\widehat{w}_p$ is a $(q-1)$-power of the correlation between the corresponding feature column $X_{:,i}$ and the dual multiplier $\lambda^\star$.

Specialization at $p = 2$. When $p = q = 2$, $\nabla f(w) = w$ and $\nabla f^\star(z) = z$. Then (A.9) reads $X^\top\lambda^\star = \widehat{w}_2$ and $X\widehat{w}_2 = Y$, which implies $XX^\top\lambda^\star = Y$ and hence $\lambda^\star = (XX^\top)^{-1}Y$. Therefore

$$\widehat{w}_2 = X^\top(XX^\top)^{-1}Y = X^+Y,$$

the minimum-ℓ_2 (Moore–Penrose) interpolator. For $p \neq 2$ the same structure persists but the map $z \mapsto \nabla f^\star(z) = \operatorname{sgn}(z)|z|^{q-1}$ is nonlinear, which is exactly what introduces the $(q-1)$-power in the subsequent spike/bulk analysis.

Why duality helps here. The dual objective

$$D(\lambda) = \langle Y, \lambda\rangle - \tfrac{1}{q}\|X^\top\lambda\|_q^q$$

separates the *data dependence* (linear in Y) from the *feature geometry* through $\|X^\top\lambda\|_q^q$. In our Gaussian design, the d coordinates of $X^\top\lambda$ split naturally into the s *spikes* (indices in S) and the $(d-s)$ *bulk*, for which we have precise ℓ_t concentration (Lemmas 7 and 8). Because D is homogeneous in a simple way along the *ray* $\lambda = tY$,

$$D(t) = t\|Y\|_2^2 - \tfrac{t^q}{q}\|X^\top Y\|_q^q,$$

we will use the *ray scale* $t_\star$ (the maximizer of $D(tY)$) as a canonical scale for $\lambda^\star$; Lemma 9 shows $\|\lambda^\star\|_2 \asymp t_\star\|Y\|_2$ and provides blockwise controls on $X^\top\lambda^\star$. The KKT map (A.10) then converts $\|X^\top\lambda^\star\|_{(q-1)r}^{q-1}$ into $\|\widehat{w}_p\|_r$, via $\||u|^{\odot(q-1)}\|_r = \|u\|_{(q-1)r}^{q-1}$, which is the backbone of the unified bound (5.2).

Concentration for Y and $X^\top Y$.

Let $m_t := \mathbb{E}|Z|^t$ for $Z \sim \mathcal{N}(0,1)$.

Lemma 4 (norm of Y)**.** *With $Y := Xw^\star + \xi$ and $\tau_s^2 := \|w^\star\|_2^2 + \sigma^2$, we have*

$$\|Y\|_2^2 = \tau_s^2\, n\,(1 + o(1)) \qquad \textit{w.h.p.}$$

More quantitatively, for every $t > 0$,

$$\Pr\Big(\big|\|Y\|_2^2 - \tau_s^2 n\big| \ge 2\tau_s^2\sqrt{nt} + 2\tau_s^2 t\Big) \;\le\; e^{-t}.$$

Proof. For each row $i \in [n]$, $(Xw^\star)_i = \sum_{j=1}^d w_j^\star X_{i,j}$ is $\mathcal{N}(0, \|w^\star\|_2^2)$ since the $X_{i,j}$ are i.i.d. $\mathcal{N}(0,1)$ and independent in j; the rows are independent. The noise $\xi_i \sim \mathcal{N}(0,\sigma^2)$ is independent of X, hence

$$Y \;\sim\; \mathcal{N}(0, \tau_s^2 I_n), \qquad \frac{\|Y\|_2^2}{\tau_s^2} \;\sim\; \chi_n^2.$$

The standard Laurent–Massart inequality for χ_n^2 variables (see e.g. *Ann. Statist.* 2000) yields, for all $t > 0$,

$$\Pr\Big(\|Y\|_2^2 - \tau_s^2 n \ge 2\tau_s^2\sqrt{nt} + 2\tau_s^2 t\Big) \le e^{-t}, \qquad \Pr\Big(\tau_s^2 n - \|Y\|_2^2 \ge 2\tau_s^2\sqrt{nt}\Big) \le e^{-t}.$$

Taking $t = cn$ gives $\|Y\|_2^2 = \tau_s^2 n(1 + o(1))$ with probability at least $1 - e^{-cn}$. □

Lemma 5 (bulk coordinates of $X^\top Y$)**.** *Conditional on Y, for each $j \notin S$,*

$$\langle X_{:,j}, Y\rangle \;\sim\; \mathcal{N}\Big(0,\; \|Y\|_2^2\Big),$$

and the variables $\{\langle X_{:,j}, Y\rangle\}_{j\notin S}$ are i.i.d. given Y. Consequently, with $m_q := \mathbb{E}|Z|^q$ for $Z \sim \mathcal{N}(0,1)$,

$$\sum_{j\notin S} \Big|\langle X_{:,j}, Y\rangle\Big|^q = (d-s)\, m_q\, \|Y\|_2^q \,\Big(1 + o(1)\Big) \;\asymp\; (d-s)\,\tau_s^q\, n^{q/2} \qquad \textit{w.h.p.}$$

Quantitatively, for any fixed $q \ge 2$ and any $u \in (0,1)$,

$$\Pr\left(\left|\frac{1}{d-s}\sum_{j\notin S}\frac{|\langle X_{:,j}, Y\rangle|^q}{\|Y\|_2^q} - m_q\right| > u \;\middle|\; Y\right) \;\le\; 2\exp\Big(-c_q(d-s)\min\{u^2, u\}\Big).$$

Proof. Fix $j \notin S$. The vector $X_{:,j} \sim \mathcal{N}(0, I_n)$ is independent of $(X_{:,k})_{k\in S}$ and ξ, hence independent of $Y = Xw^\star + \xi$, which depends only on the columns indexed by S and on ξ. Conditional on Y, by rotational invariance,

$$\langle X_{:,j}, Y\rangle \overset{d}{=} \|Y\|_2 \, Z_j, \qquad Z_j \sim \mathcal{N}(0,1),$$

and independence across $j \notin S$ follows from the independence of the columns $\{X_{:,j}\}_{j\notin S}$.

Let $W_j := |Z_j|^q - m_q$. Then W_j are i.i.d. mean-zero and sub-exponential with $\|W_j\|_{\psi_1} \le C_q$ (a standard fact for polynomial functions of a standard Gaussian, see, e.g., Vershynin's *High-Dimensional Probability*). Bernstein's inequality for sub-exponential variables gives, for any $u > 0$,

$$\Pr\left(\left|\frac{1}{d-s}\sum_{j\notin S} W_j\right| > u \;\middle|\; Y\right) \;\le\; 2\exp\Big(-c_q(d-s)\min\{u^2, u\}\Big).$$

Multiplying back by $\|Y\|_2^q$ proves the conditional concentration display. Since $(d-s) \asymp n$ by assumption, taking $u \to 0$ slowly (e.g. $u = \sqrt{(\log n)/(d-s)}$) yields

$$\sum_{j\notin S} |\langle X_{:,j}, Y\rangle|^q = (d-s)m_q \|Y\|_2^q \,(1 + o(1))$$

with probability at least $1 - Ce^{-c(d-s)} \ge 1 - Ce^{-cn}$ (unconditionally). Finally, Lemma 4 implies $\|Y\|_2^q \asymp \tau_s^q n^{q/2}$ w.h.p., completing the proof. □

Lemma 6 (Signal block with integrated uniform column-norm control)**.** *Let $X \in \mathbb{R}^{n\times d}$ have i.i.d. $\mathcal{N}(0,1)$ entries, $S \subset [d]$ with $|S| = s$, and $Y := Xw^\star + \xi$ where $\xi \sim \mathcal{N}(0, \sigma^2 I_n)$ is independent of X. Write $\tau_s^2 := \|w^\star\|_2^2 + \sigma^2$ and $W_q := \sum_{j\in S} |w_j^\star|^q$ for $q \ge 2$.*

(i) Uniform column-norm concentration (over all d columns). *There exists a universal $c \in (0,1)$ such that, for every $u > 0$,*

$$\Pr\left(\max_{1\le j\le d}\left|\frac{\|X_{:,j}\|_2^2}{n} - 1\right| > u\right) \;\le\; 2\,d\,\exp\Big(-c\,n\,\min\{u^2, u\}\Big). \tag{A.12}$$

In particular, for any fixed $\gamma > 0$,

$$u_n := \sqrt{\frac{(1+\gamma)\log d}{c\,n}} \in (0,1] \;\text{ for } n \text{ large, and } \Pr\left(\max_{j\le d}\left|\frac{\|X_{:,j}\|_2^2}{n} - 1\right| > u_n\right) \le 2\,d^{-\gamma}.$$

(ii) Spike decomposition, explicit definition of ζ_j, and q-moment bound. *For each $j \in S$, define*

$$\zeta_j \;:=\; \Big\langle X_{:,j},\; \sum_{k\in S\setminus\{j\}} w_k^\star\, X_{:,k} \;+\; \xi \Big\rangle. \tag{A.13}$$

Then

$$\langle X_{:,j}, Y\rangle \;=\; w_j^\star\, \|X_{:,j}\|_2^2 \;+\; \zeta_j. \tag{A.14}$$

Moreover, for each fixed $j \in S$,

$$\mathbb{E}[\zeta_j \,|\, X_{:,j}] = 0, \qquad \mathrm{Var}(\zeta_j \,|\, X_{:,j}) \;=\; \Big(\tau_s^2 - (w_j^\star)^2\Big)\, \|X_{:,j}\|_2^2, \tag{A.15}$$

and, conditional on $X_{:,j}$,

$$\zeta_j \;\sim\; \mathcal{N}\Big(0,\; \Big(\tau_s^2 - (w_j^\star)^2\Big)\, \|X_{:,j}\|_2^2\Big). \tag{A.16}$$

(We do not assume or use independence between the collection $\{\zeta_j\}_{j\in S}$; the proof below controls their aggregate via operator-norm bounds.) *Consequently, with probability at least $1 - 2d^{-\gamma} - Ce^{-c\sqrt{ns}}$,*

$$\sum_{j\in S} \Big|\langle X_{:,j}, Y\rangle\Big|^q = n^q\, W_q\, (1 + o(1)) \;+\; O\Big(\tau_s^q \Big(s\, n^{q/2} + s^{1+q/2}\Big)\Big), \tag{A.17}$$

where the $o(1)$ (as $n \to \infty$) and the hidden constants depend only on q (hence on p). The mixed term $\Sigma_{j\in S}|a_j|^{q-1}|b_j|$ is absorbed by Young's inequality into the $n^q W_q$ leading term and the $\Sigma_{j\in S}|b_j|^q$ remainder, with a harmless change in constants.

Proof. Part (i): For a fixed j, $Z_j := \|X_{:,j}\|_2^2 \overset{d}{=} \chi_n^2$. By Laurent–Massart, for all $x \geq 0$,

$$\Pr\Big(Z_j - n \geq 2\sqrt{nx} + 2x\Big) \leq e^{-x}, \qquad \Pr\Big(n - Z_j \geq 2\sqrt{nx}\Big) \leq e^{-x}.$$

A standard choice of x (see derivation below) yields the Bernstein-type bound

$$\Pr\Big(\Big|\frac{Z_j}{n} - 1\Big| > u\Big) \;\leq\; 2\exp\Big(-c\, n\, \min\{u^2, u\}\Big) \qquad (\forall u > 0), \tag{A.18}$$

for some universal $c \in (0,1)$. Summing over $j = 1, \ldots, d$ gives (A.12). For the explicit choice $u_n = \sqrt{(1+\gamma)\log d/(cn)} \leq 1$ (for n large),

$$2d\exp\Big(-cnu_n^2\Big) = 2d\exp(-(1+\gamma)\log d) = 2d^{-\gamma}.$$

(Derivation of the Bernstein form): If $u \in (0,1]$, choose $x = \frac{u^2 n}{8}$ to get $\Pr(Z_j - n \geq un) \leq e^{-\frac{u^2 n}{8}}$ and $x = \frac{u^2 n}{4}$ to get $\Pr(n - Z_j \geq un) \leq e^{-\frac{u^2 n}{4}}$. If $u \geq 1$, choose $x = c_0 un$ (e.g. $c_0 = 1/16$) so that $2\sqrt{nx} + 2x \leq un$, hence $\Pr(Z_j - n \geq un) \leq e^{-c_0 un}$. Combine and absorb constants into c.

Part (ii): The decomposition (A.14) is immediate from

$$Y = w_j^\star X_{:,j} + \sum_{k \in S \setminus \{j\}} w_k^\star X_{:,k} + \xi,$$

and independence/rotational invariance: conditional on $X_{:,j}$, $\langle X_{:,j}, X_{:,k} \rangle \sim \mathcal{N}(0, \|X_{:,j}\|_2^2)$ for $k \neq j$ and $\langle X_{:,j}, \xi \rangle \sim \mathcal{N}(0, \sigma^2 \|X_{:,j}\|_2^2)$, all independent. Let $a_j := w_j^\star \|X_{:,j}\|_2^2$ and $b_j := \zeta_j$ so that $\langle X_{:,j}, Y \rangle = a_j + b_j$. We show:

$$\sum_{j \in S} |a_j|^q = n^q W_q (1 + o(1)) \qquad \text{and} \qquad \sum_{j \in S} |b_j|^q \;\lesssim\; s\, \tau_s^q\, n^{q/2},$$

with the stated probability. Conditioned on the event from (i) with $u = u_n = o(1)$,

$$\max_{1 \leq j \leq d} \left| \frac{\|X_{:,j}\|_2^2}{n} - 1 \right| \leq u_n,$$

and by a mean-value bound, $\|X_{:,j}\|_2^{2q} = n^q (1 + O(u_n))$ uniformly in j. Hence

$$\sum_{j \in S} |a_j|^q = \sum_{j \in S} |w_j^\star|^q \, \|X_{:,j}\|_2^{2q} = n^q \sum_{j \in S} |w_j^\star|^q \, (1 + O(u_n)) = n^q W_q \, (1 + o(1)).$$

For any index set $T \subset [d]$, we write $X_{:,T} \in \mathbb{R}^{n \times |T|}$ for the submatrix formed by the columns $\{X_{:,j} : j \in T\}$. When convenient we abbreviate $X_{:,T}$ as X_T. For a vector $w \in \mathbb{R}^d$, w_T denotes its restriction to T, and T^c the complement of T in $[d]$. Let $G := X_S^\top X_S$ and $D := \mathrm{diag}(\|X_{:,j}\|_2^2)_{j \in S}$. Then

$$b = (b_j)_{j \in S} = (G - D)\, w_S^\star + X_S^\top \xi.$$

We bound $\|b\|_2$ and then pass to ℓ_q. Recall $b = (G - D) w_S^\star + X_S^\top \xi$, where $G := X_S^\top X_S \in \mathbb{R}^{s \times s}$ and $D := \mathrm{diag}(\|X_{:,j}\|_2^2)_{j \in S}$.

Bound on $\|(G - D) w_S^\star\|_2$. We have

$$\|(G - D) w_S^\star\|_2 \;\leq\; \|G - D\|_{\mathrm{op}} \, \|w^\star\|_2 \;\leq\; \Big(\|G - n I_s\|_{\mathrm{op}} + \|D - n I_s\|_{\mathrm{op}} \Big) \, \|w^\star\|_2. \quad \text{(A.19)}$$

Singular-value bound for $G - nI_s$. Let $s_{\max}(X_S)$ and $s_{\min}(X_S)$ denote the largest and smallest singular values of X_S. By the standard Gaussian singular-value concentration (see Vershynin, *High-Dimensional Probability*, Thm. 4.6.1), for any $t \geq 0$,

$$\mathbb{P}\Big(s_{\max}(X_S) \leq \sqrt{n} + \sqrt{s} + t, \quad s_{\min}(X_S) \geq \sqrt{n} - \sqrt{s} - t \Big) \;\geq\; 1 - 2e^{-t^2/2}. \tag{A.20}$$

Conditioned on this event,

$$\begin{aligned} \|G - nI_s\|_{\mathrm{op}} &= \max\Big\{ s_{\max}(X_S)^2 - n,\; n - s_{\min}(X_S)^2 \Big\} \\ &\leq \Big(\sqrt{n} + \sqrt{s} + t\Big)^2 - n \quad \vee \quad n - \Big(\sqrt{n} - \sqrt{s} - t\Big)^2 \\ &\leq s + 2\sqrt{ns} + 2t(\sqrt{n} + \sqrt{s}) + t^2. \end{aligned} \tag{A.21}$$

Choosing $t = \sqrt{s}$ in (A.20)–(A.21) yields, with probability at least $1 - 2e^{-s/2}$,

$$\|G - nI_s\|_{\mathrm{op}} \;\leq\; s + 2\sqrt{ns} + 2\sqrt{s}(\sqrt{n} + \sqrt{s}) + s \;\leq\; 4\sqrt{ns} + 4s. \tag{A.22}$$

Diagonal bound for $D - nI_s$. By the single-column deviation bound (A.18), for any $u > 0$ and any $j \in S$,

$$\Pr\left(\left| \frac{\|X_{:,j}\|_2^2}{n} - 1 \right| > u \right) \;\leq\; 2\exp\Big(- c\, n\, \min\{u^2, u\} \Big).$$

Union-bounding this over the s indices $j \in S$ and taking

$$u_S \;:=\; \sqrt{\frac{s}{n}}, \tag{A.23}$$

we obtain

$$\mathbb{P}\left(\max_{j \in S} \left| \frac{\|X_{:,j}\|_2^2}{n} - 1 \right| > u_S \right) \;\leq\; \begin{cases} C\, e^{-c\, s}, & s \leq n, \\ C\, e^{-c'\sqrt{ns}}, & s > n. \end{cases} \tag{A.24}$$

hence, on this event,

$$\|D - nI_s\|_{\mathrm{op}} \;=\; \max_{j \in S} \Big| \|X_{:,j}\|_2^2 - n \Big| \;\leq\; n u_S \;=\; \sqrt{ns}. \tag{A.25}$$

Combining (A.19), (A.22), and (A.25), we arrive at

$$\|(G - D)w_S^\star\|_2 \;\leq\; \Big(4\sqrt{ns} + 4s + \sqrt{ns}\Big)\, \|w^\star\|_2 \;\leq\; \Big(5\sqrt{ns} + 4s\Big)\, \|w^\star\|_2, \tag{A.26}$$

with probability at least $1 - 2e^{-s/2} - Ce^{-c'\sqrt{ns}}$.

Now we bound $\|X_S^\top \xi\|_2$. Conditionally on X_S, the vector $X_S^\top \xi$ is Gaussian with covariance

$$\Sigma \;:=\; \mathrm{Var}\Big(X_S^\top \xi \,\Big|\, X_S\Big) \;=\; \sigma^2\, G.$$

Write the eigenvalues of G as $\mu_1, \dots, \mu_s \geq 0$. Then

$$\|X_S^\top \xi\|_2^2 \;\overset{d}{=}\; \sum_{i=1}^{s} \lambda_i\, Z_i^2, \qquad \lambda_i := \sigma^2 \mu_i, \quad Z_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

The weighted χ^2 tail of Laurent–Massart (2000, Lemma 1) states that for all $x \geq 0$,

$$\mathbb{P}\left(\sum_{i=1}^{s} \lambda_i Z_i^2 \;\geq\; \sum_{i=1}^{s} \lambda_i \;+\; 2\sqrt{\Big(\sum_{i=1}^{s} \lambda_i^2\Big)\, x} \;+\; 2\Big(\max_i \lambda_i\Big)\, x \;\Big|\; X_S \right) \;\leq\; e^{-x}. \tag{A.27}$$

Since $\sum_i \lambda_i = \sigma^2 \mathrm{tr}(G)$, $\sum_i \lambda_i^2 = \sigma^4 \mathrm{tr}(G^2) \leq \sigma^4 s\, \|G\|_{\mathrm{op}}^2$, and $\max_i \lambda_i = \sigma^2 \|G\|_{\mathrm{op}}$, inserting these into (A.27) and choosing $x = s$ gives, with conditional probability $\geq 1 - e^{-s}$,

$$\|X_S^\top \xi\|_2^2 \;\leq\; \sigma^2 \Big(\mathrm{tr}(G) \;+\; 4s\, \|G\|_{\mathrm{op}} \Big). \tag{A.28}$$

We now bound $\mathrm{tr}(G)$ and $\|G\|_{\mathrm{op}}$ on the events already used in Step A. First, by (A.23)–(A.24),

$$\mathrm{tr}(G) \;=\; \sum_{j \in S} \|X_{:,j}\|_2^2 \;\leq\; s\, n\, (1 + u_S) \;=\; s\, n + s\sqrt{ns}. \tag{A.29}$$

Second, from (A.20) with $t = \sqrt{s}$,

$$\|G\|_{\mathrm{op}} \;=\; s_{\max}(X_S)^2 \;\leq\; \Big(\sqrt{n} + \sqrt{s} + \sqrt{s}\Big)^2 \;\leq\; n \;+\; 4\sqrt{ns} \;+\; 4s. \tag{A.30}$$

Plugging (A.29)–(A.30) into (A.28) and taking square roots, we obtain

$$\begin{aligned} \|X_S^\top \xi\|_2 &\leq \sigma \sqrt{s\, n + s\sqrt{ns} \;+\; 4s\,(n + 4\sqrt{ns} + 4s)} \\ &\leq \sigma \Big(\sqrt{sn} \;+\; \sqrt{s\sqrt{ns}} \;+\; 2\sqrt{sn} \;+\; 4s \Big) \\ &\leq C\, \sigma \Big(\sqrt{sn} + s \Big), \end{aligned} \tag{A.31}$$

where in the last step we used $\sqrt{s\sqrt{ns}} = s^{3/4} n^{1/4} \leq \frac{1}{2}(\sqrt{sn} + s)$.

ℓ_2 and ℓ_q bounds for b. Combining (A.26) and (A.31),

$$\|b\|_2 \;\leq\; \|(G - D) w_S^\star\|_2 + \|X_S^\top \xi\|_2 \;\leq\; C \Big(\sqrt{ns}\, \|w^\star\|_2 + s\, \|w^\star\|_2 + \sigma\sqrt{sn} + \sigma s \Big). \tag{A.32}$$

In particular, when $s \le n$ the s terms are dominated by $\sqrt{ns}$ and

$$\|b\|_2 \;\le\; C\,\tau_s\,\sqrt{sn} \qquad (\text{since } \tau_s^2 = \|w^\star\|_2^2 + \sigma^2). \tag{A.33}$$

(Refined q-moment bound via decoupling). Introduce i.i.d. "ghost" columns $\{X'_{:,j}\}_{j\in S}$ independent of (X,ξ) and set

$$\zeta'_j := \langle X'_{:,j},\, u_j\rangle, \qquad u_j := X_{:,S\setminus\{j\}}\, w^\star_{S\setminus\{j\}} + \xi.$$

By a standard decoupling inequality for Gaussian chaos of order two (de la Peña and Giné, *Decoupling: From Dependence to Independence*, 1999, Thm. 3.5.3), there exists $C_q < \infty$ (depending only on q) such that for all $t > 0$,

$$\mathbb{P}\Big(\sum_{j\in S} |\zeta_j|^q > t\Big) \;\le\; C_q\,\mathbb{P}\Big(\sum_{j\in S} |\zeta'_j|^q > t/C_q\Big).$$

Conditional on $\{u_j\}$, the variables $\{\zeta'_j\}_{j\in S}$ are independent centered Gaussians with variances $\|u_j\|_2^2$. On the singular-value and column-norm events used above (cf. (A.20) with $t = \sqrt{s}$ and (A.12)), uniformly in j,

$$\|u_j\|_2^2 \;\le\; \|X_{:,S}\|_{\mathrm{op}}^2\,\|w^\star\|_2^2 + \|\xi\|_2^2 \;\le\; C\Big(n + 4\sqrt{ns} + 4s\Big)\|w^\star\|_2^2 + C\sigma^2 n \;\le\; C\,\tau_s^2\,(n+s).$$

Hence, conditionally on $\{u_j\}$, each $|\zeta'_j|^q$ is sub-exponential with ψ_1-norm $\le C\,\tau_s^q\,(n+s)^{q/2}$. Bernstein's inequality then yields

$$\sum_{j\in S} |\zeta'_j|^q \;\le\; C\,\tau_s^q\Big(s\,n^{q/2} + s^{1+q/2}\Big) \qquad \text{with conditional probability at least } 1 - Ce^{-cs}.$$

Unconditioning and applying decoupling gives, with probability at least $1 - 2d^{-\gamma} - Ce^{-cs}$,

$$\sum_{j\in S} |b_j|^q \;=\; \sum_{j\in S} |\zeta_j|^q \;\le\; C\,\tau_s^q\Big(s\,n^{q/2} + s^{1+q/2}\Big). \tag{A.34}$$

In particular, if $s \le n$ this simplifies to $\sum_{j\in S} |b_j|^q \le C\,s\,\tau_s^q\,n^{q/2}$.

For the cross term, for $q \ge 2$ and any $a, b \in \mathbb{R}$ we have the elementary inequality

$$\Big||a+b|^q - |a|^q\Big| \;\le\; C_q\Big(|a|^{q-1}|b| + |b|^q\Big) \;\le\; C_q\Big(|a|^{q-2}b^2 + |b|^q\Big), \tag{A.35}$$

for a constant C_q depending only on q. Summing (A.35) over $j \in S$ with $a_j = w_j^\star \|X_{:,j}\|_2^2$ and $b_j = \zeta_j$, and applying Hölder,

$$\sum_{j\in S} \Big| |a_j + b_j|^q - |a_j|^q \Big| \le C_q \sum_{j\in S} |a_j|^{q-1}|b_j| + C_q \sum_{j\in S} |b_j|^q$$
$$\le C_q \Big(\sum_{j\in S} |a_j|^q \Big)^{\frac{q-1}{q}} \Big(\sum_{j\in S} |b_j|^q \Big)^{\frac{1}{q}} \; + \; C_q \sum_{j\in S} |b_j|^q. \tag{A.36}$$

Set

$$A := \sum_{j\in S} |a_j|^q, \qquad B := \sum_{j\in S} |b_j|^q.$$

Apply Young's inequality with conjugate exponents $r = \frac{q}{q-1}$ and $s = q$: for any $\varepsilon > 0$,

$$A^{\frac{q-1}{q}} B^{\frac{1}{q}} \;\le\; \frac{\varepsilon}{r} A \;+\; \frac{\varepsilon^{-(q-1)}}{s} B \;=\; \frac{q-1}{q} \varepsilon A \;+\; \frac{1}{q} \varepsilon^{-(q-1)} B. \tag{A.37}$$

With $A = n^q W_q(1 + O(u_n))$ and the bound $B \le C\, \tau_s^q (s\, n^{q/2} + s^{1+q/2})$ from (A.34), choosing a fixed $\varepsilon \in (0,1)$ (e.g. $\varepsilon = \frac{1}{2}$) absorbs the mixed term into the leading A and the B-remainder (with a harmless change of constants). Consequently,

$$\sum_{j\in S} \Big| |a_j + b_j|^q - |a_j|^q \Big| = O\Big(\tau_s^q \big(s\, n^{q/2} + s^{1+q/2} \big) \Big),$$

which yields (A.17). When $s \le n$ the remainder simplifies to $O\big(s\, \tau_s^q\, n^{q/2}\big)$. □

Combining Lemmas 5–6 yields the decomposition

$$\|X^\top Y\|_q^q \;=\; n^q\, W_q\, (1+o(1)) + (d-s)\, m_q\, \tau_s^q\, n^{q/2}\, (1+o(1)) + O\Big(\tau_s^q \big(s\, n^{q/2} + s^{1+q/2} \big) \Big) \qquad \text{w.h.p.} \tag{A.38}$$

Bulk ℓ_q-embedding and Gaussian ℓ_t relations.

Lemma 7 (uniform ℓ_q control on the bulk operator)**.** *Let $q \in [2, \infty)$ and assume $\kappa_{\text{bulk}} := \liminf_{n\to\infty} \frac{d-s}{n} > 0$. There exist constants $0 < c_q \le C_q < \infty$, depending only on $(q, \kappa_{\text{bulk}})$, such that, with probability at least $1 - Ce^{-cn}$, simultaneously for all $\lambda \in \mathbb{R}^n$,*

$$c_q\, (d-s)\, \|\lambda\|_2^q \;\le\; \sum_{j\notin S} \big| \langle X_{:,j}, \lambda \rangle \big|^q \;\le\; C_q\, (d-s)\, \|\lambda\|_2^q. \tag{A.39}$$

(Here we absorb the Gaussian absolute moment $m_q = \mathbb{E}|Z|^q$ into the constants c_q, C_q; in (A.40) we keep m_t explicit.) *Moreover, for every $t \in [1, q]$, there exist constants $0 < c_t \le C_t < \infty$, depending only on $(t, \kappa_{\text{bulk}})$, such that, w.h.p., uniformly in $\lambda \in \mathbb{R}^n$,*

$$c_t^{1/t}\,(d-s)^{1/t}\,m_t^{1/t}\,\|\lambda\|_2 \;\le\; \Big\|\big(\,|\langle X_{:,j},\lambda\rangle|\,\big)_{j\notin S}\Big\|_t \;\le\; C_t^{1/t}\,(d-s)^{1/t}\,m_t^{1/t}\,\|\lambda\|_2, \tag{A.40}$$

where $m_t := \mathbb{E}|Z|^t$ for $Z \sim \mathcal{N}(0,1)$.

Proof. Fix $\lambda \in \mathbb{R}^n$, and if $\lambda \neq 0$ write $u := \lambda/\|\lambda\|_2 \in \mathbb{S}^{n-1}$. By homogeneity,

$$\sum_{j\notin S} |\,\langle X_{:,j}|\lambda\rangle\,|^q = \|\lambda\|_2^q \sum_{j\notin S} |\,\langle X_{:,j}|u\rangle\,|^q, \tag{A.41}$$

and similarly for any $t \in [1, q]$,

$$\Big\|(|\,\langle X_{:,j}|\lambda\rangle\,|)_{j\notin S}\Big\|_t = \|\lambda\|_2 \Big(\sum_{j\notin S} |\,\langle X_{:,j}|u\rangle\,|^t\Big)^{1/t}. \tag{A.42}$$

Thus it suffices to prove the bounds for unit u.

Let $T := S^c$ and $m := |T| = d - s$. Fix $u \in \mathbb{S}^{n-1}$ and $t \in [1, q]$. Define

$$Y_j^{(t)}(u) \;:=\; \big|\langle X_{:,j}, u\rangle\big|^t, \qquad j \in T.$$

Since the columns $\{X_{:,j}\}_{j\in T}$ are i.i.d. $\mathcal{N}(0, I_n)$ and independent of u, the random variables $\{Y_j^{(t)}(u)\}_{j\in T}$ are i.i.d.

Definition 12 (Orlicz ψ_ν norm and sub-Weibull class)**.** *For $\nu \in (0, 2]$ and a real random variable Z, the Orlicz norm*

$$\|Z\|_{\psi_\nu} \;:=\; \inf\Big\{K > 0 :\; \mathbb{E}\exp\Big(\frac{|Z|^\nu}{K^\nu}\Big) \le 2\Big\}.$$

If $\|Z\|_{\psi_\nu} < \infty$, we say Z is sub-Weibull of order ν. *Special cases: $\nu = 2$ (sub-Gaussian) and $\nu = 1$ (sub-Exponential). Two basic properties we use are*

$$\mathbb{P}(|Z| > x) \;\le\; 2\exp\Big(-c\,(x/\|Z\|_{\psi_\nu})^\nu\Big) \quad (\forall x \ge 0), \tag{A.43}$$

$$\|Z - \mathbb{E}Z\|_{\psi_\nu} \;\le\; 2\,\|Z\|_{\psi_\nu}. \tag{A.44}$$

Definition 13 (Gaussian absolute moment)**.** *For $t > 0$, let $Z \sim \mathcal{N}(0,1)$ and define*

$$m_t \;:=\; \mathbb{E}|Z|^t \;=\; 2^{t/2}\,\frac{\Gamma\!\left(\frac{t+1}{2}\right)}{\sqrt{\pi}}.$$

Classification of $Y_j^{(t)}(u)$ in ψ_ν (with explicit mgf computation). Since $\langle X_{:,j}, u\rangle \sim \mathcal{N}(0,1)$, write $Z \sim \mathcal{N}(0,1)$ and set $W := |Z|^t$. For any $K > 0$,

$$\left(\frac{W}{K}\right)^{2/t} = \left(\frac{|Z|^t}{K}\right)^{2/t} = \frac{|Z|^2}{K^{2/t}}.$$

Let

$$\theta \;:=\; \frac{1}{K^{2/t}}.$$

Then

$$\mathbb{E}\exp\!\Big(\big(W/K\big)^{2/t}\Big) = \mathbb{E}\exp\!\big(\theta\, Z^2\big).$$

Compute this expectation explicitly: using the standard normal density $\varphi(z) = (2\pi)^{-1/2}e^{-z^2/2}$,

$$\begin{aligned}\mathbb{E}\big[e^{\theta Z^2}\big] &= \int_{\mathbb{R}} e^{\theta z^2}\,\varphi(z)\,dz = \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} e^{\theta z^2}\,e^{-z^2/2}\,dz \\ &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} e^{-(\frac{1}{2}-\theta)\,z^2}\,dz \;=\; \frac{1}{\sqrt{2\pi}}\cdot\sqrt{\frac{\pi}{\frac{1}{2}-\theta}} \;=\; \frac{1}{\sqrt{1-2\theta}}, \qquad \text{for } \;\theta < \tfrac{1}{2}.\end{aligned} \tag{A.45}$$

Equivalently, since $Z^2 \sim \chi^2_1$, the mgf of χ^2_1 is $(1-2\theta)^{-1/2}$ for $\theta < 1/2$, which matches (A.45).

We now choose K so that $\theta < 1/2$ and the expectation is uniformly bounded by a constant ≤ 2. Take

$$K_t \;:=\; (4t)^{t/2} \quad\Longrightarrow\quad \theta = \frac{1}{K_t^{2/t}} = \frac{1}{4t} \;<\; \frac{1}{2} \qquad (t \ge 1). \tag{A.46}$$

Then, by (A.45),

$$\mathbb{E}\exp\!\Big(\big(W/K_t\big)^{2/t}\Big) = \frac{1}{\sqrt{1-\frac{2}{K_t^{2/t}}}} = \frac{1}{\sqrt{1-\frac{1}{2t}}} \;\le\; \frac{1}{\sqrt{1-\frac{1}{2}}} = \sqrt{2} \;<\; 2, \tag{A.47}$$

where we used $t \in [1, q]$ (hence $t \ge 1$). By the definition of the Orlicz norm,

$$\|\,|Z|^t\,\|_{\psi_{2/t}} \;\le\; K_t \;=\; (4t)^{t/2}. \tag{A.48}$$

Centering preserves the class up to a factor 2 (by (A.44)), hence

$$\| \, |Z|^t - m_t \, \|_{\psi_{2/t}} \;\le\; 2K_t \;=\; 2(4t)^{t/2}. \tag{A.49}$$

Finally, define

$$\nu(t) \;:=\; \min\{1,\; 2/t\}. \tag{A.50}$$

Since $2/t \ge 1$ for $t \le 2$ and $2/t < 1$ for $t > 2$, combining (A.49) with (A.50) yields the uniform (in u) classification

$$\| \, Y_j^{(t)}(u) - m_t \, \|_{\psi_{\nu(t)}} \;\le\; K_t' \qquad \text{with } K_t' := 2(4t)^{t/2}. \tag{A.51}$$

This bound is uniform in u because $\langle X_{:,j}, u\rangle \overset{d}{=} \mathcal{N}(0,1)$ for every fixed $u \in \mathbb{S}^{n-1}$.

Empirical-mean concentration at fixed u. From (A.51) and independence across $j \in T$, a Bernstein-type inequality for sums of i.i.d. sub-Weibull(ν) variables (e.g. Theorem 3.1 in Kuchibhotla–Basu, 2018) yields, for any $\varepsilon > 0$,

$$\mathbb{P}\left(\left|\frac{1}{m}\sum_{j\in T}\left(Y_j^{(t)}(u) - m_t\right)\right| > \varepsilon\right) \;\le\; 2\exp\left\{-c_{\nu(t)}\, m\, \min\left(\frac{\varepsilon^2}{K_t'^2},\; \left(\frac{\varepsilon}{K_t'}\right)^{\nu(t)}\right)\right\}. \tag{A.52}$$

Taking $\varepsilon = \delta\, m_t$ with $\delta \in (0,1)$, and absorbing the fixed ratio m_t/K_t' (which depends only on t) into the constant, we obtain

$$\mathbb{P}\Big(\Big|\frac{1}{m}\sum_{j\in T} Y_j^{(t)}(u) - m_t\Big| > \delta\, m_t\Big) \;\le\; 2\exp\Big(-c_t\, m\, \min\{\delta^2,\; \delta^{\nu(t)}\}\Big), \tag{A.53}$$

where $c_t > 0$ depends only on t (hence only on p). In the sub-Exponential range $t \in [1,2]$, $\nu(t) = 1$ and (A.53) simplifies to

$$\mathbb{P}\Big(\Big|\frac{1}{m}\sum_{j\in T} Y_j^{(t)}(u) - m_t\Big| > \delta\, m_t\Big) \;\le\; 2\exp\Big(-c_t\, m\, \min\{\delta^2, \delta\}\Big). \tag{A.54}$$

Finally, note that

$$\mathbb{E}Y_j^{(t)}(u) \;=\; m_t, \tag{A.55}$$

by Definition 13, completing Step 1.

Now we can construct a net on the sphere and a uniform bound on that net. Let $\varepsilon \in (0, 1/8]$ be a fixed absolute constant (to be chosen below). There exists an ε-net $\mathcal{N}_\varepsilon \subset \mathbb{S}^{n-1}$ with

$$|\mathcal{N}_\varepsilon| \;\le\; \left(1 + \frac{2}{\varepsilon}\right)^n \;\le\; C_\varepsilon^n. \tag{A.56}$$

Applying (A.53) with $\delta = \delta_t \in (0, 1/4]$ (a small absolute constant depending only on t) and union-bounding over $\mathcal{N}_\varepsilon$ yields

$$\mathbb{P}\left(\exists\, v \in \mathcal{N}_\varepsilon :\ \left|\frac{1}{m}\sum_{j\in T} Y_j^{(t)}(v) - m_t\right| > \delta_t\, m_t\right) \le 2\,|\mathcal{N}_\varepsilon|\,\exp\Big(-c_t\, m\,\min\{\delta_t^2, \delta_t\}\Big)$$
$$\le 2\,\exp\Big(n\log C_\varepsilon - c_t' m\Big). \qquad \text{(A.57)}$$

Because $m \ge \kappa_{\text{bulk}} n$ and $\kappa_{\text{bulk}} > 0$, by taking δ_t fixed (e.g. $\delta_t = 1/4$) and ε fixed (e.g. $\varepsilon = 1/8$), the right-hand side of (A.57) is $\le Ce^{-cn}$. Therefore, with probability at least $1 - Ce^{-cn}$, simultaneously for all $v \in \mathcal{N}_\varepsilon$,

$$(1-\delta_t)\, m_t \ \le\ \frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|v\rangle\,|^t \ \le\ (1+\delta_t)\, m_t. \qquad \text{(A.58)}$$

We are ready to extend from the net to the whole sphere. Fix arbitrary $u \in \mathbb{S}^{n-1}$ and pick $v \in \mathcal{N}_\varepsilon$ with $\|u - v\|_2 \le \varepsilon$. For any $a, b \in \mathbb{R}$ and any $t \ge 1$, the elementary inequalities

$$|a+b|^t \ \le\ 2^{t-1}\Big(|a|^t + |b|^t\Big), \qquad |a|^t \ \le\ 2^{t-1}\Big(|a+b|^t + |b|^t\Big) \qquad \text{(A.59)}$$

hold. Applying (A.59) with $a = \langle X_{:,j}|v\rangle$ and $b = \langle X_{:,j}|u - v\rangle$, we get

$$|\,\langle X_{:,j}|u\rangle\,|^t \le 2^{t-1}\Big(|\,\langle X_{:,j}|v\rangle\,|^t + |\,\langle X_{:,j}|u-v\rangle\,|^t\Big), \qquad \text{(A.60)}$$
$$|\,\langle X_{:,j}|u\rangle\,|^t \ge 2^{1-t}\,|\,\langle X_{:,j}|v\rangle\,|^t - |\,\langle X_{:,j}|u-v\rangle\,|^t. \qquad \text{(A.61)}$$

Average (A.60) and (A.61) over $j \in T$ and divide by m to obtain

$$\frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|u\rangle\,|^t \le 2^{t-1}\left(\frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|v\rangle\,|^t + \frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|u-v\rangle\,|^t\right), \qquad \text{(A.62)}$$
$$\frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|u\rangle\,|^t \ge 2^{1-t}\,\frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|v\rangle\,|^t - \frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|u-v\rangle\,|^t. \qquad \text{(A.63)}$$

For any $w \in \mathbb{R}^n$,

$$\frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|w\rangle\,|^t = \|w\|_2^t \cdot \frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|\widehat{w}\rangle\,|^t, \qquad \widehat{w} := \frac{w}{\|w\|_2} \ \ (\text{if } w \ne 0). \qquad \text{(A.64)}$$

Define the functional and its extremal values

$$A(u) := \frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|u\rangle\,|^t, \qquad S := \sup_{u\in\mathbb{S}^{n-1}} A(u), \qquad I := \inf_{u\in\mathbb{S}^{n-1}} A(u).$$

By (A.64) and $\|u-v\|_2 \le \varepsilon$,

$$\frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|u-v\rangle\,|^t = \|u-v\|_2^t \cdot \frac{1}{m}\sum_{j\in T} |\,\Big\langle X_{:,j}\Big|\widehat{u-v}\Big\rangle\,|^t \le \varepsilon^t\, S,$$

where we used the definition of S in the last inequality. On the event (A.58) (from Step 2), $A(v) \in [(1-\delta_t)m_t, (1+\delta_t)m_t]$ for every $v \in \mathcal{N}_\varepsilon$. Plugging these into (A.62)-(A.63) yields

$$A(u) \le 2^{t-1}\Big(A(v) + \varepsilon^t S\Big),$$
$$A(u) \ge 2^{1-t}A(v) - \varepsilon^t S.$$

Taking the supremum over $u \in \mathbb{S}^{n-1}$ in the upper bound:

$$S \;\le\; 2^{t-1}\Big((1+\delta_t)m_t + \varepsilon^t S\Big) \quad\Longrightarrow\quad S \;\le\; \frac{2^{t-1}}{1-2^{t-1}\varepsilon^t}\,(1+\delta_t)\,m_t.$$

Taking the infimum over $u \in \mathbb{S}^{n-1}$ in the lower bound:

$$I \;\ge\; 2^{1-t}(1-\delta_t)m_t \;-\; \varepsilon^t S.$$

Choose fixed $\delta_t \le \frac{1}{4}$ and $\varepsilon \le \frac{1}{8}$; then

$$2^{t-1}\varepsilon^t \;=\; \frac{(2\varepsilon)^t}{2} \;\le\; \frac{(1/4)^t}{2} \;\le\; \frac{1}{8},$$

so $1-2^{t-1}\varepsilon^t \ge 7/8$ and thus

$$S \;\le\; \frac{2^{t-1}}{7/8}\,(1+\delta_t)m_t \;\le\; C_t\, m_t,$$

for a constant $C_t < \infty$ depending only on t. Substituting this bound for S back into the inequality for I gives

$$I \;\ge\; 2^{1-t}(1-\delta_t)m_t - \varepsilon^t C_t m_t \;\ge\; c_t\, m_t,$$

for some $c_t > 0$ (depending only on t). Therefore, with probability at least $1-Ce^{-cn}$,

$$c_t\, m_t \;\le\; \frac{1}{m}\sum_{j\in T} |\,\langle X_{:,j}|u\rangle\,|^t \;\le\; C_t\, m_t \qquad \text{simultaneously for all } u \in \mathbb{S}^{n-1}. \quad \text{(A.65)}$$

Multiplying (A.65) by $m = d-s$ and using (A.41) with $t = q$ yields

$$c_q\,(d-s)\,\|\lambda\|_2^q \;\le\; \sum_{j\notin S} |\,\langle X_{:,j}|\lambda\rangle\,|^q \;\le\; C_q\,(d-s)\,\|\lambda\|_2^q,$$

which is (A.39). Likewise, combining (A.65) with (A.42) gives

$$c_t^{1/t}\,(d-s)^{1/t}\,m_t^{1/t}\,\|\lambda\|_2 \;\le\; \Big\|(|\,\langle X_{:,j}|\lambda\rangle\,|)_{j\notin S}\Big\|_t \;\le\; C_t^{1/t}\,(d-s)^{1/t}\,m_t^{1/t}\,\|\lambda\|_2,$$

which is (A.40). □

Spike ℓ_t control for $X^\top Y$

Lemma 8 (spike ℓ_t control for $X^\top Y$)**.** *Fix any $t \in [1, q]$ and $\gamma > 0$. With probability at least $1 - 2d^{-\gamma} - Ce^{-cs}$,*

$$\Big\| \big(|\langle X_{:,j}, Y \rangle| \big)_{j \in S} \Big\|_t \;=\; n \, \|w^\star\|_t \Big(1 + O(u_n)\Big) \;\pm C\, \tau_s \Big(\sqrt{n}\, s^{\max\{1/t,\, 1/2\}} + s^{\,1 + (1/t - 1/2)_+} \Big), \tag{A.66}$$

where $u_n := \sqrt{(1+\gamma)\log d / (c\, n)} = o(1)$ and $(x)_+ := \max\{x, 0\}$. In particular, if $s \le n$ then the error simplifies to

$$\Big\| \big(|\langle X_{:,j}, Y \rangle| \big)_{j \in S} \Big\|_t \;=\; n \, \|w^\star\|_t \Big(1 + O(u_n)\Big) \;\pm\; C\, \tau_s \, \sqrt{n}\, s^{\max\{1/t,\, 1/2\}}. \tag{A.67}$$

All constants may depend on t (hence on p) but not on (n, d, s).

Proof. For each $j \in S$,

$$\langle X_{:,j}, Y \rangle = w_j^\star \, \|X_{:,j}\|_2^2 \;+\; \zeta_j, \qquad \zeta_j := \Big\langle X_{:,j}, \sum_{k \in S \setminus \{j\}} w_k^\star X_{:,k} + \xi \Big\rangle. \tag{A.68}$$

Conditional on $X_{:,j}$,

$$\mathbb{E}[\zeta_j \mid X_{:,j}] = 0, \qquad \mathrm{Var}(\zeta_j \mid X_{:,j}) = (\tau_s^2 - (w_j^\star)^2)\, \|X_{:,j}\|_2^2, \tag{A.69}$$

and $\zeta_j \mid X_{:,j} \sim \mathcal{N}(0, (\tau_s^2 - (w_j^\star)^2)\|X_{:,j}\|_2^2)$ by independence and rotational invariance. Define

$$a_j := w_j^\star \|X_{:,j}\|_2^2, \qquad b_j := \zeta_j, \qquad a := (a_j)_{j \in S},\; b := (b_j)_{j \in S}.$$

By the uniform column-norm bound (A.12) with $u = u_n = o(1)$, we have

$$\max_{1 \le j \le d} \left| \frac{\|X_{:,j}\|_2^2}{n} - 1 \right| \;\le\; u_n \qquad \text{with probability at least } 1 - 2d^{-\gamma}. \tag{A.70}$$

On this event,

$$\begin{aligned} \big\| (|a_j|)_{j \in S} \big\|_{\ell_t} &= \Big(\sum_{j \in S} |w_j^\star|^t \, \|X_{:,j}\|_2^{2t} \Big)^{1/t} = n \Big(\sum_{j \in S} |w_j^\star|^t \, (1 + O(u_n))^t \Big)^{1/t} \\ &= n \, \|w^\star\|_t \Big(1 + O(u_n)\Big). \end{aligned} \tag{A.71}$$

Let X_S be the $n \times s$ submatrix with columns $\{X_{:,j}\}_{j \in S}$, and set

$$G := X_S^\top X_S, \qquad D := \mathrm{diag}\Big(\|X_{:,j}\|_2^2 \Big)_{j \in S}.$$

From (A.68), in vector form

$$b = (G - D)\, w_S^\star \;+\; X_S^\top \xi. \tag{A.72}$$

We bound the two terms separately.

(i) Control of $(G - D)w_S^\star$. By the triangle inequality and operator norm submultiplicativity,

$$\|(G-D)w_S^\star\|_2 \;\le\; \|G-D\|_{\mathrm{op}}\,\|w^\star\|_2 \;\le\; \Big(\|G-nI_s\|_{\mathrm{op}} + \|D-nI_s\|_{\mathrm{op}}\Big)\|w^\star\|_2. \tag{A.73}$$

Gaussian singular-value concentration (Vershynin, HDP, Thm. 4.6.1) gives, for any $t \ge 0$,

$$\mathbb{P}\Big(s_{\max}(X_S) \le \sqrt{n} + \sqrt{s} + t, \quad s_{\min}(X_S) \ge \sqrt{n} - \sqrt{s} - t\Big) \;\ge\; 1 - 2e^{-t^2/2}. \tag{A.74}$$

On this event,

$$\begin{aligned} \|G - nI_s\|_{\mathrm{op}} &= \max\Big\{\, s_{\max}(X_S)^2 - n,\; n - s_{\min}(X_S)^2 \,\Big\} \\ &\le (\sqrt{n} + \sqrt{s} + t)^2 - n \quad \vee \quad n - (\sqrt{n} - \sqrt{s} - t)^2 \\ &\le s + 2\sqrt{ns} + 2t(\sqrt{n} + \sqrt{s}) + t^2. \end{aligned} \tag{A.75}$$

Taking $t = \sqrt{s}$ yields, with probability $\ge 1 - 2e^{-s/2}$,

$$\|G - nI_s\|_{\mathrm{op}} \;\le\; 4\sqrt{ns} + 4s. \tag{A.76}$$

By the S-only column-norm event (A.24) (with $u_S = \sqrt{s/n}$),

$$\|D - nI_s\|_{\mathrm{op}} = \max_{j \in S} \Big|\|X_{:,j}\|_2^2 - n\Big| \;\le\; nu_S \;=\; \sqrt{ns}.$$

Combining this with (A.73) and (A.76) yields

$$\|(G-D)w_S^\star\|_2 \;\le\; C\,(\sqrt{ns}+s)\,\|w^\star\|_2 \qquad \text{with probability at least } 1-2e^{-s/2}-Ce^{-c\sqrt{ns}}. \tag{A.77}$$

(ii) Control of $X_S^\top \xi$. Conditionally on X_S, one has $X_S^\top \xi \sim \mathcal{N}(0, \sigma^2 G)$. Writing $\{\mu_i\}_{i=1}^s$ for the eigenvalues of G and $\lambda_i := \sigma^2 \mu_i$, Laurent–Massart's weighted χ^2 tail (2000, Lemma 1) yields, for all $x \ge 0$,

$$\mathbb{P}\left(\sum_{i=1}^{s} \lambda_i Z_i^2 \ge \sum_i \lambda_i + 2\sqrt{\Big(\sum_i \lambda_i^2\Big)x} + 2(\max_i \lambda_i)\, x \;\Big|\; X_S\right) \;\le\; e^{-x}. \tag{A.78}$$

Using $\sum_i \lambda_i = \sigma^2 \mathrm{tr}(G)$, $\sum_i \lambda_i^2 \le \sigma^4 s \, \|G\|_{\mathrm{op}}^2$, and $\max_i \lambda_i = \sigma^2 \|G\|_{\mathrm{op}}$, and taking $x = s$ gives, with conditional probability $\ge 1 - e^{-s}$,

$$\|X_S^\top \xi\|_2^2 \;\le\; \sigma^2 \Big(\mathrm{tr}(G) + 4s \, \|G\|_{\mathrm{op}} \Big). \tag{A.79}$$

On the event (A.74) with $t = \sqrt{s}$ and (A.70),

$$\mathrm{tr}(G) = \sum_{j \in S} \|X_{:,j}\|_2^2 \;\le\; sn(1+u_n) = sn + o(sn), \quad \|G\|_{\mathrm{op}} = s_{\max}(X_S)^2 \;\le\; n + 4\sqrt{ns} + 4s. \tag{A.80}$$

Plugging (A.80) into (A.79) and taking square roots,

$$\|X_S^\top \xi\|_2 \;\le\; C \, \sigma \Big(\sqrt{sn} + s \Big) \qquad \text{with prob.} \ge 1 - 2e^{-s/2} - e^{-s}. \tag{A.81}$$

Combining (A.77), (A.81), and (A.72),

$$\|b\|_2 \;\le\; C \, \tau_s \, (\sqrt{sn} + s) \qquad \text{with prob.} \ge 1 - 2d^{-\gamma} - Ce^{-cs}. \tag{A.82}$$

For $t \in [1, 2]$, the norm monotonicity in $\mathbb{R}^s$ gives

$$\|b\|_{\ell_t} \;\le\; s^{\,1/t - 1/2} \, \|b\|_2. \tag{A.83}$$

For $t \ge 2$, $\|b\|_{\ell_t} \le \|b\|_2$. Hence, for all $t \in [1, q]$,

$$\|b\|_{\ell_t} \;\le\; s^{\,(1/t - 1/2)_+} \, \|b\|_2 \;\le\; C \, \tau_s \Big(\sqrt{n} \, s^{\max\{1/t,\, 1/2\}} + s^{\,1 + (1/t - 1/2)_+} \Big), \tag{A.84}$$

where we used (A.82). In particular, if $s \le n$ then $s^{\,1+(1/t-1/2)_+} \le \sqrt{n} \, s^{\max\{1/t,\, 1/2\}}$ and (A.84) reduces to

$$\|b\|_{\ell_t} \;\le\; C \, \tau_s \, \sqrt{n} \, s^{\max\{1/t,\, 1/2\}}. \tag{A.85}$$

Finally, by the triangle inequality,

$$\big\| (|a_j + b_j|)_{j \in S} \big\|_{\ell_t} \;\le\; \|(|a_j|)_{j \in S}\|_{\ell_t} + \|(|b_j|)_{j \in S}\|_{\ell_t}, \tag{A.86}$$

$$\big\| (|a_j + b_j|)_{j \in S} \big\|_{\ell_t} \;\ge\; \|(|a_j|)_{j \in S}\|_{\ell_t} - \|(|b_j|)_{j \in S}\|_{\ell_t}, \tag{A.87}$$

and combining with (A.71) and (A.84) (or (A.85) when $s \le n$) yields (A.66) (and (A.67)). □

Ray controls: minimal comparison and blockwise bounds

For the ray $\lambda = tY$ we have the one-dimensional dual objective

$$D(t) := \langle Y, tY\rangle - \tfrac{1}{q}\|X^\top(tY)\|_q^q = \; t\,\|Y\|_2^2 \; - \; \frac{t^q}{q}\,\|X^\top Y\|_q^q. \tag{A.88}$$

Since $D''(t) = -\,(q-1)\,t^{q-2}\,\|X^\top Y\|_q^q < 0$ for all $t > 0$, D is strictly concave on $[0,\infty)$ and admits a unique maximizer $t_\star$ given by the first-order condition $D'(t_\star) = 0$:

$$t_\star^{q-1} = \frac{\|Y\|_2^2}{\|X^\top Y\|_q^q}. \tag{A.89}$$

At this maximizer,

$$D(t_\star) = t_\star\|Y\|_2^2 - \frac{t_\star^q}{q}\|X^\top Y\|_q^q = \left(1-\frac{1}{q}\right) t_\star^q\|X^\top Y\|_q^q = \left(1-\frac{1}{q}\right)\|X^\top(t_\star Y)\|_q^q. \tag{A.90}$$

Lemma 9 (Ray controls)**.** *Let $p \in (1,2]$, $q = \frac{p}{p-1} \in [2,\infty)$, and define $t_\star$ by (A.89). With probability at least $1 - Ce^{-c(d-s)} - Ce^{-c\sqrt{ns}}$ (constants depend only on $(q, \kappa_{\mathrm{bulk}})$), the following hold simultaneously.*

(One-sided value comparison).

$$D(\lambda^\star) \;\geq\; D(t_\star) \qquad \textit{and} \qquad \|X^\top\lambda^\star\|_q^q \;\geq\; \|X^\top(t_\star Y)\|_q^q. \tag{A.91}$$

(Dual-norm scale). *There exist $0 < c_1 \leq C_1 < \infty$ depending only on $(q, \kappa_{\mathrm{bulk}})$ such that*

$$c_1\, t_\star\,\|Y\|_2 \;\leq\; \|\lambda^\star\|_2 \;\leq\; C_1\, t_\star\,\|Y\|_2. \tag{A.92}$$

(Bulk block at level $t \in [1,q]$). *For each $t \in [1,q]$ there exist $0 < c_t \leq C_t < \infty$ (depending only on $(t, \kappa_{\mathrm{bulk}})$) such that*

$$c_t^{1/t}\,(d-s)^{1/t} m_t^{1/t}\, t_\star\|Y\|_2 \;\leq\; \Big\|\big(|\langle X_{:,j}, \lambda^\star\rangle|\big)_{j\notin S}\Big\|_t \;\leq\; C_t^{1/t}\,(d-s)^{1/t} m_t^{1/t}\, t_\star\|Y\|_2, \tag{A.93}$$

where $m_t = \mathbb{E}|Z|^t$ for $Z \sim \mathcal{N}(0,1)$.

(Spike block: two-sided t-level perturbation). *For every $t \in [1,q]$,*

$$\Big\|\big(|\langle X_{:,j}, \lambda^\star\rangle|\big)_{j\in S}\Big\|_t \;=\; t_\star\Big\|\big(|\langle X_{:,j}, Y\rangle|\big)_{j\in S}\Big\|_t \;\pm\; C_2\, t_\star\,\|Y\|_2\, s^{(1/t-1/2)_+}\,(\sqrt{n}+\sqrt{s}), \tag{A.94}$$

for a constant $C_2 = C_2(q, \kappa_{\text{bulk}})$. In particular, if $s \le n$ then

$$\Big\| \big(|\langle X_{:,j}, \lambda^\star \rangle| \big)_{j \in S} \Big\|_t \;=\; t_\star \, \Big\| \big(|\langle X_{:,j}, Y \rangle| \big)_{j \in S} \Big\|_t \;\pm\; C_3 \, t_\star \, \tau_s \, \sqrt{n} \, s^{\max\{1/t,\, 1/2\}}. \quad \text{(A.95)}$$

In the last display we used $\|Y\|_2 = \tau_s \sqrt{n}\,(1 + o(1))$ from Lemma 4.

Proof. We work on the intersection of the high-probability events supplied by Lemma 7 (both (A.39) and (A.40)), Lemma 5, and the singular-value bound (A.20); this intersection has probability at least $1 - Ce^{-cn}$.

(One-sided value comparison (A.91)). By optimality of $\lambda^\star$ and the definition of $t_\star$,

$$D(\lambda^\star) \;\ge\; D(t_\star).$$

Using the Fenchel-Young identity at the optimum (see (A.11)) and (A.90),

$$D(\lambda^\star) = \Big(1 - \tfrac{1}{q}\Big) \|X^\top \lambda^\star\|_q^q, \qquad D(t_\star) = \Big(1 - \tfrac{1}{q}\Big) \|X^\top (t_\star Y)\|_q^q,$$

hence (A.91).

(Dual-norm scale (A.92)). **Lower bound.** From $D(\lambda^\star) \ge D(t_\star)$ and (A.90),

$$D(\lambda^\star) \;\ge\; \Big(1 - \tfrac{1}{q}\Big) t_\star \|Y\|_2^2.$$

Since $D(\lambda^\star) \le \langle Y, \lambda^\star \rangle \le \|Y\|_2 \|\lambda^\star\|_2$, we get

$$\|\lambda^\star\|_2 \;\ge\; \Big(1 - \tfrac{1}{q}\Big)\, t_\star \, \|Y\|_2.$$

Upper bound. Let

$$S(\lambda) := \sum_{j \in S} |\langle X_{:,j}, \lambda \rangle|^q, \qquad B(\lambda) := \sum_{j \notin S} |\langle X_{:,j}, \lambda \rangle|^q.$$

From (A.11),

$$D(\lambda^\star) = \Big(1 - \tfrac{1}{q}\Big) \Big(S(\lambda^\star) + B(\lambda^\star) \Big).$$

By Lemma 7 (left inequality in (A.39)),

$$B(\lambda^\star) \;\ge\; c_q \, (d - s) \, \|\lambda^\star\|_2^q.$$

Combining with $D(\lambda^\star) \le \|Y\|_2\|\lambda^\star\|_2$ gives

$$\Big(1 - \tfrac{1}{q}\Big) c_q (d-s)\, \|\lambda^\star\|_2^{q-1} \;\le\; \|Y\|_2. \tag{A.96}$$

Next, Lemma 5 yields

$$\sum_{j \notin S} |\langle X_{:,j}, Y\rangle|^q = (d-s)\, m_q\, \|Y\|_2^q\, (1 + o(1)),$$

so $\|X^\top Y\|_q^q \ge c\,(d-s)\,\|Y\|_2^q$. From (A.89),

$$\big(t_\star \|Y\|_2\big)^{q-1} = \frac{\|Y\|_2^{q+1}}{\|X^\top Y\|_q^q} \;\le\; \frac{1}{c} \cdot \frac{\|Y\|_2}{(d-s)}.$$

Comparing with (A.96) gives $\|\lambda^\star\|_2^{q-1} \le C\,(t_\star\|Y\|_2)^{q-1}$ and hence $\|\lambda^\star\|_2 \le C_1\, t_\star\, \|Y\|_2$.

(Bulk block (A.93)). Apply Lemma 7 at level t (two-sided inequality (A.40)) with $\lambda = \lambda^\star$:

$$c_t^{1/t}(d-s)^{1/t} m_t^{1/t}\, \|\lambda^\star\|_2 \;\le\; \Big\| \big(|\langle X_{:,j}, \lambda^\star\rangle|\big)_{j \notin S} \Big\|_t \;\le\; C_t^{1/t}(d-s)^{1/t} m_t^{1/t}\, \|\lambda^\star\|_2.$$

Substitute $\|\lambda^\star\|_2 \asymp t_\star \|Y\|_2$ from (A.92).

(Spike block (A.94)-(A.95)). Set $h := \lambda^\star - t_\star Y$. Then

$$X_{:,S}^\top \lambda^\star \;=\; t_\star\, X_{:,S}^\top Y \;+\; X_{:,S}^\top h.$$

For any $t \ge 1$, the triangle inequality gives

$$\Big\| (|\langle X_{:,j}, \lambda^\star\rangle|)_{j \in S} \Big\|_t \;\le\; t_\star \Big\| (|\langle X_{:,j}, Y\rangle|)_{j \in S} \Big\|_t \;+\; \|X_{:,S}^\top h\|_{\ell_t},$$

and the analogous lower bound with a minus sign. By norm monotonicity in $\mathbb{R}^s$ and operator norm submultiplicativity,

$$\|X_{:,S}^\top h\|_{\ell_t} \;\le\; s^{(1/t - 1/2)_+}\, \|X_{:,S}^\top h\|_2 \;\le\; s^{(1/t-1/2)_+}\, s_{\max}(X_{:,S})\, \|h\|_2.$$

From (A.20) with $t = \sqrt{s}$, $s_{\max}(X_{:,S}) \le C(\sqrt{n} + \sqrt{s})$ w.h.p., and from (A.92),

$$\|h\|_2 \;=\; \|\lambda^\star - t_\star Y\|_2 \;\le\; \|\lambda^\star\|_2 + t_\star \|Y\|_2 \;\le\; (C_1 + 1)\, t_\star\, \|Y\|_2.$$

Putting these together yields (A.94). If $s \le n$, Lemma 4 gives $\|Y\|_2 = \tau_s \sqrt{n}(1 + o(1))$ and

$$s^{(1/t-1/2)_+}(\sqrt{n} + \sqrt{s}) \;\le\; 2\sqrt{n}\, s^{\max\{1/t, 1/2\}},$$

which implies (A.95). □

A.2.3 Proof of Theorem 12

With these lemmas in place, we are ready to prove Theorem 12.

Proof of Theorem 12. We work on the intersection of the high-probability events provided by Lemmas 4, 5, 6, 7, 8, and 9; this event has probability at least $1 - Ce^{-c(d-s)} - Ce^{-c\sqrt{ns}} - 2d^{-\gamma}$, consistent with Remark A.2. All constants implicit in $\asymp$ depend only on $(q, \kappa_{\text{bulk}})$.

Along the ray $\lambda = tY$, the one-dimensional dual objective

$$D(t) \;=\; t\,\|Y\|_2^2 - \frac{t^q}{q}\,\|X^\top Y\|_q^q$$

is strictly concave with unique maximizer given by the first-order condition (see (A.89))

$$t_\star^{q-1} \;=\; \frac{\|Y\|_2^2}{\|X^\top Y\|_q^q}. \tag{A.97}$$

By Lemma 4, $\|Y\|_2^2 = \tau_s^2 n(1 + o(1))$, and by the decomposition (A.38),

$$\|X^\top Y\|_q^q \;=\; n^q W_q\,(1 + o(1)) \;+\; (d - s)\,m_q\,\tau_s^q\,n^{q/2}\,(1 + o(1)) \;+\; O\Big(s\,\tau_s^q\,n^{q/2}\Big).$$

Substituting into (A.97) yields

$$t_\star^{q-1} \;\asymp\; \frac{\tau_s^2 n}{n^q W_q + \Big((d - s)\,m_q + O(s)\Big)\,\tau_s^q n^{q/2}} \qquad \text{w.h.p.} \tag{A.98}$$

By strong duality and Fenchel-Young (see (A.11)),

$$\sup_\lambda D(\lambda) \;=\; \left(1 - \frac{1}{q}\right)\|X^\top \lambda^\star\|_q^q \;=\; \left(1 - \frac{1}{q}\right)\|\widehat{w}_p\|_p^p. \tag{A.99}$$

Evaluating D on the ray at $t_\star$ and using $D(\lambda^\star) \geq D(t_\star)$ gives

$$\|\widehat{w}_p\|_p^p \;=\; \|X^\top \lambda^\star\|_q^q \;\geq\; \|X^\top (t_\star Y)\|_q^q \;=\; t_\star^q\,\|X^\top Y\|_q^q \;=\; \frac{\|Y\|_2^{\frac{2q}{q-1}}}{\|X^\top Y\|_q^{\frac{q}{q-1}}}. \tag{A.100}$$

Moreover, by Cauchy–Schwarz and (A.92),

$$\|X^\top \lambda^\star\|_q^q = \langle Y, \lambda^\star \rangle \;\leq\; \|Y\|_2\,\|\lambda^\star\|_2 \;\lesssim\; t_\star\,\|Y\|_2^2 \;=\; t_\star^q\,\|X^\top Y\|_q^q.$$

Combining with (A.100) we obtain the two-sided scale

$$\|\widehat{w}_p\|_p^p \;=\; \|X^\top \lambda^\star\|_q^q \;\asymp\; t_\star^q\,\|X^\top Y\|_q^q.$$

Using the coordinatewise KKT map (A.10),

$$\widehat{w}_p \;=\; \nabla f^\star(X^\top\lambda^\star) \;=\; \operatorname{sgn}(X^\top\lambda^\star)\odot|X^\top\lambda^\star|^{q-1}.$$

Hence, for any $r\in[1,p]$,

$$\|\widehat{w}_p\|_r \;=\; \|X^\top\lambda^\star\|_{(q-1)r}^{q-1}. \tag{A.101}$$

Split the $(q-1)r$-norm over the spike block S and the bulk block S^c and note that $\|u\|_t^t=\|u_S\|_t^t+\|u_{S^c}\|_t^t$ implies $\|u\|_t\asymp\max\{\|u_S\|_t,\|u_{S^c}\|_t\}$:

$$\|\widehat{w}_p\|_r \;\asymp\; \max\Big\{\,\|(|\langle X_{:,j},\lambda^\star\rangle|)_{j\in S}\|_{(q-1)r}^{q-1},\quad \|(|\langle X_{:,j},\lambda^\star\rangle|)_{j\notin S}\|_{(q-1)r}^{q-1}\,\Big\}. \tag{A.102}$$

(We used $\max\{a,b\}\le(a^t+b^t)^{1/t}\le 2^{1/t}\max\{a,b\}$ *for* $t\ge1$.*)*

Set $t:=(q-1)r\le q$. By the spike-ray perturbation from Lemma 9 (see (A.95) when $s\le n$),

$$\Big\|\big(\,|\langle X_{:,j},\lambda^\star\rangle|\,\big)_{j\in S}\Big\|_{\ell_t} \;=\; t_\star\,\Big\|\big(\,|\langle X_{:,j},Y\rangle|\,\big)_{j\in S}\Big\|_{\ell_t} \;\pm\; C\,t_\star\,\tau_s\sqrt{n}\,s^{\max\{1/t,\,1/2\}}. \tag{A.103}$$

(If $s>n$, use the general form (A.94); the conclusion below is unchanged up to constants since $(\sqrt{n}+\sqrt{s})\,s^{(1/t-1/2)_+}\le 2\sqrt{n}\,s^{\max\{1/t,1/2\}}+s^{\,1+(1/t-1/2)_+}$, which is captured by the final "spike remainder" term.) By Lemma 8 at level t,

$$\Big\|\big(\,|\langle X_{:,j},Y\rangle|\,\big)_{j\in S}\Big\|_{\ell_t} \;=\; n\,\|w^\star\|_t\,(1+o(1)) \;\pm\; C\,\tau_s\sqrt{n}\,s^{\max\{1/t,\,1/2\}}. \tag{A.104}$$

Combining (A.103)-(A.104) and using $(a+b)^{q-1}\le 2^{q-2}(a^{q-1}+b^{q-1})$ for $a,b\ge0$, we obtain the following uniform two-sided bounds (recall $t=(q-1)r\le q$):

$$\Big\|\big(|\langle X_{:,j},\lambda^\star\rangle|\big)_{j\in S}\Big\|_{\ell_t}^{q-1}\le C\Big\{\,t_\star^{\,q-1}\,n^{\,q-1}\,\|w^\star\|_t^{q-1} \;+\; (t_\star\tau_s\sqrt{n})^{\,q-1}\,s^{\,(q-1)\max\{1/t,\,1/2\}}\,\Big\}, \tag{A.105}$$

$$\Big\|\big(|\langle X_{:,j},\lambda^\star\rangle|\big)_{j\in S}\Big\|_{\ell_t}^{q-1}\ge c\Big(\,t_\star\,n\,\|w^\star\|_t \;-\; C\,t_\star\,\tau_s\sqrt{n}\,s^{\max\{1/t,\,1/2\}}\,\Big)_+^{q-1}. \tag{A.106}$$

Applying the mean-value inequality to the map $z\mapsto z^{\,q-1}$,

$$|(x\pm y)^{q-1}-x^{q-1}|\le C\,(x^{q-2}y+y^{q-1}),$$

with $x = t_\star n\|w^\star\|_t$ and $y = Ct_\star\tau_s\sqrt{n}\,s^{\max\{1/t,\,1/2\}}$, we obtain

$$\Big\|\Big(|\langle X_{:,j},\lambda^\star\rangle|\Big)_{j\in S}\Big\|_{\ell_t}^{q-1} \;=\; t_\star^{q-1}\,n^{\,q-1}\,\|w^\star\|_t^{q-1}\,(1{+}o(1)) \;\pm C\,(t_\star\tau_s\sqrt{n})^{\,q-1}\,s^{\,\max\{\,(q-1)/2,\;(q-1)/t\,\}}. \tag{A.107}$$

Recalling $t = (q-1)r$ and $\|w^\star\|_t \asymp \|w^\star\|_{(q-1)r}$, we obtain the spike contribution stated in (5.2). *(For completeness: specializing (A.94) to $t = q$ together with Lemma 8 at $t = q$ yields the same rate and remainder exponent as in (A.107).)*

By Lemma 9 (bulk control (A.93)) together with (A.92),

$$\|(|\langle X_{:,j},\lambda^\star\rangle|)_{j\notin S}\|_{(q-1)r} \;\asymp\; (d-s)^{1/((q-1)r)}\,t_\star\,\|Y\|_2.$$

Raising to the $(q-1)$-th power and using $\|Y\|_2 \asymp \tau_s\sqrt{n}$ (Lemma 4),

$$\|(|\langle X_{:,j},\lambda^\star\rangle|)_{j\notin S}\|_{(q-1)r}^{q-1} \;\asymp\; (d-s)^{1/r}\left(t_\star\,\tau_s\sqrt{n}\right)^{q-1}. \tag{A.108}$$

Plug (A.107) and (A.108) into (A.102). This yields

$$\|\widehat{w}_p\|_r \;\asymp\; \max\left\{\,t_\star^{q-1}\,n^{\,q-1}\,\|w^\star\|_{(q-1)r}^{q-1},\;\; (d{-}s)^{1/r}\left(t_\star\,\tau_s\sqrt{n}\right)^{q-1},\;\; s^{\,\max\{1/r,\,(q-1)/2\}}\left(t_\star\,\tau_s\sqrt{n}\right)^{q-1}\right\},$$

which is exactly the three-term unified bound in (5.2). When $r < 2(p-1)$ and $(d-s) \gtrsim s$, the third term is absorbed by the bulk term, recovering the two-term maximum.

In the proportional regime $(d-s) \asymp \kappa_{\text{bulk}}\,n$, balance the two leading terms in $\|X^\top Y\|_q^q$ (cf. (A.38)) to define

$$n^q W_q \;\asymp\; (d-s)\,\tau_s^q\,n^{q/2} \quad\Longleftrightarrow\quad n^{q/2} \;\asymp\; \kappa_{\text{bulk}}\,\frac{\tau_s^q}{W_q} \quad\Longleftrightarrow\quad n_\star \;\asymp\; \left(\kappa_{\text{bulk}}\,\frac{\tau_s^q}{W_q}\right)^{\frac{2}{q-2}},$$

which matches (5.3).

(i) Dual spike-dominated regime $n \gg n_\star$. Then $\|X^\top Y\|_q^q \asymp n^q W_q$ and (A.98) gives

$$t_\star^{q-1} \;\asymp\; \frac{\tau_s^2 n}{n^q W_q} \;=\; \frac{\tau_s^2}{W_q}\,n^{-(q-1)}. \tag{A.109}$$

Consequently

$$(d-s)^{1/r}\left(t_\star\,\tau_s\sqrt{n}\right)^{q-1} \asymp \frac{\tau_s^{q+1}}{W_q}\,n^{\frac{1}{r}-\frac{1}{2(p-1)}}, \tag{A.110a}$$

$$s^{\max\{1/r,(q-1)/2\}}\left(t_\star\,\tau_s\sqrt{n}\right)^{q-1} \asymp \frac{\tau_s^{q+1}}{W_q}\,s^{\max\{1/r,(q-1)/2\}}\,n^{-\frac{1}{2(p-1)}}. \tag{A.110b}$$

In particular, when $r \le 2(p-1)$ the two "bulk-type" terms are of the same order (and are dominated by the spike main when $r \ge 2(p-1)$); this recovers (5.4).

(ii) Dual bulk-dominated regime $n \ll n_\star$. Then $\|X^\top Y\|_q^q \asymp (d-s)\tau_s^q n^{q/2}$ and

$$t_\star^{q-1} \asymp \frac{\tau_s^2 n}{(d-s)\tau_s^q n^{q/2}} = \frac{\tau_s^{2-q}}{(d-s)}\,n^{1-\frac{q}{2}}. \tag{A.111}$$

Therefore

$$(d-s)^{1/r}\left(t_\star\,\tau_s\sqrt{n}\right)^{q-1} \asymp \kappa_{\text{bulk}}^{\frac{1}{r}-1}\,\tau_s\,n^{\frac{1}{r}-\frac{1}{2}}, \tag{A.112a}$$

$$s^{\max\{1/r,(q-1)/2\}}\left(t_\star\,\tau_s\sqrt{n}\right)^{q-1} \asymp \kappa_{\text{bulk}}^{-1}\,\tau_s\,s^{\max\{1/r,(q-1)/2\}}\,n^{-1/2}. \tag{A.112b}$$

Taking the maximum together with the spike main term gives (5.5) whenever the third term is absorbed; otherwise the third term with exponent $\max\{1/r,(q-1)/2\}-1/2$ may dominate.

This completes the proof of (5.2) (three-term form), the energy scale (A.100), hence the proof of Theorem 12. □

A.2.4 Two concrete corollaries: single spike and flat support

We keep $p \in (1,2]$, $q = \frac{p}{p-1} \in [2,\infty)$, $r \in [1,p]$, and $\kappa_{\text{bulk}} = \liminf (d-s)/n > 0$. Recall the unified bound from Theorem 12. We will repeatedly use the identity

$$\|\widehat{w}_p\|_r \asymp \max\Big\{\, t_\star^{q-1}\,n^{q-1}\,\|w^\star\|_{(q-1)r}^{q-1},\ (d-s)^{1/r}\left(t_\star\,\tau_s\sqrt{n}\right)^{q-1}, \tag{A.113}$$

$$s^{\max\{1/r,\ (q-1)/2\}}\left(t_\star\,\tau_s\sqrt{n}\right)^{q-1}\Big\}, \tag{A.114}$$

together with

$$t_\star^{q-1} = \frac{\|Y\|_2^2}{\|X^\top Y\|_q^q}, \qquad n_\star \asymp \left(\kappa_{\text{bulk}}\,\frac{\tau_s^q}{W_q}\right)^{\frac{2}{q-2}}, \qquad W_q = \sum_{j\in S}|w_j^\star|^q, \quad \tau_s^2 = \|w^\star\|_2^2 + \sigma^2. \tag{A.115}$$

Case (i): single spike ($s = 1$). Let the support be $\{j_0\}$ and write $a := |w^\star_{j_0}| > 0$. Then

$$W_q = a^q, \qquad \|w^\star\|_{(q-1)r} = a, \qquad \tau_s^2 = a^2 + \sigma^2. \tag{A.116}$$

The transition scale simplifies to

$$n_\star \asymp \left(\kappa_{\text{bulk}} \frac{(a^2+\sigma^2)^{q/2}}{a^q}\right)^{\frac{2}{q-2}}. \tag{A.117}$$

In (A.113), the spike remainder is dominated by the bulk term since

$$\frac{\text{spike remainder}}{\text{bulk}} = (d-1)^{-1/r} \ll 1 \quad \text{for large } d. \tag{A.118}$$

Dual spike-dominated ($n \gg n_\star$). Using the phase form (5.4), we obtain

$$\|\widehat{w}_p\|_r \asymp \begin{cases} \dfrac{(a^2+\sigma^2)^{\frac{q+1}{2}}}{a^q}\, n^{\frac{1}{r}-\frac{1}{2(p-1)}}, & r \le 2(p-1), \\ \dfrac{a^2+\sigma^2}{a}, & r > 2(p-1). \end{cases} \tag{A.119}$$

Dual bulk-dominated ($n \ll n_\star$). Using (5.5),

$$\|\widehat{w}_p\|_r \asymp \max\left\{\kappa_{\text{bulk}}^{\frac{1}{r}-1}\,(a^2+\sigma^2)^{1/2}\, n^{\frac{1}{r}-\frac{1}{2}},\ \kappa_{\text{bulk}}^{-1}\,(a^2+\sigma^2)^{\frac{2-q}{2}}\, a^{\,q-1}\, n^{\frac{q}{2}-1}\right\}. \tag{A.120}$$

(The third term in (5.5) equals $\kappa_{\text{bulk}}^{-1}\tau_s n^{-1/2}$ and is dominated by the first term for large n.)

Case (ii): flat signal on its support. Assume $w^\star_j = a\, s_j$ for all $j \in S$ with $|s_j| = 1$ and $|S| = s$. Then

$$\|w^\star\|_2 = \sqrt{s}\,|a|, \qquad W_q = s\,|a|^q, \qquad \|w^\star\|_{(q-1)r} = s^{\frac{1}{(q-1)r}}\,|a|, \qquad \tau_s^2 = s\,a^2 + \sigma^2. \tag{A.121}$$

The transition scale grows linearly in s:

$$n_\star \asymp \left(\kappa_{\text{bulk}} \frac{(sa^2+\sigma^2)^{q/2}}{s\,|a|^q}\right)^{\frac{2}{q-2}} = \kappa_{\text{bulk}}^{\frac{2}{q-2}}\, s \left(1 + \frac{\sigma^2}{sa^2}\right)^{\frac{q}{q-2}}. \tag{A.122}$$

Dual spike-dominated ($n \gg n_\star$). From (5.4),

$$\|\widehat{w}_p\|_r \asymp \begin{cases} \dfrac{(sa^2+\sigma^2)^{\frac{q+1}{2}}}{s\,|a|^q}\, n^{\frac{1}{r}-\frac{1}{2(p-1)}}, & r \le 2(p-1), \\ s^{\frac{1}{r}-1}\,\dfrac{sa^2+\sigma^2}{|a|}, & r > 2(p-1). \end{cases} \tag{A.123}$$

In the noiseless case ($\sigma = 0$),

$$r > 2(p-1): \quad \|\widehat{w}_p\|_r \asymp s^{1/r}\,|a|, \qquad r \le 2(p-1): \quad \|\widehat{w}_p\|_r \asymp s^{\frac{q-1}{2}}\,|a|\,n^{\frac{1}{r}-\frac{1}{2(p-1)}}. \tag{A.124}$$

Dual bulk-dominated ($n \ll n_\star$). From (5.5),

$$\|\widehat{w}_p\|_r \asymp \max\Big\{\kappa_{\text{bulk}}^{\frac{1}{r}-1}\,(sa^2+\sigma^2)^{1/2}\,n^{\frac{1}{r}-\frac{1}{2}},\ \kappa_{\text{bulk}}^{-1}\,(sa^2+\sigma^2)^{\frac{2-q}{2}}\,s^{1/r}|a|^{\,q-1}\,n^{\frac{q}{2}-1}, \tag{A.125}$$

$$\kappa_{\text{bulk}}^{-1}\,(sa^2+\sigma^2)^{1/2}\,s^{\max\{1/r,\,(q-1)/2\}}\,n^{-1/2}\Big\}. \tag{A.126}$$

When $r \le 2(p-1)$ and $s \lesssim (d-s)$, the third term is absorbed by the first (Remark A.2.1).

B

Additional appendices for Chapter 3

B.1 More related work

B.1.1 Preliminaries: two kinds of questions generalization and two types of inductive bias

In this supplementary section we expand on our briefer discussion of related work in the Introduction of the main paper. The question of why and how DNNs generalize in the overparameterized regime has generated a vast literature. To organize our discussion, we follow [Mingard et al., 2021] and first distinguish two kinds of questions about generalization in overparameterized DNNs:

1) The question of over-parameterized generalization: Why do DNNs generalize at all in the overparameterized regime, where classical learning theory suggests they should heavily overfit.

2) The question of fine-tuned generalization: Given that a DNN already generalizes reasonably well, how can detailed architecture choice, optimizer choice, and hyperparameter tuning further improve generalization?

Question 2) is the main focus of a large tranche of the literature on generalization, and for good reason. In order to build state-of-the-art (SOTA) DNNs, even a few percent accuracy improvement (taking image classification as an example) is important in practice. Improved generalization performance can be achieved in

many ways, including local adjustments of the DNNs structure (e.g. convolutional layers, pooling layers, shortcut connections etc.), hyperparameter tuning (learning rate, batch size etc.), or choosing different optimizers (e.g. vanilla SGD versus entropySGD [Chaudhari et al., 2019] or Adam Kingma and Ba [2014].

In this paper, however, we are primarily interested in question 1). As pointed out, for example famously in [Zhang et al., 2016a], but also by many researchers before that [1], DNNs can be proven to be highly expressive, so that the number of hypotheses that can fit a training data set S, but generalize poorly, is typically many orders of magnitude larger than the number that can actually generalize. And yet in practice DNNs do not tend to overfit much, and can generalize well, which implies that DNNs must have some kind of *inductive bias* [Shalev-Shwartz and Ben-David, 2014] toward hypotheses that generalise well on unseen data.

Following the framework of [Mingard et al., 2021], we use the language of functions (rather than that of hypotheses, see also Section B.2.) to distinguish two major potential types of inductive bias.

A) The inductive bias upon upon random sampling of parameters over a parameter distribution $P_w(\mathbf{w})$. In other words, given a DNN architecture, loss function etc. and a measure over parameters $P_w(\mathrm{w})$ (which can be taken to be the initial parameter distribution for an optimiser, but is more general), this bias occurs when certain types of functions more likely to appear upon random sampling of parameters than others. This inductive bias can be expressed in terms of a prior over functions $P(f)$, or in terms of a posterior $P_B(f|S)$ when the functions are conditioned, for example, on obtaining zero error on training set S.

B) The inductive bias induced by optimizers during a training procedure. In other words, given an inductive bias upon initialization (from **A)**, does the training procedure induce a further inductive bias on what functions a DNN expresses? One way of measuring this second form of inductive bias is to calculate the probability $P_{opt}(f|S)$ that an DNN trained to zero error on training set

[1] For example, Leo Breiman, included the question of overparameterised generalization in DNN back in in 1995 as one of the main issues raised by his reflections on 20 years of refereeing for Neurips [Breiman, 1995]),

S with optimizer *opt* (typically a variant of SGD) expresses function f, and to then compare it to the Bayesian posterior probability $P_B(f|S)$ that this function obtains upon random sampling of parameters Mingard et al. [2021]. In principle $P_B(f|S)$ expresses the inductive bias of type A), so any differences between $P_{opt}(f|S)$ and $P_B(f|S)$ could be due to inductive biases of type B).

These two sources of inductive bias can be relevant to both questions above about generalization. Wwe emphasise that our taxonomy of two questions about generalization, and two types of inductive bias is just one way of parsing these issues. We make these first order distinctions to help clarify our discussion of the literature, and are aware that there are other ways of teasing out these distinctions.

B.1.2 Related work on flatness

The concept "flatness" of the loss function of DNNs can be traced back to Hinton and van Camp [1993] and Hochreiter and Schmidhuber [1997a]. Although these authors did not provide a completely formal mathematical definition of flatness, Hochreiter and Schmidhuber [1997a] described flat minima as "a large connected region in parameter space where the loss remains approximately constant", which requires lower precision to specify than sharp minima. They linked this idea to the minimum description length (MDL) principle [Rissanen, 1978], which says that the best performing model is the one with shortest description length, to argue that flatter minima should generalize better than sharp minima. More generally, flatness can be interpreted as a complexity control of the hypotheses class introduced by algorithmic choices.

The first thing to note is that flatness is a property of the functions that a DNN converges on. In other words, the basic argument above is that flatter functions will generalize better, which can be relevant to both questions 1) and 2) above.

It is a different question to ask whether a certain way of finding functions (say by optimising a DNN to zero error on a training set) will generate an inductive bias towards flatter functions. In Hochreiter and Schmidhuber [1997a], the authors proposed an algorithm to bias towards flatter minima by minimizing the training

loss while maximizing the log volume of a connected region of the parameter space. This idea is similar to the recent suggestion of entropy-SGD Chaudhari et al. [2019], where the authors also introduced an extra regularization to bias the optimizer into wider valleys by maximizing the "local entropy".

In an influential paper, Keskar et al. [2016] reported that the solutions found by SGD with small batch sizes generalize better than those found with larger batch sizes, and showed that this behaviour correlated with a measure of "sharpness" (sensitivity of the training loss to perturbations in the parameters). Sharpness can be viewed as a measure which is the inverse of the flatness introduced by Hinton and van Camp [1993] and Hochreiter and Schmidhuber [1997a]. This work helped to popularise the notion that SGD itself plays an important role in providing inductive bias, since differences in generalization performance and in sharpness correlated with batch size. In follow-on papers others have showed that the correlation with batch size is more complex, as some of the improvements can be mimicked by changing learning rates or number of optimization steps for example, see [Hoffer et al., 2017, Goyal et al., 2017, Smith et al., 2017a, Neyshabur et al., 2017b]. Nevertheless, these changes in generalization as a function of optimizer hyperparameters are important things to understand because they are fundamentally type B inductive bias. Because the changes in generalization performance in these papers tend to be relatively small, they mainly impinge on question 2) for fine-tuned generalization. Whether these observed effects are relevant for question 1) is unclear from this literature.

Another strand of work on flatness has been through the lens of generalization bounds. For example, Neyshabur et al. [2017b] showed that sharpness by itself is not sufficient for ensuring generalization, but can be combined, through PAC-Bayes analysis, with the norm of the weights to obtain an appropriate complexity measure. The connection between sharpness and the PAC-Bayes framework was also investigated by Dziugaite and Roy [2017b], who numerically optimized the overall PAC-Bayes generalization bound over a series of multivariate Gaussian distributions (different choices of perturbations and priors) which describe the KL-divergence term appearing in the second term in the combined generalization

bound by Neyshabur et al. [2017b]. For more discussion of this literature on bounds and flatness, see also the recent review Valle-Pérez and Louis [2020].

Rahaman et al. [2018] also draw a connection to flatness through the lens of Fourier analysis, showing that DNNs typically learn low frequency components faster than high frequency components. This frequency argument is related to the input-output sensitivity picture, which is systematically investigated in Novak et al. [2018a].

There is also another wide-spread belief that SGD trained DNNs are implicitly biased towards having small parameters norms or large margin, intuitively inspired by classical ridge regression and SVMs. Bartlett et al. [2017c] presented a margin-based generalization bound that depends on spectral and $L_{2,1}$ norm of the layer-wise weight matrices of DNNs. Neyshabur et al. [2017a] later proved a similar spectral-normalized margin bound using PAC-Bayesian approach rather than the complex covering number argument used in Bartlett et al. [2017c]. Liao et al. [2018] further strengthen the theoretical arguments that an appropriate measure of complexity for DNNs should be based on a product norm by showing the linear relationship between training/testing cross entropy loss of normalized networks. Jiang et al. [2018] also empirically studied the role of margin bounds.

In a recent important large-scale empirical work on different complexity measures by Jiang et al. [2019a], 40 different complexity measures are tested when varying 7 different hyperparameter types over two image classification datasets. They do not introduce random labels so that data complexity is not thoroughly investigated. Among these measures, the authors found that sharpness-based measures outperform their peers, and in particular outperform norm-based measures. It is worth noting that their definition of "worst case" sharpness is similar to Definition 5 but normalized by weights, so they are not directly comparable. In fact, their definition of worst case sharpness in the PAC-Bayes picture is more close to the works by Petzka et al. [2019], Rangamani et al. [2019], Tsuzuku et al. [2019] which focus on finding scale-invariant flatness measure. Indeed enhanced performance are reported in these works. However, these measures are only scale-invariant when the

scaling is layer-wise. Other methods of re-scaling (e.g. neuron-wise re-scaling) can still change the metrics. Moreover, the scope of Jiang et al. [2019a] is concentrated on the practical side (e.g. inductive bias of type B) and does not consider data complexity, which we believe is a key ingredient to understanding the inductive bias needed to explain question 1) on generalization.

Finally, in another influential paper, Dinh et al. [2017a] showed that many measures of flatness, including the sharpness used in Keskar et al. [2016], can be made to vary arbitrarily by re-scale parameters while keeping the function unchanged. This work has called into question the use of local flatness measures as reliable guides to generalization, and stimulated a lot of follow on studies, including the present paper where we explicitly study how parameter-rescaling affects measures of flatness as a function of epochs.

B.1.3 Related work on the infinite-width limit

A series of important recent extensions of the seminal proof in Neal [1994] - that a single-layer DNN with random iid weights is equivalent to a Gaussian process (GP) [Mackay, 1998] in the infinite-width limit - to multiple layers and architectures (NNGPs) have recently appeared [Lee et al., 2017, Matthews et al., 2018, Novak et al., 2018b, Garriga-Alonso et al., 2019, Yang, 2019]. These studies on NNGPs have used this correspondence to effectively perform a very good approximation to exact Bayesian inference in DNNs. When they have compared NNGPs to SGD-trained DNNs the generalization performances have generally shown a remarkably close agreement. These facts require rethinking the role SGD plays in question 1) about generalization, given that NNGPs can already generalize remarkably well without SGD at all.

B.1.4 Relationship to previous papers using the function picture

The work in this paper builds on a series of recent papers that have explored the function based picture in random neural networks. We briefly review these works

to clarify their connection to the current paper.

Firstly, in [Valle-Pérez et al., 2018], the authors demonstrated empirically that upon random sampling of parameters, DNNs are highly biased towards functions with low complexity. This behaviour does not depend very much on $P_w(\mathbf{w})$ for a range of initial distributions typically used in the literature. Note that this behaviour does start to deviate from what was found in [Valle-Pérez et al., 2018], when the system enters a chaotic phase, which can be reached with for tanh or erf non-linearities and for $P_w(\mathbf{w})$ with a relatively large variance Yang and Salman [2019]. They show more specifically that the bias towards simple functions is consistent with the "simplicity bias" from Dingle et al. [2018, 2020], which was inspired by the coding theorem from algorithmic information theory (AIT) [Li and Vitanyi, 2008], first derived by Levin [1974] . The idea of simplicity bias in DNNs states that if the parameter-function map is sufficiently biased, then the probability of the DNN producing a function f on input data drops exponentially with increasing Kolmogorov complexity $K(f)$ of the function f. In other words, high $P(f)$ functions have low $K(f)$, and high $K(f)$ functions have low $P(f)$. A key insight from [Dingle et al., 2018, 2020] is that $K(f)$ can be approximated by an appropriate measure $\tilde{K}(f)$ and still be used to make predictions on $P(f)$, even if the true $K(f)$ is formally incomputable. Recently Mingard et al. [2019] and De Palma et al. [2018] gave two separate non-AIT based theoretical justifications for the existence of simplicity bias in DNNs. In other words, this line of work suggests that DNNs have an intrinsic bias towards simple functions upon random sampling of parameters, and in our taxonomy, that is bias of type A).

If simplicity bias in DNNs matches "natural" data distributions, then, at least upon random sampling of parameters, this should help facilitate good generalization. Indeed, it has been shown that data such as MNIST or CIFAR-10 is relatively simple [Lin et al., 2017, Goldt et al., 2019, Spigler et al., 2019], suggesting that an inductive bias toward simplicity will assist with good generalization.

A second paper upon which the current one builds is [Mingard et al., 2021], where extensive empirical test (for a range of architectures (FCN, CNN, LSTM), datasets

(MNIST, Fashion-MNIST, CIFAR-10,ionosphere, IMDb moviereview dataset), and SGD variants (vanilla SGD, Adam, Adagrad, RMSprop, Adadelta), as well as for different batch sizes and learning rates) were done of the hypothesis that:

$$P_{opt}(f|S) \approx P_B(f|S). \tag{B.1}$$

Here $P_{opt}(f|S)$ is the probability that an optimiser (SGD or one of its variants) converges upon a function f after training to zero training error on a training set S. By training over many different parameter initializations, $P_{opt}(f|S)$ can be calculated. Similarly, the Bayesian posterior probability $P_B(f|S)$ is defined as the probability that upon random sampling of parameters, a DNN expresses function f, conditioned on zero error on S. The functions were, as in the current paper, a restriction to a given training set S and test set E. Since the systems always had zero error on the training set, functions could be compared by what they produced on the test set (for example, the set of labels on the images for image classification). It was found that the hypothesis (B.1.4) held remarkably well to first order, for a wide range of systems. At first sight this similarity is surprising, given that the procedures to generate $P_{opt}(f|S)$ (training with an optimiser such as SGD) is completely different from those for $P_B(f|S)$ (where GP techniques and direct sampling were used), which knows nothing of optimisers at all. The fact that these two probabilities are so similar suggests that any inductive bias of type B, which would be a bias beyond what is already present in $P_B(f|S)$, is relatively small. While this conclusion does not imply that there are no induced biases of type B), and clearly there are since hyperparameter tuning affects fine-tuned generalization, it does suggest that the main source of inductive bias needed to explain 1), the question of why DNNs generalize in the first place, is found in the inductive biases of type A), which are already there in $P_B(f|S)$. In [Mingard et al., 2021], the authors propose that, for highly biased priors $P(f)$, that SGD is dominated by the large differences in basin size for the different functions f, and so finds functions with probabilities dominated by the initial distribution. A

similar effect was seen in evolutionary systems Schaper and Louis [2014], Dingle et al. [2015] where it was called the arrival of the frequent.

In addition, in [Mingard et al., 2021], the authors observed for one system that $-\log(P_B(f|S))$ scaled linearly with the generalization error on E for a wide range of errors. This preliminary result provided inspiration for the current paper where we directly study the correlation between the prior $P(f)$ and the generalization error.

The third main function based paper that we build upon is [Valle-Pérez and Louis, 2020] which provides a comprehensive analysis of generalization bounds. In particular, it studies in some detail the Marginal Likelihood PAC-Bayes bound, first presented in Valle-Pérez et al. [2018], which is predicts a direct link between the generalization error and the log of the marginal likelihood $P(S)$. $P(S)$ can be interpreted as the total prior probability that a function is found with zero error on the training set S, upon random sampling of parameters of the DNN. The performance of the bound was tested for challenges such as varying amounts of data complexity, different kinds of architectures, and different amounts of training data (learning curves). For each challenge it works remarkably well, and to our knowledge no other bound has been tested this comprehensively. Again, the good performance of this bound, which is agnostic about optimisers, suggest that a large part of the answer to question 1) can be found in the inductive bias of type A), e.g. that found upon initialization. The bound is not accurate enough to explain smaller effects relevant for fine-tuning generalization, which can originate from other sources such as a difference in optimiser hyperparameters. These conclusions are consistent with the different approach in this paper, where we use the prior $P(f)$ (which knows nothing about SGD) and show that it also correlates with predicted test error for DNNS trained with SGD and its variants. We do propose a simpler bound that is consistent with the observed scaling, but more work is needed to get anywhere near the rigour found in [Valle-Pérez and Louis, 2020] for the full marginal likelihood bound.

Finally, we note that in all three of these papers, GPs are used to calculate marginal likelihoods, posteriors, and priors. Technical details of how to use GPs can be found clearly explained there.

The current paper *builds* on this body of work and uses some of the techniques described therein, but it is distinct. Firstly, our measurements on flatness are new, and our claim that the prior $P(f)$ correlates with generalization, while indirectly present in [Mingard et al., 2021] was not developed there at all as that paper focuses on the posterior $P_B(f|S)$, and did not use the attack set trick to vary functions that are consistent with S, and so is tackling a different question (namely how much extra inductive bias comes from using SGD over the inductive bias already present in the Bayesian posterior). The attack set trick means that $P(S)$ does not change, while clearly the generalisation error (or expected test error) does change, so the marginal likelihood bound is not predictive here.

B.2 Parameter-function map and neutral space

The link between the parameters of a DNN and the function it expresses is formally described by the parameter-function map:

Definition 14 (Parameter-function map)**.** *Consider the model defined in Definition 6, if the model takes parameters within a set $W \subseteq \mathbb{R}^n$, then the parameter-function map $\mathcal{M}$ is defined as*

$$\mathcal{M} : W \to \mathcal{F}$$

$$\mathbf{w} \mapsto f_{\mathbf{w}}.$$

where $f_{\mathbf{w}}$ denotes the function parameterized by $\mathbf{w}$.

The parameter-function map, introduced in [Valle-Pérez et al., 2018], serves as a bridge between a parameter searching algorithm (e.g. SGD) and the behaviour of a DNN in function space. In this context we can also define the:

Definition 15 (Neutral space)**.** *For a model defined in Definition 14, and a given function f, the neutral space $\mathcal{N}_f \subseteq W$ is defined as*

$$\mathcal{N}_f := \{\mathbf{w} \in W : \mathcal{M}(\mathbf{w}) = f\}.$$

The nomenclature comes from genotype-phenotype maps in the evolutionary literature [Manrubia et al., 2020], where the space is typically discrete, and a neutral set refers to all genotypes that map to the same phenotype. In this context, the Bayesian prior $P(f)$ can be interpreted as the probabilistic volume of the corresponding neutral space.

B.3 Clarification on definition of functions and prior

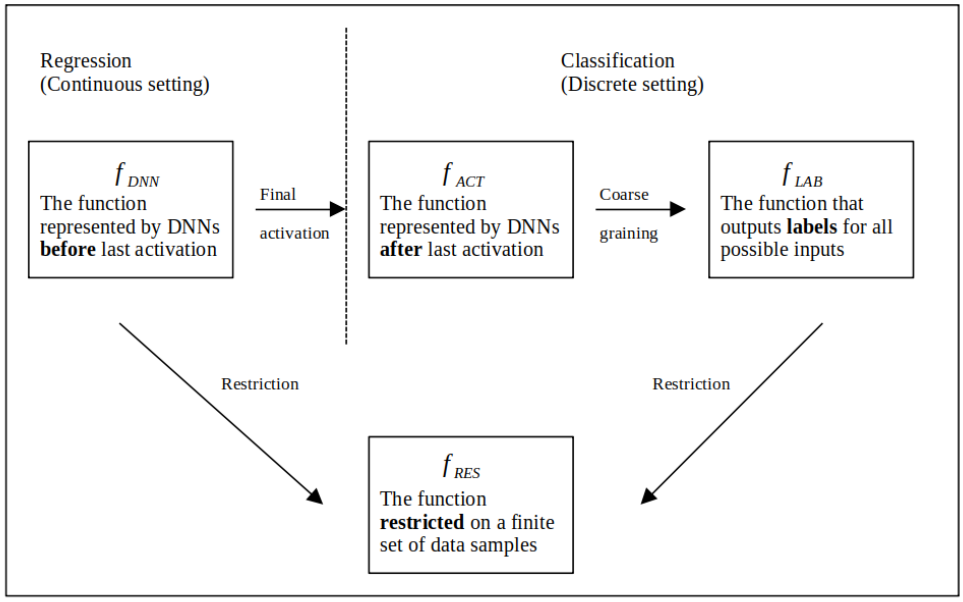


Figure B.1: The diagram of different definitions for functions represented by DNNs.

The discussion of "functions" represented by DNNs can be confusing without careful definition. In Fig. B.1 we list four different interpretations of "functions" commonly seen in literature which also are directly related to our work. These interpretations cover both regression and classification settings. Let $\mathcal{X}$ be an arbitrary input domain and $\mathcal{Y}$ be the output space. According to different interpretations of the function represented by a DNN, $\mathcal{Y}$ will be different, for the same choice of $\mathcal{X}$ and DNN.

Definition 16 (f_{DNN})**.** *Consider a DNN whose input domain is $\mathcal{X}$. Then f_{DNN} belongs to a class of functions $\mathcal{F}_{\text{DNN}}$ which define the mapping between $\mathcal{X}$ to the pre-activation of the last layer of DNN, which lives in $\mathbb{R}^d$:*

$$f_{\text{DNN}} \in \mathcal{F}_{\text{DNN}} : \mathcal{X} \to \mathbb{R}^d$$

d is the width of the last layer of DNN.

In standard Gaussian process terminology, f_{DNN} is also called *latent function* [Rasmussen, 2003]. This is the function we care about in regression problems.

In the context of supervised learning, we have to make some assumptions about the characteristics of $\mathcal{F}_{\text{DNN}}$, as otherwise we would not know how to choose between functions which are all consistent with the training sample but might have hugely different generalization ability. This kind of assumptions are called *inductive bias*. One common approach of describing the inductive bias is to give a prior probability distribution to $\mathcal{F}_{\text{DNN}}$, where higher probabilities are given to functions that we consider to be more likely. For DNNs, $\mathcal{F}_{\text{DNN}}$ is a set of functions over an (in general) uncountably infinite domain $\mathcal{X}$. There are several approaches to define probability distributions over such sets. Gaussian processes represent one approach, which generalizes Gaussian distributions to function spaces. If we ask only for the properties of the functions at a finite number of points, i.e. restriction of $\mathcal{F}_{\text{DNN}}$ to $C : \{c_1, \ldots, c_n\} \subset \mathcal{X}$ (see Definition 6), then inference with a Gaussian process, reduces to inference with a standard multidimensional Gaussian distribution. This is an important property of Gaussian process called *consistency*, which helps in making computations with Gaussian processes feasible. As shown in Section B.4, we can readily compute with this GP prior over $\mathcal{F}_{\text{DNN}}$ as long as it is restricted on a finite data set. Later in Definition 19 we will formally define the restricted function f_{RES}.

In classification tasks, we typically get a data sample from $\mathcal{X} \times \mathcal{Y}$, where without loss of generality $\mathcal{Y}$ has the form of $\mathcal{Y} = \{1, \ldots, k\}$ where k is the number of classes. For simplicity, we further assume binary classification where $\mathcal{Y} = \{0, 1\}$ Note in the scope of binary classification we have the last layer width of $d = 1$. To grant the outputs of the function represented by a DNN a probability interpretation, we need

the outputs lie in the interval $(0, 1)$. One way of doing so is to "squash" the outputs of f_{DNN} to $(0, 1)$ by using a final *activation*, typically a logistic or sigmoid function $\lambda(z) = (1 + \exp(-z))^{-1}$. Subsequently we have the definition of f_{ACT} in Fig. B.1:

Definition 17 (f_{ACT})**.** *Consider the setting and f_{DNN} defined in Definition 16 where $d = 1$, and a logistic activation $\lambda(z) = (1+\exp(-z))^{-1}$. Then f_{ACT} is defined as :*

$$f_{\text{ACT}} := f_{\text{DNN}} \circ \lambda : \mathcal{X} \to (0, 1)$$

where $\circ$ denotes function composition. we also define the space of f_{ACT} as

$$\mathcal{F}_{\text{ACT}} = \{f_{\text{ACT}} \text{ for every } f_{\text{DNN}} \in \mathcal{F}_{\text{DNN}}\}$$

In real life classification datasets, we typically do not have access to the probability of an input classified as one certain label, but the labels instead. When we discuss functions represented by DNNs in classification, we usually mean the *coarse-grained* version of $f_{\text{ACT}} \in \mathcal{F}_{\text{ACT}}$, meaning we group all outputs to 1 if the probability of predicting the inputs as being label "1" is greater or equal than 0.5, and 0 otherwise. Mathematically, we define f_{LAB} as:

Definition 18 (f_{LAB})**.** *Consider the setting and f_{ACT} defined in Definition 17 and a threshold function*

$$\tau(z) = \begin{cases} 1 \text{ if } z \geq 0.5 \\ 0 \text{ otherwise} . \end{cases}$$

Then we define f_{LAB} and the space $\mathcal{F}_{\text{LAB}}$ as:

$$f_{\text{LAB}} = f_{\text{ACT}} \circ \tau : \mathcal{X} \to \{0, 1\}$$

$$\mathcal{F}_{\text{LAB}} = \{f_{\text{LAB}} \text{ for every } f_{\text{ACT}} \in \mathcal{F}_{\text{ACT}}\}$$

The Definition 18 allows us to describe the function represented by a DNN in binary classification as a binary string consisting of "0" and "1", whose length is equal to the size of input domain set $|\mathcal{X}|$. As explained earlier, in classification we also want to put a prior over $\mathcal{F}_{\text{LAB}}$ and use this prior as our belief about the task before seeing any data.

Finally, as we mentioned above, to make computations tractable, we restrict the domain to a finite set of inputs. We use the definition of restriction in Definition 6 to formally define the "functions" we mean and practically use in our paper:

Definition 19 (f_{RES})**.** *Consider a DNN whose input domain is $\mathcal{X}$ with a last layer width $d = 1$. Let $C = \{c_1, \ldots, c_n\} \subset \mathcal{X}$ be any finite subset of $\mathcal{X}$ with cardinality $n \in \mathbb{N}$. The restriction of function space $\mathcal{F} \in \{\mathcal{F}_{\text{DNN}}, \mathcal{F}_{\text{LAB}}\}$ to C is denoted as $\mathcal{F}^C$, and is defined as the space of all functions from C to $\mathcal{Y}$ realizable by functions in $\mathcal{F}$. We denote with f_{RES} elements of their corresponding spaces of restricted functions. Specifically, in regression:*

$$f_{\text{RES}} \in \mathcal{F}^C_{\text{DNN}} : C \to \mathbb{R}$$

and in binary classification:

$$f_{\text{RES}} \in \mathcal{F}^C_{\text{LAB}} : C \to \{0, 1\}$$

Note that in Definition 19 we only consider scalar outputs in the regression setting. For multiple-output functions, one approach is to consider d Gaussian processes and compute the combined kernel [Alvarez et al., 2011].

In statistical learning theory, the function spaces $\mathcal{F}_{\text{DNN}}$ and $\mathcal{F}_{\text{LAB}}$ are also called *hypotheses classes*, with their elements called *hypotheses* [Shalev-Shwartz and Ben-David, 2014]. It is important to note that our definition of prior and its calculation is based on the restriction of the hypotheses class to the concatenation of training set and test set $S + E$. Mathematically, this means the prior of a function $P(f)$ we calculated in the paper is precisely $P(f_{\text{RES}})$, except for the Boolean system in Section 3.5.1, where the input domain $\mathcal{X}$ is discrete and small enough to enumerate (this can also be thought of as the trivial restriction). As explained above, this restriction is inevitable if we want to compute the prior over $\mathcal{F}_{\text{DNN}}$ or $\mathcal{F}_{\text{LAB}}$. A simple example on MNIST [LeCun et al., 1998] can also help to gain a intuition of the necessity of such restriction, where all inputs would include the set of 28x28 integer matrices whose entries take values from 0-255, which gives 256^{784} possible inputs. This indicates that for real-life data distributions the number

of all possible inputs is hyper-astronomically large, if not infinite. Nevertheless, In some cases, such as the Boolean system described in Valle-Pérez et al. [2018] and treated in section 3.5.1, there is no need for such restriction because it is feasible to enumerate all possible inputs: there are only 7 Boolean units which give $2^7 = 128$ possible data sample. However, even in such cases, the number of possible functions is still large ($2^{128} \approx 10^{38}$).

B.4 Gaussian process approximation of the prior

In this section, we sketch out how we calculated the prior of a function $P(f)$ [Valle-Pérez et al., 2018, Mingard et al., 2021]. As in those papers, we use Gaussian processes, which have been shown to be equivalent to DNNs in the limit of infinite layer width [Neal, 1994, Lee et al., 2017, Matthews et al., 2018, Tan, 2008, Rasmussen, 2003]. These neural network GPs (NNGPs) have been shown to accurately approximate the prior over functions $P(f)$ of finite-width Bayesian DNNs [Valle-Pérez et al., 2018, Matthews et al., 2018, Mingard et al., 2021].

For the NNGPs, a GP prior is placed on the pre-activations z of the last layer of the neural network (before a final non-linearity, e.g. softmax, is applied), meaning that for any finite inputs set $\mathbf{x} = \{x_1, \ldots, x_n\}$, the random output vector (pre-activations) $\boldsymbol{z} = [z(x_1), \ldots, z(x_n)]^T$ has a Gaussian distribution. Note that in this paper, the the last layer has a single activation since we only focus on binary classification. This setting is corresponding to the definition of function restriction is Definition 19, with $\mathbf{z} \in \mathbb{R}^n$. Without loss of generality, we can assume such a process has a zero mean. The prior probability of the outputs $\boldsymbol{z}$ can be calculated as:

$$P(\boldsymbol{z}) = \frac{1}{(2\pi)^{\frac{n}{2}}\Sigma^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\boldsymbol{z}^T\Sigma^{-1}\boldsymbol{z}\right) \tag{B.2}$$

Σ is the covariance matrix (often called kernel), whose entries are defined as $\Sigma(x_i, x_j) \equiv \mathbb{E}[z(x_i), z(x_j)]$. Neal [1994] gave the basic form of kernel Σ in single hidden layer case, where Σ depends on the variance of weights and biases (σ_w and σ_b). In DNNs with multiple hidden layers, the kernel for layer l can be calculated recursively by induction, assuming the layer $l-1$ is a GP [Lee et al., 2017,

Matthews et al., 2018]. The kernel for fully connected ReLU-activated networks has a well known *arc-cosine kernel* analytical form [Cho and Saul, 2009], which we used in all FCNs in our work.

For ResNet50, the analytical form of GP kernel is intractable. Instead, we use a Monte Carlo empirical kernel [Novak et al., 2018b], and apply one step of the fully connected GP recurrence relation [Lee et al., 2017], taking advantage of the fact that the last layer of ResNet50 is fully connected. Mathematically, the empirical kernel can be expressed as:

$$\tilde{\Sigma}\left(x_i, x_j\right) := \frac{\sigma_w^2}{Mn} \sum_{m=1}^{M} \sum_{c=1}^{n} \left(h_{\mathbf{w}_m}^{L-1}(x_i)\right)_c \left(h_{\mathbf{w}_m}^{L-1}\left(x_j\right)\right)_c + \sigma_b^2 \tag{B.3}$$

where $\left(h_{\mathbf{w}_m}^{L-1}(x)\right)_c$ is the activation of c-th neuron in the last hidden layer (L is the total number of layers) for the network parameterized by the m-th sampling of parameters $\mathbf{w}_m$, M is the number of total Monte Carlo sampling, n is the width of the final hidden layer, and σ_w, σ_b are the weights and biases variance respectively. In our experiments, M is set to be $0.1 \times (|S| + |E|)$.

After calculating $P(\boldsymbol{z})$ with the corresponding kernel, the prior over (coarse-grained) restriction of functions $P(f)$ can be calculated through likelihood $P(f|\boldsymbol{z})$, which in our case is just a Heaviside function representing a hard sign nonlinearity. As non-Gaussian likelihood produces an intractable $P(f)$, we used Expectation Propagation (EP) algorithm for the approximation of $P(f)$ [Rasmussen, 2003]. This same EP approximation was used in Mingard et al. [2021] where it is discussed further. We represent the function f by the input-output pairs on the concatenation of training set and test set $S + E$.

B.5 Implementing parameter re-scaling

In this section we describe in detail how we implement the alpha scaling in DNNs first proposed by Dinh et al. [2017a]. The widely used rectified linear activation (ReLU) function

$$\phi_{\text{rect}}(x) = \max(x, 0)$$

exhibits the so-called "non-negative homogeneity" property:

$$\forall(z,\alpha) \in \mathbb{R} \times \mathbb{R}^{+}, \phi_{\text{rect}}(z\alpha) = \alpha\phi_{\text{rect}}(z)$$

The action of a L-layered deep feed-forward neural network can be written as:

$$y = \phi_{\text{rect}}\left(\phi_{\text{rect}}\left(\ldots\phi_{\text{rect}}\left(x \cdot W_1 + b_1\right)\ldots\right) \cdot W_{L-1} + b_{L-1}\right) \cdot W_L + b_L$$

in which

- x is the input vector
- W_L is the weight matrix of the L-th layer
- b_L is the bias vector of the L-th layer

To simplify notation, we have not included the final activation function, which may take any form (softmax or sigmoid etc.) without modification of the proceeding arguments. Generalizing the original arguments from Dinh et al. [2017a] slightly to include bias terms, we exploit the non-negative homogeneity of the ReLU function to find that a so-called "α-scaling" of one of the layers will not change its behaviour. Explicitly applying this to the i-th layer yields:

$$\left(\phi_{\text{rect}}\left(x \cdot \alpha W_i + \alpha b_i\right)\right) \cdot \frac{1}{\alpha} W_{i+1} = \left(\phi_{\text{rect}}\left(x \cdot W_i + b_i\right)\right) \cdot W_{i+1} \tag{B.4}$$

Clearly, the transformation described by $(W_i, b_i, W_{i+1}) \rightarrow \left(\alpha W_i, \alpha b_i, \frac{1}{\alpha} W_{i+1}\right)$ will lead to an observationally equivalent network (that is, a network whose output is identical for any given input, even if the weight and bias terms differ).

Since the α scaling transformation does not change the function, it does not change the prior of the function. However, for large enough α, as shown for example in Fig. 3.5, we see that SGD can be "knocked" out of the current neutral space because of the large gradients that are induced by the α scaling. This typically leads to the prior suddenly surging up, because the random nature of the perturbation means that the system is more likely to land on large volume functions. However, we always observe that the prior then drops back down quite quickly as SGD

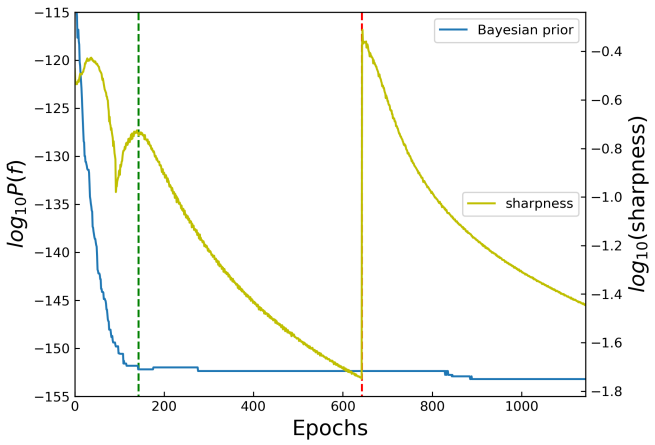


Figure B.2: The effect of alpha scaling on prior and sharpness. At each epoch we calculate the sharpness and the prior for our FCN on MNIST system with $|S| = 500$. The green dashed line denotes where zero-training error is reached and post-training starts. The red dashed line denotes the epoch where α-scaling takes place with $\alpha = 5.0$. Here the value of α is not big enough to "knock" the optimizer out of the neutral space, upon alpha scaling, in contrast to Fig. 3.5. As expected, we observe no change in prior upon alpha scaling (note that prior can change on overtraining if a slightly different function is found by SGD). The sharpness shows a larger peak upon alpha-scaling, as expected. See Section 3.9.

reaches zero training error again. On the other hand, as shown in Fig. B.2, when the value of α is smaller it does not knock SGD out of the neutral space, and so the prior does not change at all. Nevertheless, the sharpness still exhibits a strong spike due to the the alpha scaling.

Although not in the scope of this work, it is worth noting that the alpha scaling process in Convolutional Neural Networks (CNNs) with batch normalization [Ioffe and Szegedy, 2015] layer(s) is somewhat different. Because a batch normalization layer will eliminate all affine transformations applied on its inputs, one can arbitrarily alpha scale the layers before a batch normalization layer without needing to of compensate in following layer, provided the scaling is linear.

C

Additional appendices for Chapter 4

C.1 Measure families referenced in Chapter 4

We adopt the categories and normalizations used by prior experimental studies of generalization measures [Jiang et al., 2019b, Dziugaite et al., 2020b]; for exact constants and implementation choices, see App. C.6 of Dziugaite et al. [2020b]. Let a d-layer network have layer weights $\{W_i\}_{i=1}^d$ with initialization $\{W_i^0\}$; write $\|\cdot\|_F$ and $\|\cdot\|_2$ for Frobenius and spectral norms; let n denote training-set size and γ a robust (e.g., 10th-percentile) training margin.

- **Frobenius distances.** Layerwise distances from initialization and norm aggregates, e.g.

$$C_{\text{FrobDist}} \;=\; \sqrt{\sum_{i=1}^{d} \|W_i - W_i^0\|_F^2 \,/\, n}\,, \qquad C_{\text{param}} \;=\; \sqrt{\sum_{i=1}^{d} \|W_i\|_F^2 \,/\, n}\,.$$

- **Inverse margin.** A margin-based surrogate,

$$C_{\text{inv-margin}} \;\propto\; \frac{\sqrt{n}}{\gamma}\,.$$

- **Spectral metrics.** Products/means and distances in spectral norm, e.g.

$$C_{\prod \text{spec}} \;=\; \sqrt{\prod_{i=1}^{d} \|W_i\|_2^2 \,/\, n}\,, \qquad C_{\text{DistSpecInit}} \;=\; \sqrt{\sum_{i=1}^{d} \|W_i - W_i^0\|_2^2 \,/\, n}\,.$$

- **Combined spectral–Frobenius ratio.** We follow the combined ratio used in prior large-scale studies (App. C.6 of Dziugaite et al., 2020a), denoted `FRO_OVER_SPEC`, which normalizes Frobenius quantities by spectral ones to reduce raw scale effects. We report it alongside its constituents in our plots.

- **PAC-Bayes families and flatness proxies.** Bounds/proxies parameterized by posterior radii σ (and magnitude-aware σ_0), e.g.

$$C_{\text{PACBAYES-ORIG}} = \frac{1}{\sqrt{n}}\sqrt{\frac{\|w\|_2^2}{4\sigma^2} + \log\left(\frac{n}{\delta}\right) + 10},$$
$$C_{\text{Flatness}} = \frac{1}{\sigma\sqrt{n}}, \qquad C_{\text{MAG-Flatness}} = \frac{1}{\sigma_0\sqrt{n}}.$$

- **Path norms.** With $w = \text{vec}(W_1, \ldots, W_d)$ and $f_{w^2}(\mathbf{1})[i]$ the ith logit when all weights are squared elementwise and the input is all ones,

$$C_{\text{path.norm}} = \sqrt{\sum_i f_{w^2}(\mathbf{1})[i] \;/\; n}, \qquad C_{\text{path.norm-over-margin}} = \sqrt{\sum_i f_{w^2}(\mathbf{1})[i] \;/\; (\gamma^2 n)}.$$

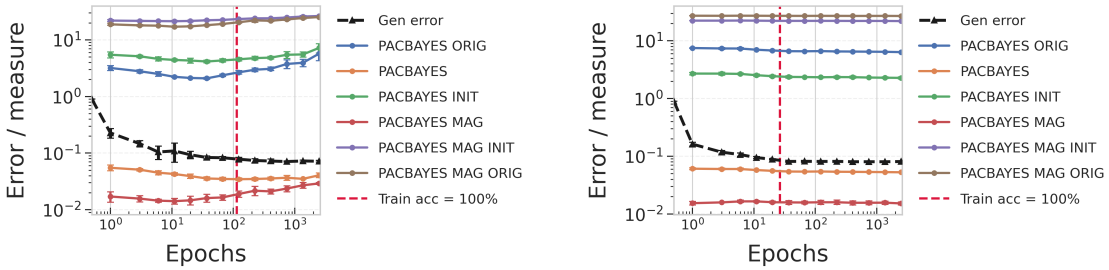
- **VC-dimension proxy (parameter count).** A coarse parameter-count surrogate,

$$C_{\text{params}} = \sqrt{\sum_{i=1}^{d} k_i^2\, c_{i-1}(c_i + 1) \;/\; n}\,,$$

with kernel sizes k_i and channel counts c_i.

C.2 Additional temporal behavior results: optimizer sensitivity across measure families

This section extends the temporal analysis in Section 4.4 by holding the task and hyperparameters fixed and changing only the optimizer. We train ResNet–50 on FashionMNIST with learning rate 0.01 and no early stopping, and we track each measure across epochs. All panels use a logarithmic epoch axis; this makes the early regime and successive orders of magnitude more legible, while very late additive-only epochs occupy little horizontal extent unless they span a substantial multiplicative range. The red dashed vertical line in every panel marks the first epoch at which training accuracy reaches 100%; on a log axis this event is still



(a) Adam **(b)** SGD with momentum

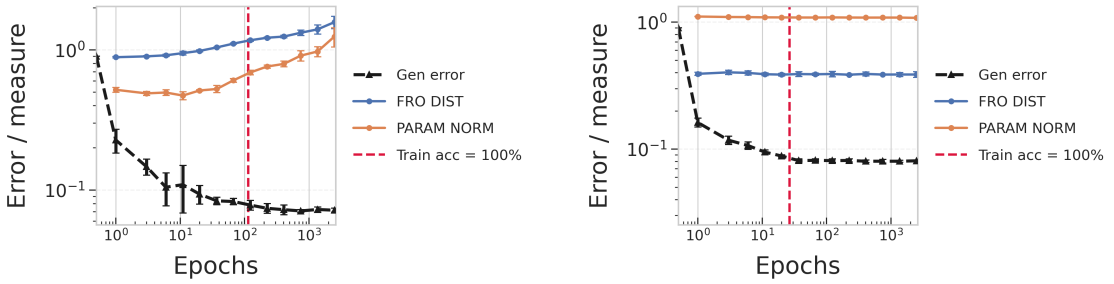
Figure C.1: Temporal behavior for PAC–Bayes variants on ResNet–50/FashionMNIST at fixed learning rate 0.01. The epoch axis is logarithmic and the red dashed vertical line marks the first 100% training-accuracy epoch. ADAM exhibits slow post-accuracy growth (notably in `PACBAYES_ORIG` and `PACBAYES_INIT`), whereas SGD with momentum keeps the family flat once the dashed line is crossed.

easy to spot, but short post-interpolation intervals can appear visually narrow if they do not cover a large multiplicative window. To avoid duplication with the main text, we omit the path-norm panels here and focus on complementary families whose behavior further illustrates optimizer sensitivity.

The PAC–Bayes family provides a clean illustration of this theme. Under ADAM, multiple bounds show slow, persistent growth after interpolation; on a log-time axis this appears as a steady positive slope across late decades of epochs, most clearly for `PACBAYES_ORIG` and `PACBAYES_INIT`. With momentum SGD, the same curves remain essentially flat within error bars once the dashed line is crossed, and on the log axis they sit nearly horizontal, emphasizing stability rather than drift.

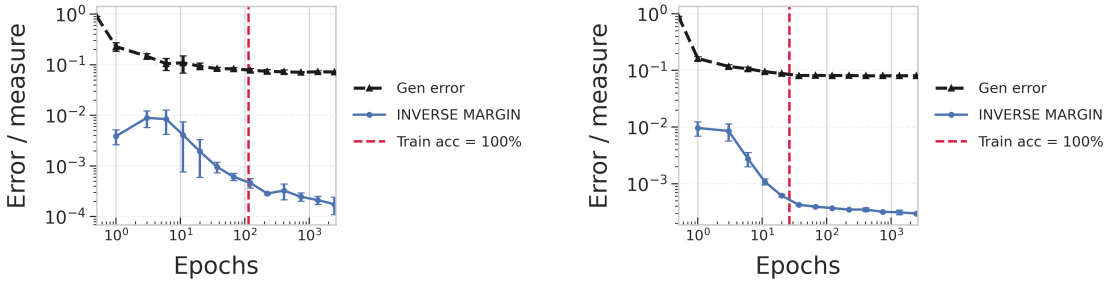
Measures tied to weight scale show the starkest divergence. In the Frobenius panel, ADAM drives both the distance to initialization and the parameter norm upward almost monotonically after the model has interpolated; on a log-time axis this shows up as a persistent positive slope across late epochs. By contrast, momentum SGD leaves both traces effectively horizontal once the dashed line is passed, highlighting a stable plateau.

Not every family bends under this perturbation. Both optimizers rapidly reduce the inverse-margin surrogate early in training and then hold it near a floor. The log-time axis spreads out the initial drop, making the shared shape and



(a) Adam **(b)** SGD with momentum

Figure C.2: Frobenius distance and parameter norm through time. The epoch axis is logarithmic. ADAM produces continued growth in both `FRO_DIST` and `PARAM_NORM` after 100% training accuracy, while SGD with momentum holds them near a constant level.



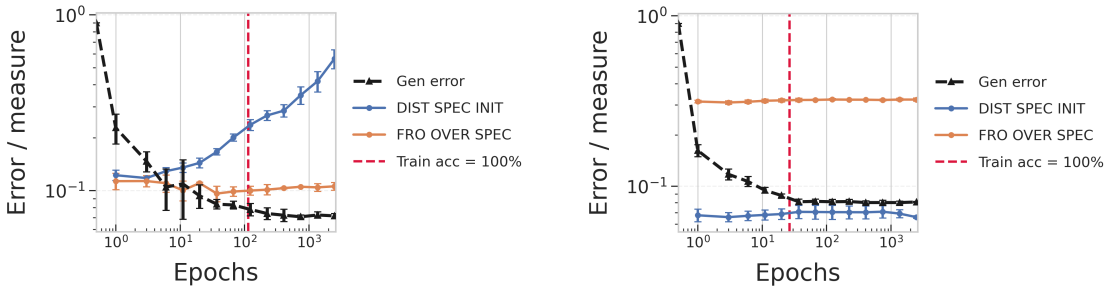
(a) Adam **(b)** SGD with momentum

Figure C.3: Inverse-margin surrogate through time on a logarithmic epoch axis. Both optimizers shrink the measure quickly and then hover near a floor; ADAM stabilizes at a slightly higher level but the overall shape is shared.

timing transparent; ADAM settles at a slightly higher level, but the trajectories otherwise coincide.

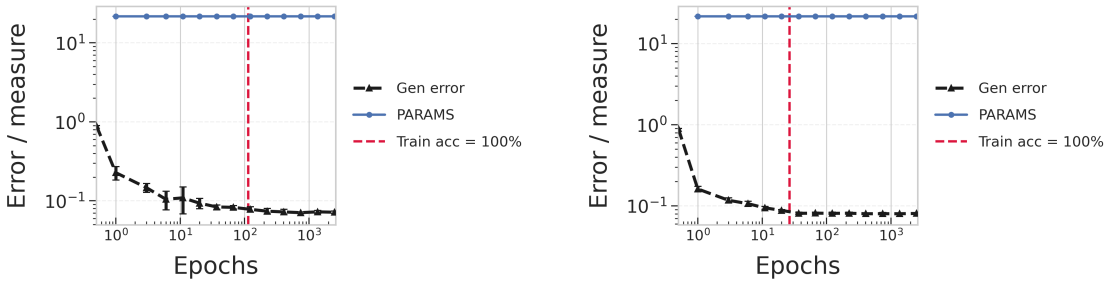
Spectral surrogates reveal a subtler but consistent imprint. Under ADAM, the distance from initialization in spectral norm drifts upward over time; on log time the slope is small but positive beyond the dashed line. Under momentum SGD, the same quantity gently decreases from a plateau, appearing as a mild negative slope. The ratio `FRO_OVER_SPEC` also separates in level, hinting that the optimizer reshapes how mass is distributed across singular directions even when predictive performance is unchanged.

As a neutral reference, the VC-dimension proxy behaves identically across optimizers by construction, and the accompanying generalization error follows the same calm trajectory. The log-time axis makes this invariance explicit: the traces



(a) Adam **(b)** SGD with momentum

Figure C.4: Spectral metrics through time. The epoch axis is logarithmic. ADAM drives `DIST_SPEC_INIT` upward after interpolation, while momentum SGD yields a gentle decline; `FRO_OVER_SPEC` diverges slightly in level.

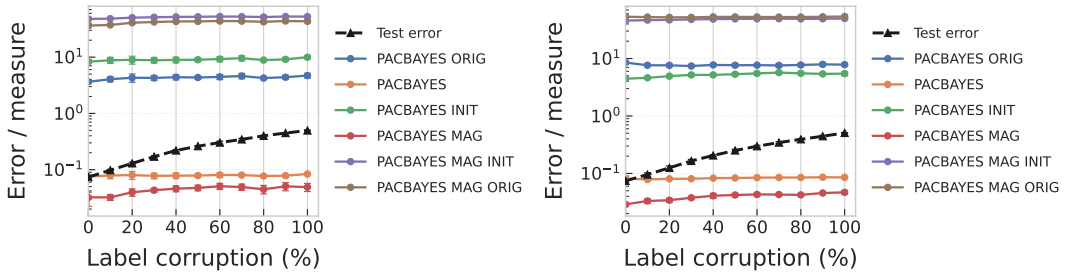


(a) Adam **(b)** SGD with momentum

Figure C.5: VC-dimension proxy and generalization error through time on a logarithmic epoch axis. The parameter-count surrogate is identical for both optimizers and the error trajectories are similarly calm, providing a neutral reference.

remain overlapped across decades of epochs, serving as anchors that remind us not all measures are sensitive to the optimizer change.

Taken together, these appendix figures broaden the temporal evidence. Several measure families that depend directly on weight scale or spectrum—Frobenius norms, spectral surrogates, and parts of the PAC–Bayes suite—react strongly to an optimizer swap despite matching accuracy, while others such as margin-based quantities and the VC proxy remain largely stable. Reading these results in aggregate helps separate measure-intrinsic behavior from optimizer-driven drift and, with the log-time view, clarifies whether apparent motion reflects genuine multiplicative change or merely late-stage additive updates. For temporal behavior of path norms, see Section 4.4, where those panels are discussed in detail.



(a) Adam, LR = 0.01 **(b)** Adam, LR = 0.001

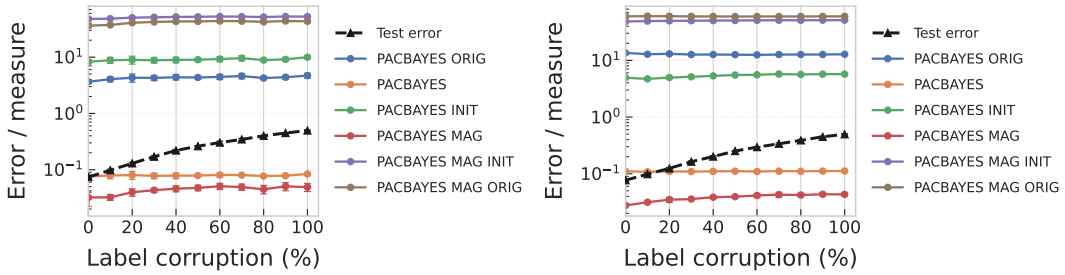
Figure C.6: PAC–Bayes measures vs. label corruption under Adam. With LR=10^{-2}, the family rises steadily with corruption; with LR=10^{-3} it sits higher overall and shows a shallow U–shape. *All panels use 10,000 training samples.*

C.3 Additional label-corruption results: PAC–Bayes and Path Norms

This appendix gathers the label–corruption results referenced in the main text for both PAC–Bayes–style measures and Path Norms.

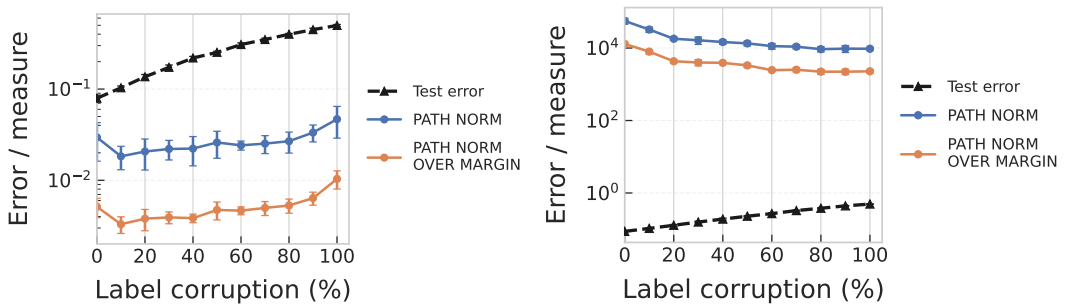
PAC–Bayes measures. Under Adam, lowering the learning rate from 10^{-2} to 10^{-3} transforms a clean, steadily rising ramp (e.g., `PACBAYES_ORIG` from $\sim$ 3.6 past 4.7) into a higher–lying but *shallower* U–shape centred around $\sim$ 8. At fixed learning rate 10^{-2}, swapping Adam for SGD produces a striking level shift: SGD yields a *high plateau* ($\sim$ 12–13) with little curvature, while Adam traces a *low*, clearly increasing arc ($\sim$ 3.6–4.7). Even within the PAC–Bayes family, members respond differently by optimizer: `PACBAYES_INIT` grows aggressively under Adam but only mildly under SGD. These patterns illustrate an (often) *insensitive* or contradictory relationship between the measure and increasing label corruption—another facet of fragility.

Path norms under SGD (momentum). Mirroring the Adam case in the main text, SGD with momentum exhibits an equally striking flip (Fig. C.8): at LR=10^{-3}, the path-norm trajectory lives in the 10^4–10^5 band and *decays* steadily with corruption; at LR=10^{-1}, the scale *crashes* to nearly zero and the curve *climbs*



(a) Adam, LR = 0.01 **(b)** SGD (momentum), LR = 0.01

Figure C.7: PAC–Bayes measures vs. label corruption at fixed LR (10^{-2}), swapping only the optimizer. Adam yields a low, steadily rising family (e.g., `PACBAYES_ORIG` $\approx$ 3.6–4.7), whereas SGD holds a high plateau ($\sim$ 12–13) with minimal curvature; `PACBAYES_INIT` grows far more under ADAM than under SGD. *All panels use 10,000 training samples.*



(a) LR = 0.1 **(b)** LR = 0.001

Figure C.8: Path norms vs. label corruption with *SGD (momentum)*; panels differ only in learning rate. At LR=10^{-3} the curve sits in 10^4–10^5 and decays with corruption; at LR=10^{-1} it lives near zero and rises. Both the direction and the dynamic range flip—another instance of qualitative mismatch across a minimal training change. *All panels use 10,000 training samples.*

monotonically. Trend and scale both invert. If a reader tried to infer "harder data $\Rightarrow$ larger path norm" from one panel, the other would immediately contradict it.

C.4 Stress-testing generalization measures with pixel permutations

A useful generalization measure should pass two basic stress tests: (i) it should be *insensitive* to symmetry-preserving transformations that do not change the intrinsic task, yet (ii) *sensitive* when task-relevant information is destroyed. Pixel permutations provide such a testbed: applying the *same* permutation to the training

and test sets preserves label–input relationships (a symmetry for fully connected networks), whereas applying *independent* random permutations to train and test destroys spatial structure and any usable signal.

We evaluate three families of measures alongside test error on MNIST with two architectures: a fully connected network (FCN) and a ResNet-50. We vary the optimizer (SGD vs. ADAM) and the early-stopping criterion (best cross-entropy "CE" vs. best accuracy "Acc") to expose hyperparameter sensitivity. Results are summarized in Tables C.1 and C.2.

FCN (permutation symmetry holds). When the same pixel permutation is applied to both train and test, the FCN effectively sees an unchanged task. As expected, test error remains essentially constant (≈ 0.03 across optimizers/stopping), and the PAC-Bayesian marginal likelihood (ML) bound is flat (0.142 throughout). This indicates desirable *robustness* to symmetry-preserving changes. In contrast, under independent random permutations, test error rises to $\approx$ 0.49–0.50 (near random guessing), and the ML bound increases to 0.532—appropriately reflecting reduced learnability. The path norm and the original PAC-Bayes variant (PACBAYES-ORIG) trend upward in this harder setting, but they also exhibit large spread across optimizers/stopping; e.g., the path norm spans from 0.105 to 6.632 under random shuffling, smaller than some unshuffled cases, revealing *fragility* to seemingly minor training choices.

ResNet-50 (permutation symmetry broken). Because convolutional inductive biases depend on spatial locality, even a single shared permutation distorts the data geometry and degrades performance (test error increases from $\approx$ 0.014–0.025 to $\approx$ 0.041–0.060; the ML bound rises from 0.101 to 0.170). With independent random permutations, test error again moves to $\approx$ 0.44–0.46 and the ML bound to ≈ 0.504. The path norm is particularly volatile here, spanning *orders of magnitude* across training choices (e.g., 0.016 vs. 3375.355), underscoring substantial *measure fragility*.

Table C.1: Pixel permutations with **MNIST + FCN** (training set size n=10,000). Same-permutation preserves the FCN's permutation symmetry; independent (random) permutations destroy learnable signal.

Optimizer / stopping	**Test err.**	**PAC-Bayes ML**	**Path norm**	**PACBAYES-ORIG**
Unshuffled pixels				
SGD / CE	0.033	0.142	0.080	1.534
ADAM / CE	0.034	0.142	0.173	1.208
SGD / Acc	0.032	0.142	0.081	1.535
ADAM / Acc	0.031	0.142	0.379	1.187
Same permutation on train & test				
SGD / CE	0.032	0.142	0.080	1.558
ADAM / CE	0.034	0.142	0.184	1.202
SGD / Acc	0.032	0.142	0.081	1.534
ADAM / Acc	0.032	0.142	0.314	1.183
Independent random permutations (train & test)				
SGD / CE	0.497	0.532	0.124	2.232
ADAM / CE	0.492	0.532	5.736	1.791
SGD / Acc	0.498	0.532	0.105	2.400
ADAM / Acc	0.488	0.532	6.632	1.837

CE: best checkpoint by cross-entropy; **Acc**: best checkpoint by accuracy.

Takeaways. Under symmetry-preserving changes (FCN + same-permutation), a robust measure should not move; under signal destruction (independent permutations), it should reflect the loss of learnability. The PAC-Bayesian ML bound behaves in this manner in both architectures, whereas the path norm and PACBAYES-ORIG can vary dramatically with optimizer/stopping, masking the underlying data effect. Reporting such sensitivity is crucial when proposing or comparing generalization measures, in line with our paper's emphasis on diagnosing and documenting *fragility*.[1]

[1]While label corruption is another standard knob for data complexity, here we focus on pixel permutations to isolate architectural symmetry vs. information destruction.

Table C.2: Pixel permutations with **MNIST + ResNet-50** (n=10,000). Convolutional inductive biases depend on spatial locality, so even a single shared permutation harms performance.

Optimizer / stopping	**Test err.**	**PAC-Bayes ML**	**Path norm**	**PACBAYES-ORIG**
Unshuffled pixels				
SGD / CE	0.022	0.101	2980.415	15.396
ADAM / CE	0.016	0.101	0.173	4.498
SGD / Acc	0.025	0.101	2972.887	15.125
ADAM / Acc	0.014	0.101	0.022	4.417
Same permutation on train & test				
SGD / CE	0.060	0.170	3375.355	13.534
ADAM / CE	0.046	0.170	0.016	4.074
SGD / Acc	0.056	0.170	3001.528	12.529
ADAM / Acc	0.041	0.170	0.021	3.909
Independent random permutations (train & test)				
SGD / CE	0.460	0.504	116.602	13.77
ADAM / CE	0.437	0.504	0.081	4.157
SGD / Acc	0.460	0.504	67.497	12.475
ADAM / Acc	0.437	0.504	0.084	4.119

CE: best checkpoint by cross-entropy; **Acc**: best checkpoint by accuracy.

D

Additional appendices for Chapter 5

D.1 Extending the ℓ_r-Scaling Theorem to Diagonal Linear Networks

This section is a blueprint for porting our main ℓ_r-scaling theorem from the minimum-ℓ_p interpolator to predictors selected by training *diagonal linear networks* (DLNs) with arbitrary depth. The goal is to reuse the entire spike+bulk argument with minimal surgery by swapping in the right implicit regularizer and the right one-dimensional balance. The guidance below covers both the two-layer case and the general depth-D case, aligning with the characterization of implicit bias in DLNs proved by Woodworth et al. [2020].

In our $\min \ell_p$ analysis, the predictor among all interpolators is selected by a separable power potential, and the proof runs through a dual "link" that maps the ray variable back to primal coordinates. DLNs fit exactly the same template:

- For two layers, the implicit regularizer is the hypentropy-type separable potential, and the link is the corresponding odd, strictly increasing map (Woodworth et al., Thm. 1). Non-uniform initialization simply reweights coordinates multiplicatively throughout.

- For depth $D \geq 3$, the implicit regularizer is again separable but with a depth-dependent link; Woodworth et al. (Thm. 3) identify the unique depth-D link

and its inverse. Practically, you can treat it as "the D-link" playing the role occupied by the power map in $\min \ell_p$ and by the hypentropy link at $D = 2$.

No other structural change is needed: once the link is fixed, every step of our proof goes through with the same spike/bulk decomposition and the same ray reduction.

As in the $\min \ell_p$ proof, restrict the dual variable to the ray spanned by the labels and determine a single scale t from a strictly monotone one-dimensional balance. Conceptually:

- In the *kernel-like window* (small arguments of the link on both spike and bulk), the link linearizes and the entire analysis collapses to the $p = 2$ case *verbatim*. This is the "lazy" regime.
- In the *rich-like window* (arguments large on the bulk and/or a dominant spike), the nonlinearity of the link controls the transition. For two layers, the balance yields a Lambert–W controlled scale; for $D \geq 3$, the depth-D link gives a faster, polynomial-in-initialization transition. You do not need a closed form—just the monotonicity and the small/large-argument asymptotics.

Bulk block. Replace the power moment used in the $\min \ell_p$ bulk bound by the depth-appropriate scalar functional that averages the link across a standard Gaussian coordinate. Operationally:

- Define a *bulk scalar* by applying the DLN link at the ray scale to a single Gaussian coordinate and taking its ℓ_r moment (to the $1/r$). This plays the exact role of $m_t^{1/t}$ in the $\min \ell_p$ proof.
- Use the same Gaussian embedding for the bulk design to lift this scalar to the full bulk contribution. In the kernel-like window you recover the $p = 2$ scaling exactly; in the rich-like window you get the accelerated depth-D growth predicted by the link's large-argument behavior.
- Keep track of the global scaling coming from the link's overall prefactor (this carries the initialization scale); it multiplies both bulk and spike-remainder terms.

Spike block. On the spike coordinates, keep the original two-part structure:

- *Spike-main:* apply the link to the mean shift determined by the signal; if a single coordinate dominates the one-dimensional balance, the selected predictor saturates at the spike scale and becomes essentially independent of the initialization (up to lower-order logarithmic or depth-dependent corrections).
- *Spike-remainder:* control the residual Gaussian fluctuation by the same operator-norm and concentration events as in the $\min \ell_p$ proof; its ℓ_r size is the bulk scalar (at the ray scale) times $s^{\max\{1/r,\,1/2\}}$, again multiplied by the link's global prefactor.

When spikes are *meek* relative to the bulk (no dominant coordinate), the spike block linearizes and you are back in the $p = 2$ laws.

Unified bound. After these replacements, the final display has the identical three-term structure:

$$\textit{DLN predictor's } \ell_r \textit{ size} = \textit{maximum of } (\text{spike-main, bulk, spike-remainder}),$$

with each term obtained from the $\min \ell_p$ counterpart by: (i) replacing the power link with the DLN link; (ii) inserting the link's global prefactor; and (iii) using the DLN bulk scalar in place of the power moment. In the kernel-like window this reproduces the $p = 2$ version *exactly*; in the rich-like window you get either bulk-controlled growth (Lambert–W for two layers; depth-accelerated for $D \geq 3$) or spike saturation.

Depth and initialization intricacy.

- **Depth $D \geq 3$.** The depth-D link is odd, strictly increasing, and has a simple linearization at the origin and an explicit rational form away from it (Woodworth et al., Thm. 3). This yields the same kernel-like reduction and a sharper rich-like transition than at $D = 2$. You never need its closed form—only its monotonicity and asymptotics.

- **Non-uniform initialization.** The per-coordinate *shape* of the initialization simply reweights the separable potential and carries multiplicatively through the link. Every bound inherits these weights in a purely multiplicative way (Woodworth et al., Thm. 1).

- **Limits.** Large initialization recovers the minimum-ℓ_2 norm predictor; vanishing initialization recovers the minimum-ℓ_1 predictor (with the usual caveats on how small "small" must be). These are the DLN analogues of the kernel and rich limits and hold for all depths covered above.

A handy dictionary for porting the proof. To translate any display or lemma from the min ℓ_p analysis to DLNs, we can make the following substitutions:

1. **Power link $\to$ DLN link**: replace the power map by the depth-appropriate link (hypentropy at two layers; the depth-D link from Woodworth et al. otherwise), including its global prefactor.

2. **Ray scale $\to$ DLN balance**: keep the same one-dimensional, strictly monotone balance along the label ray; solve it numerically or via asymptotics (linear in the kernel-like window; Lambert–W at two layers and power-law at depth $D \geq 3$ in the rich-like window).

3. **Bulk scalar**: replace the power moment by the ℓ_r moment of the DLN link applied to a single Gaussian coordinate at the ray scale; lift via the Gaussian embedding exactly as before.

4. **Spike block**: reuse the deterministic-plus-Gaussian decomposition, the operator-norm and concentration events, and the same ℓ_r geometry; only the link and its global prefactor change.

With the substitutions above, the ℓ_r-scaling analysis for the minimum-ℓ_p interpolator transfers directly to DLNs of any depth. The proof structure, the spike/bulk decomposition, and the final three-term form remain identical; only the link and its scalar balance change. Two layers inherit a Lambert–W bulk scale; deeper networks

transition faster with initialization due to their depth-D link. In the kernel-like window, everything collapses to the $p = 2$ bounds almost word-for-word.