Doctoral Thesis in Computer Science

On Implicit Smoothness Regularization in Deep Learning

MATTEO GAMBA

KTH ROYAL INSTITUTE OF TECHNOLOGY

On Implicit Smoothness Regularization in Deep Learning

MATTEO GAMBA

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Thursday the 7th November 2024, at 15:00 p.m. in Kollegiesalen, Brinellvägen 6, Stockholm.

Doctoral Thesis in Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden 2024

TRITA-EECS-AVL-2024:80
ISBN 978-91-8106-077-5

Printed by: Universitetsservice US-AB, Sweden 2024

To Matilda

Abstract

State of the art neural networks provide a rich class of function approximators, fueling the remarkable success of gradient-based deep learning on complex high-dimensional problems, ranging from natural language modeling to image and video generation and understanding. Modern deep networks enjoy sufficient expressive power to shatter common classification benchmarks, as well as interpolate noisy regression targets. At the same time, the same models are able to generalize well whilst perfectly fitting noisy training data, even in the absence of external regularization constraining model expressivity. Efforts towards making sense of the observed *benign overfitting* behaviour uncovered its occurrence in overparameterized linear regression as well as kernel regression, extending classical empirical risk minimization to the study of minimum norm *interpolators*. Existing theoretical understanding of the phenomenon identifies two key factors affecting the generalization ability of interpolating models. First, overparameterization – corresponding to the regime in which a model counts more parameters than the number of constraints imposed by the training sample – effectively reduces model variance in proximity of the training data. Second, the structure of the learner – which determines how patterns in the training data are encoded in the learned representation – controls the ability to separate signal from noise when attaining interpolation. Analyzing the above factors for deep finite-width networks respectively entails characterizing the mechanisms driving feature learning and norm-based capacity control in practical settings, thus posing a challenging open problem. The present thesis explores the problem of capturing effective complexity of finite-width deep networks trained in practice, through the lens of model function geometry, focusing on factors *implicitly* restricting model complexity. First, model expressivity is contrasted to effective nonlinearity for models undergoing double descent, highlighting constrained effective complexity afforded by overparameterization. Second, the geometry of interpolation is studied in the presence of noisy targets, observing robust interpolation over volumes of size controlled by model scale. Third, the observed behavior is formally tied to parameter-space curvature, connecting parameter-space geometry to the input space's. Finally, the thesis concludes by investigating whether the findings translate to the context of self-supervised learning, relating the geometry of representations to downstream robustness, and highlighting trends in keeping with neural scaling laws. The present work isolates input-space smoothness as a key notion for characterizing effective complexity of model functions expressed by overparameterized deep networks.

Sammanfattning

Toppmoderna neurala nätverk erbjuder en rik klass funktionsapproximatorer, vilket stimulerar den anmärkningsvärda utvecklingen av gradientbaserad djupinlärning för komplexa högdimensionella problem, allt från modellering av naturligt språk till bild- och videogenerering och förståelse. Moderna djupa nätverk har tillräckligt mycket expressiv kraft för att kunna slå vanliga klassificeringsbenchmarks, samt interpolera brusiga regressionsmål. Samma modeller kan generalisera väl samtidigt som de kan anpassas perfekt till brusig träningsdata, även i frånvaro av extern regularisering som begränsar modellens uttrycksförmåga. Ansträngningar för att förstå det observerade så kallade *benign overfitting*-beteendet har påvisat dess förekomst i överparameteriserad linjär regression såväl som i kärn-baserad regression, vilket utvidgar klassisk empirisk riskminimering till studiet av miniminorm *interpolatorer*. Befintlig teoretisk förståelse av fenomenet identifierar två nyckelfaktorer som påverkar generaliseringsförmågan hos interpolerande modeller. För det första reducerar överparameterisering - motsvarande regimen där en modell har fler parametrar än antalet villkor som ställs av träningsproven - effektivt modellvariansen i närheten av träningsdatan. För det andra styr inlärningens struktur - som bestämmer hur mönster i träningsdata kodas i den inlärda representationen - förmågan att separera signal från brus när interpolering uppnås. Att analysera ovanstående faktorer för nätverk med djup ändlig bredd innebär att karakterisera de mekanismer som driver funktionsinlärning och normbaserad kapacitetskontroll i praktiska sammanhang, vilket utgör ett utmanande öppet problem. Den föreliggande avhandlingen utforskar problemet med att fånga den effektiva komplexiteten hos djupa nätverk med ändlig bredd som tränas i praktiken, sett genom linsen av modellfunktionens geometri, med fokus på faktorer som *implicit* begränsar modellens komplexitet. För det första kontrasteras modellexpressivitet till effektiv olinjäritet för modeller som genomgår så kallad double descent, vilket framhäver begränsad effektiv komplexitet som ges av överparameterisering. För det andra studeras interpolationens geometri i närvaro av brusiga mål, och observerar robust interpolation över volymer av storlekar bestämda av modellskalan. För det tredje kopplas det observerade beteendet formellt till parameter-rymdens krökning, vilket kopplar parameterrymdens geometri till indatarymdens. Slutligen avslutas avhandlingen med att undersöka huruvida resultaten kan översättas till kontexten av självövervakad inlärning, relaterar representationernas geometri till nedströms robusthet, och belyser trender i linje med neurala skalningslagar. Det föreliggande arbetet isolerar indatarymdens jämnhet som ett nyckelbegrepp för att karakterisera effektiv komplexitet hos modellfunktioner uttryckta av överparameteriserade djupa nätverk.

Acknowledgements

My PhD journey and my time in Sweden have been blessed by many inspiring people, who I had the honour and fortune of interacting with, and who I will always hold close to my heart.

I am forever grateful to my advisor, Mårten, for giving me the opportunity to pursue a topic I am deeply passionate about, for your kindness, guidance and patience through the long journey of the PhD. I wish to thank Hossein for inspiring me to work on deep learning, for your support in navigating academic life and for the many discussions about science, movies and life. I have learned a lot from both of you, in ways that no words can quantify. I would like to extend my deepest gratitude to Stefan, for your endless passion for geometry, for always asking the tough questions when discussing research, and for starting me on my topic. Thank you Josephine, for welcoming me to the deep learning group at RPL, for teaching me the value of well-tested implementations, and for a fantastic collaboration on the geometry of linear regions. Thank you Atsuto for our collaboration on feature contraction, as well as many fun discussions and lunches together.

Over the years, I was lucky to enjoy the company of several amazing office mates, who kept me sane through the ups and downs of academic life. Thank you Federico, Luca, and Erik for sharing this journey with me from the very beginning. Our interactions made me grow as a person as well as a researcher. Thank you also Federico and Erik for our amazing trip to California. Thank you David M for always being open to talk about new ideas, Math, and plants.

Thank you Ali, for taking me under your wing at the beginning of my studies, and for showing me that it is possible to start a family while pursuing a research career. Thank you Shuangshuang, Wenjie, and Zehang for teaching me about Chinese culture and food, for your humbleness and vast knowledge, and for the many hours spent together.

Thank you Chris, for our long discussions about optimization and Hessians, as well as for career advice. Thank you Amir, Wenyin, Heng, Klas, Mohammad, and Bharath for making the Monday meetings engaging and for bringing new energy to our group.

Thank you to my colleagues and friends at RPL, who made this journey special. To Giovanni, for bringing clarity through your extensive knowledge, for your passion for meaningful research, and for teaching me how to identify the core of an issue before attacking a problem. Thank you Alfredo, for the many conversations about the meaning of science, and for being a history nerd like me. Thank you Alberta, Yonk, Miguel, Aniss, and Jens, for the fun times together. Thank you Marco, for convincing me to play the guitar again after so many years, and for our evenings playing the Blues together.

Thank you Gustav, for making our trip to New Orleans memorable. Thank you Vladislav, for inspiring me to pursue a more theoretical path throughout my PhD. Thank you Adrian, for our work together on ReLU networks.

Thank you Olga, Akshaya, Özer and Georg, for the many fun evenings playing board games together, and to Xiaomeng for guiding me through Seoul, and for inviting me to your D&D sessions with Juan and Jim. Thank you David B, for our trip together to Aberdeen, and to Farzaneh for bringing positive energy to the lab and for our trip to Vienna.

Thank you Amir, Qingwen, Joonatan, and Vladimir for making GPU administration fun. Thank you Patric, Christian, and Grace for making RPL a big welcoming family.

Outside of RPL, I wish to thank Arna and Krishna, for a fantastic and ongoing collaboration, and Amaya for being a good friend and for our trips to Aberdeen, New Orleans, and Vienna.

During my years in Stockholm, I found a second family in my close friends, with whom I spent almost ten years together. Thank you Naresh, for always being there for me, and for our discussions about neuroscience; to Halit for being a true friend through the good and the hard times; to Erica, for being my Swedish sister, and to Eva for navigating our earlier years in Stockholm together. Thank you Björn and Jan, for welcoming me to Sweden. Thank you Pinar and Christoph, for being my family before I had a family, and for our fun vacations together. Thank you Stefano, for inspiring me to continue my studies, and to Jacopo for all the good parties. Thank you to Gabri, for your friendship and our discussions about polytopes, and to Andrea, Karin, Chiara, Massi, and Vani, for all these years together.

Thank you Juan, Marco, Palby, Arun, Sasi, rms, and Puria, for inspiring me, teaching me resilience, and introducing me to the scientific community through free software. Thank you to my parents, for always believing in me and for their unconditional support, for teaching me to pursue what I believe in, and the value of hard work. Thank you Berto, for teaching me how to programme and for making me passionate about science, and thank you to my brothers for their love and unwavering support. Thank you to my grandfather and my late grandmother. I will always cherish the times we spent on the phone together.

Above all, thank you Kimberley, for being my companion of 12 years, for knowing me better than I do, and for making this very journey possible through your selflessness and love. Thank you to my loving Matilda, who always has a smile for me, for giving a new meaning to my life.

Contents

List of acronyms and abbreviations

ERM	Empirical Risk Minimization. 7–9
ID	In-Distribution. 61–64, 69
MC	Monte Carlo. 29, 31, 32
MSE	Mean Squared Error. 9, 40
NTK	Neural Tangent Kernel. 4
OOD	Out-of-Distribution. 14, 60–64, 68–70
PSD	Positive Semi-Definite. 55
ReLU	Rectified Linear Unit. 13, 17–19, 21, 23, 25, 26, 35–37
SGD	Stochastic Gradient Descent. 7, 42
SSL	Self-Supervised Learning. 6, 13, 14, 49–53, 57–60, 62, 68, 70
VC	Vapnik-Chervonenkis. 5, 26

Part I

Overview

Chapter 1

Introduction

The past decade has set deep networks apart from other machine learning algorithms in two important ways. On the one hand, the synergy between deep networks and large-scale datasets has fueled the success and widespread adoption of deep learning for discriminative [1]–[7], generative [8]–[15], as well as reinforcement learning problems [16]–[21]. On the other hand, deep networks stand out for missing a comprehensive foundational theory accounting for their generalization ability [22]–[26]. In principle, a pragmatic theory of deep learning should be able to quantitatively predict, up to bounded error, the performance of a given network architecture on a specified task or data distribution, and prescribe recipes for improving performance, stability or efficiency of a model. According to no-free-lunch theorems [27], [28], the theory should account for inductive biases embedded in the model and error function considered, and their alignment with structural properties of the task [29]. Consequently, a comprehensive theory should allow to quantify robustness of a deep learning algorithm to certain distribution shifts [30], [31], for instance noisy and adversarially crafted input [32], [33], as well as noisy training targets [34]. Importantly, such a theory could allow to identify and isolate failure modes of a learning algorithm [35]–[37], and provide guarantees on safety of a model in the wild. Finally, a more direct theoretical understanding would allow to design novel architectures, regularization strategies, as well as faster optimizers, and guide practitioners in allocating resources when tackling a new problem [38], [39]. Efforts towards understanding the performance, limitations, and inner workings of deep networks can be broadly categorized into two groups, both of which have substantially shaped the current understanding of the field. On the one hand, experiment-driven approaches study emerging properties and structure of modern deep networks trained on real-world datasets, focusing on state of the art models and training practices, and accounting for the influence of specific hyperparameters and design choices. Foundational approaches, on the other hand, strip down a complex practical problem in order to build a minimal model of the phenomenon of interest, against which hypotheses can be formally defined and tested.

Towards A Foundational Theory of Deep Learning

Ultimately, a foundational theory should provide an essential model of deep learning, capturing the classes of problems that can be successfully cast as neural network optimization and can be approximately solved by first order methods [40]. In the pursuit of building a rigorous foundation of deep learning, *minimal models* allow to focus on selected aspects of the problem, which are often emphasized through asymptotic or mean-field analysis [41], while disregarding its more complex instantiations, *e.g.* the impact of a particular neural network topology or training hyperparameter, or by approximating a discrete problem with a continuous one [42]. While minimal models may provide a more rigid characterization of what is in practice a noisy stochastic process, they serve as a mathematical metaphor with which the problem can be understood and more precise conjectures can be formulated. Recent advances have provided minimal models of certain aspects of deep learning:

- The Neural Tangent Kernel (NTK) [43] establishes a connection between infinitely-wide neural networks and ridgeless kernel regression, allowing to study the evolution of deep networks under (kernel) gradient descent [44], [45], providing a closed form for the optimization trajectories. In the kernel regression duality, wide networks operate in the so called *lazy regime*, whereby parameters change negligibly from initialization [43], thereby failing to capture the rich regime [46] of feature learning. Particularly, the kernel regime fails to match the performance of finite-width networks in some settings [23], supporting the hypothesis that feature learning is an important component of deep learning [47]. To this end, recent works isolate tasks that are easily solved by finite-width networks, but not in the NTK regime [48]–[50].

- *Random matrix theory* [51] allows to describe deep networks at initialization [51], [52], as well as random feature models [53], with recent works indicating that "good" conditioning of the loss landscape is determined by the network architecture and parameter initialization scheme [54], emphasizing their role over the stochastic optimizer's in ensuring generalization. At the same time, Martin and Mahoney [55] report the emergence of heavy-tailed eigenspectra in the weight tensors of generalizing networks in connection with implicit model regularization in the feature-rich regime [56], noting the phenomenon is controlled on a minimal model by optimizer hyperparameters.

- Finally, *mean-field theory* provides a limiting case for infinitely-wide networks under a different parameterization of width-scaling, going beyond the lazy regime and exhibiting some degree of feature learning [57], [58].

Beyond current minimal models, a fundamental open question lies in understanding the emergence of structured representations in finite-width networks [59], which in large-scale classification settings [60] learn transferable semantic features [61]–[64], as well as show *emerging abilities* in large-scale models, *i.e.* desirable properties not directly enforced in the underlying optimization problem [22], [65], [66]. At present,

developing a minimal model of the problem remains an active area of research [59], [67]–[69]. Importantly, understanding the emergence of structures may help reveal the mechanisms underpinning generalization, motivating further empirical studies.

The Role of Empiricism in Shaping Theory

Empirical studies have played a considerable role in shaping current understanding of deep learning, by exposing emergent behaviour and structure of trained state-of-the-art models. Over the years, a number of *surprising findings* have been reported [33], [66], [70]–[80], exposing limitations of classical theory [78], [81], and yielding practical advances [82]–[86] including novel training paradigms [14], [32]. Remarkably, deep networks trained on supervised tasks have sufficient expressive power to perfectly fit arbitrary labelings of the training data [78], [87], [88], while at the same time achieving state-of-the-art performance on several natural datasets, suggesting that expressivity is effectively constrained when networks are trained on natural data. An important consequence is that classical complexity measures, such as Vapnik-Chervonenkis (VC) dimension [89] and Rademacher complexity [90], fail to adequately capture the generalization performance of modern deep networks, by estimating complexity *uniformly* over the entire hypothesis space, namely the space of functions theoretically expressible by the model. Indeed, overwhelming evidence shows that complexity is effectively constrained in practice, and that trained networks are endowed with stronger structure than what theoretically afforded by their high expressive power [91], and what captured by minimal models.

- According to the *lottery ticket hypothesis*, the final performance of a trained network can be retained and even improved when a large fraction of the weights are removed via iterative pruning and retraining [86], so long as the unpruned parameters are reset to their respective value at initialization before retraining. This suggests that, while depth and overparameterization are beneficial for successful training [92], only a very sparse subnetwork is responsible for the final model performance.

- Perhaps more strikingly, the performance of a trained network is largely unaffected if entire layers are reset to their value at initialization, a phenomenon known as *module criticality*. However, if those layers are removed and a shallower network is trained from scratch, the final performance is lower than that of the deeper model [77], [93].

- The loss landscape of deep networks exhibits *linear mode connectivity*, whereby the solutions of different training runs of the same architecture are connected by a linear path with low loss barriers [71], [74], [94], [95]. Averaging two connected solutions may lead to better performance [82], and allows for semantic manipulation of the features of fine-tuned models [85].

- Neural networks may exhibit *delayed generalization*, whereupon training initially results in memorization of the training data and random test perfor-

mance. However, under certain conditions, prolonged training may result in a sudden improvement in test performance, giving rise to *grokking* [72], [96], [97]. The phenomenon can be induced in practical settings by controlling the norm of the weights at initialization, and has been connected to the lazy regime of training [98]. Grokking has also been observed beyond generalization performance, with recent work highlighting the emergence of *delayed adversarial robustness* upon prolonged training [70].

The above evidence indicates that deep networks trained in practice express a more restricted set of solutions, characterized by emerging structures. Hence, understanding the generalization ability of deep networks is intimately tied to measuring *effective complexity*, *i.e.* capturing and modelling the hypothesis class of functions expressed in practice by networks trained on real-world data, in contrast to the space of functions theoretically expressible by a given architecture [55], [76], [99].

Guiding Questions

The present thesis focuses on overparameterized deep networks trained on supervised learning tasks, with the goal of characterizing effective complexity of deep networks trained in practice, thus focusing on the feature-rich regime. An important open question explored throughout the thesis is understanding when deep networks trained until *interpolation* on noisy data generalize. To investigate the question, a promising venue is identified in capturing structure and regularity emerging from training. Focusing on classification tasks, the problem is approached as a natural phenomenon [100], by measuring the effect of interventions on models trained in practice, thereby grounding the study on empirical evidence. A second important question lies in exploring mechanisms constraining effective complexity, in relation to the model architecture and its interaction with optimization. Finally, the problem of capturing generalization is extended to Self-Supervised Learning (SSL) [101]–[103], an unsupervised training paradigm yielding competitive performance [104] with supervised learning on vision and language tasks.

1.1 On the Ill-Posed Nature of Deep Learning

Isolating factors that control and constrain effective complexity entails several theoretical challenges. First, from a dynamics standpoint, the high-dimensional loss landscape of deep networks is non-convex, admitting multiple local minima [105]–[107], as well as saddles [108]. Second, from an expressivity perspective, modern architectures are *overparameterized*, i.e. they enjoy more degrees of freedom than the number of constraints imposed by the training sample [24], [87]. Indeed, overparameterized network optimization admits multiple solutions interpolating the training data, with vastly different generalization ability [72], [78], [97], contradicting uniform convergence assumptions [81]. Thus, the training of multi-layer networks is ill-posed, in that a single problem instance admits multiple possible solutions.

Definition 1.1.1 (Well-posed problem [109])**.** *A problem is said to be well-posed if, for every problem instance, there exists a unique solution which continuously depends on the initial conditions. Otherwise, the problem is ill-posed.*

The ill-posed nature of deep learning makes direct theoretical analysis challenging, since individual instances of a learning problem may admit multiple solutions with different generalization ability. However, in practical settings, first-order optimization can consistently recover generalizing solutions on a multitude of tasks [110], [111], suggesting the existence of additional mechanisms controlling complexity.

Explicit regularization A common strategy to address ill-posedness of a problem is Tikhonov regularization [112], which allows to restrict the space of solutions by adding constraints to the problem. If the constraints are formulated in terms of a controllable (hyper-)parameter, for instance expressed via a Lagrangian multiplier, then regularization is said to be *explicit* and its strength can be manually tuned. At present, different explicit regularizers are employed by practitioners to ensure stable training and boost performance of deep learning algorithms [113]–[115]. At the same time, deep networks attain non-trivial generalization performance also in the absence of explicit regularization [88], suggesting that the true factors underpinning generalization are implicitly embedded in the model architecture, weight initialization, and their interplay with the optimization procedure [25], [78], [116].

Implicit regularization The choice of model architecture and optimization algorithm can restrict or bias learning, effectively encoding a preference towards a particular family of solutions, thereby exerting *implicit* regularization [25]. Understanding implicit regularization in machine learning is an important problem even beyond neural networks [117], [118]. In some settings, characterizing an implicit regularization mechanism allows to formulate it as an explicit one, whereby regularization strength can be understood and controlled [119]. The neural network architecture as well as Stochastic Gradient Descent (SGD) are each thought to exert implicit regularization on training, encoding priors that are well aligned with the physical world [120]–[122], and biasing optimization trajectories towards stable solutions [123]. A central question to this thesis is exploring the emergence of regularization, interpreted as structure encoded in a model's parameters resulting in reduced expressivity. The focus of the works hereby presented is on isolating regularity of functions expressed in practice by deep networks, in relationship to the model architecture and its interaction with the optimizer. In the following, the formalism of Empirical Risk Minimization (ERM) is adopted to introduce the research questions explored in the thesis.

Empirical Risk Minimization

For a population distribution $\mathbb{P}(\mathbf{x}, y)$ jointly defined over a space of input-response pairs $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, supervised learning involves estimating the conditional dis-

tribution $\mathbb{P}(y|\mathbf{x})$ given access to samples from the population. The relationship between $\mathbf{x}$ and y is typically modelled by a function $f : \mathcal{X} \to \mathcal{Y}$ called *statistical hypothesis*, belonging to a larger set of functions $\mathcal{H}$ called *hypothesis space*. In simplified settings, uncertainty in the observed data due to error is expressed by the relationship $y = f^*(\mathbf{x}) + \varepsilon$, for conditionally independent noise ε with zero mean and variance $\sigma^2 > 0$. Given a non-negative measure $\mathcal{L}(f, \mathbf{x}, y)$ expressing the error incurred by a particular hypothesis f when predicting $f(\mathbf{x})$ with ground-truth y, the risk associated with the hypothesis is defined as

$$\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}(\mathbf{x},y)}[\mathcal{L}(f(\mathbf{x}), y)] \tag{1.1}$$

The optimal predictor f^* is defined as the one minimizing the risk

$$f^* = \operatorname*{argmin}_{f\in\mathcal{H}} \mathcal{R}(f) \tag{1.2}$$

However, since the population distribution is in general not accessible, a fixed set of i.i.d. data points $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is used to select f based on the *empirical risk*

$$\hat{\mathcal{R}}(f) = \mathbb{E}_{\mathcal{D}}[\mathcal{L}(f, \mathbf{x}, y)] \tag{1.3}$$

providing the name empirical risk minimization. For a training set $\mathcal{D}$ and hypothesis space $\mathcal{H}$, a *learning algorithm* $\mathcal{A}_\mathcal{H}$ selects a hypothesis f with low empirical risk

$$\hat{f} = \mathcal{A}_\mathcal{H}(f) = \operatorname*{argmin}_{f\in\mathcal{H}} \hat{\mathcal{R}}(f) \tag{1.4}$$

aiming for f to be as close as possible to f^*. In deep learning, ERM is cast as an optimization problem [40], whereby the hypothesis space $\mathcal{H} = \{\mathbf{f}_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y} \text{ s.t. } \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ is associated with a neural network architecture with parameter $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. The learning algorithm aims to select a function with low risk $\mathcal{R}(\mathbf{f}_{\boldsymbol{\theta}})$ by minimizing

$$\hat{\mathbf{f}}_{\boldsymbol{\theta}} = \operatorname*{argmin}_{\mathbf{f}_{\boldsymbol{\theta}}\in\mathcal{H}} \hat{\mathcal{R}}(\mathbf{f}_{\boldsymbol{\theta}}) \tag{1.5}$$

in turn solving the optimization problem

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \hat{\mathcal{R}}(\mathbf{f}_{\boldsymbol{\theta}}) \tag{1.6}$$

where the error function $\mathcal{L}$ is replaced by a surrogate loss amenable to optimization. Throughout the thesis, the input space and output space are usually identified with $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}^k$, while the parameter space with $\boldsymbol{\Theta} = \mathbb{R}^p$, with p denoting the number of learned model parameters (all weights and biases). Finally, in classification settings, the ground truth label $y \in [k]$ can be cast as an element $\mathbf{y} \in \mathbb{R}^k$ with $\mathbf{y} = \mathbf{e}_y$ denoting the y-th standard basis vector. The population and empirical risk are defined using the 0/1 loss $\mathcal{L}_{0/1}(\hat{y}, y) = [\hat{y} = y]$, with $\hat{y} = \operatorname{argmax}_{j\in[k]}(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}))_j$, while the surrogate loss $\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)$ is typically given by cross-entropy

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y}) = -\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i)_y - \log\sum_{j=1}^{k}\exp(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i)_j)\right) \tag{1.7}$$

or Mean Squared Error (MSE)

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y}) = \frac{1}{2n} \sum_{i=1}^{n} \|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i) - \mathbf{y}_i\|^2 \tag{1.8}$$

Generalization in the ERM setting is determined by how closely the learned model behaves w.r.t. the optimal model outside of the training sample (in distribution, out-of-sample generalization), and is measured via the *excess risk*

$$\mathcal{R}(\mathbf{f}_{\boldsymbol{\theta}}) - \mathcal{R}(\mathbf{f}^*_{\boldsymbol{\theta}}) \tag{1.9}$$

In general, attaining low empirical risk does not guarantee good generalization. In fact, the empirical risk minimizer may overfit noise in the training data, and thus perform worse on unseen data. Formally, for the case of MSE, the excess risk can be decomposed as the sum of three terms

$$\mathbb{E}_{(\mathbf{x},y),\varepsilon}[(y - \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}))^2] = \left(\mathbf{f}^*_{\boldsymbol{\theta}}(\mathbf{x}) - \mathbb{E}[\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})]\right)^2 + \mathbb{E}\Big[\left(\mathbb{E}[\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})] - \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})\right)^2\Big] + \sigma^2 \tag{1.10}$$

namely (squared) model bias, model variance, and irreducible error caused by the noise ε. For a notion of *model complexity* – estimating the expressive power of a given hypothesis by measuring a proxy quantity – the excess risk decomposition allows to interpret the error committed by an empirical risk minimizer in terms of the trade-off between model bias and variance [124]. A high-bias model may underfit the training data by imposing too restrictive assumptions on the hypothesis class, while a high-variance model may overfit the training data due to high expressivity. As model complexity increases, the population error follows a U-shaped curve, with lowest value observed at the optimal bias/variance trade-off. Historically, the number of model parameters [125] and training epochs [119] have been used as an intuitive proxy for complexity of linear regression models and neural networks.

Minimum-norm interpolators When overparameterized models are trained until zero empirical risk, Equation 1.4 admits infinitely many solutions. Formally, the level sets $\mathcal{L}^{-1}(0)$ represent a collection of manifolds in parameter space, forming connected components [126] each expressing interpolating solutions. The study of interpolating solutions is thus typically restricted to certain classes of functions, with the goal of accounting for inductive biases embedded in the network architecture and learning algorithm. Ideally, according to the principle of *Occam's razor*, one seeks the model of least complexity with enough expressivity to explain the data [127], for some measurable notion of "complexity". Extending the ERM formalism, inductive biases are typically incorporated via a choice of norm in the hypothesis space $\mathcal{H}$, thus endowing it with a (complete) metric structure. Particularly, for a given choice of norm $\|\cdot\|_{\mathcal{H}}$, one is interested in studying the *minimum-norm interpolator*, namely the hypothesis satisfying

$$\hat{\mathbf{f}}_{\boldsymbol{\theta}} = \underset{\substack{\mathbf{f}_{\boldsymbol{\theta}} \in \mathcal{H}: \\ \hat{\mathcal{R}}(\mathbf{f}_{\boldsymbol{\theta}}) = \mathbf{0}}}{\operatorname{argmin}} \|\mathbf{f}_{\boldsymbol{\theta}}\|_{\mathcal{H}} \tag{1.11}$$

Hence, overparameterized learning is cast as a problem of *joint interpolation and regularization*. As discussed in the previous section, regularization may come from explicit mechanisms, or implicitly. For instance, gradient descent recovers the minimum ℓ_2 norm interpolator on convex losses when the parameter is initialized at zero [128]. Similarly, the normal equations of linear regression provide the minimum ℓ_2 norm interpolator, and gradient descent recovers the normal equations in overparameterized linear regression [129]. In general, a priori from *how* a solution to Equation 1.11 is found, the generalization ability of the minimum norm interpolator depends on the structure of the data, the (learned) features and the loss function [44], [130]. A central concept in understanding the behaviour of minimum norm interpolators is the *double descent* phenomenon [131], introduced below.

Double Descent

To account for the generalization ability of large overparameterized models, Belkin, Hsu, Ma, *et al.* [131] propose the *double descent* curve of the test error, extending the study of complexity beyond the bias-variance trade-off of classical models. If the number of model parameters is interpreted as an intuitive complexity measure, when model complexity increases the test error is observed to follow the classical bias/variance trade-off curve, peaking when the resulting model is complex enough to perfectly fit the training data. The smallest such model is known as the *interpolation threshold* along the model-scale axis. As model complexity increases further, the test error decreases a second time, extending classical analysis to the *overparameterized regime*, whereby models operate by interpolating the training data. Double descent, also called *benign* or *harmless overfitting*, has been widely studied for overparameterized linear regression, both in the finite [132]–[134] and asymptotic [135] regime as well as in the kernel setting [44]. The phenomenon has also been consistently reproduced experimentally for deep learning on natural data [136]. However, a full theoretical characterization for finite-width networks is still missing. In the following, the formalism of linear regression is used to frame the research questions explored for deep networks in the thesis.

Insights from Overparameterized Linear Regression

The overparameterized linear regression problem asks to recover the unknown parameter $\theta^* \in \mathbb{R}^d$ given n d-dimensional i.i.d. observations $X \in \mathbb{R}^{n \times d}$ as well as noisy targets $\mathbf{y} = X\theta^* + \boldsymbol{\varepsilon} \in \mathbb{R}^n$, for $d \gg n$. The noise $\boldsymbol{\varepsilon}$ is assumed to originate from an isotropic normal distribution $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, I_n\sigma^2)$ independent from X, with variance σ^2. The minimum ℓ_2 norm interpolator solving the search problem

$$\hat{\theta} = \underset{\substack{\theta \in \mathbb{R}^d \\ \text{s.t. } X\theta = \mathbf{y}}}{\operatorname{argmin}} \|\theta\|_2 \tag{1.12}$$

is given in closed-form by the normal equations [124]

$$\hat{\theta} = X^T \big(XX^T\big)^{-1}\mathbf{y} \tag{1.13}$$

For unseen $(\mathbf{x}, y)$, the ideal interpolator θ^* attains minimum error, with excess risk

$$R(\hat{\theta}) = \mathbb{E}_{\mathbf{x}}\big[\big(\mathbf{x}^T(\theta^* - \hat{\theta})\big)^2\big] \tag{1.14}$$

Let $\Sigma = \mathbb{E}_{\mathbf{x}}[\mathbf{x}\mathbf{x}^T]$ denote the data covariance, assuming centered data. Bartlett, Long, Lugosi, *et al.* [133] decompose the excess into [133]

$$\mathbb{E}_{\boldsymbol{\varepsilon}}[R(\hat{\theta})] \geq \theta^{*T} B \theta^* + \sigma^2 \operatorname{tr}(V) \tag{1.15}$$

for

$$B = \big(I_d - X^T(XX^T)^{-1}X\big)\Sigma\big(I_d - X^T(XX^T)^{-1}X\big) \tag{1.16}$$

$$V = \big(XX^T\big)^{-1}X\Sigma X^T\big(XX^T\big)^{-1} \tag{1.17}$$

whereby both bias B and variance V depend on the data covariance Σ, which effectively controls harmless interpolation in the overparameterized regime. Specifically, the ability of interpolators to generalize to unseen data whilst interpolating noise $\boldsymbol{\varepsilon}$ is determined by the structure of the problem, governed by Σ, as well as by how signal and noise are encoded in the parameter $\hat{\theta}$. How Σ affects generalization in the interpolating regime has been object of active research [132], [133], [135], [137], [138]. In the random feature setting, if the data has independent sub-Gaussian rows [53], [132], as d increases, then the contribution of the variance term to the excess risk in Equation 1.15 scales as $\frac{n\sigma^2}{d}$, thereby reducing the impact of the noise variance σ^2. Finally, the contribution of the bias term – measuring the deviation of the solution $\hat{\theta}$ from the ideal interpolator θ^* – is controlled by the alignment of $\hat{\theta}$ with the covariance Σ eigenspaces. In fact, if the signal is encoded in the top eigendirections of $\hat{\theta}$, then overparameterization controls the bias by distributing the noise over the remaining eigendirections [132]. Moreover, if the data covariance has power-law eigenspectrum, the *ideal interpolator* can be manually constructed as the one encoding the signal in the top eigenspaces of $\hat{\theta}$, and the noise in the tail of the eigenspectrum [133], ensuring that both bias and variance decrease, whenever $d \gg n$. In fact, for infinite dimensional data, it can be shown that this form of interpolator guarantees optimal performance [138]. The present short overview of double descent in linear regression highlights a few important messages. First, *minimum norm* interpolation does not a priori guarantee benign overfitting, and the phenomenon depends on the degree of overparameterization (growth of d w.r.t. n). Second, harmless interpolation is determined by the structure of the learner, *i.e.* whether the minimum norm interpolator is able to separate signal from noise so that the latter is distributed over the tail of the eigenspectrum. Importantly, these observations have been reproduced also in the kernel setting [47], substantiating the idea that the structure of the learner matters. Therefore, in order to understand

generalization in deep learning in the feature rich setting, three components are of interest.

Research Question 1.1.2 (Open questions)**.**

1. *Formalize and study a data-dependent notion of complexity for non-linear models in relation to interpolation.*

2. *Understanding the mechanisms constraining said complexity for interpolating models.*

3. *Understanding how model size affects complexity, in relation to the mechanisms controlling it.*

The following section concludes the introduction by presenting the challenges associated with each of the above open questions, and how the present thesis contributes towards addressing them.

1.2 Challenges and Research Questions

Generalizing the above framework to deep networks in the non-asymptotic setting poses several challenges. Referring to Equation 1.12, the problem of minimum-norm interpolation for deep networks is formulated as

$$\hat{\mathbf{f}}_{\boldsymbol{\theta}} = \underset{\substack{\boldsymbol{\theta}\in\boldsymbol{\Theta} \text{ s.t.} \\ \mathcal{L}_{0/1}(\mathbf{f}_{\boldsymbol{\theta}},X,\mathbf{y})=0}}{\operatorname{argmin}} \|\mathbf{f}_{\boldsymbol{\theta}}\| \tag{1.18}$$

for some appropriate functional norm $\|\mathbf{f}_{\boldsymbol{\theta}}\|$, to be identified. The task involves several sub-questions, detailed below.

I. The hypothesis space is now a more general *functional space* $\mathcal{H} = \{\mathbf{f}_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^k \text{ for } \boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^p\}$, dependent on the model architecture and choice of activation function. Crucially, in contrast to linear regression, the input data dimensionality d remains fixed as model size p varies.

II. *Interpolation is now attained non-linearly*, and beyond the training data the learned model function is potentially affected by instability, analogously to the Runge phenomenon for polynomial fitting [139].

III. For an error function $\mathcal{L}(\mathbf{f}_{\boldsymbol{\theta}}, \mathbf{x}, y)$, the search for optimal parameters entails navigating a *non-convex loss landscape*, whereby curvature is governed by the Hessian $H = \mathbb{E}_{(\mathbf{x},y)} \frac{\partial^2}{\partial\boldsymbol{\theta}_i \partial\boldsymbol{\theta}_j} \mathcal{L}(\mathbf{f}_{\boldsymbol{\theta}}, \mathbf{x}, y)$, rather than the data covariance Σ (which now has fixed dimensionality).

IV. The learning problem depends on the loss landscape curvature as well as parameter initialization. Studying emerging regularization should account for the interaction between optimizer and model architecture.

V. Implicit regularization of the learned hypothesis $\mathbf{f}_{\boldsymbol{\theta}}$ should involve an appropriate *functional norm* $\|\mathbf{f}_{\boldsymbol{\theta}}\|$ on $\mathcal{H}$ (interpreted as a metric space), in turn entailing integration $\int_{\Omega} \|\mathbf{f}_{\boldsymbol{\theta}}\|$ over the unknown data distribution (or a subset of its support), with $\Omega \subset \mathbb{R}^d$ denoting the distribution support.

VI. Finally, as opposed to linear regression, a closed-form solution for the generalization error, as well as for the minimum norm interpolator, is a priori not available, requiring different strategies to attack the problem.

The present thesis studies the above research questions through the lens of smoothness of neural networks' model functions $\mathbf{f}_{\boldsymbol{\theta}}$ w.r.t. their inputs – a notion related to Lipschitz continuity, in turn representing bounded variation for continuous functions – for which there exists $K > 0$ such that

$$\|\mathbf{f}(\mathbf{x}_1) - \mathbf{f}(\mathbf{x}_2)\|_q \leq K\|\mathbf{x}_1 - \mathbf{x}_2\|_p \quad \forall\ \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d \tag{1.19}$$

where $\|\cdot\|_q$ and $\|\cdot\|_p$ respectively denote output-space and input-space norms. The main goal of the thesis is to *develop a notion of regularization generalizing the case of overparameterized linear regression to hypothesis spaces over non-linear functions, accounting for the model architecture, and tracking double descent.*
The notion of minimum norm interpolation also appears in theoretical studies of SSL, a wide family of unsupervised representation learning algorithms that circumvent the need for labelled data by learning model functions which attempt to maximize invariance to perturbations of the input. The invariance-based nature of SSL algorithms intimately connects them to metric properties of the data, connecting invariance-based learning objectives to local smoothness of the model function. Building on this connection, the thesis concludes by investigating implications of its findings on supervised learning to the SSL setting, in relation to robustness of learned representations.

1.3 Contributions of the Thesis

The following presents the thesis contributions, in relation to the above open questions. Focusing on the hypothesis class of networks equipped with the Rectified Linear Unit (ReLU) activation function (Problem I), **Paper A** studies emerging regularity of hypotheses expressed in practice, in relation to estimates of expressivity of the hypothesis class. **Paper B-C** focus on the input-space geometry of interpolation (Problem II) in relation to double descent, by studying the loss landscape in **Paper B** and the output (logit) space of neural networks in **Paper C**. Exploring in more detail the hypothesis space (Problem I), **Paper C** discusses

the role of hierarchical representations in relation to interpolation of the training data. Furthermore, mechanism of implicit regularization are related to the curvature of parameter space (Problem III), as well as the interaction between gradient descent and the model architecture (Problem IV). Taken together **Papers A-C** suggest a notion of regularization that generalizes the problem of interpolation+regularization from linear regression to non-linear functional spaces (Problem V). Finally, **Paper D** adopts the metrics developed in **Paper C** to study the geometry of representations learned with SSL on vision tasks, in relation to representation robustness, highlighting that current SSL methods operate in a regime compatible with underparameterized supervised learning. **Paper E** extends the observations by studying Out-of-Distribution (OOD) generalization for SSL methods based on data-augmentation, observing that OOD robustness behaves in keeping with scaling laws for underparameterized supervised learning [38].

List of Included Papers

This thesis is based on the following papers:

A: **M. Gamba**, A. Chmielewski-Anders, J. Sullivan, H. Azizpour, M. Björman. *Are all Linear Regions Created Equal?* In 'International Conference on Artificial Intelligence and Statistics (AISTATS)' (pp. 6573-6590). PMLR. 2022.

B: **M. Gamba**, E. Englesson, M. Björkman, H. Azizpour. *Deep Double Descent via Smooth Interpolation*. In 'Transaction on Machine Learning Research'. 2023.

C: [1] **M. Gamba**, H. Azizpour, M. Björkman. *On the Lipschitz Constant of Deep Networks and Double Descent*. In '34th British Machine Vision Conference (BMVC)'. 2023.

D: **M. Gamba**, A. Ghosh, K. K. Agrawal, B. A. Richards, H. Azizpour, M. Björkman. *Different Faces of Model Scaling in Supervised and Self-Supervised Learning*. In 'ICLR Workshop on Bridging the Gap Between Theory and Practice'. 2024

E: **M. Gamba**, K. K. Agrawal, A. Ghosh, B. A. Richards, H. Azizpour, M. Björkman. *When Does Self-Supervised Pre-Training Yield Robust Representations?* Preprint. 2024

[1] The version included in the thesis corrects errata of the published paper.

The following is a list of additional papers contributed by the author, but not included in the thesis.

X-1: **M. Gamba**, H. Azizpour, M. Björkman. *Overparameterization Implicitly Regularizes Input-Space Smoothness* [2]. In 'NeurIPS Workshop on Interpolation Regularizers'. 2022.

X-2: **M. Gamba**, S. Carlsson, H. Azizpour, M. Björkman. *Hyperplane Arrangements of Trained ReLU ConvNets are Biased*. ArXiv preprint arXiv:2003.07797. 2020.

X-3: **M. Gamba**, H. Azizpour, S. Carlsson, M. Björkman. *On the Geometry of Rectifier Convolutional Neural Networks*. In 'Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops', 2019.

[2] This is a preliminary version of **Paper C**.

Chapter 2

Geometry of Interpolation in Deep Learning

The thesis' exploration of effective complexity of deep networks begins by studying the hypothesis space of models equipped with the ReLU activation function [140], $x \mapsto \varphi(x) = \max(0, x)$ for $x \in \mathbb{R}$ (problem I.).

2.1 Hypothesis Space of ReLU Networks

Feed-forward ReLU networks $\mathbf{f}_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^k$ compose l affine layers of the form

$$A^\ell(\mathbf{x}) = \theta^\ell \mathbf{x} + \mathbf{b}^\ell \tag{2.1}$$

with the element-wise application of ReLU, for $\mathbf{x} \in \mathbb{R}^{d_{\ell-1}}$, $\theta^\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$, $\mathbf{b}^\ell \in \mathbb{R}^{d_\ell}$, $\ell \in [l] =: 1, \ldots, l$; $d_0 := d$, and $d_l := k$. By hierarchically composing affine layers with the continuous piece-wise linear function φ, ReLU networks express continuous piece-wise affine functions of the input data $\mathbf{x} \in \Omega \subseteq \mathbb{R}^d$ known as *affine spline operators* [141]

$$\mathbf{f}^\ell(\mathbf{x}) = (A^\ell \circ \varphi \circ A^{\ell-1} \circ \ldots \circ \varphi \circ A^1)(\mathbf{x}) \tag{2.2}$$

with $\mathbf{f}^l(\mathbf{x}) = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ denoting the full network. For each layer $\ell = 1, \ldots, l$, the affine function A^ℓ defines a collection of d_ℓ hyperplanes $\{\mathbf{x} \in \mathbb{R}^{d_{\ell-1}} : \theta_i^\ell \mathbf{x} + \mathbf{b}_i^\ell = 0\}$ in its preactivation space. When ℓ layers are composed hierarchically, the hyperplanes defined by layer ℓ cut through those defined by earlier layers j [142], for $j = 1, \ldots, \ell - 1$. In the network's input space Ω, each hidden unit $\mathbf{f}_i^\ell$ defines a *bent hyperplane* [76] $\mathbf{h}_i^\ell = \{\mathbf{x} \in \Omega : \mathbf{f}_i^\ell(\mathbf{x}) = 0\}$ for $i \in [d_\ell]$, inducing an arrangement

$$\mathcal{C} = \Omega \setminus \bigcup_{\ell \in [l]} \bigcup_{i \in [d_\ell]} \mathbf{h}_i^\ell \tag{2.3}$$

The resulting configuration partitions the input domain Ω into disjoint convex polytopes $\mathcal{P} = \{P_\varepsilon\}_\varepsilon$ known as *activation regions* [142], [143], with $\mathcal{C} = \cup_\varepsilon P_\varepsilon$.

Importantly, on each polytope P_ε, the network $\mathbf{f}_{\boldsymbol{\theta}}$ computes a single affine function

$$A_{(\varepsilon)}(\mathbf{x}) = \theta_{(\varepsilon)}\mathbf{x} + \mathbf{b}_{(\varepsilon)} \tag{2.4}$$

so that

$$\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{\varepsilon} \left(\theta_{(\varepsilon)}\mathbf{x} + \mathbf{b}_{(\varepsilon)}\right) \mathbb{1}_{P_\varepsilon}(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega \tag{2.5}$$

whereby $\mathbb{1}_{P_\varepsilon}(\mathbf{x}) = [\mathbf{x} \in P_\varepsilon]$ denotes the indicator function. For each activation region P_ε, the corresponding affine function can be computed in closed form [141]

$$\begin{aligned} A_{(\varepsilon)}(\mathbf{x}) = \theta^l \left(\prod_{\ell=l-1}^{1} \theta^\ell D^\ell(\mathbf{x}) \right) \mathbf{x} \\ + \theta^l \sum_{\ell=1}^{l-1} \left(\prod_{j=l-1}^{\ell+1} \theta^j D^j(\mathbf{x}) \right) \left(D^\ell(\mathbf{x}) \mathbf{b}^\ell \right) + \mathbf{b}^l \end{aligned} \tag{2.6}$$

where $D^\ell(\mathbf{x})$ is a $d_\ell \times d_\ell$ diagonal matrix, with $D^\ell_{ii}(\mathbf{x}) = 1$ if $\mathbf{f}^\ell_i(\mathbf{x}) > 0$ and 0 otherwise. The concatenated patterns $(D^1_{11}(\mathbf{x}), \ldots, D^1_{d_1 d_1}(\mathbf{x}), \ldots, D^l_{d_l d_l}(\mathbf{x}))$ provide a unique signature identifying the activation region of $\mathbf{x}$, known as binary *activation pattern* [76], [144]. Clearly, points within the same activation region share the same activation pattern, and activation regions are indeed defined as the geometric locus of points sharing the same activation pattern.

Estimating the Partition $\mathcal{P}$

Finally, the present introduction to the geometry of ReLU networks concludes by discussing the problem of computing the partition $\mathcal{P}$. For compact sets $\Omega \subset \mathbb{R}^d$, corresponding to bounded input domains (*e.g.* RGB pixels), the activation region partition $\mathcal{P}$ can be computed analytically by tracking, for each layer $\ell \in [l]$ and for each hidden unit $i \in [d_\ell]$, how the affine function $\mathbf{f}^\ell_i$ cuts through existing activation regions defined by earlier layers [76], [145]. Alternatively, a mixed-integer programme can be defined and solved numerically, as proposed in [146] (Theorem 11). To build some intuition about the complexity of estimating $\mathcal{P}$ for m-dimensional Ω, one can rely on a classical result from Zaslavsky on standard hyperplane arrangements ($l = 1$) [147], for which, given n hyperplanes in general position in $\mathbb{R}^m$

$$|\mathcal{P}| = \sum_{j=0}^{m} \binom{n}{j} \tag{2.7}$$

Hence, the complexity of exact region counting algorithms (for single layer networks) roughly scales as a degree-m polynomial in the number of hyperplanes n, thus suffering from the curse of dimensionality for high dimensional Ω [148].

Expressivity of ReLU Networks

Affine spline operators parameterized by ReLU networks are endowed with universal approximation ability of continuous functions [149]–[151]. In function approximation problems, the ability of a ReLU network to represent functions of non-zero curvature is tied to its capacity of expressing many activation regions [91], [152]. Particularly, in classification tasks, non-linear expressivity allows to model complex decision boundaries, for data that is not linearly separable. Non-linear expressivity of a ReLU network can be theoretically quantified by estimating the number of activation regions that the architecture can express on the input domain. Several works propose upper [144], [146], [153], [154] as well as lower bounds [142], [143] on the number of regions expressed by different network architectures, interpreted as a measure of the theoretical expressive power of a model. Importantly, this avenue of research allows to understand the theoretical benefits of parameterizing a class of functions $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^k$ of fixed domain/codomain by using deeper networks, with exponential gains in expressive power compared to shallower models [91], [152]. Subsequent works shift the focus from theoretical expressivity of a deep architecture to exploiting the number of activation regions in order to estimate complexity of networks trained in practice. Importantly, for ReLU networks at random initialization [7], Hanin and Rolnick [76] theoretically show that the density of activation regions on compact domains $\Omega \subset \mathbb{R}^d$ is bounded in expectation by a factor that depends polynomially on the number of neurons $z = \sum_{\ell=1}^{l} d_\ell$, and exponentially on the input dimensionality d, whenever $z \geq d$, in keeping with Zaslavsky's theorem for standard hyperplane arrangements [147]. Intriguingly, the bound does not depend on the network architecture, but only on the total number of neurons.

2.2 Effective Complexity of Interpolating Affine Splines

Studies discussed thus far treat activation regions expressed by a fixed architecture uniformly, implicitly assuming that partitions $\mathcal{P}$ of $\mathbb{R}^d$ counting higher region density are associated with a model function $\mathbf{f}_{\boldsymbol{\theta}}$ of higher non-linearity. However, as intuitively depicted in Figure 2.1, for affine splines of non-vanishing curvature, effective model non-linearity also depends on the functions $A_\varepsilon(\mathbf{x})$ expressed by each affine component. In line with this observation, **Paper A** studies effective complexity of deep ReLU networks trained in practice, by contrasting the density of activation regions to the effective non-linearity of the corresponding affine spline.

Research Question 2.2.1. *Is the density of activation regions a reliable estimator of model non-linearity?*

Specifically, estimating effective non-linearity is decomposed into two tasks:

Task 1. Exactly computing the partition $\mathcal{P}$ on a compact subdomain of Ω.

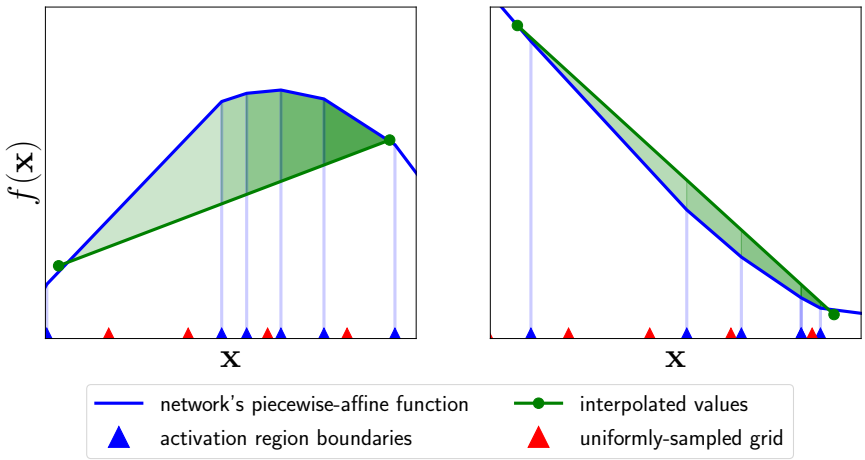


Figure 2.1: For affine spline operators of non-vanishing curvature (blue line) with corresponding activation regions (projections onto the x-axis, with boundaries marked by blue triangles), **the same number of activation regions may express functions of different local non-linearity** (green area).

Task 2. Measuring effective model non-linearity by studying the affine components $A_{(\varepsilon)}(\mathbf{x})$ on $\mathcal{P}$. Again with reference to Figure 2.1, the non-linearity measure should account for the slope $\theta_{(\varepsilon)}$ and the offset $\mathbf{b}_{(\varepsilon)}$ of each affine component, as well as the volume over which the function $A_{(\varepsilon)}$ is defined.

Additionally, the contribution of certain activation regions to a model's non-linearity may be negligible. In fact, large networks can be successfully distilled into smaller models [155], and a large fraction of model parameters can be pruned without affecting model performance [156]. This observation motivates the following question.

Research Question 2.2.2. *In the interpolating regime, the assumption that each activation region equally contributes to a model's effective non-linearity may not hold. How does model non-linearity scale vis-à-vis overparameterization?*

Indeed, while a higher number of activation regions is indicative of a model expressing a relatively more fine-grained partition of the input domain, neighbouring activation regions may approximately express the same affine function [157], thereby showing reduced local non-linearity. Such redundancy is expected to be exacerbated in the overparameterized regime, which is the focus of **Paper A**.

Activation Region Discovery

The first step towards precisely quantifying non-linearity of a ReLU network trained in practice is computing the partition $\mathcal{P}$. As seen in the previous section, estimating the partition $\mathcal{P}$ of Ω is computationally intractable for large networks and high-dimensional input spaces. Several works thus restrict the computation to bounded one-dimensional [76], [99], [144], [146] or two-dimensional domains [76], [145], [158]. Formally, if $\Gamma \subset \Omega$ is a low-dimensional sub-domain of the input space, the problem is restricted to measuring non-linearity over $\mathcal{B} = \{P_\varepsilon \cap \Gamma : P_\varepsilon \in \mathcal{P}\}$. Particularly, if Γ is convex, then by convexity of each $\mathcal{P}_\varepsilon$, $\mathcal{B}$ is itself an activation region partition. Finally, a numerical alternative to exactly computing $\mathcal{B}$ can be found in generating a grid of equally-spaced input points (e.g. along a trajectory or plane in $\mathbb{R}^d$), and counting the number of unique activation patterns observed [99]. This method, depicted by the red triangles in Figure 2.1 suffers from two main drawbacks, namely the need to know a priori an upper bound on the number of activation regions one expects to encounter (in order to select how fine-grained the sampling is), whilst at the same time incurring the risk of undersampling regions smaller than the selected precision. Particularly, grid-based sampling does not allow to precisely estimate the activation region boundaries within $\mathcal{B}$, thus not fulfilling Task 1.

Contribution: Activation Region Discovery Algorithm

In order to scale computation to ReLU networks used in practice, **Paper A** considers one-dimensional compact domains of $\mathbb{R}^d$, parameterized by piece-wise linear trajectories. The first contribution is an activation region discovery algorithm, allowing to compute the partition $\mathcal{B}$ along a specified direction $\mathbf{d}$, based at a point $\mathbf{x}_0$. In a nutshell, given a starting point $\mathbf{x}_0$ belonging to activation region B_0, and an endpoint $\mathbf{x}_n \in B_n$, the algorithm computes the displacement λ_0 to cross the closest boundary of B_0 along the direction vector $\mathbf{d} = \mathbf{x}_n - \mathbf{x}_0 \in \mathbb{R}^d$. The procedure is iterated until the region B_n containing point $\mathbf{x}_n$ is reached.

Preconditions By recalling that each layer $\ell = 1, \dots, l$ defines d_ℓ hyperplanes $\{\mathbf{x} \in \mathbb{R}^{d_{\ell-1}} : \theta_i^\ell \mathbf{x} + \mathbf{b}_i^\ell = 0\}$ in its preactivation space, let $\mathbf{x}^\ell = \varphi(\theta^\ell \mathbf{x}^{\ell-1} + \mathbf{b}^\ell)$ denote the post-activation of the ℓ-th layer, with $\mathbf{x}^0 := \mathbf{x} \in \mathbb{R}^d$. For $\mathbf{x}_t \in \mathbb{R}^d$, let $\mathbf{x}_t^\ell = \mathbf{x}^\ell(\mathbf{x}_t)$ denote the image of $\mathbf{x}_t$ in $\mathbb{R}^{d_\ell}$, for $t = 0, \dots, n$. Then, each hyperplane $i \in [d_\ell]$, defines a positive halfspace $\theta_i^\ell \mathbf{x}^{\ell-1} + \mathbf{b}_i^\ell > 0$, as well as a negative one. To denote this, let $\text{sign}(\theta_i^\ell \mathbf{x}_t^{\ell-1} + \mathbf{b}_i^\ell) = 1$ if $\mathbf{x}_t^{\ell-1}$ lies in the positive halfspace of the i-th hyperplane at layer ℓ, and $\text{sign}(\theta_i^\ell \mathbf{x}_t^{\ell-1} + \mathbf{b}_i^\ell) = 0$ if $\mathbf{x}_t^{\ell-1}$ lies in the negative halfspace. The case for which $\mathbf{x}_t^{\ell-1}$ lies on the hyperplane is discussed separately. Finally, let $\mathbf{d}_t^{\ell-1} = \prod_{j=1}^{\ell-1} D^j(\mathbf{x}_t)\theta^j \mathbf{d}$ denote the direction $\mathbf{d}$ projected onto the preactivation space $\mathbb{R}^{d_{\ell-1}}$, with $D^j(\mathbf{x}_t)$ defined according to Equation 2.6.

Iteration At iteration t, the smallest displacement λ_t to cross the boundary of B_t is computed by solving the linear problem defined in Equation 2.10, for each

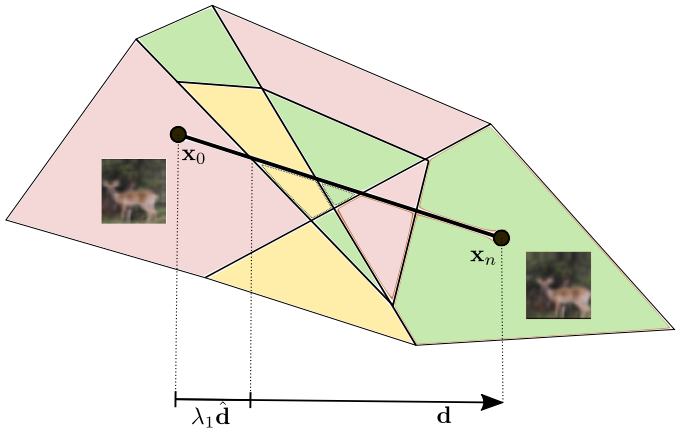


Figure 2.2: Iteration of the activation region discovery algorithm along a direction vector connecting two input data points.

layer $\ell = 1, \ldots, l$, and for each hyperplane $i = 1, \ldots d_\ell$.

$$z = \text{sign}\left(\theta_i^\ell \mathbf{x}_t^{\ell-1} + \mathbf{b}_i^\ell\right) \in \{0, 1\} \tag{2.8}$$

$$\lambda_i^\ell = \underset{\lambda \in \mathbb{R} \cup \{\pm\infty\}}{\text{argmin}} \; (-1)^z \left(\theta_i^\ell (\mathbf{x}_t^{\ell-1} + \lambda \mathbf{d}_t^{\ell-1}) + \mathbf{b}_i^\ell\right) \geq 0 \tag{2.9}$$

$$= \underset{\lambda \in \mathbb{R} \cup \{\pm\infty\}}{\text{argmin}} \; (-1)^z \lambda \geq (-1)^{z+1} \frac{\theta_i^\ell \mathbf{x}_t^{\ell-1} + \mathbf{b}_i^\ell}{\theta_i^\ell \mathbf{d}^{\ell-1}} \tag{2.10}$$

If $\theta_i^\ell \mathbf{x}_t^{\ell-1} + \mathbf{b}_i^\ell = 0$, then $\mathbf{x}_t^{\ell-1}$ lies on the i-th hyperplane of layer ℓ and therefore moving along $\mathbf{d}$ requires computing the shortest distance to all other hyperplanes. Hence, all $\lambda_t^\ell = 0$ are discarded by the algorithm. Finally, the minimum non-zero displacement λ_t is selected, and a step is taken in the direction $\mathbf{d}$

$$\lambda_t = \min_{\substack{\ell \in [l] \\ i \in [d_\ell]}} \{\lambda_i^\ell : |\lambda_i^\ell| \geq \tau\} \tag{2.11}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \lambda_t \mathbf{d} \tag{2.12}$$

The procedure halts whenever the endpoint $\mathbf{x}_n$ is reached.

Numerical considerations In order to execute the algorithm with finite precision (e.g. 64-bit) a sensitivity threshold $0 < \tau \ll 1$ is defined to ensure that an activation region boundary is always crossed at each iteration t. This is reflected in Equation 2.11. Figure 2.2 shows one iteration of the algorithm, for a straight line in input space connecting two samples of the CIFAR-10 dataset [159]. Importantly, the algorithm can be efficiently computed with a single forward pass through the

network for each iteration t, and by solving the linear problem 2.10 iteratively for each layer. The computational complexity $\mathcal{O}(ln)$ scales linearly with the number of regions discovered and the number of layers l. Importantly, the linear programme can be easily batched to process an entire layer at a time, as well as multiple inputs $\mathbf{x}_0$ and directions $\mathbf{d}$, making it possible to study large convolutional networks such as ResNet [7] and VGG [160] on high dimensional input datasets. Indeed, at the time of publication, **Paper A** presented the first large-scale study of activation regions for ResNets and large ConvNets.

Implications If the direction $\mathbf{d}$ is parameterized by a line path $\boldsymbol{\gamma} : \mathcal{I} = [0, 1] \to \mathbb{R}^d$, with $\boldsymbol{\gamma}(s) = \mathbf{x}_0 + s\mathbf{d}$, for $0 \leq s \leq 1$ then, by convexity of activation regions, $\mathcal{P}$ induces a partition $\mathcal{B}_{\mathcal{I}} = \{s_t \in \mathcal{I} : 0 = s_0 < \ldots < s_n = 1\}$ of $\mathcal{I}$, dividing $\mathcal{I}$ into intervals $\mathcal{I}_t = [s_t, s_{t+1}]$, with $|\mathcal{I}_t| = \lambda_t$ corresponding to the normalized length of region B_t along $\mathbf{d}$. Importantly, this allows to compute and cheaply store the entry and exit points of $\boldsymbol{\gamma}$ into each activation region along $\mathbf{d}$. In the following, the quantities are used to introduce a simple measure of non-linearity of a network.

Contribution: Effective Non-linearity Measure

A second contribution of **Paper A** is a non-linearity measure tailored for interpolating affine spline operators. For high-dimensional input spaces, the measure implicitly relies on the manifold hypothesis [161], [162].

Definition 2.2.3 (Manifold Hypothesis)**.** *Natural data lies in proximity of low-dimensional manifolds embedded in the Euclidean input space.*

Intuition Let $\mathbf{f}_{\boldsymbol{\theta}}$ denote a ReLU network trained to perfectly fit a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, so that the train error $\mathbb{E}_{\mathcal{D}}\mathcal{L}_{0/1}(\boldsymbol{\theta}, \mathbf{x}, y) = 0$. For a training point $\mathbf{x}_i \in \mathcal{D}$, let $\tilde{\mathbf{x}}_i$ denote a second data point, lying in proximity of $\mathbf{x}_i$. Then, according to the manifold hypothesis, if $\mathbf{d} = \tilde{\mathbf{x}}_i - \mathbf{x}_i$ with small enough $\|\mathbf{d}\|_2$, then $\mathbf{d}$ is tangential to the data manifold at $\mathbf{x}_i$. Then, interpolation of $\tilde{\mathbf{x}}_i$ can be attained by the affine function $A(\mathbf{x}) = \theta\mathbf{x}+\mathbf{b}$ expressed by $\mathbf{f}_{\boldsymbol{\theta}}$ at $\mathbf{x}_i$, so that $\mathbf{f}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_i) = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i+\mathbf{d}) = \theta(\mathbf{x}_i + \mathbf{d}) + \mathbf{b}$. Hence, the minimum norm affine spline interpolator should express a function of vanishing curvature between $\mathbf{x}_i$ and $\tilde{\mathbf{x}}_i$. Conversely, for a network $\mathbf{f}_{\boldsymbol{\theta}}$ interpolating a dataset $\mathcal{D}$, by using perturbed points $\tilde{\mathbf{x}}_i$ it is possible to probe the network locally to each $\mathbf{x}_i$ to measure how far it deviates from an interpolator with locally vanishing curvature. Figure 2.3 presents an illustration of the method: for each pair of values $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{f}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_i) \in \mathbb{R}^k$ (denoted by green dots in the two top panels), the non-linearity of the model between the corresponding points $\mathbf{x}_i, \tilde{\mathbf{x}}_i$ can be estimated by measuring the model's deviation (shaded green area) from a locally flat interpolator (green line), defined in proximity of the training data $\mathbf{x}_i \in \mathcal{D}$.

Non-linearity measure Let $A_t(\mathbf{x}) = \theta_t\mathbf{x} + \mathbf{b}_t$ denote the affine function expressed by $\mathbf{f}_{\boldsymbol{\theta}}$ on the activation region containing point $\mathbf{x}_t$. For two endpoints $\mathbf{x}_0 \in \mathcal{D}$

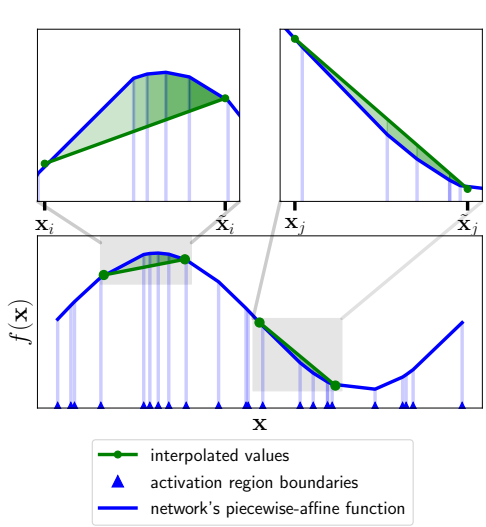


Figure 2.3: Proposed **non-linearity measure**, capturing local curvature by computing the distance from a local affine spline interpolator of zero curvature.

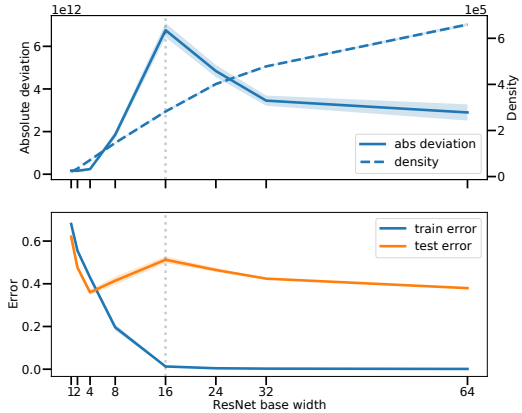


Figure 2.4: For ResNets of increasing width (x-axis) with test error undergoing double descent (bottom panel), activation region density (top panel) monotonically increases with model size, whereas **the proposed non-linearity measure mirrors the test error,** peaking when the models become large enough to interpolate the training data.

and $\tilde{\mathbf{x}}_0 \in \mathbb{R}^d$, let $\boldsymbol{\gamma} : \mathcal{I} \to \mathbb{R}^d$ be the line path parameterized by $\boldsymbol{\gamma}(s) = \mathbf{x}_0 + s\mathbf{d}$, for $\mathbf{d} = \tilde{\mathbf{x}}_0 - \mathbf{x}_0$ and $0 \leq s \leq 1$. Finally, let $A_r(\mathbf{x}) = \theta_r \mathbf{x} + \mathbf{b}_r$ be the affine function on the activation region of $\tilde{\mathbf{x}}_0$ and define the function $\mathbf{a}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^k$ s.t.

$$\mathbf{a}(\boldsymbol{\gamma}(s)) := (1-s)(\boldsymbol{\theta}_0 \boldsymbol{\gamma}(s) + \boldsymbol{b}_0) + s(\boldsymbol{\theta}_r \boldsymbol{\gamma}(s) + \boldsymbol{b}_r) \tag{2.13}$$

Definition 2.2.4 (Absolute deviation along path)**.** *Using Equation 2.13, local non-linearity of an interpolating affine spline* $\mathbf{f}_{\boldsymbol{\theta}}$ *along* $\boldsymbol{\gamma}$ *can be measured by*

$$\int_{\boldsymbol{\gamma}} \|\mathbf{f}_{\boldsymbol{\theta}} - \mathbf{a}\| \tag{2.14}$$

Using the proposed activation region discovery algorithm, absolute deviation along $\boldsymbol{\gamma}$ can be efficiently computed as

$$\sum_{t=1}^{r} \int_{s_{t-1}}^{s_t} \|\boldsymbol{\theta}_t \boldsymbol{\gamma}(s) + \mathbf{b}_t - \mathbf{a}(\boldsymbol{\gamma}(s))\| \|\dot{\boldsymbol{\gamma}}(s)\| ds \tag{2.15}$$

It is worth observing that, since $s_t - s_{t-1} = \lambda_t$, absolute deviation takes into account the length of activation regions A_t along $\mathbf{d}$. Furthermore, in contrast to previous

studies of function variation [99], the proposed measure takes into account non-linearity arising from the bias parameters, as well as variation of the visited affine functions, fulfilling Task 2. Indeed, ReLU networks with fixed random weight tensors and learned biases are universal approximators [163]. For each training point $\mathbf{x}_i$, the measure can be trivially extended to compute non-linearity along piece-wise linear paths $\boldsymbol{\gamma}_i = \cup_{j=1}^{p} \boldsymbol{\gamma}_i^j$ connecting sequences of points $\mathbf{x}_i =: \tilde{\mathbf{x}}_i^0, \tilde{\mathbf{x}}_i^1, \ldots, \tilde{\mathbf{x}}_i^p$. For each training point $\mathbf{x}_i$ and piece-wise linear path $\boldsymbol{\gamma}_i$, a single scalar non-linearity measure can be aggregated by computing the expected absolute deviation

$$\mathbb{E}_{\mathbf{x}_i \sim \mathcal{D}} \frac{1}{\text{len}(\boldsymbol{\gamma}_i)} \int_{\boldsymbol{\gamma}_i} \|\mathbf{f}_{\boldsymbol{\theta}} - \mathbf{a}\| \tag{2.16}$$

Disentangling Non-linearity from Region Density

The proposed measure and activation region discovery algorithm allow to disentangle effective non-linearity from the underlying partition $\mathcal{P}$ in proximity of the training data, for networks trained in practice. Importantly, this enables the study of the effect of overparameterization on spline affine operators expressed by ReLU networks operating in the interpolating regime. In a series of experiments, ConvNets and ResNets are trained to interpolate tasks of increasing complexity, whereby the network would benefit from expressing a more fine-gradient activation region partition. Importantly, since the models considered are trained until zero training error is reached, underfitting can be ruled out as a confounding factor whenever reduced non-linearity is observed. Key experimental observations are summarized below.

- For models trained to interpolate training data with a fraction of the labels randomly perturbed (from 20% to 100% perturbed samples) [78] absolute deviation is able to distinguish affine operators interpolating increasingly noisy data, while activation region density provides an unreliable predictor of effective non-linearity. Similarly, absolute deviation is able to distinguish model functions *smoothed* with training-time data augmentation from those trained without data augmentation, while activation region density largely fails.

- Activation region density poorly correlates with absolute deviation, highlighting that effective model non-linearity is poorly explained by the underlying activation regions partition in the overparameterized setting. Importantly, while prior work identifies reduced effective model complexity with expected activation region density being lower than what suggested by theoretical expressivity bounds [76], **Paper A** observes only weak correlation between activation region density and effective non-linearity.

- The proposed non-linearity measure undergoes double descent when model size is controlled (Figure 2.4). Crucially, while activation region density monotonically increases with model size, effective non-linearity is reduced in the

overparameterized regime, undergoing a phase transition near the interpolation threshold. Hence, while the density of activation regions increases for larger models, neighbouring activation regions approximately express the same affine function along data-driven trajectories.

- Finally, taken together, the above observations provide empirical evidence in support of the VC-dimension not being suited to study deep networks. By recalling that, for a hypothesis class expressing (standard) hyperplane arrangements, the VC-dimension scales like the number of cells in the arrangement (in turn given by Zaslavsky's theorem, c.f.r. Equation 2.7), and by recalling that Hanin and Rolnick [76] provide depth-independent bounds on the expected activation region density matching Zaslavsky's theorem, then Figure 2.4 suggests that the VC-dimension of the network monotonically increases with model size, thereby failing to capture generalization.

The present fine-grained study of affine spline operators parameterized by ReLU networks, vis-à-vis their implicit activation region partition, highlights that reduced effective complexity in the overparameterized regime is associated with increased local redundancy of activation regions, corresponding to reduced effective non-linearity. The observation emphasizes the importance of variation-based measures in understanding emerging regularity in the overparameterized regime. Referring again to Figure 2.3, the proposed measure of non-linearity implicitly captures variation of ReLU networks from a locally flat interpolator across activation regions. However, the measure does not distinguish non-linearity arising from small (but frequent) oscillations of the model function from non-linearity due to high local curvature. **Paper B** addresses this shortcoming by separately studying curvature and function variation in input space, in relation to interpolation of noisy data.

2.3 Loss Landscape Geometry and Interpolation

The ability of deep networks to perfectly fit noisy training data while retaining generalization on i.i.d. test data suggests that generalizing models fit noisy samples whilst correctly predicting the ground truth when extrapolating away from those points. In other words, a generalizing model is able to deviate from the ground truth function *locally* to a noisy training point, without affecting the model's predictions outside of a neighbourhood of the noisy data. This observation is intuitively depicted for a *regularized* interpolating polynomial in Figure 2.5a. To characterize this behaviour for deep networks, **Paper B** is concerned with studying the geometry of interpolation in the presence of noisy training targets (problem II.).

Geometry of Interpolation

To study interpolation for deep networks, this section draws inspiration from the function approximation literature, identifying useful tools to analyze interpolation

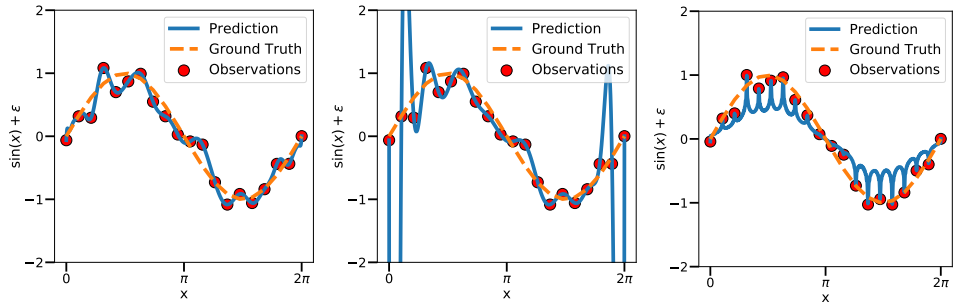


Figure 2.5: Intuition from polynomial interpolation. a) A *regularized* polynomial of high degree interpolating noisy training data without considerably deviating from the underlying ground truth signal. b) The Runge phenomenon, observed for unregularized polynomials interpolating uniformly-spaced data. c) High-degree interpolating polynomial exhibiting sharp variation in a neighbourhood of each interpolated point. **Paper B** argues that overparameterization implicitly controls the geometry of interpolation. While models near the interpolation threshold behave consistently with the polynomial case b), large networks *smoothly* interpolate both clean and noisy data, and thus improved generalization in the interpolating regime is tied to smoothness of the loss w.r.t. the input variable.

geometry. A rich class of non-linear interpolators is provided by polynomials, by virtue of the Stone-Weierstrass theorem [164].

Theorem 2.3.1 (Stone-Weierstrass approximation theorem [164])**.** *For a continuous function $f : [a, b] \to \mathbb{R}$, for every $\varepsilon > 0$, $\exists$ a polynomial p_n of degree n s.t.*

$$\sup_{x \in [a,b]} |f(x) - p_n(x)| < \varepsilon \tag{2.17}$$

For a set of interpolation points $\{(x_i, f(x_i))\}_{i=1}^{m+1}$, a polynomial degree n and a polynomial basis are typically chosen, allowing to construct the interpolating polynomial minimizing a particular notion of error. An important consequence of the choice of basis is the *Runge phenomenon* [139], depicted in Figure 2.5b, where a m-degree polynomial interpolating $m + 1$ equidistant points oscillates near the endpoints of the interval. Another interesting case is noted in Figure 2.5c, whereupon a high-degree polynomial interpolates the noisy data while showing high-norm gradients, thereby attaining a poor fit outside of the interpolation points. As in the general case, the choice of norm imposes a prior on the class of interpolators, restricting the solution space. If the ground-truth function admits a k-th order derivative, then the best approximation error is controlled by $|f(x) - p_n(x)| \leq \frac{\pi}{2} \frac{1}{(n+1)^k} |f^{(k)}|$. Then, to control the gradients and attain good extrapolation, a common solution is to in-

crease the polynomial degree and impose smoothness constraints of the form [165]

$$\frac{1}{2m}\sum_{i=1}^{m}(f(x_i)-p_n(x_i))^2+\lambda\int_a^b|p_n^{(k)}(x)|dx \tag{2.18}$$

Similarly, spline interpolators $a(x)$ admit a smoothing spline formulation, defined via *roughness* (curvature) penalization [166]

$$\frac{1}{2m}\sum_{i=1}^{m}(f(x_i)-a(x_i))^2+\lambda\int_a^b f''(x)^2dx \tag{2.19}$$

typically imposed via Lagrangian multipliers $\lambda\in\mathbb{R}$. Figure 2.5a shows an example of a high-degree polynomial fitted with smoothness regularization. Building on the polynomial intuition, a natural question is thus how interpolation is attained by overparameterized deep networks. Possible scenarios include models characterized by high sensitivity (Figure 2.5c), unstable functions analogous to the Runge phenomenon (Figure 2.5b) or smooth interpolators akin to regularized polynomials (Figure 2.5a). To investigate the phenomenon, **Paper B** presents a study of the input-space geometry of the loss landscape for models interpolating noisy datasets. Importantly, the following question is explored.

Research Question 2.3.2. *What is the effect of overparameterization on the loss-landscape geometry of interpolating models? How do interpolating models behave locally to interpolated samples with clean and corrupted targets, respectively?*

Measuring Variation

For a network $\mathbf{f}_{\boldsymbol{\theta}}$ of trained parameter $\boldsymbol{\theta}$, geometry of interpolation is studied by exploring the input-space landscape of the loss $\mathcal{L}(\mathbf{x},y):=\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x},y)$ for fixed $\boldsymbol{\theta}$. Local smoothness of the loss landscape is measured by computing the Jacobian norm

$$J(\mathbf{x},y)=\|\mathbf{J}(\mathbf{x},y)\|_F^2:=\|\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x},y)\|_F^2 \tag{2.20}$$

measuring sensitivity of the loss to infinitesimal perturbations local to $\mathbf{x}$. To capture local curvature at a point, the Hessian norm is computed via the functional

$$H(\mathbf{x},y)=\left\|\frac{\partial^2}{\partial\mathbf{x}\partial\mathbf{x}^T}\mathcal{L}(\mathbf{x},y)\right\|_F^2 \tag{2.21}$$

Again appealing to the manifold hypothesis (Definition 2.2.3), Hessian estimation is restricted to curvature directions near the data manifold. Inspired by the Hessian eigenmaps embedding method [167] as well as the rugosity estimator [168], computation of the Hessian norm at each point $(\mathbf{x}_i,y_i)$ is restricted to directions tangential to the data manifold at $\mathbf{x}_i$. Formally, for any point $(\mathbf{x}_i,y_i)$ with corresponding Jacobian $\mathbf{J}(\mathbf{x}_i,y_i)$, m neighbours $\mathbf{x}_i+\delta\mathbf{u}_j$ are generated by randomly

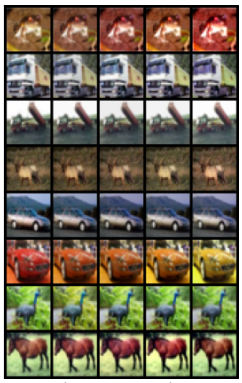


Figure 2.6: Random colour augmentations used to estimate the tangent Hessian norm. Each row represents a set of augmentations, with the first image showing the corresponding base sample.

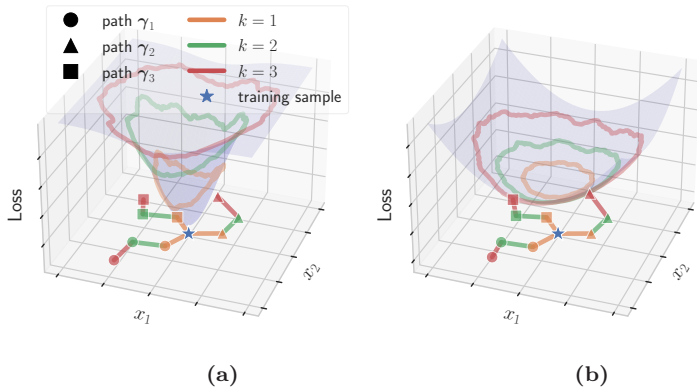


Figure 2.7: Geodesic MC integration. For each base training point, p paths approximating on-manifold geodesics are formed by connecting a sequence of augmentations of increasing strength, covering volumes of increasing size in the loss landscape around each training point. The goal is to compare points that are a) sharply interpolated from those that are b) smoothly interpolated. For a base training point (blue star), each path $\boldsymbol{\gamma}$ joins different augmentations of the base sample for augmentation strength $k = 1, 2, 3$, controlling the path length.

sampling a displacement vector $\mathbf{u}_j$ using weak data augmentation, for $j \in [m]$. For each sampled $\mathbf{u}_j$, the tangent Hessian $\mathbf{H}(\mathbf{x}_i, y_i)$ projected along the direction $\mathbf{x}_i + \mathbf{u}_j$, is estimated via the finite difference $\frac{1}{\delta}\mathbf{J}(\mathbf{x}_i, y_i) - \frac{1}{\delta}\mathbf{J}(\mathbf{x}_i + \delta\mathbf{u}_j, y_i)$.

$$H(\mathbf{x}_i, y_i) = \frac{1}{m\delta^2} \sum_{j=1}^{m} \left\| \mathbf{J}(\mathbf{x}_i, y_i) - \mathbf{J}(\mathbf{x}_i + \delta\mathbf{u}_j, y_i) \right\|_F^2 \tag{2.22}$$

Weak colour transformations are employed to sample neighbouring points $\mathbf{x}_i + \mathbf{u}_j$, since (weak) photometric transformations are guaranteed to be fully on-manifold. A visualization of the colour transformations is presented in Figure 2.6.

Contribution: Geodesic Monte Carlo Estimator

To characterize the loss landscape geometry, the infinitesimal measures of Equations 2.20 and 2.22 are extended to volumetric measures centered at each interpolated point $\mathbf{x}_i$, with the goal of capturing smoothness and curvature when traveling away from the point. The procedure leverages Monte Carlo (MC) integration and involves two steps, namely integrating the Jacobian and tangent Hessian norms over geodesic paths, and averaging the per-path measures to obtain a volume estimate. Figure 2.7 illustrates the method, which allows to distinguish between a *sharp loss*

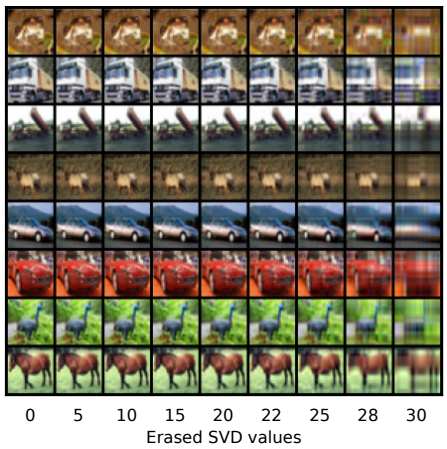
landscape at each point (Figure 2.7a) – characterized by high local curvature and high loss variation – and a *wide loss landscape*, with lower curvature (Figure 2.7b).

Generating Paths Let $\mathcal{T}_{\mathbf{s}} = \{T_{\mathbf{s}} : \mathbb{R}^d \to \mathbb{R}^d\}$, represent a family of smooth transformations (data augmentation) acting on the input space and governed by parameter $\mathbf{s}$, controlling the strength $s = \|\mathbf{s}\|_2$ as well as the direction $\frac{\mathbf{s}}{\|\mathbf{s}\|_2}$ of the augmentation in $\mathbb{R}^d$. In general, the parameter $\mathbf{s}$, interpreted as a suitably distributed random variable, models the randomness of the transformation. Randomly sampling $\mathbf{s}$ yields a value $\mathbf{s}_k$ corresponding to a fixed transformation $T_{\mathbf{s}_k}$ of strength s_k. For instance, for affine translations, $\mathbf{s}_k$ represents a random radial direction sampled from a hypersphere $\mathbb{S}^{d-1}$ centered at $\mathbf{x}_i$, with strength s_k denoting the magnitude of the translation (*e.g.* 4-pixel shift). For photometric transformations, $\mathbf{s}_k$ may model the change in brightness, contrast, hue, and saturation, with total strength s_k. The generation of p on-manifold paths $\boldsymbol{\gamma}_i^j$ emanating from $\mathbf{x}_i$, for $j \in [p]$, proceeds as follows. First, a sequence of $m+1$ strengths $s_0 < s_1 < \ldots < s_m$ is selected, with $s_0 = 0$ denoting the identity transformation. Then, a sequence of augmentations $\mathbf{x}_i^0 \prec \mathbf{x}_i^1 \prec \ldots \prec \mathbf{x}_i^m$ is joined into a piece-wise linear path $\boldsymbol{\gamma}_i^j : [0,1] \to \mathbb{R}^d$, with $\mathbf{x}_i^k = T_{\mathbf{s}_k}(\mathbf{x})$, using transformations $T_{\mathbf{s}_k}$ of increasing strength $s_0, \ldots, s_m$. Each path $\boldsymbol{\gamma}_i^j$ approximates an on-manifold trajectory by a sequence of Euclidean segments $\mathbf{x}_i^k \mathbf{x}_i^{k-1}$, for $k = [m]$. The maximum augmentation strength s_m controls the distance traveled from $\mathbf{x}_i$, while the number m of strengths used controls how fine-grained is the Euclidean approximation. Finally, for each training sample, multiple paths $\Gamma_i = \{\boldsymbol{\gamma}_i^j\}_{j=1}^p$ are generated by a new sequence of transformations $T_{\mathbf{s}_0}, \ldots, T_{\mathbf{s}_m}$, obtained by sampling a corresponding sequence of strengths $\mathbf{s}_0, \ldots, \mathbf{s}_m$ so that the respective magnitudes match for all p paths.

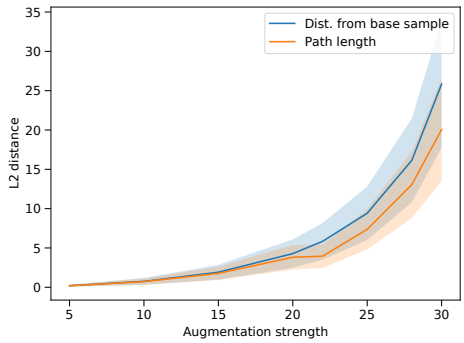
Near-Manifold Augmentations An important requirement for generating on-manifold paths is that the sequence of augmentations $\mathbf{x}_i^0 \prec \mathbf{x}_i^1 \prec \ldots \prec \mathbf{x}_i^m$ should lie in proximity of the corresponding base training sample $\mathbf{x}_i$, which is interpreted as requiring that each $\mathbf{x}_i^k$ is highly correlated with its predecessor $\mathbf{x}_i^{k-1}$. To achieve this, a weak augmentation strategy proposed by Yu, Long, and Hopcroft [169] is used, whereby each colour channel $\mathbf{x}_{ic}$ of the base training sample is factorized via singular value decomposition $\mathbf{x}_{ic} = U_i^c D_i^c V_i^c$, and a subset of the smallest singular values is erased, producing a matrix $\tilde{D}_i^c$ of lower rank. The resulting $\mathbf{x}_{ic}^k = U_i^c \tilde{D}_i^c V_i^c$ constitutes the transformed input, visualized in Figure 2.8.

Volume integration Once a sequence of paths $\{\boldsymbol{\gamma}_i^j\}_{j=1}^p$ is generated for $\mathbf{x}_i$, path-based smoothness and sharpness are computed by integrating over each path $\boldsymbol{\gamma}_i^j$, and normalizing the measure by the length $\text{len}(\boldsymbol{\gamma}_i^j)$ of each path:

$$\frac{1}{p} \sum_{j=1}^{p} \left(\frac{1}{\text{len}(\boldsymbol{\gamma}_i^j)} \int_{\boldsymbol{\gamma}_i^j} \boldsymbol{\sigma}(\mathbf{x}, y_i) d\mathbf{x} \right)^{\frac{1}{2}} \tag{2.23}$$



(a)



(b)

Figure 2.8: Strategy used to generate on-manifold paths. (a) Each row represents SVD augmentations of increasing strength, with the first column representing the base sample used to generate the corresponding augmentations in each row. (b) Average ℓ_2 distance from the base samples, for augmentations of increasing strength.

where $\boldsymbol{\sigma}$ represents an infinitesimal sharpness measure, namely the Jacobian and tangent Hessian norm functionals at $(\mathbf{x}, y_i)$. The same method can also be applied to accuracy and cross-entropy loss to evaluate consistency and confidence of the models predictions over volumes. Figures 2.7a and 2.7b illustrate geodesic MC integration. For each training point, p geodesic paths are generated, each anchored to the data manifold by m augmentations. Integrating infinitesimal measures over each path returns a MC sample of sharpness along $\boldsymbol{\gamma}_i^j$. Finally, volumetric sharpness is estimated by MC integration over p samples. Importantly, the number p of paths is fixed throughout all experiments, representing the number of MC samples for volume-based integration. Finally, a scalar, mean-field estimate is obtained by averaging over the training set $\mathcal{D}$ in Equation 2.25.

Definition 2.3.3 (Volume-based smoothness)**.**

$$\textit{smoothness} = \frac{1}{p}\mathbb{E}_{\mathcal{D}} \sum_{j=1}^{p} \left(\frac{1}{\text{len}(\boldsymbol{\gamma}_i^j)} \int_{\boldsymbol{\gamma}_i^j} J(\mathbf{x}, y_i) d\mathbf{x} \right)^{\frac{1}{2}} \tag{2.24}$$

$$= \frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} \left(\frac{1}{\text{len}(\boldsymbol{\gamma}_i^j)} \int_{\boldsymbol{\gamma}_i^j} J(\mathbf{x}, y_i) d\mathbf{x} \right)^{\frac{1}{2}} \tag{2.25}$$

An analogous construction provides volume-based sharpness, whereby the Jacobian norm functional $J(\mathbf{x}, y_i)$ is replaced with the tangent Hessian norm's $H(\mathbf{x}, y_i)$.

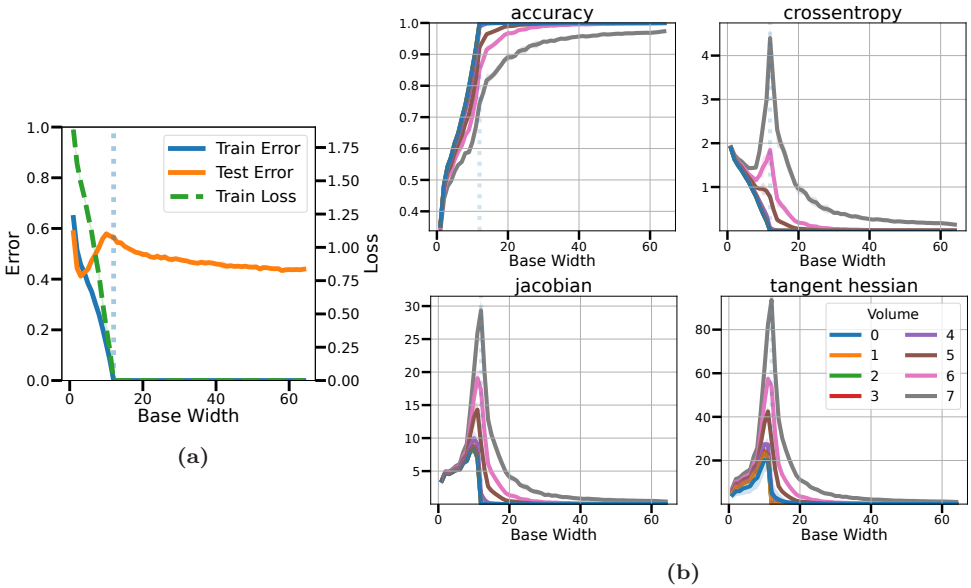


Figure 2.9: a) **Double descent curve for the test error** for ConvNets trained on CIFAR-10 with 20% noisy labels. b) Average metrics integrated over volumes of increasing size. Volumes are denoted by the number k of weak augmentations used to generate each geodesic path. All models are trained for 4k epochs.

Signatures of Overfitting

Equipped with the tools to probe the input-space geometry of the loss landscape, **Paper B** presents a study of deep networks undergoing model-wise and epoch-wise double descent [136] when trained on datasets with a fraction of the training labels corrupted. Similarly to the experimental setting of **Paper A**, all explicit regularization (batch normalization, weight decay, data augmentation, dropout, etc.) is disabled in order to remove confounders, unless explicitly stated.

Loss Landscape Curvature as a Signature of Harmful Overfitting Figure 2.9 reproduces the double descent curve of the test error for a series of ConvNets trained on CIFAR-10 [159] with 20% of the training labels randomly corrupted with symmetric noise [78]. The smallest model size able to attain zero train error is marked with a dotted vertical line and is called *interpolation threshold* [131]. Figure 2.9b presents training accuracy, cross-entropy, as well as volumetric smoothness and curvature, each integrated over volumes of increasing size, respectively corresponding to longer paths emanating from each base training sample. For each plot, a volume of 0 indicates infinitesimal measures, where neither MC integration nor geodesic integration are performed. All other volume sizes are marked by the num-

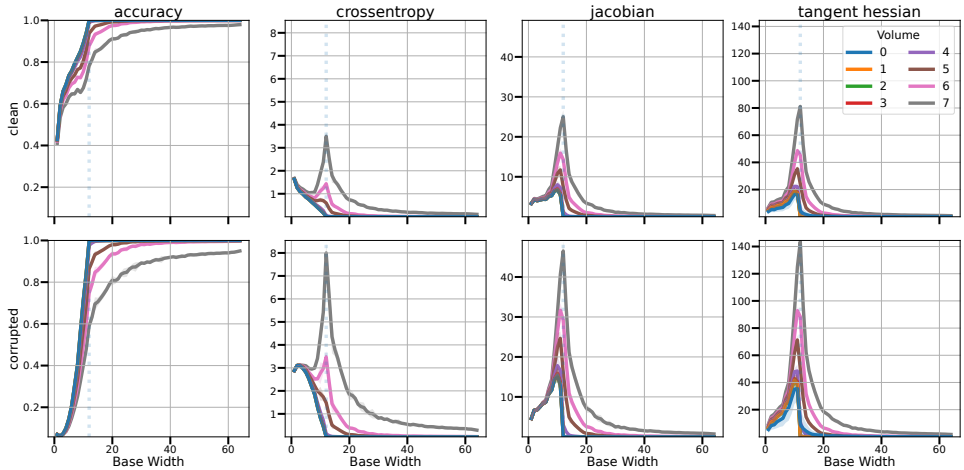


Figure 2.10: Separation between clean and noisy samples. Average accuracy, cross-entropy, Jacobian and Hessian norms integrated over volumes of increasing size (augmentations per path) around clean (top) and noisy (bottom) subsets $\mathcal{D}$.

ber k of augmentations used to construct the geodesic paths γ_i^j (c.f.r. Figure 2.7). As model size grows, interpolating models are able to perfectly fit all training samples, predicting the interpolated label with *high confidence*, corresponding to low cross-entropy loss at the training point (volume zero). When traveling away from the interpolated training points (thus increasing integration volume), cross-entropy loss, local sensitivity as well as curvature rapidly increase, with models near the interpolation threshold rapidly losing accuracy and confidence akin to the polynomial example of Figure 2.5c. This is accompanied by an increase in the test error, thus exhibiting *harmful overfitting* [35]. Finally, large overparameterized models are able to fit all training points with high accuracy and confidence, paired with a smooth and flat loss landscape, similar to the regularized polynomial (Figure 2.5a).

Smooth Interpolation of Noisy Labels Next, the smoothness and sharpness measures are computed separately on the fraction $\tilde{\mathcal{D}} \subset \mathcal{D}$ of noisily-labeled samples, and on the clean samples $\mathcal{D} \setminus \tilde{\mathcal{D}}$. Figure 2.10 reports the loss landscape metrics for clean samples (top row) and noisy samples (bottom row). First, small underfitting models attain non-trivial generalization performance by fitting mostly the cleanly-labelled samples, in line with studies showing that deep networks favour learning samples in a specific order [37], [170]. Second, while cross-entropy at a point monotonically decreases in the interpolating regime, both loss sensitivity and sharpness peak near the interpolation threshold, even for infinitesimal volumes, with larger peaks observed for incorrectly-labelled samples. Hence, whenever models are operating near the interpolation threshold (as may be the case with large datasets

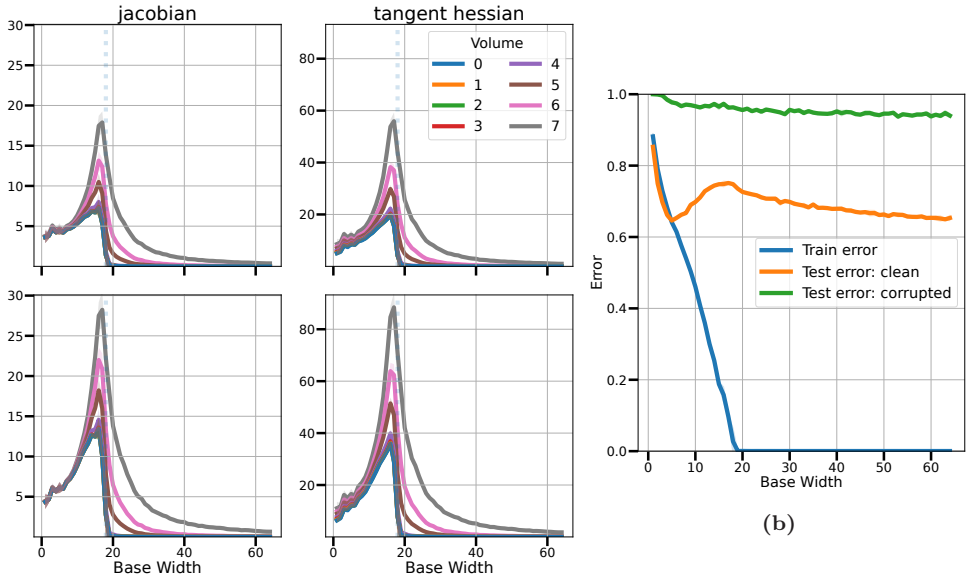


Figure 2.11: Decoupling smoothness from generalization. Randomly selected classes of CIFAR-100 are corrupted with asymmetric label noise, perturbing 80% training labels within each class, for a total of 20% corrupted samples. This enables splitting the test set classes into those associated with corrupted train samples and uncorrupted classes.

and large models), input space curvature may provide a signature of noisy samples that should be discarded. Indeed, a recent work subsequent to **Paper B** uses the observation to detect memorization at the end of training [171]. Finally, as models become largely overparameterized, both clean and noisy samples are smoothly interpolated, with corrupted labels being confidently predicted over large volumes.

Impact of Asymmetric Noise Finally, asymmetric noise is employed to decouple smooth interpolation from generalization, by training ResNets on CIFAR-100 with only selected classes randomly perturbed. Specifically, 20 randomly selected classes of the CIFAR-100 training split are corrupted with asymmetric label noise, perturbing 80% training labels within each class, for a total of 20% corrupted training samples. This modification of the training setup allows to split the *test set* into classes that have been corrupted at training time, and unperturbed classes. Figure 2.11 presents the experimental results. Similarly to the symmetric noise case, overparameterized models present a smooth and flat loss landscape around clean (top row) and corrupted samples (bottom row), with highest peaks in sharpness around corrupted samples near the interpolation threshold. At test time, the double descent trend for the test error is still clearly observed for the unperturbed classes,

while the phenomenon is removed from the heavily corrupted classes. This observation shows that large overparameterized networks express smooth interpolators, and that double descent occurs only when smoothness is aligned with generalization. After establishing that large overparameterized networks express smooth interpolators of the training data, akin to (explicitly regularized) smoothing splines and roughness-constrained polynomials, several questions remain open.

Research Question 2.3.4 (Problems III-V)**.** *What mechanisms underlie implicit smoothness regularization? How does smoothness develop throughout training, and what is the role played by the model architecture in relation to the optimizer?*

2.4 Implicit Smoothness Regularization

The experimental work presented so far isolates input-space smoothness, as measured by the input Jacobian norm, as a signature of reduced effective complexity in the overparameterized regime, suggesting that large overparameterized networks express functions of bounded variation. Formally, bounded variation of a continuous function is expressed via Lipschitz continuity.

Definition 2.4.1 (Lipschitz continuity)**.** *Let* $(\mathbb{R}^d, \|\cdot\|_p)$ *and* $(\mathbb{R}^k, \|\cdot\|_q)$ *denote two metric spaces, with distance respectively induced by the norms* $\|\cdot\|_p$ *and* $\|\cdot\|_q$*. A function* $\mathbf{f} : \Omega \subseteq \mathbb{R}^d \to \mathbb{R}^k$ *is* K*-Lipschitz continuous if* $\exists K > 0$ *such that*

$$\|\mathbf{f}(\mathbf{x}_1) - \mathbf{f}(\mathbf{x}_2)\|_q \leq K\|\mathbf{x}_1 - \mathbf{x}_2\|_p \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in \Omega \tag{2.26}$$

Typically, one refers to the Lipschitz constant of $\mathbf{f}$ as the smallest $K > 0$ for which the condition holds. If $\mathbf{f}$ is differentiable almost everywhere, Equation 2.26 implies

$$K = \sup_{\mathbf{x} \in \Omega} \|\nabla_{\mathbf{x}} \mathbf{f}\|_q \tag{2.27}$$

Investigating the Emergence of Input Smoothness

Paper C studies the emergence of bounded model function variation in relation to double descent, and investigates some of the mechanisms controlling the phenomenon. A first natural question is whether the non-monotonic trends observed for input-space smoothness in the loss landscape in **Paper B** originate from the loss function $\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}, y)$ or the underlying model function $\mathbf{f}_{\boldsymbol{\theta}}$. Indeed, as observed in **Paper A** for ReLU networks, reduced model function variation can be appreciated for interpolating affine spline models. If $p = q = 2$, computing K for the model function is equivalent to estimating the operator norm

$$K = \sup_{\mathbf{x} \in \Omega} \sup_{\substack{\mathbf{u} \in \mathbb{R}^d: \\ \|\mathbf{u}\|_2 = 1}} \|\langle \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}), \mathbf{u} \rangle\|_2 \tag{2.28}$$

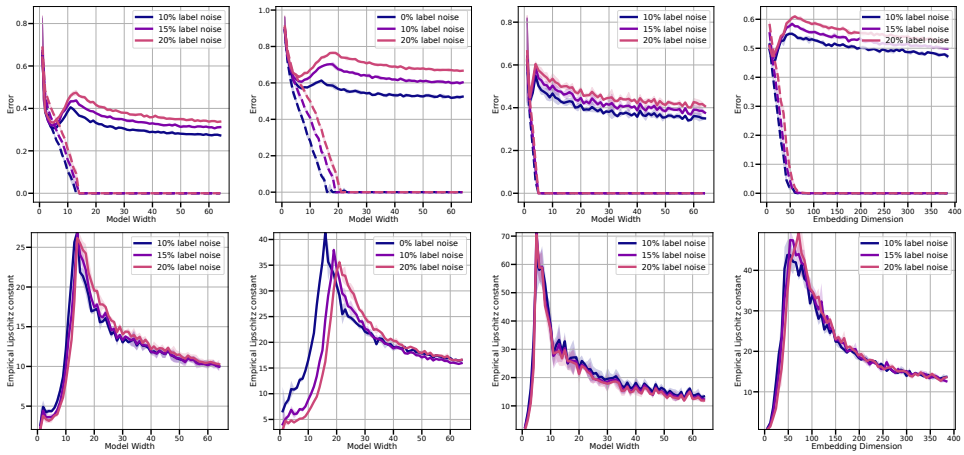


Figure 2.12: (Top) **Train error** (dashed) and **test error** (solid) for the experimental setting of **paper C**, with the test error undergoing double descent as model size increases. (Left to right) ConvNets trained on CIFAR-10 (left) and CIFAR-100 (mid-left), ResNets trained on CIFAR-10 (mid-right) and Vision Transformers on CIFAR-10 (right). (Bottom) **Empirical Lipschitz constant** for the same models. The Lipschitz lower bound depends non-monotonically on model size, strongly correlating with double descent, showing that overparameterization promotes regularization of the learned model functions via increased local Lipschitz continuity.

whereby the differential map $\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^k$ is interpreted as a linear operator. Importantly, by recalling Equation 2.5, computing the Lipschitz constant of ReLU networks is equivalent to

$$K = \sup_{\varepsilon \in [|\mathcal{P}|]} \sup_{\substack{\mathbf{u} \in \mathbb{R}^d : \\ \|\mathbf{u}\|_2 = 1}} \|\theta_{(\varepsilon)}\mathbf{u}\|_2 \tag{2.29}$$

making more apparent the intractable nature of Lipschitz constant computation for large models. Indeed, precisely computing the Lipschitz constant of ReLU networks is deemed NP-hard [172]. To this end, a lower bound on the Lipschitz constant

$$\left(\mathbb{E}_{\mathcal{D}} \|\nabla_{\mathbf{x}} \mathbf{f}_{\boldsymbol{\theta}}\|_2^2\right)^{\frac{1}{2}} := \left(\frac{1}{n}\sum_{i=1}^{n} \sup_{\mathbf{u}:\|\mathbf{u}\|\neq 0} \frac{\|\boldsymbol{\theta}_{(\epsilon(\mathbf{x}_i))}\mathbf{u}\|_2^2}{\|\mathbf{u}\|_2^2}\right)^{\frac{1}{2}} \leq K \tag{2.30}$$

as well as an upper bound are considered

$$K = \sup_{\mathbf{x}\in\Omega} \|\nabla_{\mathbf{x}} \mathbf{f}_{\boldsymbol{\theta}}\| \leq \sup_{\mathbf{x}\in\Omega} \prod_{\ell=1}^{l} \|D^{\ell}(\mathbf{x})\boldsymbol{\theta}^{\ell}\| \leq \sup_{\mathbf{x}\in\Omega} \prod_{\ell=1}^{l} \|\boldsymbol{\theta}^{\ell}\| = \prod_{\ell=1}^{l} \|\boldsymbol{\theta}^{\ell}\|_2 \tag{2.31}$$

Notably, in line with **Paper A** and **B**, the lower bound provides an average-sensitivity estimate in proximity of the training data for ReLU networks, thereby

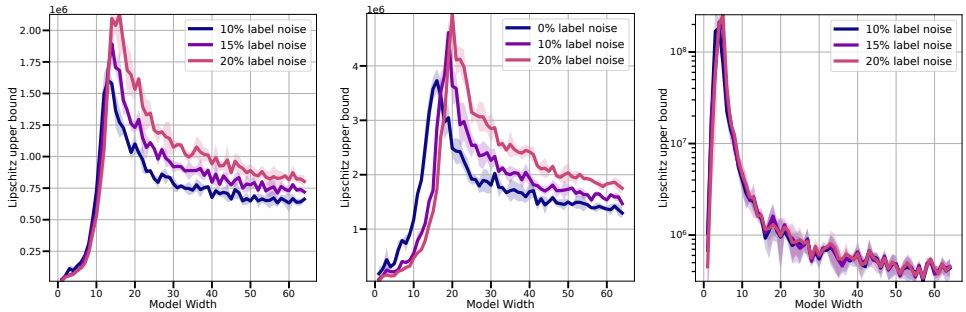


Figure 2.13: Upper bound on the true Lipschitz constant, undergoing double descent as model size increases. From left to right: ConvNets trained on CIFAR-10 (left), CIFAR-100 (middle) and ResNets trained on CIFAR-10 (right).

capturing smoothness of interpolation, this time in logit space. Furthermore, Equation 2.31 appears in several generalization bounds for deep networks [173]–[175], although its relation with double descent was not investigated prior to **Paper C**. Figure 2.12 presents the model scaling behaviour of Equation 2.30 for ConvNets, ResNets as well as Vision Transformers [3]. The top row shows the test error undergoing double descent as models become overparameterized, while the bottom row displays the empirical Lipschitz constant lower bound, which mirrors the non-monotonic trend of the test error, showing that reduced complexity in the overparameterized regime is directly captured by the model function. Figure 2.13 extends the observation to the Lipschitz upper bound of Equation 2.31. Finally, Figure 2.14 studies the contribution of cross entropy loss to its input Jacobian

$$\nabla_{\mathbf{x}} \mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}, y) = (\boldsymbol{\sigma}(\mathbf{x}) - \mathbf{e}_y) \nabla_{\mathbf{x}} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) \tag{2.32}$$

where $\mathbf{e}_y = (\delta_{cy})_{c=1}^{k}$ is the one-hot encoding of label y, $\boldsymbol{\sigma}(\mathbf{x})$ is the softmax of $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$, and $\delta_{ij} = [i = j]$ is the Kronecker delta. Importantly, the term $\|\boldsymbol{\sigma}(\mathbf{x}) - \mathbf{e}_y\|$ is related to the model's confidence in the prediction [176] $p = 1 - \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{\sigma}(\mathbf{x}_i) - \mathbf{e}_{y_i}\|$, which saturates for large uncalibrated models [177], yielding high confidence predictions at convergence. Hence, the non-monotonic trend of input-space smoothness can be ascribed to the model function $\mathbf{f}_{\boldsymbol{\theta}}$, which is thus the focus of the study henceforth.

Contribution: Geometry of Neural Network Gradients

To understand the observed trend beyond convergence, **Paper C** takes a closer look at the structure of model gradients in relationship to the network architecture, by extending earlier results for the first layer of a network [178] to networks of arbitrary depth, highlighting a layer-wise regularization mechanism embedded in the hierarchical structure of deep networks. Again referring to ReLU networks, let $\mathbf{x}^{\ell} := \varphi(\mathbf{z}^{\ell}) = \varphi(\theta^{\ell} \mathbf{x}^{\ell-1} + \mathbf{b}^{\ell})$ denote the post-activation of layer ℓ, with pre-

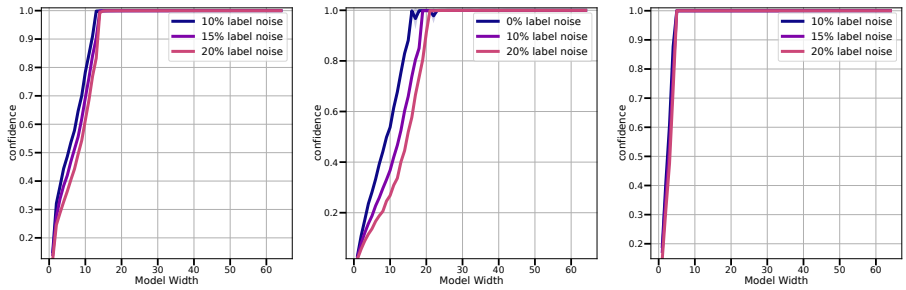


Figure 2.14: Prediction confidence as a function of model size, for ConvNets trained on CIFAR-10 (left), CIFAR-100 (middle) and ResNet18s trained on CIFAR-10. For all experimental settings, model confidence monotonically depends on model size.

activation $\mathbf{z}^\ell$ and input $\mathbf{x}^{\ell-1}$, for $\ell \in [l-1]$, with $\mathbf{x}^0 := \mathbf{x} \in \mathbb{R}^d$. By the chain rule, for each layer ℓ, the parameter gradient of $\mathbf{f}_{\boldsymbol{\theta}}$ takes the form

$$\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \theta^\ell} = \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \mathbf{z}^\ell}^T \mathbf{x}^{\ell-1 T} \tag{2.33}$$

obtained as the product of the upstream gradient $\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \mathbf{z}^\ell}$ with the local gradient. At the same time,

$$\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \mathbf{x}^{\ell-1}} = \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \mathbf{z}^\ell} \theta^\ell \tag{2.34}$$

but then, the following relation ties the input gradients to the parameter gradients

$$\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \theta^\ell} \mathbf{x}^{\ell-1} = \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \mathbf{z}^\ell}^T \|\mathbf{x}^{\ell-1}\|_2^2 \tag{2.35}$$

$$\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \mathbf{x}^{\ell-1}} = \frac{\mathbf{x}^{\ell-1 T}}{\|\mathbf{x}^{\ell-1}\|_2^2} \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \theta^\ell}^T \theta^\ell \tag{2.36}$$

Finally, taking the norm on both sides and applying Cauchy-Schwartz proves

Theorem 2.4.2. *If* $\|\theta^\ell\|_2 > 0$ *then for each* $\ell \in [l]$

$$\left\| \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \mathbf{x}^{\ell-1}} \right\|_2^2 \frac{\|\mathbf{x}^{\ell-1}\|_2^2}{\|\theta^\ell\|_2^2} \leq \left\| \nabla_{\theta^\ell} \mathbf{f}_{\boldsymbol{\theta}} \right\|_2^2 \tag{2.37}$$

where $\nabla_{\theta^\ell} \mathbf{f}_{\boldsymbol{\theta}} := \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \theta^\ell}$.

Hence, the parameter gradients at each layer control input smoothness at the layer, highlighting the role of hierarchical compositionality of deep networks in promoting input smoothness. Importantly, by noticing that $\nabla_{\mathbf{x}} \mathbf{f}_{\boldsymbol{\theta}} = \frac{\partial \mathbf{f}_{\boldsymbol{\theta}}}{\partial \mathbf{z}^\ell} \theta^\ell \nabla_{\mathbf{x}} \mathbf{x}^{\ell-1}$, Corollary 2.4.2.1 immediately follows

Corollary 2.4.2.1. *If* $\|\boldsymbol{\theta}_\ell\|_2 > 0$ *and* $\|\nabla_{\mathbf{x}}\mathbf{x}_{\ell-1}\|_2 > 0$*, then*

$$\|\nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}\|_2^2 \frac{\|\mathbf{x}^{\ell-1}\|_2^2}{\|\theta^\ell\|_2^2\|\nabla_{\mathbf{x}}\mathbf{x}^{\ell-1}\|_2^2} \leq \|\nabla_{\boldsymbol{\theta}^\ell}\mathbf{f}_{\boldsymbol{\theta}}\|_2^2 \tag{2.38}$$

The above result highlights an important aspect of implicit regularization through model depth. Throughout training, at every iteration t, each parameter update $\|\theta^\ell(t) - \theta^\ell(t-1)\|_2 \propto \|\nabla_{\theta^\ell}\mathbf{f}_{\boldsymbol{\theta}}\|_2$ and so the amount of change in input-smoothness of the model function expressed by the network at iteration t is upper bounded by the displacement $\theta^\ell(t) - \theta^\ell(t-1)$ for that parameter, establishing a link between parameter-space dynamics and input-space dynamics. The above results motivate two important questions.

Research Question 2.4.3. *What factors implicitly control parameter gradients? How does the model architecture affects regularization of* $\nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}$*?*

To study the first question, the problem is lifted to the loss landscape. Specifically, by recalling Equation 2.32, an analogous result to Corollary 2.4.2.1 follows.

Corollary 2.4.3.1. *If* $\|\theta^\ell\|_2 > 0$ *and* $\|\nabla_{\mathbf{x}}\mathbf{x}^{\ell-1}\|_2 > 0$*, then*

$$\|\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{f}_{\boldsymbol{\theta}}, \mathbf{x}, y)\|_2^2 \frac{\|\mathbf{x}^{\ell-1}(\mathbf{x})\|_2^2}{\|\boldsymbol{\theta}_\ell\|_2^2\|\nabla_{\mathbf{x}}\mathbf{x}^{\ell-1}(\mathbf{x})\|_2^2} \leq \|\nabla_{\boldsymbol{\theta}_\ell}\mathcal{L}(\mathbf{f}_{\boldsymbol{\theta}}, \mathbf{x}, y)\|_2^2 \tag{2.39}$$

Contribution: Local Geometry of the Parameter Space

To study the effect of parameter gradients on input-space smoothness, the former are related to the loss landscape geometry in proximity of a critical point. By adopting a linear stability perspective [179], [180], the loss $\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)$ is approximated in a neighbourhood of a critical point $\boldsymbol{\theta}^*$ via a second-order Taylor expansion

$$\mathbb{E}_{\mathcal{D}}\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^T H(\boldsymbol{\theta} - \boldsymbol{\theta}^*) + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2) \tag{2.40}$$

where the first order term vanishes at the critical point $\boldsymbol{\theta}^*$, as does the zero-th order term for interpolating models, and $H = \mathbb{E}_{\mathcal{D}}\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)$ represents the expected Hessian of the training loss. Let $\mathcal{L}_i(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}, \mathbf{x}_i, y_i)$ for $(\mathbf{x}_i, y_i) \in \mathcal{D}$, then

$$H = \frac{1}{n}\sum_{i=1}^{n} \nabla^2_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}) \tag{2.41}$$

$$= \frac{1}{n}\sum_{n=1}^{n} \nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i)^T \nabla^2_{\mathbf{f}_{\boldsymbol{\theta}}}\mathcal{L}_i(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i) \tag{2.42}$$

$$+ \frac{1}{n}\sum_{i=1}^{n} \nabla_{\mathbf{f}_{\boldsymbol{\theta}}}\mathcal{L}_i(\boldsymbol{\theta})\nabla^2_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i) \tag{2.43}$$

By noting that $\nabla_{\mathbf{f}_{\boldsymbol{\theta}}}\mathcal{L}_i(\boldsymbol{\theta}) \to \mathbf{0}$ when $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2 \to 0$ for interpolating models [181]–[183], the expected loss Hessian amounts to the cross term

$$H = \frac{1}{n}\sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i)^T \nabla^2_{\mathbf{f}_{\boldsymbol{\theta}}}\mathcal{L}_i(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i) + \mathcal{O}(\mathcal{L}(\boldsymbol{\theta})) \tag{2.44}$$

Finally, the analysis concludes by relating input-space smoothness to parameter space curvature as follows. Let $\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}} = (\nabla_{\theta^1}\mathbf{f}_{\boldsymbol{\theta}}, \dots, \nabla_{\theta^l}\mathbf{f}_{\boldsymbol{\theta}})$, so that $\|\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}\|_F^2 = \sum_{\ell=1}^{l} \|\nabla_{\theta^\ell}\mathbf{f}_{\boldsymbol{\theta}}\|_F^2$. Furthermore, let $c(\mathbf{x}) := \left(\sum_{\ell=1}^{l} \frac{\|\mathbf{x}^{\ell-1}(\mathbf{x})\|_2^2}{\|\theta^\ell\|_2^2 \|\nabla_{\mathbf{x}}\mathbf{x}^{\ell-1}(\mathbf{x})\|_2^2} \right)$.

Mean Squared Error To prove the following result, the MSE $\mathbb{E}_{\mathcal{D}}\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}, y) = \frac{1}{2n}\sum_{i=1}^{n}(f_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i)^2$ is considered. By noting that $\nabla^2_{\mathbf{f}_{\boldsymbol{\theta}}}\mathcal{L}_i(\boldsymbol{\theta}) = I$ for MSE in Equation 2.42, then the expected Hessian amounts to

$$H = \frac{1}{n}\sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i)^T \nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i) \tag{2.45}$$

Then, by revisiting Corollary 2.4.2.1, for MSE it holds

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\|\nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}\|_2^2 c(\mathbf{x}) &\leq \mathbb{E}_{\mathcal{D}}\|\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}\|_F^2 && (2.46)\\
&= \mathbb{E}_{\mathcal{D}}\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}^T && (2.47)\\
&= \mathbb{E}_{\mathcal{D}}\operatorname{tr}(\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}^T\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}) && (2.48)\\
&= \operatorname{tr}(H) + \mathcal{O}(\mathcal{L}(\boldsymbol{\theta})) + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2) && (2.49)
\end{aligned}$$

Cross-Entropy Loss For the case of cross-entropy, it holds $\nabla^2_{\mathbf{f}_{\boldsymbol{\theta}}}\mathcal{L}_i(\boldsymbol{\theta}) = \operatorname{diag}(\boldsymbol{\sigma}(\mathbf{x}_i)) - \boldsymbol{\sigma}(\mathbf{x}_i)\boldsymbol{\sigma}(\mathbf{x}_i)^T$. Then, revisiting Corollary 2.4.3.1

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\|\nabla_{\mathbf{x}}\mathcal{L}(\boldsymbol{\theta}\mathbf{x}, y)\|_2^2 c(\mathbf{x}) &\leq \mathbb{E}_{\mathcal{D}}\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)\|_F^2 && (2.50)\\
&= \mathbb{E}_{\mathcal{D}}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)^T && (2.51)\\
&= \mathbb{E}_{\mathcal{D}}\operatorname{tr}(\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)^T\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)) && (2.52)\\
&\leq \mathcal{L}''_{\max}(\boldsymbol{\theta})\operatorname{tr}(H) + \mathcal{O}(\mathcal{L}(\boldsymbol{\theta})) + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2) && (2.53)
\end{aligned}$$

with $\mathcal{L}''_{\max}(\boldsymbol{\theta}) = \max_{i\in[n],j,l\in[k]} \operatorname{diag}(\boldsymbol{\sigma}(\mathbf{x}_i))_{jl} - \boldsymbol{\sigma}_j(\mathbf{x}_i)\boldsymbol{\sigma}_l(\mathbf{x}_i)^T$ Lastly, by setting $\mathcal{L}''_{\max}(\boldsymbol{\theta}) = 1$ for MSE, the results are combined in the following statement.

Theorem 2.4.4. *Under the conditions of Corollary 2.4.3.1 for every layer $\ell \in [l]$*

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\|\nabla_{\mathbf{x}}\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)\|_2^2 c(\mathbf{x}) \leq \mathcal{L}''_{\max}(\boldsymbol{\theta})\,\Delta(\mathcal{L}(\boldsymbol{\theta}))\\
+ \mathcal{O}(\mathcal{L}(\boldsymbol{\theta})) + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2)
\end{aligned} \tag{2.54}$$

with $\Delta(\mathcal{L}(\boldsymbol{\theta})) = \operatorname{tr}(H)$ denoting the Laplace mean curvature operator.

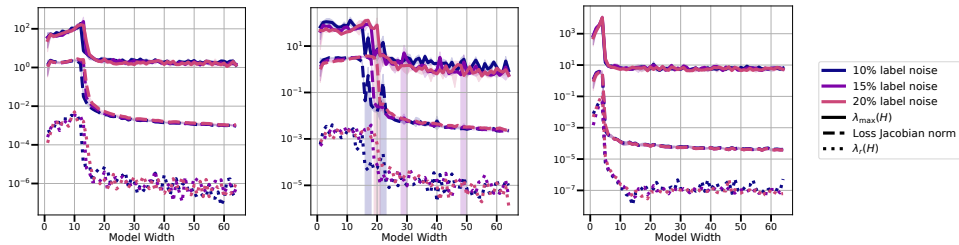


Figure 2.15: Maximum and minimum curvature for the loss in parameter space, and **input-space loss Jacobian norm**. From left to right: ConvNets trained on CIFAR-10 (left), CIFAR-100 (middle) and ResNets trained on CIFAR-10 (right). In all settings, minimum and maximum parameter-space curvature strongly correlate with double descent, peaking at the interpolation threshold, and highlighting a nonlinear dependence on model size.

The result connects input-space smoothness to parameter-space geometry, via average Hessian curvature in proximity of a critical point, highlighting the implicit regularization effect of the loss landscape geometry in proximity of an optimum for interpolating models. A related contemporary work to **Paper C** presents a similar result, while further explicity incorporating bias parameters, as well as proposing an upper bound on the parameter gradients governed by a gradient-flow interpretation of discrete step-size gradient descent [184]. Figure 2.15 presents the (loss landscape) parameter-space Hessian trace as well input-space loss Jacobian norm, showing that both quantities undergo a phase transition in the overparameterized regime. Furthermore, to draw connections to the asymptotic training limit, the smallest non-zero Hessian eigenvalue λ_r is also reported. The smallest eigenvalue closely mirrors the trend of the input-space Jacobian norm, suggesting that in the prolonged training limit, the non-monotonic behaviour is more closely modeled by λ_r. Indeed, a precise connection to the prolonged training regime could be drawn by noticing that the dynamics of gradient descent are controlled by the smallest non-zero Hessian eigenvalue, once the training dynamics have converged along all other eigendirections. By recalling the quadratic form approximation of the loss in Equation 2.40, at iteration t the parameter gradient is given by $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})(t) = H(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*)$, where the direction vector $\boldsymbol{\theta} - \boldsymbol{\theta}^*$ directs gradient descent towards the (local) optimum $\boldsymbol{\theta}^*$, with step size $\eta > 0$. The parameter updates thus take the form $\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta H(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*)$. Adding and subtracting $\boldsymbol{\theta}^*$ provides a relationship for the optimization error

$$\begin{aligned}(\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}^*) &= (\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*) - \eta H(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*) \\ &= (I - \eta H)(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*) \\ &= (I - \eta H)^{t+1}(\boldsymbol{\theta}(0) - \boldsymbol{\theta}^*)\end{aligned} \tag{2.55}$$

Finally, by decomposing the expected Hessian $H = U\Lambda U^T$, for $\Lambda = \text{diag}((\lambda_i))$ with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_p$, and studying the dynamics in the eigenspace of H

$$\begin{aligned} U^T(\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}^*) &= (U^T U - \eta U^T H U) U^T(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*) \\ &= \text{diag}((1-\eta\lambda_1)^{t+1}, \ldots, (1-\eta\lambda_p)^{t+1}) U^T(\boldsymbol{\theta}(0) - \boldsymbol{\theta}^*) \end{aligned} \tag{2.56}$$

Assuming linear stability [179], [180], then the dynamics in the long training limit $t \gg 1$ are governed by $(1-\eta\lambda_r)^t(\boldsymbol{\theta}(0) - \boldsymbol{\theta}^*)$, where λ_r is the smallest non-zero eigenvalue. The connection to the smallest non-zero Hessian eigenvalue is well aligned with recent results from related work on double descent, which provide a lower bound on the test error in terms of λ_r in the asymptotic training limit $t \to \infty$ [181]. For a more precise discussion of stability and convergence of gradient descent, as well as SGD, the reader is referred to Mori, Ziyin, Liu, *et al.* [182]. Finally, the result in **Paper C** addresses Problem III, extending observations on the data covariance for linear regression and random feature models [129], [132], [133] to finite-width deep networks, whereby curvature of the loss landscape is controlled by the parameter-space Hessian rather that the data covariance. In the remainder of the section, experimental results are presented to discuss implications of the theoretical results covered thus far.

Input Smoothness Throughout Training

The result of Corollary 2.4.2.1 suggest that, at each training iteration t, each displacement $\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)$ in parameter space controls the expected change in input-smoothness of the model function expressed by the network. To understand the effect of overparameterization on input smoothness, **Paper C** hypothesizes that larger models, endowed with higher model capacity, are able to fit the data with fewer gradient updates, thus incurring in lower deviation of the model function from initialization. This intuition is aligned with the theory of deep linear networks, whereby overparameterization is observed to have an implicit acceleration effect on training convergence [185]. However, results in the deep linear case are independent from model width, which instead plays an important role in affecting double descent for non-linear models [136]. While in the infinite width limit parameters change arbitrarily little from initialization [43], [45], characterizing the behaviour in the rich regime of feature learning poses an open question. The following claim summarizes the hypothesized acceleration effect of overparameterization.

Claim 2.4.5. *Under small-norm initialization and appropriate hyperparameters (ensuring convergence), overparameterized models attain faster interpolation, thereby effectively constraining model complexity as captured by the input Jacobian norm.*

To explore this intuition, Figure 2.16 tracks Equation 2.30 throughout training, in connection with the train error, expressing the fraction of data correctly fitted by the model at each iteration. Let ω_0 denote the smallest model width attaining zero training error. At initialization, all models have model-function input

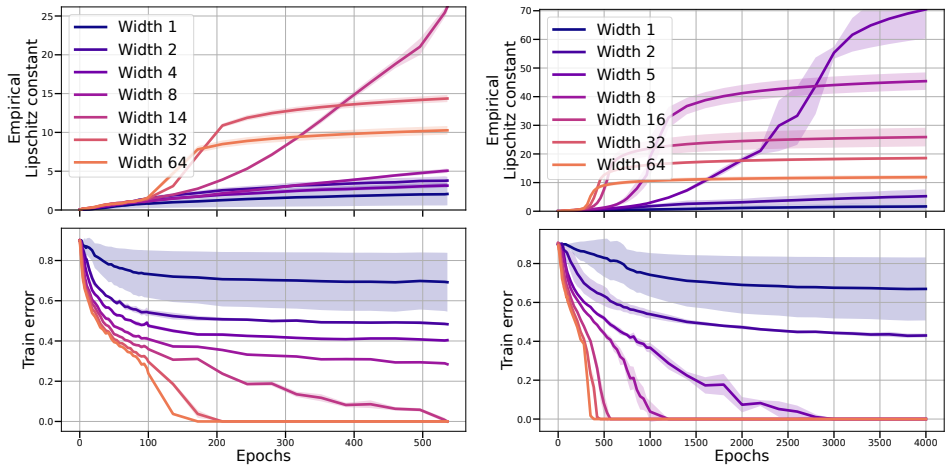


Figure 2.16: Model-function input Jacobian norm over epochs (top) and **Train error** (bottom) for ConvNets (left) and ResNets (right). The smallest models respectively attaining interpolation have width $\omega = 14$ (left panel) and $\omega = 5$ (right panel).

Jacobian close to zero, thus expressing a low-variation low-complexity function at initialization. Throughout training, three distinct behaviours are observed.

- Underparameterized models ($\omega \ll \omega_0$) are unable to interpolate the entire training set, and their training error as well as input Jacobian quickly plateau, remaining stable therefrom. Increasing size among small models reduces their training error, at the cost of increased Jacobian norm.
- Models near the interpolation threshold ω_0 – peaking in test error and Jacobian norm (cfr. Figure 2.12) – are able to achieve interpolation, *only when given considerable training budget*. Correspondingly, the Jacobian norm monotonically increases over training as the training error is reduced, resulting in models achieving worst sensitivity and worst test error.
- In contrast, overparameterized models ($\omega \gg \omega_0$) quickly interpolate the training set, with the largest models requiring fewer epochs to do so. This is matched by reduced growth of the input Jacobian norm which almost plateaus after interpolation is attained.

The seemingly unbounded input Jacobian norm of models near ω_0 suggests that the observations reported in Hardt, Recht, and Singer [186] – for which prolonged training may hurt generalization performance – are pertaining only to models near ω_0. In fact, larger models can be trained for considerably long without a comparable increase in complexity (4k epochs in Figure 2.16). To further strengthen the observation, Figure 2.17 reports the input Jacobian norm of the model function, as

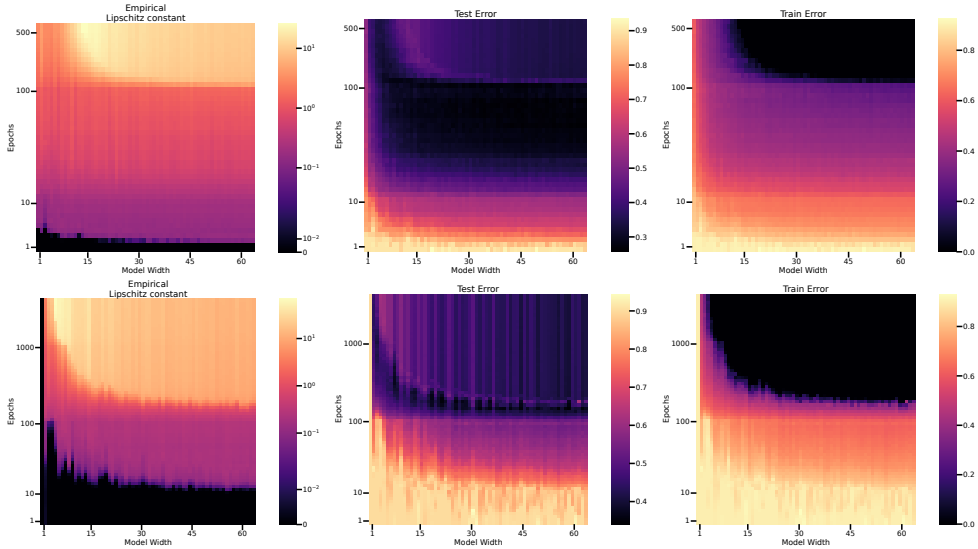


Figure 2.17: (Top left) **Empirical Lipschitz constant** (color) **as a function of training epochs** ($y - axis$) and model size (x-axis). (Top middle) **Test error** for ConvNets on CIFAR-10 with 20% noisy training labels. (Top right) **Test error** for ConvNets on CIFAR-10 with 20% noisy training labels. (Bottom) Analogous plots for ResNet18s trained on the same dataset.

well as train and test error against model size (x-axis) and training epochs (y-axis). Consistently with the line plots of Figure 2.16, small models maintain a small input Jacobian throughout training, while models near the interpolation threshold accumulate a large Jacobian norm after prolonged training. Finally, large models attain relatively low sensitivity, plateauing earlier as model size increases past the interpolation threshold. Interestingly, the heatmaps allow to visually track second-order information, captured by the rate of change of the input Jacobian. Consistently with the trends reported in **Paper B**, for all models the initial increase in Jacobian norm – occurring during "early" training (up until epoch 100 for ConvNets and 400 for ResNets) – is matched by a rapid decrease in test error. During mid-training (epoch $100 < t < 200$ for ConvNets, and $400 < t < 500$ for ResNets) the rate of increase of the Jacobian norm changes according to model size. Small models plateau in their Jacobian norm, train and test error, and remain stable thereafter. Models near the interpolation threshold start slowly increasing the Jacobian norm as they slowly interpolate the training set, with corresponding increase in test error, showcasing the "malign overfitting" phenomenon [133], Strikingly, large models quickly interpolate the training set, causing relative increase in the Jacobian norm, inversely correlating with model size. Throughout this phase of "accelerated interpolation" the test error undergoes epoch-wise double descent [136]. Crucially, while

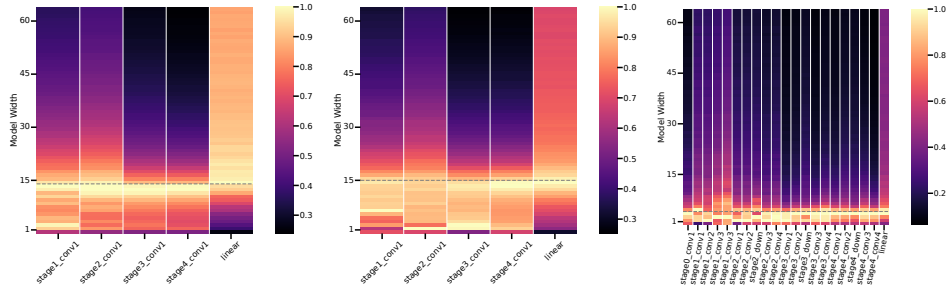


Figure 2.18: Distance from initialization for each layer of ConvNets trained on noisy CIFAR-10 (left), CIFAR-100 (middle), and ResNet18s trained on noisy CIFAR-10 (right). For each ConvNet and most ResNet layers, distance from initialization follows double descent, peaking at the interpolation threshold (dashed), suggesting global boundedness of the model function beyond training data for large models.

for all models the Jacobian norm monotonically increases over epochs, its *rate* of growth correlates with *epoch-wise* double descent for the test error.

Effective Complexity Beyond the Training Set

Referring again to Figure 2.16, this section discusses implications of the observed trends for effective model complexity beyond the training data. By recalling that the parameters of neural networks are typically initialized to small values around zero [7], [187], the input Jacobian norm of all models is close to zero at the beginning of training (c.f.r. Equation 2.6 for ReLU networks). This corresponds to all models expressing a low sensitivity, small output-norm function at initialization, albeit with low generalization performance (typically close to random chance). Second, during training, fitting the dataset requires all models' Lipschitz constant to grow [174]. This is also reflected by a corresponding increase in input Jacobian norm. When zero error is reached ($\omega \geq \omega_0$), the input Jacobian norm approximately plateaus, thereafter only slowly increasing over epochs. Recalling that large models interpolate faster, this finding suggests that large models may achieve interpolation via small (but meaningful) deviation from initialization, realizing an overall smooth function even beyond the training set. To assess this hypothesis, **Paper C** concludes by tracking the rescaled distance from initialization

$$\frac{\|\theta^\ell(t^*) - \theta^\ell(0)\|_F}{\|\theta^\ell(0)\|_F} \tag{2.57}$$

of each layer $\ell \in [l]$, where t^* denotes the last training epoch. Figure 2.18 presents distance from initialization (colour) as model width increases (y-axis), for each layer (x-axis), for ConvNets (left) and ResNets (right). For almost all layers, the quantity follows double descent as model width increases, peaking near the interpolation threshold (dashed line), and matching the epoch-wise trend reported in

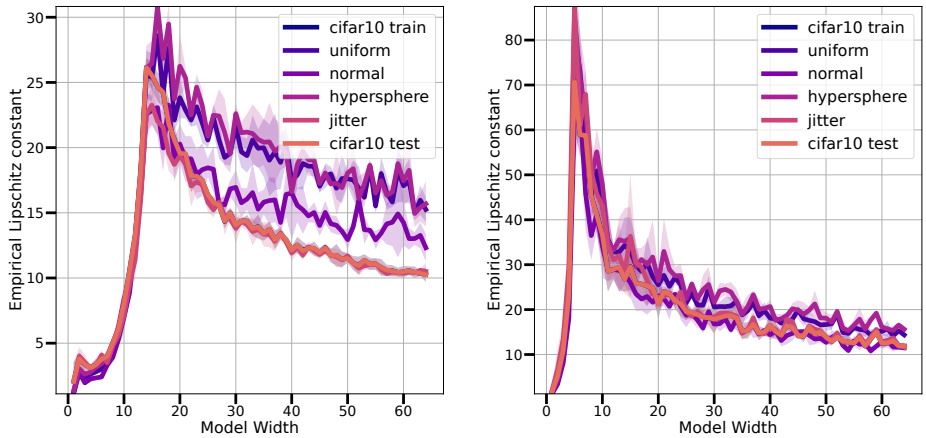


Figure 2.19: Model-function **input Jacobian estimated on the test set**, as well as random noise sampled from the input space, for ConvNets (left) and ResNets (right).

Figure 2.16. This experiment substantiates the interpretation that faster interpolation, as promoted by overparameterization, results in model functions which have overall low-complexity. The observation extends Neyshabur, Li, Bhojanapalli, *et al.* [24], who initially reported that distance from initialization decreases for overparameterized models, by further showing that the statistic is non-monotonic in model size, peaking near the interpolation threshold. Together with the observed low mean curvature of large models shown in Figure 2.15, this finding shares potential connection to the linear mode connectivity phenomenon [188], by which low-loss paths that connect solutions obtained by optimization of the same model and task have been found in practice. To estimate model variation away from the training data, Figure 2.19 tracks the model function input Jacobian norm for models trained on CIFAR-10, probing the networks by computing Equation 2.30 on unseen test data as well as on random noise lying far from the support of the data distribution. Intriguingly, the input Jacobian norm remains bounded even far from $\mathcal{D}$, and the model-wise trend follows double descent, peaking at the interpolation threshold. This finding supports the view that implicit acceleration, as afforded by overparameterization, may essentially control global model function complexity. Finally, the observation bears important connections to the Interpolation Information Criterion [189], that interprets small-norm parameter initialization as a geometric prior (towards low norm solutions), and expresses distance of a model from said prior as distance of the converged parameter from initialization.

Implications for Implicit Regularization

Revisiting Problem V, the work presented so far suggests a formulation of effective complexity for non-linear hypothesis spaces spanned by overparameterized networks. Under small-norm initialization and appropriate training hyperparameters, minimum ℓ_2-norm interpolation could be cast as an implicitly regularized problem

$$\mathbf{f}_{\hat{\boldsymbol{\theta}}} = \underset{\substack{\boldsymbol{\theta}\in\Theta \text{ s.t.} \\ \mathcal{L}_{0/1}(\boldsymbol{\theta},X,\mathbf{y})=0}}{\operatorname{argmin}} \mathbb{E}_{\mathcal{D}}\|\nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}\|_2 \tag{2.58}$$

framing the hypothesis space $\mathcal{H}$ as a Sobolev space, whereupon interpolation with deep networks is implicitly biased towards solutions with low Jacobian norm. Recalling the minimum ℓ_2-norm interpolator for overparameterized linear regression (c.f.r. Chapter 1)

$$\hat{\boldsymbol{\theta}} = \underset{\substack{\boldsymbol{\theta}\in\mathbb{R}^d: \\ \mathbf{y}=X\boldsymbol{\theta}}}{\operatorname{argmin}} \|\boldsymbol{\theta}\|_2 \tag{2.59}$$

it is observed that indeed Equation 2.58 encompasses also hypothesis spaces of linear models. Importantly, for non-linear models, the notion becomes a data-dependent norm, where integration is taken over the training set $\mathcal{D}$. The implicit regularization model of Equation 2.58 mirrors the explicit regularization strategies used for fitting smoothing splines (c.f.r. Equation 2.19). Empirical evidence from **Paper B** suggests that also second order gradients (local curvature) are implicitly regularized by overparameterization, opening the question of whether Corollary 2.4.2.1 could be extended to higher order derivatives. Finally, the suggested regularization model opens a broader question of understanding for which problems interpolation coupled with implicit smoothness regularization is a good prior ensuring generalization.

Chapter 3

Beyond Supervised Learning

The success of deep networks on discriminative problems extends beyond supervised learning. Recent advances in representation learning [161] have revamped the notion of Self-Supervised Learning [103], [190], providing a broad family of unsupervised learning algorithms capable of extracting representations that match the performance of supervised learning on downstream classification [104], [191], propelling the rise of foundation models [11], [65], [66], [192], [193]. In place of approximating a supervised input-label mapping $\mathbf{x} \mapsto y$, SSL methods map unlabelled input data to embeddings that aim to capture semantic relationships between perturbed versions of the input. *Generative approaches* rely on learning to reconstruct masked input, conditionally to either input space or latent space information [1], [4], [11], [194]. *Invariance-based approaches* augment the training dataset using a family of input perturbations (e.g. rotations, translations, cropping, photometric transformations, etc.) and leverage symmetries in the augmented data to map semantically related inputs close to each other in embedding space. Among invariance-based approaches, *contrastive* methods attain invariance by mapping positively related embeddings close to each other, while separating them from negatively related ones [195], [196]. *Non-contrastive* methods operate exclusively by matching positively related embeddings, while avoiding trivial constant solutions via regularization [197], [198], or self-distillation with teacher-student networks [66], [199]. The present chapter focuses on invariance-based SSL algorithms and studies downstream robustness by exploring smoothness of the learned representations.

3.1 Preliminaries

Formally, for an encoder network $\mathbf{f}_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^p$, SSL operates on a set of unlabelled input samples $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$ with $(\mathbf{x}_1, \dots, \mathbf{x}_n)^T \in \mathbb{R}^{n \times d}$ to produce a representation $(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_1), \dots, \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_n))^T \in \mathbb{R}^{n \times p}$ by training the encoder with an optional projection head $\mathbf{h}_{\boldsymbol{\Phi}} : \mathbb{R}^p \to \mathbb{R}^q$ returning embeddings $(\mathbf{h}_{\boldsymbol{\Phi}} \circ \mathbf{f}_{\boldsymbol{\theta}})((\mathbf{x}_1, \dots, \mathbf{x}_n))$. The learned encoder $\mathbf{f}_{\boldsymbol{\theta}}$ can later be used by a simpler model $\mathbf{g} : \mathbb{R}^p \to \mathbb{R}^k$ to

solve a downstream task by training $\mathbf{g} \circ \mathbf{f}_{\boldsymbol{\theta}}$ using labelled data while either keeping the encoder $\mathbf{f}_{\boldsymbol{\theta}}$ frozen (*linear probing*) or *fine-tuning* $\mathbf{g} \circ \mathbf{f}_{\boldsymbol{\theta}}$ end-to-end. A prominent SSL training paradigm relies on data augmentation [200], [201] to generate m perturbed views of the training data and enforcing the corresponding embeddings to be approximately invariant under the chosen family of augmentations. Formally, for a perturbation distribution $\Delta_{\mathcal{X}\times\mathcal{X}}$ over the input space $\mathcal{X} := \mathbb{R}^d$, perturbed versions $\xi(\mathbf{x})$ of $\mathbf{x}$ are drawn from $\Delta_{(\xi|\mathbf{x})}$, augmenting the training set to $(\mathbf{x}_1, \dots, \mathbf{x}_n)^T \mapsto (\xi_1(\mathbf{x}_1), \dots, \xi_m(\mathbf{x}_1), \dots, \xi_1(\mathbf{x}_n), \dots, \xi_m(\mathbf{x}_n))^T \in \mathbb{R}^{nm\times d}$. Hereafter, $\mathbf{X} \in \mathbb{R}^{nm\times d}$ is used to denote the augmented training set, with corresponding features $\mathbf{Z} \in \mathbb{R}^{nm\times p}$. Sometimes, related samples are for convenience denoted by $\mathbf{x}_i^j := \xi_j(\mathbf{x}_i)$, for $j \in [m]$ and $i \in [n]$, with corresponding features $\mathbf{z}_i^j$. Finally, the relation between perturbed samples is represented by the *augmentation graph* [202] with adjacency matrix $\mathbf{G} \in \mathbb{R}^{nm\times nm}$, with $\mathbf{G}_{ii} = 0\ \forall i$ and $\mathbf{G}_{ij} > 0$ if samples $\mathbf{X}_i$ and $\mathbf{X}_j$ are semantically related, for $i \neq j$.

3.2 Invariance-Based Objectives

Several SSL training objectives $\mathcal{L} : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}_0^+$ can be expressed as the weighted sum of an invariance term and a regularization term

$$\mathcal{L} = \mathcal{L}_{\text{inv}} + \mathcal{L}_{\text{reg}} \tag{3.1}$$

Cabannes, Kiani, Balestriero, *et al.* [203] propose a theoretical SSL objective that captures both contrastive and non-contrastive methods by generalizing the contrastive loss of [202] to encompass the non-contrastive VICReg loss [197]

$$\mathcal{L} = \beta \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \mathbb{E}_{\xi_i,\xi_j} \Big[d\big(\mathbf{f}_{\boldsymbol{\theta}}(\xi_i(\mathbf{x})), \mathbf{f}_{\boldsymbol{\theta}}(\xi_j(\mathbf{x}))\big)^2 \Big] + \|\mathbb{E}_{\xi}[\mathbf{f}_{\boldsymbol{\theta}}(\xi)\mathbf{f}_{\boldsymbol{\theta}}(\xi)^T] - \mathbf{I}_p\|_2^2 \tag{3.2}$$

where $d(\cdot,\cdot)$ is a (pseudo-)distance between embeddings. Equation 3.2 allows to recover popular SSL objectives. By decomposing the second term on the RHS into a trace term [1] and an orthogonalization constraint

$$\sum_{i=1}^{p} (\rho(\mathbf{Z}_{:i}, \mathbf{Z}'_{:i}) - 1)^2 + \sum_{i=1}^{p} \sum_{j\neq i} \rho(\mathbf{Z}_{:i}, \mathbf{Z}'_{:j})^2 \tag{3.3}$$

for $\beta > 0$, one recovers the VICReg objective [197], while setting $\beta = 0$ yields Barlow Twins [198]. Finally, SimCLR [196] is equivalent to setting $\beta = 1$ and using

[1] Additionally, Equation 3.3 is rewritten by computing the cosine similarity between all positively related pairs according to $\mathbf{G}$. This is accomplished by replacing $\rho(\mathbf{Z}_{:i}, \mathbf{Z}_{:j})$ with $\rho(\mathbf{Z}_{:i}, \mathbf{Z}'_{:j})$, and by constructing [204] $\mathbf{Z}, \mathbf{Z}' \in \mathbb{R}^{nm^2\times p}$ as $\mathbf{Z} = (\xi_1(\mathbf{x}_1), \dots, \xi_m(\mathbf{x}_1), \dots, \xi_1(\mathbf{x}_1), \dots, \xi_m(\mathbf{x}_1), \dots, \xi_1(\mathbf{x}_n), \dots, \xi_m(\mathbf{x}_n), \dots, \xi_1(\mathbf{x}_n), \dots \xi_m(\mathbf{x}_n))^T$, $\mathbf{Z}' = (\xi_1(\mathbf{x}_1), \dots, \xi_1(\mathbf{x}_1), \dots, \xi_m(\mathbf{x}_1), \dots \xi_m(\mathbf{x}_1), \dots, \xi_1(\mathbf{x}_n), \dots, \xi_1(\mathbf{x}_n), \dots, \xi_m(\mathbf{x}_n), \dots \xi_m(\mathbf{x}_n))^T$.

the following similarity score

$$d(\mathbf{Z}_i, \mathbf{Z}_j) = -\log\left(\frac{\exp(\rho(\mathbf{Z}_i, \mathbf{Z}_j)/\tau)}{\sum_{h=1\ h\neq i}\exp(\rho(\mathbf{Z}_i, \mathbf{Z}_h)/\tau)}\right) \tag{3.4}$$

with temperature parameter $\tau > 0$. A typical choice of ρ is given by the cosine similarity for SimCLR and Barlow Twins, and by the standard inner product between (centered) $\mathbf{Z}_i$ and $\mathbf{Z}_j$ for VICReg. Importantly, for contrastive objectives as well as VICReg, the invariance term pushes embeddings of semantically related samples close to each other in feature space. At the same time, for non-contrastive losses such as Barlow Twins and constrained VICReg ($\beta = 0$) invariance is encoded purely by maximizing cross-correlation of related embeddings (Equation 3.3), while retaining global feature informativeness via the regularization term [198].
For linear encoders $\mathbf{f}_\theta = \theta \in \mathbb{R}^{d\times p}$, with $\mathbf{Z} = \mathbf{X}\theta$, Balestriero and LeCun [204] provide closed-form solutions for several contrastive and non-contrastive SSL objectives, allowing to formulate invariance-based SSL as a regression problem. Taking for example $\beta = 0$, allows to obtain a formulation of the Barlow Twins loss as a constrained optimization problem

$$\mathcal{L} = \left\|\frac{1}{nm}\theta^T\mathbf{X}^T\mathbf{X}\theta - \mathbf{I}_p\right\|^2 \tag{3.5}$$

whereby the unconstrained Lagrangian formulation recovers the original Barlow Twins loss (Equation 3.3). The linear regression interpretation allows to immediately see that learning invariance in SSL is equivalent to recovering the left singular vectors of particular loss-specific cross-correlation (or covariance) operators (and thus that the optimization problem is rotation invariant [202]). Furthermore, if the covariance $\mathbf{X}^T\mathbf{X}$ is diagonalized as $U^T\mathbf{X}^T\mathbf{X}U = \Lambda$, minimizing Equation 3.5 amounts to whitening the covariance via θ, and thus recovering in θ the inverse covariance singular values $\Lambda^{-\frac{1}{2}}$. Importantly, for an encoder network of width p, the problem of minimum norm interpolation can be cast as recovering the top p eigendirections of the covariance operator [203], [205], [206].
By explicitly learning invariance to input perturbations, augmentation-based SSL naturally attains a degree of robustness to noise. Equation 3.5 frames learning as an approximation problem [205], whereby invariance is attained by minimizing feature variation. In the general setting and in keeping with standard regression, the quality of the approximation is determined by the expressivity of the encoder (controlled by p and the model architecture), as well as the structure of the learning problem, controlled by $\mathbf{X}^T\mathbf{X}$, and the augmentation graph $\mathbf{G}$. This interpretation begs two important questions.

Research Question 3.2.1. *How does model scaling (controlling p) affect the approximation quality, and thus the attained invariance? How well does invariance transfer downstream?*

Paper D explores the question through the lens of effective representation rank.

3.3 Geometry of Self-Supervised Representations

Before exploring representation robustness, the linear evaluation protocol of SSL is briefly recalled. Given a supervised downstream task, assessing the performance of pre-trained representations typically involves composing a linear classification head $\mathbf{g} : \mathbb{R}^p \to \mathbb{R}^k$ of parameter $\mathbf{W}$ with the encoder network, and training the resulting classifier on the task while freezing the encoder's parameters. The evaluation setting presents two important drawbacks. First, it relies on access to labelled downstream data, which might be expensive to collect. Second, the process of downstream evaluation can be time and resource consuming, especially during the development stage of a model, when multiple hyperparameter settings and network configurations have to be tested. In order to measure the quality of representations without relying on labelled downstream data, a series of recent works establishes a positive correlation between downstream performance and notions of *effective rank* of the representation covariance estimated on the pre-training dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j) - \overline{\mathbf{f}}_{\boldsymbol{\theta}})(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j) - \overline{\mathbf{f}}_{\boldsymbol{\theta}})^T \tag{3.6}$$

centered at $\overline{\mathbf{f}}_{\boldsymbol{\theta}} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j)$. Compared to the standard notion of rank of a matrix, real-valued estimates of effective rank aim to capture feature diversity by measuring concentration of energy over the available p feature dimensions [53], [207], as a function of the eigenspectrum $\lambda(\Sigma) = \{\lambda_1 \geq \ldots \geq \lambda_r\}$ for $r \leq \min\{mn-1, p\}$. Agrawal, Mondal, Ghosh, *et al.* [208] empirically observe that the covariance eigenspectrum of SSL representations is well approximated by a power law

$$\lambda_i \propto i^{-\alpha} \tag{3.7}$$

and correlate its *spectral decay coefficient* $\alpha > 0$ with downstream performance, showing that best downstream generalization is attained for $\alpha \approx 1$. He and Ozay [209] study the emergence of power-law eigenspectra in self-supervised representations of vision datasets, in relation to model hyperparameters such as the depth of the projector head. Finally, Garrido, Balestriero, Najman, *et al.* [210] estimate effective rank based on the entropy of the normalized eigenspectrum [207]

$$\text{RankMe}(\Sigma) = \exp\Big(-\sum_{j=1}^{r} p_j \log p_j\Big) \tag{3.8}$$

for $p_j = \frac{\sigma_j}{\|\boldsymbol{\sigma}\|_1}$, where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_r)$ and $\sigma_j = \sqrt{\lambda_j}$ are the singular values [2] of Σ. Intuitively, models with high effective rank are able to efficiently encode

[2] In [210] the effective rank measure is computed using the singular values of the *embeddings*' covariance rather than the representation covariance Σ. However, the authors explicitly assume a monotonic relationship between the effective rank of the representation and that of the embeddings, implying that either can be employed for computing order correlations.

diverse features from the pre-training set, that a simple linear readout layer could later exploit to solve a specific downstream task. A second intuition relies on the problem of representation collapse in SSL, whereby an encoder may attain trivial invariance by mapping every input to a constant [211]. Hence, high representation rank positively correlates with the model avoiding collapse. Finally, the authors of `RankMe`, justify the connection between high representation rank and downstream performance, by invoking Cover's theorem [212], for which non-linear separability of a set of data points assigned to random groups is more likely in high-dimensions, and further noting that linear probes are unable to increase the representation rank.

Exploring Representation Robustness

Building on the intuition that expressive encoder architectures are in principle able to attain lower approximation error and consequently stronger invariance, **Paper D** explores whether notions of effective rank of Σ are indicative of *downstream robustness* for high-dimensional representations. A key quantity expressing robustness of an encoder $\mathbf{f}_{\boldsymbol{\theta}}$ to local perturbations is the expected Jacobian norm

$$J = \mathbb{E}_{\mathcal{D}} \|\nabla_{\mathbf{x}} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})\|_2 \tag{3.9}$$

capturing sensitivity of the representation to perturbations of the input data. In turn, encoder robustness affects that of a classifier $\mathbf{g}$, since

$$\|\nabla_{\mathbf{x}} \mathbf{g}(\mathbf{x})\|_2 \leq \|\mathbf{W}\|_2 \|\nabla_{\mathbf{x}} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})\|_2 \tag{3.10}$$

Furthermore, tracking Equation 3.9 allows to draw direct connections to the representation covariance Σ, which is the object of studies of rank-based representation quality metrics. Current self-supervised objectives encode rank-seeking terms in order to prevent representation collapse, either implicitly [194], [196], [199] or explicitly [197], [198]. In turn, for models with large capacity, high-rank at scale may promote retaining nuisance factors from the training data, thus affecting representation sensitivity. At the same time, as seen in **Paper C**, robustness of large networks improves in the overparameterized regime [213], where redundancy of representations increases. Hence, connecting the geometry of Σ to downstream robustness entails two methodological questions.

Research Question 3.3.1. *What is the connection between effective representation rank and robustness? For fixed sample size $|\mathcal{D}|$, how does robustness scale in relation to encoder size?*

Contribution: Effective Rank and Input Sensitivity

The first step towards studying representation robustness is relating Equation 3.9 to effective rank of Σ. By recalling that, for each sample $\mathbf{x}_i \in \mathcal{D}$, augmentation-based SSL methods map m views $\mathbf{x}_i^1, \ldots, \mathbf{x}_i^m$ of $\mathbf{x}_i$ to related representations $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j)$,

Paper D provides a decomposition of the feature covariance Σ into two terms. The first one measures local variation within each set of representations of related views, while the second measures variation of representations associated with different base inputs $\mathbf{x}_i$. For each base input $\mathbf{x}_i$, let $\overline{\mathbf{x}}_i = \frac{1}{m}\sum_{j=1}^{m}\mathbf{x}_i^m$ denote the centroid of the set of positives $\{\mathbf{x}_i^1, \ldots, \mathbf{x}_i^m\}$. Then, a local linearization [3] of $\mathbf{f}_{\boldsymbol{\theta}}$ is considered such that

$$\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i) + \nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i)(\mathbf{x} - \overline{\mathbf{x}}_i) + o(\|\mathbf{x} - \overline{\mathbf{x}}_i\|^2) \tag{3.11}$$

using the shorthand $\nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i) := \nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})|_{\mathbf{x}=\overline{\mathbf{x}}_i}$. Then, $\frac{1}{m}\sum_{j=1}^{m}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j) \approx \mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i)$ and

$$\frac{1}{m}\sum_{j=1}^{m}\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j) - \mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i)\big)\big(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j) - \mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i)\big)^T = \nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i)\Sigma^{(i)}\nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i)^T \tag{3.12}$$

where $\Sigma^{(i)} = \frac{1}{m}\sum_{j=1}^{m}(\mathbf{x}_i^j - \overline{\mathbf{x}}_i)(\mathbf{x}_i^j - \overline{\mathbf{x}}_i)^T$ is the input space covariance for positive views $\mathbf{x}_i^j$, for $j \in [m]$. Finally, by the law of total covariance, Σ can be decomposed into

$$\begin{aligned}
\Sigma &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j) - \overline{\mathbf{f}}_{\boldsymbol{\theta}})(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j) - \overline{\mathbf{f}}_{\boldsymbol{\theta}})^T \\
&= \frac{1}{n}\sum_{i=1}^{n}\nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i)\Sigma^{(i)}\nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i)^T + \frac{1}{n}\sum_{i=1}^{n}(\mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i) - \overline{\mathbf{f}}_{\boldsymbol{\theta}})(\mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i) - \overline{\mathbf{f}}_{\boldsymbol{\theta}})^T \\
&= \Sigma_{\text{intra}} + \Sigma_{\text{inter}}
\end{aligned} \tag{3.13}$$

The first term in the decomposition measures the expected covariance within representations corresponding to views of the same input, and the second captures covariance between representations of different base samples. Concretely, $\boldsymbol{\Sigma}_{\text{intra}}$ captures local variation of related representations around the corresponding centroid $\mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i)$, thus measuring invariance attained by the representation. If the invariance term in Equation 3.2 is written in terms of the squared ℓ_2 distance

$$\mathcal{L}_{\text{inv}} = \beta\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\mathbb{E}_{\xi_i,\xi_j}\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j) - \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j)\|_2^2 \tag{3.14}$$

by the triangle inequality it is easy to see that reducing the distance between each $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_i^j)$ and the corresponding centroid $\mathbf{f}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_i)$ reduces the approximation error, in turn improving invariance. The second term instead measures separability of representations corresponding to different base samples, whereby large variance denotes well separated representations. Importantly, the two covariances also appear in theoretical analyses of *neural collapse* of deep representations [75] in supervised learning as well as in the Fisher discriminant [124], whereby $\text{tr}\left(\Sigma_{\text{intra}}^{-1}\Sigma_{\text{inter}}\right)$ is

[3] **Paper D** provides a more nuanced local linearization of the feature space, by interpreting each cluster of positively related features as an object manifold. Here, a Euclidean approximation is used for conciseness of exposition.

measured to capture the relative degree of invariance and separability attained by a representation. Importantly, while in supervised learning class membership is used to identify related input $\mathbf{x}_i^1, \ldots, \mathbf{x}_i^m$, the present formulation does not rely on labels, and instead considers input related if they are different views of the same base training point. In the following, a link between the features Jacobian and the spectral decay coefficient (Equation 3.7) is presented.

Assumption 3.3.2. *The eigenspectrum of the feature covariance matrix Σ is well approximated by a power law with coefficient $\alpha > 0$.*

The analysis begins by noting that both Σ_{intra} and Σ_{inter} are symmetric Positive Semi-Definite (PSD), and thus admit only non-negative real eigenvalues. To simplify notation, the following assumes that Σ_{inter} is full rank. In the general case, the sequel can be easily adapted by discarding the null-space of Σ_{inter} before proceeding. Let $V^T \Sigma_{\text{inter}} V = \Lambda_{\text{inter}}$ denote the transformation that diagonalizes Σ_{inter}, with unitary matrix $V \in \mathbb{R}^{p \times p}$. Denote $Z = V \Lambda_{\text{inter}}^{-\frac{1}{2}}$ the corresponding whitening transformation. Then, the matrix $Z^T \Sigma_{\text{intra}} Z$ is symmetric PSD, and is in turn diagonalized by $U^T Z^T \Sigma_{\text{intra}} Z U = \Lambda_{\text{intra}}$. Combining the two observation yields

$$U^T Z^T \Sigma Z U = U^T Z^T \Sigma_{\text{intra}} Z U + U^T Z^T \Sigma_{\text{inter}} Z U \tag{3.15}$$
$$= \Lambda_{\text{intra}} + I_p \tag{3.16}$$

Finally, let $\lambda_k(A)$ denote the k-th eigenvalue of a square matrix A, and $r = \text{rank}(A)$. Then, the relationship $\lambda_k(A)\lambda_r(B) \leq \lambda_k(AB) \leq \lambda_k(A)\lambda_1(B)$ is used to prove

$$\lambda_k(U^T Z^T \Sigma Z U) = \lambda_k(Z^T \Sigma Z) = \lambda_k(\Lambda_{\text{inter}}^{-1} \Sigma) \tag{3.17}$$
$$= \lambda_k(I_p + \Lambda_{\text{intra}}) \tag{3.18}$$

from which

$$\lambda_k(\Sigma)\lambda_p(\Lambda_{\text{inter}}^{-1}) \leq \lambda_k(I_p + \Lambda_{\text{intra}}) \leq \lambda_k(\Sigma)\lambda_1(\Lambda_{\text{inter}}^{-1}) \tag{3.19}$$

By recalling that Σ_{intra} depends on the input Jacobian (Equation 3.12), and by virtue of assumption 3.3.2, then

$$\lambda_k(I_p + \Lambda_{\text{intra}}) \leq k^{-\alpha} \lambda_1(\Lambda_{\text{inter}}^{-1}) \tag{3.20}$$

connecting the eigenspectrum of Σ_{intra} to the spectral decay coefficient α. Then, if $\alpha \to \infty$ (corresponding to low effective rank), the k-th eigenvalue of Σ_{intra} will rapidly decay. Conversely, by

$$\lambda_k(\Lambda_{\text{inter}}^{-1}) r^{-\alpha} \leq \lambda_k(I_p + \Lambda_{\text{intra}}) \tag{3.21}$$

if $r = \text{rank}(\Sigma) \leq \min\{mn - 1, p\}$ is large and $\alpha \to 0$, then $\lambda_k(\Sigma_{\text{intra}})$ will increase. Hence, the spectral decay α and the eigenspectrum of Σ_{intra}, dependent on the Jacobian $\nabla_{\mathbf{x}} \mathbf{f}_{\boldsymbol{\theta}}$, are inversely related.

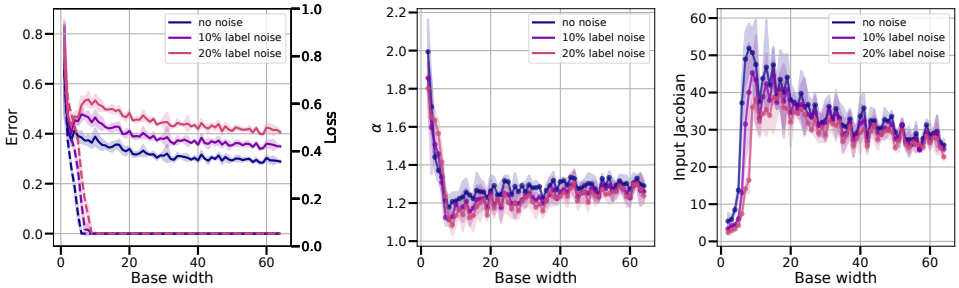


Figure 3.1: Spectral decay and feature smoothness for supervised learning. (Left to right) Test error (solid) and train error (dashed) for ResNets of increasing base width trained on CIFAR-10 as well as noisy CIFAR-10; Spectral decay coefficient; Input Jacobian norm of features $\mathbf{f}_{\theta}$ learned via standard supervised learning. The spectral decay of features as well as their input sensitivity mirror double descent as model size increases, establishing a negative correlation between global and local representation geometry measures, and motivating the use of the metrics in studying SSL representations.

Spectral Decay in Supervised Learning

To validate the connection between effective rank and feature sensitivity, the measures are first computed in a supervised learning setting. The purpose of the experiment is twofold. First, establishing a qualitative (and quantitative) correlation between α and J, and second, to assess whether the correlation holds both in the underparameterized and overparameterized regime, which are both covered by standard supervised settings. Figure 3.1 tracks the spectral decay α of the penultimate layer's feature covariance, as well as the average input Jacobian norm for the same features. In line with **Paper C**, the input Jacobian is able to mirror double descent for the test error, also matched by the spectral decay coefficient, which inversely correlates with the Jacobian norm. In the *underparameterized* regime, the input Jacobian increases up until the interpolation threshold is reached, accompanied by a slower decay of the features effective rank. Finally, in the overparameterized regime, feature sensitivity is reduced, followed by a decrease in effective rank (faster spectral decay). Importantly, in keeping with theoretical analysis of underparameterized networks [174], it is noted that fitting the training data requires a model to increase its Lipschitz constant (in turn affecting the penultimate layer's features Jacobian). As a consequence, when model scale increases within the underparameterized regime, larger models are able to fit a larger fraction of the training set, with corresponding increase in sensitivity (larger Lipschitz constant) and effective rank. The monotonic increase in input sensitivity and effective rank spans all models in the underparameterized regime, until the interpolation threshold is reached, exposing a limitation of both metrics, which are unable to discriminate between underfitting and overfitting models until interpolation is reached.

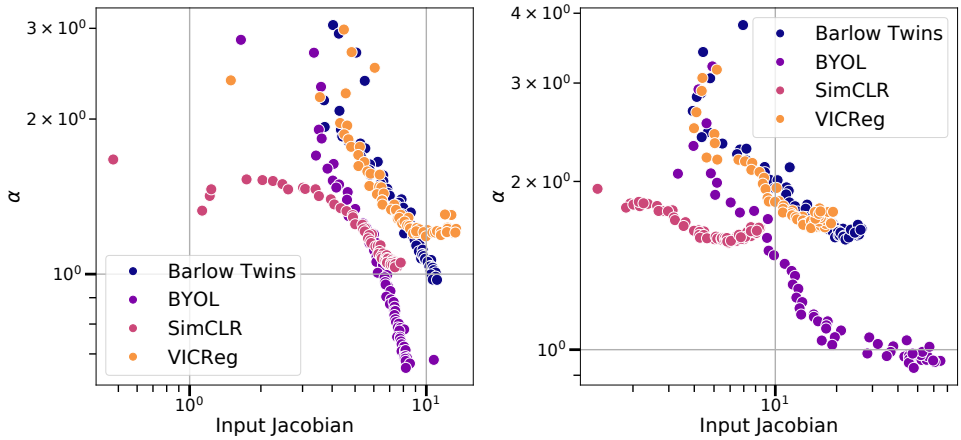


Figure 3.2: Spectral decay and feature smoothness for SSL, for ResNet18 backbones pre-trained with different SSL objectives on CIFAR-10 (left) and STL-10 (right), as model size varies, extending the inverse relation between α and the feature Jacobian to the Self-Supervised setting.

Self-Supervised Representations

After experimentally establishing a correlation between effective rank of features and their input sensitivity, a strong correlation between the two metrics is established in the SSL setting, for representations trained on a ResNet backbone with Barlow Twins [198], BYOL [199], SimCLR [196], as well as VICReg [197]. Figure 3.2 presents a strong visual correlation between effective rank and feature sensitivity (quantitatively validated in **Paper D**), extending the finding to the SSL setting.

Effect of Model Scaling

Finally, the relationship between effective rank and feature sensitivity is explored across model scales. Figure 3.3 reports downstream test performance, effective rank and feature sensitivity for ResNet backbones as the number of parameters in the encoder architecture $\mathbf{f}_{\boldsymbol{\theta}}$ varies. For all SSL algorithms considered, a monotonic trend for all three metrics is observed, with downstream performance improving as model size increases. The observed trend positions the SSL methods and architectures considered in the *underparameterized* regime of learning, whereby increasing model capacity affords better performance at the cost of increased model sensitivity (relative to smaller models). Thus, consistently with the supervised learning setting, one might expect the Jacobian norm and effective rank to be unable to discriminate between underfitting models – which may attain a degree of robustness due to their limited ability to overfit – and overfitting models with increased sensitivity in the downstream evaluation setting.

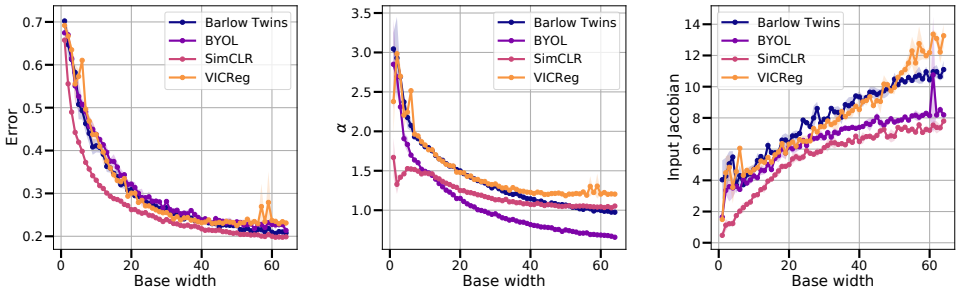


Figure 3.3: SSL across model scales. (Left to right) Test error for linear probes trained on SSL features with ResNet18 encoder on CIFAR-10; Spectral decay coefficient; Input Jacobian norm of SSL features $\mathbf{f}_{\boldsymbol{\theta}}$. With reference to Figure 3.1, performance as well as effective rank of SSL models behaves consistently with the underparameterized regime of supervised learning.

Tracking Robustness

To conclude, **Paper D** measures robustness of SSL pre-training to adversarially generated data, by relying on *poisoned datasets* [214], [215]. Data poisoning is a technique aimed at producing adversarially-corrupted datasets that are hard to learn by supervised baselines. By including class-dependent high-frequency noise in each image, poisoning adds spurious patterns to the data, that a classifier might pick on in order to discriminate the classes. However, since the patterns are not occurring in the population distribution, models trained on poisoned data show poor generalization performance. Adversarially-poisoned images are generated by using projected-gradient attacks [216] on a large model pre-trained with supervised learning, and are observed to be transferable across similar architectures [217]. Since poisoned datasets contain class-specific spurious patterns, it is conjectured that large-scale SSL backbones trained using rank-seeking objectives are pushed towards encoding those spurious patterns in their representation. Then, in the linear evaluation setting, a readout layer might pick on the spurious features to classify the downstream data. To test the hypothesis, a series of ResNet18 backbones is pre-trained on a version of CIFAR-10 poisoned by projected-gradient attack on a pre-trained ResNet50 architecture. Then, two sets of linear readout layers are trained on top of the frozen encoder, one on the standard CIFAR-10 training set, and one on the poisoned CIFAR-10. Finally, downstream performance is evaluated on the clean CIFAR-10 test set. Figure 3.4 shows the validation performance of the two sets of linear probes, in relation to their effective rank as measured by the spectral decay coefficient and `RankMe` (Equation 3.8). First, it is observed that downstream performance on clean data monotonically improves with model size, suggesting that high-rank SSL representations are able to separate 'signal' from 'noise' in their representations. Second, the effect of adversarial noise can be observed in the downstream performance of linear probes trained on the poisoned

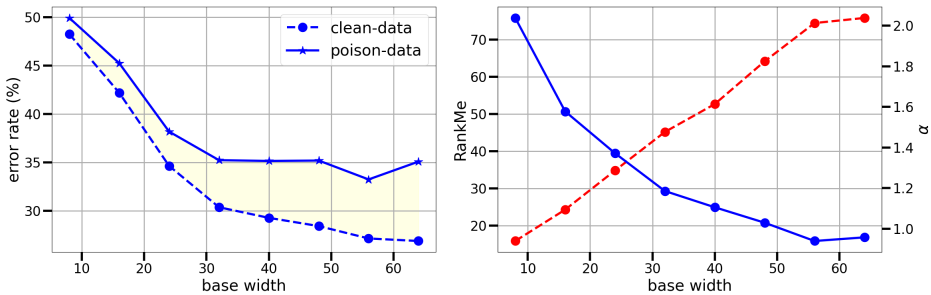


Figure 3.4: Measuring robustness of SSL pre-training across model scales. Tracking effective rank is insufficient to select models that are high performing and robust. (Left) Linear probe evaluation performance for features pre-trained on poisoned data. Linear probes are evaluated both on the clean test set (dashed line) and poisoned data (solid line). (Right) Tracking effective rank as model size increases, using RankMe (red line) and the spectral decay coefficient (blue line).

data, whereby performance is affected for large enough models, showing an increasing gap between robustness and performance as model size grows. Intriguingly, robustness *plateaus* for large models, consistently over half of the model scales considered (base width > 32). Lastly, it is observed that measures of effective rank are unable to capture the plateau in robustness, indicating that tracking effective rank may be insufficient to measure downstream robustness. In, conclusion, in the range of model scales considered, SSL methods behave consistently with the underparameterized regime of supervised learning, whereby increasing model scale affords better downstream performance. However, different from supervised learning, there exists model scales after which downstream performance and robustness respectively plateau, whereas effective rank continues increasing, suggesting a saturation threshold for downstream robustness. Importantly, measuring effective rank of the feature distribution is not sufficient for tracking downstream robustness, raising the need for dedicated measures targeting SSL representations.

3.4 Exploring Representation Robustness

The present section expands the study of downstream robustness of SSL under model scaling, to investigate the connection between invariance-based learning and robustness. Recalling Equation 3.2, augmentation-based SSL algorithms extract representations of the data by minimizing the distance between related embeddings, while preventing representation collapse. As discussed in section 3.2, learning invariance is equivalent to recovering the eigendirections of appropriate data-covariance operators (dependent on the choice of loss function). The degree of invariance attained on the training data by the learner is directly tied to the ap-

proximation error, expressing how closely the features capture variation of the data. Within this setting, an important open question towards the safe deployment of SSL methods is *understanding to what extent the attained invariance transfers downstream*, with OOD generalization providing an important class of problems in which invariance to input perturbations has been connected to improved robustness.

Formally, *in the context of supervised learning*, theoretical frameworks of OOD robustness treat a deep network classifier as the composition of a linear predictor $\mathbf{g}$ with non-linear features $\mathbf{f}_{\boldsymbol{\theta}}$. For a collection of input $\mathcal{E}_{\text{avail}} = \{(\mathbf{x}^e, y) : e = 1, \dots, E\}$ governing the supervised training data across different environmental conditions, the problem of generalizing to unseen data sampled from a wider set of distributions $\mathcal{E}_{\text{all}} \supseteq \mathcal{E}_{\text{avail}}$ is related to the variation of features $\mathbf{f}_{\boldsymbol{\theta}}$ over $\mathcal{E}_{\text{avail}}$ [30]. Variation is given by

$$\mathcal{V}(\mathbf{f}_{\boldsymbol{\theta}}, \mathcal{E}_{\text{avail}}) = \max_{y \in 1,\dots,k} \sup_{e_i, e_j} d\big(\mathbb{P}(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}^{e_i})|y), \mathbb{P}(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}^{e_j})|y)\big) \tag{3.22}$$

where $d(\cdot, \cdot)$ represent a symmetric distance between distributions and y denotes supervised class information. Assuming that OOD data $\mathcal{E}_{\text{all}}$ magnifies variation of the features $\mathbf{f}_{\boldsymbol{\theta}}$, a non-negative, non-decreasing *expansion function* $s : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ fixing 0, represents the increase in feature variation $s(\mathcal{V}(\mathbf{f}_{\boldsymbol{\theta}}, \mathcal{E}_{\text{avail}})))$ incurred in $\mathcal{E}_{\text{all}}$. Based on the expanded variation, Ye, Xie, Cai, *et al.* [30] mandate that, for learnable tasks, reducing feature variation on $\mathcal{E}_{\text{avail}}$ can contrast the expansion function, thus improving the chances of OOD generalization.

In the context of SSL, invariance-based objectives promote a stronger notion of convergence than Equation 3.22, making SSL representations potential strong candidates for learning robust features. More precisely, Equation 3.22 can be cast in the context of SSL by identifying the ID domains $\mathcal{E}_{\text{avail}}$ with the distribution of perturbations (data augmentation) used to generate multiple views of the unlabelled training data $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$

$$\mathcal{V}(\mathbf{f}_{\boldsymbol{\theta}}, \mathcal{E}_{\text{avail}}) = \max_{\mathbf{x} \in \mathcal{D}} \sum_{\xi_i, \xi_j} d\big(\mathbf{f}_{\boldsymbol{\theta}}(\xi_i(\mathbf{x})), \mathbf{f}_{\boldsymbol{\theta}}(\xi_j(\mathbf{x}))\big) \tag{3.23}$$

where distributional distance is replaced by the pairwise distance between all views of each unperturbed sample $\mathbf{x}$. Such point-wise notion of distance is stronger than distributional distance and it controls the latter. In summary, the above framework allows to connect feature invariance to robustness, via the approximation error. With reference to the regression formulation of SSL (Equation 3.5), the observation allows to set up an experimental framework for investigating factors affecting downstream generalization, namely by explicitly controlling those affecting the approximation error. **Paper E** explores OOD robustness of vision-based SSL encoders, by controlling the encoder dimensionality p, the number of base training samples n, as well as the number of views m of each base training point, and presents an empirical study of OOD generalization for representative contrastive and non-contrastive methods. To empirically explore OOD generalization, a family of SSL encoders are trained on an unlabelled dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$ of size n. After models are pre-trained, performance on supervised In-Distribution downstream

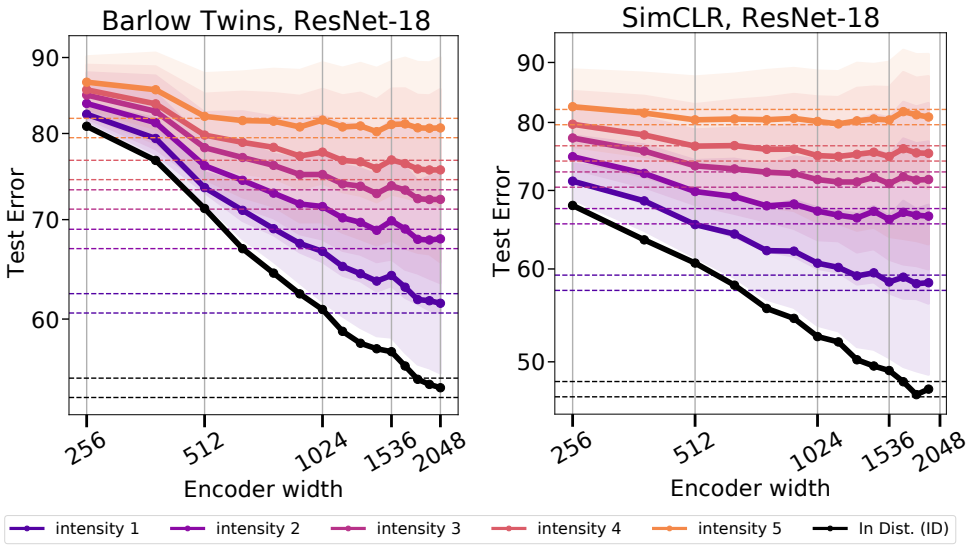


Figure 3.5: ID and OOD generalization under model scaling of ResNet encoders on CIFAR-100, for increasing OOD noise and fixed dataset size. In-Distribution (ID) performance for all models (black curves) monotonically improves with model scale. Strikingly, OOD robustness (averaged across 11 noise distributions and 5 seeds) exhibits the bottleneck behaviour associated with underparameterized learning. Specifically, across noise intensities (solid colored lines), downstream robustness plateaus after a noise-dependent model scale is reached. Horizontal dotted lines, matching the colour of the noise intensity curves, are placed at ±1% of the largest model's performance for that noise intensity, to visually identify the plateau.

tasks is assessed by training a linear probe $\mathbf{g} : \mathbb{R}^p \to \mathbb{R}^k$ on top of the frozen representation, using the labelled version of the data $\mathcal{D}_y = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$. Finally, OOD robustness is quantified by evaluating $\mathbf{g} \circ \mathbf{f}_{\boldsymbol{\theta}}$ using noisy versions of the input data $\tilde{\mathcal{D}}_y = \{(\tilde{\mathbf{x}}_i, y_i)\}_{i=1}^n$, whereby each training point is corrupted with several OOD noise distributions, each with increasing controlled intensity [218].

Contribution: OOD Robustness Under Scaling Laws

The ID performance of state of the art neural networks behaves according to phenomenological scaling laws, whereby when the number of model parameters and dataset size jointly scale, performance improves following a power law [38], [219], [220]. If either dataset size or model scale bottleneck each other, ID performance breaks from the scaling law and plateaus [221]. For large datasets and small models, bottlenecks may occur when the model has exhausted capacity to fit additional data, whereas for large models and small datasets, plateauing is representative of underfitting [38]. In the following, the occurrence of analogous behaviour is estab-

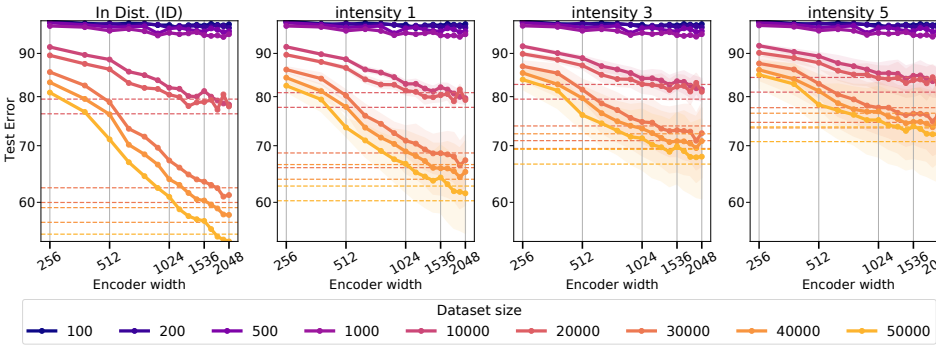


Figure 3.6: ID and OOD performance on CIFAR-100 for non-contrastive Barlow Twins under joint model and dataset scaling. Dotted horizontal lines mark the onset of the plateau in OOD robustness for different dataset sizes. While ID performance consistently improves for increasing model and dataset size, all models plateau on OOD task when trained on larger datasets.

lished for OOD robustness of augmentation-based SSL. Figure 3.5 studies ResNet encoders evaluated on a corrupted version of CIFAR-100 [218], whereby 11 noise distributions are used to perturb the training data. Furthermore, each noise is applied with 5 different intensities, for a total of 11×5 noisy versions of the dataset. The setup allows to study expected OOD robustness (across the noise distributions) of each model, as noise intensity is controlled. For both non-contrastive as well as contrastive learning algorithms, ID performance considerably improves when encoder networks enjoy increased expressivity. In contrast, OOD robustness follows a different scaling law, whereupon low noise intensities (namely, intensity 1) scale similarly to ID performance, while robustness to stronger noise plateaus. The onset of the plateau across model scales is dependent on the noise intensity, with stronger noise causing smaller models to enter the plateau. The observation establishes scaling law behavior for OOD robustness, and opens the question of identifying bottlenecking quantities that may cause the observed behaviour. To do so, **Paper E** studies the effect of dataset size and model scale.

Robustness Under Joint Scaling

Jointly scaling dataset and model size allows to reproduce the bottleneck phenomenon for ID performance on small datasets (Figure 3.6), while larger dataset sizes afford performance improvements across all model scales. Moving to OOD data causes models across all dataset sizes to incur in plateauing, with more pronounced effects for strong noise intensities. By reporting bottlenecks with respect to both model and dataset size, the experiment more clearly establishes scaling-law behaviour for OOD robustness. Importantly, while non-contrastive learning shows improved performance across all dataset scales, the situation is strikingly

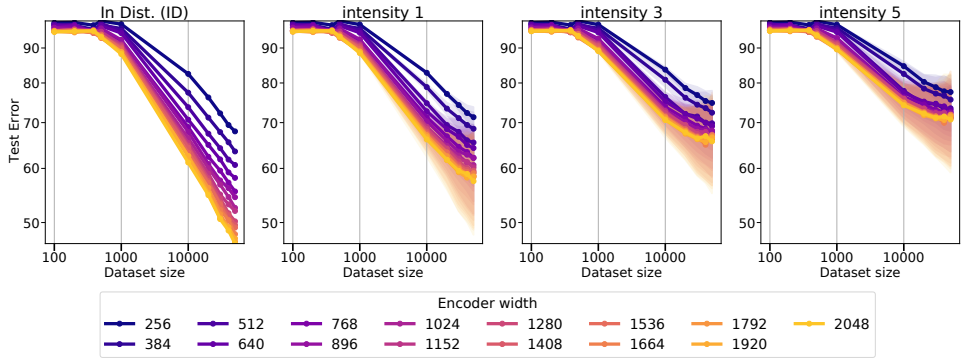


Figure 3.7: Robustness of contrastive learning plateaus in the OOD setting across all model scales, showing a relative loss of robustness for large models.

different for contrastive learning (SimCLR). Figure 3.7 highlights a strong bottleneck behaviour on OOD data for SimCLR across all models for large dataset sizes, suggesting that increasing the number of unperturbed training points fails to counteract the growth of feature variation in the OOD setting for SimCLR. In contrast, for the same datasets and models, ID performance improves without any bottlenecks, excluding a plateau due to ID underfitting.

The observation can be interpreted in terms of approximation error. By increasing the number of unperturbed training points (and correspondingly the augmentation graph $\mathbf{G}$), the quadratic form expressing invariance (Equation 3.2) counts additional eigendirections, than an expressive enough model can recover. On the one hand, if the additional eigendirections overlap with the OOD test set $\mathcal{E}_{\text{all}}$, then one could expect improved robustness to follow. On the other hand, if the additional eigendirections do not contribute to robustness, then an improvement in ID performance may be still observed, without a corresponding improvement in robustness. Building on this intuition, **Paper E** explores how increasing dataset size with unperturbed samples affects downstream robustness, vis-à-vis increasing the number of synthetic samples via data augmentation.

Synthetic versus Natural Data

Up to this point, the pre-training pipelines used $m = 2$ augmentations for each unperturbed sample to construct the augmentation graph at each iteration. **Paper E** concludes by studying the role of data augmentation and dataset size n. For a fixed baseline dataset of 10k unperturbed samples, different encoders are trained with $m = 2$ and $m = 4$ augmentations. Furthermore, the dataset size is doubled to 20k unperturbed samples and $m = 4$ augmentations, as well as 10k samples with 8 augmentations and 40k samples with 2 augmentations. The latter three settings provide a fixed total dataset size of mn, whereby the proportion of natural and aug-

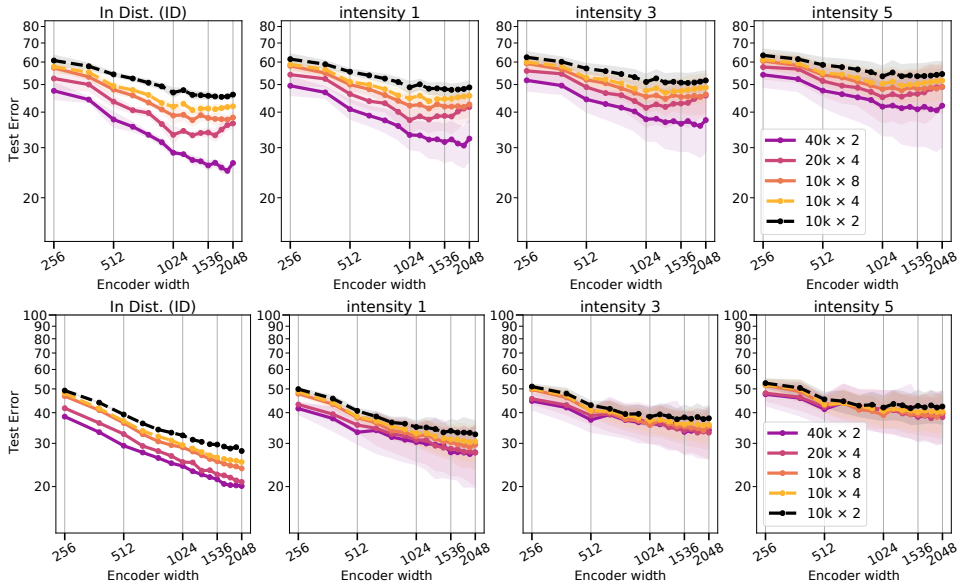


Figure 3.8: Increasing dataset size with natural samples vs augmentations impacts OOD robustness differently. For non-contrastive methods (top row), extending a dataset with natural data provides the strongest improvement in performance, both ID and OOD. In contrast, contrastive learning mostly benefits from novel natural samples for ID performance, while data augmentation and natural data similarly contribute to OOD robustness across all noise intensities.

mented samples varies. Figure 3.8 reports model scaling results for different dataset configurations, for non-contrastive (top row) and contrastive (bottom row) methods. Increasing dataset size for Barlow Twins provides considerable performance gains, both ID and OOD. While large models still exhibit plateauing behaviour for strong noise intensity, all models benefit from accessing additional unperturbed samples, showing considerable improvements compared to settings with more synthetic samples. The same behaviour for ID data can be observed for SimCLR. In the OOD setting however, robustness plateaus across all configurations, showing only marginal benefits of using unperturbed samples vs increasing the number of augmentations.

The experiments hereby presented establish previously unreported scaling-law phenomena for OOD robustness, highlighting the need to isolate and characterize the factors affecting OOD robustness. Importantly, earlier attempts to produce a robustness metric based on the feature covariance Σ failed to generalize to the model scaling setting considered. In line with **Paper D**, the models studied behave consistently with the underparameterized regime of supervised learning, whereby traditional, unnormalized measures of rank (input Jacobian norm as well as effective

rank of features) increase with model scale, due to the larger models attaining improved fitting of the data. Identifying proper normalization strategies remains at the time of writing a direction for future work. Nevertheless, the empirical exploration of **Paper E** establishes strong trends, paving the way for larger scale studies as well as a full theoretical characterization of the plateauing phenomenon for robustness.

Chapter 4

Conclusions and Future Work

The present chapter concludes the thesis, by summarizing the main contributions, as well as discussing limitations and identifying directions for future work.

4.1 Summary of Contributions

Deep networks provide a versatile class of function approximators, able to learn structured representations from natural data embedded in high-dimensional Euclidean spaces. Over thirty years of technical advances have provided large, complex architectures, which strike a balance between expressivity and trainability with first order optimization methods, despite the non-convex nature of parameter estimation. In the pursuit of characterizing the hypothesis class of models that attain good generalization, the thesis presented a study of emerging regularity of the input/output mappings expressed by deep discriminative networks in practice, identifying input smoothness as a key signature of reduced effective complexity in the overparameterized regime. The main focus of **Paper A-C** has been to establish a connection between implicit norm-based capacity control in connection with overparameterization and the ability of neural networks to perfectly fit the training data. Focusing on affine spline operators parameterized by ReLU networks, **Paper A** presented an empirical characterization of the activation regions partition of overparameterized models, identifying local redundancy of activation regions and the corresponding reduced non-linearity as a signature of implicit regularization for trained networks. This was done by introducing an adaptive activation region discovery algorithm, as well as a measure of non-linearity implicitly based on local curvature. **Paper B** extended the study to the loss landscape of neural networks, proposing a connection between local smoothness and curvature of the input space in connection with interpolation of noisy training data, and extrapolation over high-dimensional volumes. The study was carried out by proposing an approximate geodesic Monte Carlo integration method, that allowed to estimate the input-space curvature of the loss landscape in proximity of interpolated training data, as well as

to assess extrapolation away from the data. **Paper C** provided a formalization of the findings of the prior two works, by connecting the proposed input-smoothness metrics to a more principled notion of local input smoothness related to Lipschitz continuity, and then establishing a connection between implicit smoothness regularization and the dynamics of gradient descent in proximity of a stationary point of the loss in parameter space. The construction is based on relating parameter gradients to input-space gradients across the layers of neural networks, establishing a structural relationship between model depth and regularization beyond norms. Additionally, in the context of linear regression, the proposed smoothness metric reverts to the widely studied parameter norm, partly reconciling linear hypothesis spaces with non-linear ones. The smoothness metric has also been connected to effective rank of representations, and has been used in **Paper D** to empirically observe that certain classes of contrastive and non-contrastive Self-Supervised Learning methods behave consistently with the underparameterized regime of supervised learning. The study highlighted a previously unobserved scaling regime for downstream robustness of SSL representations, whereby downstream robustness bottlenecks for large enough models, hinting at a deeper scaling law phenomenon with respect to latent factors indirectly controlled by model scale and dataset size. Building on the connection between invariance-based learning and representation robustness, **Paper E** presented a study of OOD generalization for the linear evaluation setting of augmentation-based SSL methods substantiating and establishing a scaling law behaviour for OOD robustness. In the context of underparameterized learning, whereby interpolation is not attained, representation smoothness remains a relevant metric of the quality of representations. However, its direct interpretation is harder due to the lack of normalization, which would in principle allow to distinguishing between "undertraining" and the preference of a model to learn simple functions.

4.2 Limitations

The main focus of **Papers A-C** was to propose scalable metrics for tracking emerging regularity in overparameterized learning. At the time of publication, **Paper A** provided the first large-scale study of activation regions for ResNets. By focusing on practical training settings, **Papers A-B** provide post-hoc measures that correlate with the generalization ability of deep networks, *assuming the model is trained until interpolation*, and falling short of providing generalization bounds based on the proposed metrics. Additionally, while the proposed metrics are able to track double descent for the test error, **Paper A** and **C** are unable to distinguish between underfitting and overfitting in the underparameterized regime. While this is a common limitation of several post-hoc metrics, which are meant for relative model comparisons [222], generalization measures should in principle provide intrinsic notions of regularity, without relying on extrinsic information (e.g. knowing the relative performance of a different instantiation of the model). While **Paper B**

was not directly used for model selection, characterizing local sharpness of the loss landscape allows to identify "easy" and "hard" examples, for instance originating from mislabelled samples. Indeed, a subsequent work to **Paper B** uses local curvature information to detect memorized samples [171], substantiating the results of **Paper B**. As previously highlighted, a further limitation of smoothness-based metrics is that they scale proportionally to the (effective) rank of the model function, becoming harder to interpret in the underfitting regime. A solution to the problem is to identify proper normalization of the metrics, in order to compare different models on a unified scale. In this regard, spectrally normalized notions of margins [174] may provide inspiration for constructing complexity metrics that are more directly applicable to the underparameterized regime. A further limitation of local analysis of hypotheses expressed by deep networks is that focusing on a single parameterization of the model function fails to account for the many hidden symmetries characterizing deep learning [223]. Particularly, the connection between input-space and parameter space gradients could be extended to account for classes of equivalent functions, and the corresponding orbits they form in parameter space. Finally, a major challenge of understanding and interpreting neural networks lies in characterizing the alignment between the learned model functions and the data distribution, both in terms of learning invariance useful for classification, as well as in terms of understanding the mechanisms underpinning feature learning in the finite-width regime in relation to the model architecture. While the study of the problem beyond linear models is still in its infancy [59], [224], understanding the input-feature alignment may open the door to deeper studies of representation learning, allowing to design optimal augmentation strategies, as well as learning objectives for faster and more robust training. In this regard, a limitation of **Papers D-E** is that they only indirectly track feature alignment between augmented ID data and OOD data, as well as any implicit bias of the model architecture in learning specific eigendirections.

4.3 Future Work

Several exciting directions for future work lay ahead. In the context of supervised learning, the connection between parameter gradients and input gradients studied in **Paper C** could be extended beyond convergence, casting parameter space trajectories into equivalent trajectories in the model function output space through a local linear approximation of the function. Particularly, such a study could potentially allow to connect parameter space phenomena, such as linear mode connectivity [188] and layer-wise mode connectivity [71] to related structures in input space. Furthermore, implicit smoothness regularization could be connected to the emergence of delayed robustness [70] in ReLU networks, whereby models gain adversarial robustness after prolonged training past the point of interpolation. The phenomenon strikes resemblance with neural collapse [75], whereby smoothness of the model function emerges from a collapse of within-class variation of the training

data. Importantly, Humayun, Balestriero, and Baraniuk [70] observe a simplification of the activation regions partition in the late stage of training, suggesting a promising connection with the findings of **Paper A**, as well as to the input-gradient and parameter-gradient connection studied in **Paper C**. Particularly, research in this direction could establish a connection between effective rank of the model function and its smoothness, providing a connection to the structure of robust learners beyond metric-based properties (e.g. Lipschitz continuity). In the context of SSL, empirically characterizing the manifold of learned features may allow to establish a deeper connection to downstream robustness, and particularly a statistical OOD detection test, whereby candidate points could be rejected based on how far they lie from the feature manifold. Importantly, the measure itself might allow to gain deeper insights into the nature of the plateauing behaviour observed in **Paper D-E**, potentially allowing to identify directions in feature space that should be covered by the feature manifold, and even select augmentation policies based on manifold coverage (based on external validation sets). Such measure could also allow to differentiate between the robustness behaviour of non-contrastive and contrastive learning observed in **Paper E**.

Chapter 5

Summary of Included Papers

This chapter contains abstracts of the included papers and contributions by the author of the thesis.

Paper A

Are all Linear Regions Created Equal?

M. Gamba, A. Chmielewski-Anders, J. Sullivan, H. Azizpour, M. Björkman.
In *International Conference on Artificial Intelligence and Statistics (AISTATS)*
(pp. 6573-6590). PMLR. 2022.

Abstract: The number of linear regions has been studied as a proxy of complexity for ReLU networks. However, the empirical success of network compression techniques like pruning and knowledge distillation, suggest that in the overparameterized setting, linear regions density might fail to capture the effective nonlinearity. In this work, we propose an efficient algorithm for discovering linear regions and use it to investigate the effectiveness of density in capturing the nonlinearity of trained VGGs and ResNets on CIFAR-10 and CIFAR-100. We contrast the results with a more principled nonlinearity measure based on function variation, highlighting the shortcomings of linear regions density. Furthermore, interestingly, our measure of nonlinearity clearly correlates with model-wise deep double descent, connecting reduced test error with reduced nonlinearity, and increased local similarity of linear regions.

Contributions by the author: Proposed the non-linearity measure as well as the experimental methodology of the paper. Extended a proof-of-concept implementation of the activation region discovery algorithm to handle large-scale neural networks. Executed all experiments and wrote the paper (excluding the illustration figure).

Paper B

Deep Double Descent via Smooth Interpolation

M. Gamba, E. Englesson, M. Björkman, H. Azizpour.
In *Transactions on Machine Learning Research*. 2023.

Abstract: The ability of overparameterized deep networks to interpolate noisy data, while at the same time showing good generalization performance, has been recently characterized in terms of the double descent curve for the test error. Common intuition from polynomial regression suggests that overparameterized networks are able to sharply interpolate noisy data, without considerably deviating from the ground-truth signal, thus preserving generalization ability. At present, a precise characterization of the relationship between interpolation and generalization for deep networks is missing. In this work, we quantify sharpness of fit of the training data interpolated by neural network functions, by studying the loss landscape w.r.t. to the input variable locally to each training point, over volumes around cleanly- and noisily-labelled training samples, as we systematically increase the number of model parameters and training epochs. Our findings show that loss sharpness in the input space follows both model- and epoch-wise double descent, with worse peaks observed around noisy labels. While small interpolating models sharply fit both clean and noisy data, large interpolating models express a smooth loss landscape, where noisy targets are predicted over large volumes around training data points, in contrast to existing intuition.
Contributions by the author: Proposed the methodology and its mathematical formulation. Implemented the methodology and the majority of the experiments. Carried out all experiments, wrote the majority of the paper (excluding parts of the related works section, and the illustration figure).

Paper C

On the Lipschitz Constant of Deep Networks and Double Descent

M. Gamba, H. Azizpour, M. Björkman.
In *34th British Machine Vision Conference (BMVC)*. 2023.

Abstract: Existing bounds on the generalization error of deep networks assume some form of smooth or bounded dependence on the input variable, falling short of investigating the mechanisms controlling such factors in practice. In this work, we present an extensive experimental study of the empirical Lipschitz constant of deep networks undergoing double descent, and highlight non-monotonic trends strongly correlating with the test error. Building a connection between parameter-space and input-space gradients for SGD around a critical point, we isolate two important factors – namely loss landscape curvature and distance of parameters from initialization – respectively controlling optimization dynamics around a critical point and bounding model function complexity, even beyond the training data. Our study presents novels insights on implicit regularization via overparameterization, and effective model complexity for networks trained in practice.
Contributions by the author: Proposed and designed methodology, carried out all experiments and wrote the paper.

Paper D

Different Faces of Model Scaling in Supervised and Self-Supervised Learning

M. Gamba, A. Ghosh, K. K. Agrawal, B. A. Richards, H. Azizpour, M. Björkman.
In *ICLR Workshop on Bridging the Gap Between Theory and Practice*. 2024

Abstract: The quality of the representations learned by neural networks depends on several factors, including the loss function, learning algorithm, and model architecture. In this work, we use information geometric measures to assess the representation quality in a principled manner. We demonstrate that the sensitivity of learned representations to input perturbations, measured by the spectral norm of the feature Jacobian, provides valuable information about downstream generalization. On the other hand, measuring the coefficient of spectral decay observed in the eigenspectrum of feature covariance provides insights into the global representation geometry. First, we empirically establish an equivalence between these notions of representation quality and show that they are inversely correlated. Second, our analysis reveals varying roles of scaling model size in improving generalization. Increasing model width leads to higher discriminability and relatively reduced smoothness in the self-supervised regime, compatibly with the underparameterized regime of supervised learning. Interestingly, we report no observable double descent phenomenon in SSL with non-contrastive objectives for commonly used parameterization regimes, which opens up new opportunities for tight asymptotic analysis. Taken together, our results provide a loss-aware characterization of the different role of model scaling in supervised and self-supervised learning.

Contributions by the author: Provided the mathematical formalism, wrote the majority of the paper (excluding parts of the experiments section), carried out all experiments (excluding the data poisoning experiment).

Paper E

When Does Self-Supervised Pre-Training Yield Robust Representations?

M. Gamba, K. K. Agrawal, A. Ghosh, B. A. Richards, H. Azizpour, M. Björkman.
Preprint. 2024

Abstract: Self-Supervised Learning (SSL) provides a powerful class of learning algorithms for extracting representations of unlabelled data. A common learning paradigm relies on generating multiple views of the training data by perturbing inputs with data augmentation, effectively enforcing the representation to attain invariance to certain input perturbations. While encoding invariance in this way has been shown to reliably improve downstream performance, its impact on Out of Distribution (OOD) generalization is underexplored. In particular, invariance-based learning objectives enforce low feature variation under selected input perturbations, which is a fundamental desideratum when dealing with downstream distribution shifts. Building on this connection, this work explores OOD robustness of SSL representations when data is corrupted with noise of increasing intensity, under different model scales and dataset sizes. Strikingly, our experiments suggest that, for fixed training set, increasing encoder capacity consistently improves in-distribution performance, whereas OOD robustness plateaus.Furthermore, increasing training set size either virtually (via data augmentation) or by increasing the number of unperturbed samples improves OOD robustness across all model scales, delaying the onset of the plateau. While increasing dataset size with unperturbed samples consistently improves downstream performance as well as robustness, data augmentation in the low-samples regime offers a strong alternative when acquiring unperturbed data is impractical.

Contributions by the author: Provided the mathematical formalism, wrote the majority of the paper, carried out all experiments by extending the codebase of A. G. and K. K. A.

References

[1] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[2] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations", *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale", in *International Conference on Learning Representations*, 2020.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[6] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks", in *International Conference on Learning Representations*, 2016.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[8] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling", in *The Eleventh International Conference on Learning Representations*, 2023.

[9] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models", *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[10] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution", *Advances in neural information processing systems*, vol. 32, 2019.

[11] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training", *Technical report*, 2018.

[12] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions", *Advances in neural information processing systems*, vol. 31, 2018.

[13] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics", in *International conference on machine learning*, PMLR, 2015, pp. 2256–2265.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets", *Advances in neural information processing systems*, vol. 27, 2014.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes", in *International Conference on Learning Representations*, 2013.

[16] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models", in *International Conference on Learning Representations*, 2020.

[17] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks", in *International conference on machine learning*, PMLR, 2017, pp. 1126–1135.

[18] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies", *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.

[19] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning", in *International conference on machine learning*, PMLR, 2016, pp. 1995–2003.

[20] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization", in *International conference on machine learning*, PMLR, 2015, pp. 1889–1897.

[21] X. Guo, S. Singh, H. Lee, R. L. Lewis, and X. Wang, "Deep learning for real-time atari game play using offline monte-carlo tree search planning", *Advances in neural information processing systems*, vol. 27, 2014.

[22] S. Arora and A. Goyal, "A theory for emergence of complex skills in language models", *arXiv preprint arXiv:2307.15936*, 2023.

[23] J. Lee, S. Schoenholz, J. Pennington, B. Adlam, L. Xiao, R. Novak, and J. Sohl-Dickstein, "Finite versus infinite neural networks: An empirical study", *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 156–15 172, 2020.

[24] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro, "The role of over-parametrization in generalization of neural networks", in *International Conference on Learning Representations*, 2018.

[25] B. Neyshabur, R. Tomioka, and N. Srebro, "In search of the real inductive bias: On the role of implicit regularization in deep learning", in *International Conference on Learning Representations Workshop Track*, 2015.

[26] ——, "Norm-based capacity control in neural networks", in *Conference on Learning Theory*, PMLR, 2015, pp. 1376–1401.

[27] D. H. Wolpert, "The supervised learning no-free-lunch theorems", *Soft computing and industry: Recent applications*, pp. 25–42, 2002.

[28] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization", *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[29] T. M. Mitchell, "The need for biases in learning generalizations", 1980.

[30] H. Ye, C. Xie, T. Cai, R. Li, Z. Li, and L. Wang, "Towards a theoretical framework of out-of-distribution generalization", *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 519–23 531, 2021.

[31] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization", in *International Conference on Learning Representations*, 2020.

[32] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples", *arXiv preprint arXiv:1412.6572*, 2014.

[33] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks", *arXiv preprint arXiv:1312.6199*, 2013.

[34] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey", *IEEE transactions on neural networks and learning systems*, vol. 34, no. 11, pp. 8135–8153, 2022.

[35] N. Mallinar, J. Simon, A. Abedsoltan, P. Pandit, M. Belkin, and P. Nakkiran, "Benign, tempered, or catastrophic: Toward a refined taxonomy of overfitting", *Advances in Neural Information Processing Systems*, vol. 35, pp. 1182–1195, 2022.

[36] Y. Geifman and R. El-Yaniv, "Selectivenet: A deep neural network with an integrated reject option", in *International conference on machine learning*, PMLR, 2019, pp. 2151–2159.

[37] M. Toneva, A. Sordoni, R. T. des Combes, A. Trischler, Y. Bengio, and G. J. Gordon, "An empirical study of example forgetting during deep neural network learning", in *International Conference on Learning Representations*, 2018.

[38] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models", *arXiv preprint arXiv:2001.08361*, 2020.

[39] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, *et al.*, "Large scale distributed deep networks", *Advances in neural information processing systems*, vol. 25, 2012.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors", *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[41] R. W. Batterman, "Asymptotics and the role of minimal models", *The British Journal for the Philosophy of Science*, vol. 53, no. 1, pp. 21–38, 2002.

[42] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks", in *International Conference on Learning Representations*, 2018.

[43] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks", *Advances in neural information processing systems*, vol. 31, 2018.

[44] B. Adlam and J. Pennington, "The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization", in *International Conference on Machine Learning*, PMLR, 2020, pp. 74–84.

[45] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, "Wide neural networks of any depth evolve as linear models under gradient descent", *Advances in neural information processing systems*, vol. 32, 2019.

[46] M. Geiger, L. Petrini, and M. Wyart, "Landscape and training regimes in deep learning", *Physics Reports*, vol. 924, pp. 1–18, 2021.

[47] M. Belkin, S. Ma, and S. Mandal, "To understand deep learning we need to understand kernel learning", in *International Conference on Machine Learning*, PMLR, 2018, pp. 541–549.

[48] M. Refinetti, S. Goldt, F. Krzakala, and L. Zdeborová, "Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed", in *International Conference on Machine Learning*, PMLR, 2021, pp. 8936–8947.

[49] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari, "Limitations of lazy training of two-layers neural network", *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[50] Z. Allen-Zhu and Y. Li, "What can resnet learn efficiently, going beyond kernels?", *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[51] J. Pennington and Y. Bahri, "Geometry of neural network loss surfaces via random matrix theory", in *International Conference on Machine Learning*, PMLR, 2017, pp. 2798–2806.

[52] J. Pennington and P. Worah, "The spectrum of the fisher information matrix of a single-hidden-layer neural network", *Advances in neural information processing systems*, vol. 31, 2018.

[53] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices", in *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012, 210–268.

[54] L. Xiao, Y. Bahri, J. Sohl-Dickstein, S. Schoenholz, and J. Pennington, "Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks", in *International Conference on Machine Learning*, PMLR, 2018, pp. 5393–5402.

[55] C. H. Martin and M. W. Mahoney, "Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning", *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 7479–7551, 2021.

[56] C. H. Martin, T. Peng, and M. W. Mahoney, "Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data", *Nature Communications*, vol. 12, no. 1, p. 4122, 2021.

[57] G. Yang and E. J. Hu, "Feature learning in infinite-width neural networks", *arXiv preprint arXiv:2011.14522*, 2020.

[58] S. Mei, A. Montanari, and P.-M. Nguyen, "A mean field view of the landscape of two-layer neural networks", *Proceedings of the National Academy of Sciences*, vol. 115, no. 33, E7665–E7671, 2018.

[59] E. Nichani, A. Damian, and J. D. Lee, "Provable guarantees for nonlinear feature learning in three-layer neural networks", *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[60] R. Balestriero and Y. LeCun, "How learning by reconstruction produces uninformative features for perception", in *Forty-first International Conference on Machine Learning*, 2024.

[61] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5188–5196.

[62] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[63] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks", in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, Springer, 2014, pp. 818–833.

[64] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network", *University of Montreal*, vol. 1341, no. 3, p. 1, 2009.

[65] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[66] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers", in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.

[67] L. Zhu, C. Liu, A. Radhakrishnan, and M. Belkin, "Catapults in sgd: Spikes in the training loss and their impact on generalization through feature learning", in *Forty-first International Conference on Machine Learning*, 2024.

[68] E. Michaud, Z. Liu, U. Girit, and M. Tegmark, "The quantization model of neural scaling", *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[69] A. Lewkowycz, Y. Bahri, E. Dyer, J. Sohl-Dickstein, and G. Gur-Ari, "The large learning rate phase of deep learning: The catapult mechanism", *arXiv preprint arXiv:2003.02218*, 2020.

[70] A. I. Humayun, R. Balestriero, and R. Baraniuk, "Deep networks always grok and here is why", in *Forty-first International Conference on Machine Learning*, 2024.

[71] Z. Zhou, Y. Yang, X. Yang, J. Yan, and W. Hu, "Going beyond linear mode connectivity: The layerwise linear feature connectivity", *Advances in Neural Information Processing Systems*, vol. 36, pp. 60 853–60 877, 2023.

[72] A. Power, Y. Burda, H. Edwards, I. Babuschkin, and V. Misra, "Grokking: Generalization beyond overfitting on small algorithmic datasets", *arXiv preprint arXiv:2201.02177*, 2022.

[73] J. Cohen, S. Kaur, Y. Li, J. Z. Kolter, and A. Talwalkar, "Gradient descent on neural networks typically occurs at the edge of stability", in *International Conference on Learning Representations*, 2021.

[74] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin, "Linear mode connectivity and the lottery ticket hypothesis", in *International Conference on Machine Learning*, PMLR, 2020, pp. 3259–3269.

[75] V. Papyan, X. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training", *Proceedings of the National Academy of Sciences*, vol. 117, no. 40, pp. 24 652–24 663, 2020.

[76] B. Hanin and D. Rolnick, "Deep relu networks have surprisingly few activation patterns", in *Advances in Neural Information Processing Systems*, 2019, pp. 359–368.

[77] C. Zhang, S. Bengio, and Y. Singer, "Are all layers created equal?", *ICML Workshop Deep Phenomena*, 2019.

[78] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization", *International Conference on Learning Representations*, 2018.

[79] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world", in *Artificial intelligence safety and security*, Chapman and Hall/CRC, 2018, pp. 99–112.

[80] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[81] V. Nagarajan and J. Z. Kolter, "Uniform convergence may be unable to explain generalization in deep learning", *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[82] S. Ainsworth, J. Hayase, and S. Srinivasa, "Git re-basin: Merging models modulo permutation symmetries", in *The Eleventh International Conference on Learning Representations*, 2023.

[83] M. Armenta, T. Judge, N. Painchaud, Y. Skandarani, C. Lemaire, G. Gibeau Sanchez, P. Spino, and P.-M. Jodoin, "Neural teleportation", *Mathematics*, vol. 11, no. 2, p. 480, 2023.

[84] B. Zhao, N. Dehmamy, R. Walters, and R. Yu, "Symmetry teleportation for accelerated optimization", *Advances in neural information processing systems*, vol. 35, pp. 16 679–16 690, 2022.

[85] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, *et al.*, "Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time", in *International conference on machine learning*, PMLR, 2022, pp. 23 965–23 998.

[86] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks", in *International Conference on Learning Representations*, 2018.

[87] M. Belkin, D. J. Hsu, and P. Mitra, "Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate", in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018.

[88] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, *et al.*, "A closer look at memorization in deep networks", in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 233–242.

[89] V. Vapnik, "On the uniform convergence of relative frequencies of events to their probabilities", in *Doklady Akademii Nauk USSR*, vol. 181, 1968, pp. 781–787.

[90] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results", *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 463–482, 2002.

[91] M. Telgarsky, "Benefits of depth in neural networks", in *29th Annual Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, vol. 49, Columbia University, New York, USA: PMLR, 2016, pp. 1517–1539.

[92] G. Urban, K. J. Geras, S. E. Kahou, O. Aslan, S. Wang, A. Mohamed, M. Philipose, M. Richardson, and R. Caruana, "Do deep convolutional nets really need to be deep and convolutional?", in *International Conference on Learning Representations*, 2022.

[93] N. Chatterji, B. Neyshabur, and H. Sedghi, "The intriguing role of module criticality in the generalization of deep networks", in *International Conference on Learning Representations*, 2020.

[94] S. I. Mirzadeh, M. Farajtabar, D. Gorur, R. Pascanu, and H. Ghasemzadeh, "Linear mode connectivity in multitask and continual learning", in *International Conference on Learning Representations*, 2022.

[95] R. Entezari, H. Sedghi, O. Saukh, and B. Neyshabur, "The role of permutation invariance in linear mode connectivity of neural networks", in *International Conference on Learning Representations*, 2021.

[96] Z. Liu, E. J. Michaud, and M. Tegmark, "Omnigrok: Grokking beyond algorithmic data", in *The Eleventh International Conference on Learning Representations*, 2022.

[97] H. Mehta, A. Cutkosky, and B. Neyshabur, "Extreme memorization via scale of initialization", in *International Conference on Learning Representations*, 2020.

[98] T. Kumar, B. Bordelon, S. J. Gershman, and C. Pehlevan, "Grokking as the transition from lazy to rich training dynamics", in *The Twelfth International Conference on Learning Representations*, 2024.

[99] R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, "Sensitivity and generalization in neural networks: An empirical study", in *International Conference on Learning Representations*, 2018.

[100] H. Sedghi, S. Bengio, K. Hata, A. Madry, A. Morcos, B. Neyshabur, M. Raghu, A. Rahimi, L. Schmidt, and Y. Xiao, "Identifying and understanding deep learning phenomena", in *ICML 2019 Workshop*, 2019.

[101] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey", *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 4037–4058, 2020.

[102] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning", *Technologies*, vol. 9, no. 1, p. 2, 2020.

[103] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods", in *33rd annual meeting of the association for computational linguistics*, 1995, pp. 189–196.

[104] S. Abbasi Koohpayegani, A. Tejankar, and H. Pirsiavash, "Compress: Self-supervised learning by compressing representations", *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 980–12 992, 2020.

[105] I. Safran and O. Shamir, "Spurious local minima are common in two-layer ReLU neural networks", in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 4433–4441.

[106] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey, "Three factors influencing minima in sgd", *arXiv preprint arXiv:1711.04623*, 2017.

[107] G. Swirszcz, W. M. Czarnecki, and R. Pascanu, "Local minima in training of neural networks", *arXiv preprint arXiv:1611.06310*, 2016.

[108] H. He, G. Huang, and Y. Yuan, "Asymmetric valleys: Beyond sharp and flat local minima", *Advances in neural information processing systems*, vol. 32, 2019.

[109] J. Hadamard, "Sur les problèmes aux dérivées partielles et leur signification physique", *Princeton university bulletin*, pp. 49–52, 1902.

[110] K. Kawaguchi, J. Huang, and L. P. Kaelbling, "Effect of depth and width on local minima in deep learning", *Neural computation*, vol. 31, no. 7, pp. 1462–1498, 2019.

[111] K. Kawaguchi and Y. Bengio, "Depth with nonlinearity creates no bad local minima in resnets", *Neural Networks*, vol. 118, pp. 167–174, 2019.

[112] A. N. Tikhonov and V. Arsenin, "Solutions of ill-posed problems", 1977.

[113] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[114] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks", in *International conference on machine learning*, PMLR, 2013, pp. 1310–1318.

[115] A. E. Hoerl and R. W. Kennard, "Ridge regression: Applications to nonorthogonal problems", *Technometrics*, vol. 12, no. 1, pp. 69–82, 1970.

[116] P.-y. Chiang, R. Ni, D. Y. Miller, A. Bansal, J. Geiping, M. Goldblum, and T. Goldstein, "Loss landscapes are all you need: Neural network generalization can be explained without the implicit bias of gradient descent", in *The Eleventh International Conference on Learning Representations*, 2022.

[117] N. Razin and N. Cohen, "Implicit regularization in deep learning may not be explainable by norms", *Advances in neural information processing systems*, vol. 33, pp. 21 174–21 187, 2020.

[118] S. Gunasekar, B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro, "Implicit regularization in matrix factorization", *Advances in neural information processing systems*, vol. 30, 2017.

[119] A. Ali, J. Z. Kolter, and R. J. Tibshirani, "A continuous-time view of early stopping for least squares regression", in *The 22nd international conference on artificial intelligence and statistics*, PMLR, 2019, pp. 1370–1378.

[120] E. Abbe, E. B. Adsera, and T. Misiakiewicz, "The merged-staircase property: A necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks", in *Conference on Learning Theory*, PMLR, 2022, pp. 4782–4887.

[121] E. Abbe, E. Boix-Adsera, M. S. Brennan, G. Bresler, and D. Nagaraj, "The staircase property: How hierarchical structure can guide deep learning", *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 989–27 002, 2021.

[122] H. W. Lin, M. Tegmark, and D. Rolnick, "Why does deep and cheap learning work so well?", *Journal of Statistical Physics*, vol. 168, pp. 1223–1247, 2017.

[123] D. Barrett and B. Dherin, "Implicit gradient regularization", in *International Conference on Learning Representations*, 2021.

[124] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[125] S. Geman, E. Bienenstock, and R. Doursat, "Neural Networks and the Bias/Variance Dilemma", *Neural Computation*, vol. 4, no. 1, pp. 1–58, Jan. 1992, ISSN: 0899-7667. DOI: 10.1162/neco.1992.4.1.1.

[126] Z. Li, T. Wang, and S. Arora, "What happens after sgd reaches zero loss?–a mathematical framework", in *International Conference on Learning Representations*, 2021.

[127] D. J. MacKay, "Bayesian interpolation", *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[128] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro, "The implicit bias of gradient descent on separable data", *Journal of Machine Learning Research*, vol. 19, no. 70, pp. 1–57, 2018.

[129] I. Kuzborskij, C. Szepesvári, O. Rivasplata, A. Rannen-Triki, and R. Pascanu, "On the role of optimization in double descent: A least squares study", *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 567–29 577, 2021.

[130] L. Chen, Y. Min, M. Belkin, and A. Karbasi, "Multiple descent: Design your own generalization curve", *Advances in Neural Information Processing Systems*, vol. 34, pp. 8898–8912, 2021.

[131] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off", *Proceedings of the National Academy of Sciences*, vol. 116, no. 32, pp. 15 849–15 854, 2019.

[132] V. Muthukumar, K. Vodrahalli, V. Subramanian, and A. Sahai, "Harmless interpolation of noisy data in regression", *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 67–83, 2020.

[133] P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler, "Benign overfitting in linear regression", *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30 063–30 070, 2020. DOI: 10.1073/pnas.1907378117.

[134] M. Belkin, D. Hsu, and J. Xu, "Two models of double descent for weak features", *SIAM Journal on Mathematics of Data Science*, vol. 2, no. 4, pp. 1167–1180, 2020.

[135] T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani, "Surprises in high-dimensional ridgeless least squares interpolation", *Annals of statistics*, vol. 50, no. 2, p. 949, 2022.

[136] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep double descent: Where bigger models and more data hurt", in *International Conference on Learning Representations*, 2019.

[137] S. Mei and A. Montanari, "The generalization error of random features regression: Precise asymptotics and the double descent curve", *Communications on Pure and Applied Mathematics*, vol. 75, no. 4, pp. 667–766, 2022.

[138] J. B. Simon, D. Karkada, N. Ghosh, and M. Belkin, "More is better: When infinite overparameterization is optimal and overfitting is obligatory", in *The Twelfth International Conference on Learning Representations*, 2024.

[139] C. Runge *et al.*, "Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten", *Zeitschrift für Mathematik und Physik*, vol. 46, no. 224-243, p. 20, 1901.

[140] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines", in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[141] R. Balestriero and R. G. Baraniuk, "A spline theory of deep learning", in *International Conference on Machine Learning*, PMLR, 2018, pp. 374–383.

[142] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks", in *Advances in Neural Information Processing Systems*, 2014, pp. 2924–2932.

[143] R. Pascanu, G. F. Montufar, and Y. Bengio, "On the number of inference regions of deep feed forward networks with piece-wise linear activations", 2013.

[144] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, "On the expressive power of deep neural networks", in *International Conference on Machine Learning*, 2017, pp. 2847–2854.

[145] X. Zhang and D. Wu, "Empirical studies on the properties of linear regions in deep neural networks", in *International Conference on Learning Representations*, 2020.

[146] T. Serra, C. Tjandraatmadja, and S. Ramalingam, "Bounding and counting linear regions of deep neural networks", in *International Conference on Machine Learning*, 2018, pp. 4558–4566.

[147] R. P. Stanley *et al.*, "An introduction to hyperplane arrangements", *Geometric combinatorics*, vol. 13, no. 389-496, p. 24, 2004.

[148] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.

[149] P. Kidger and T. Lyons, "Universal Approximation with Deep Narrow Networks", in *Proceedings of Thirty Third Conference on Learning Theory*, J. Abernethy and S. Agarwal, Eds., ser. Proceedings of Machine Learning Research, vol. 125, PMLR, 2020, pp. 2306–2327.

[150] B. Hanin, "Universal function approximation by deep neural nets with bounded width and relu activations", *Mathematics*, vol. 7, no. 10, p. 992, 2019.

[151] H. Lin and S. Jegelka, "Resnet with one-neuron hidden layers is a universal approximator", *Advances in neural information processing systems*, vol. 31, 2018.

[152] R. Eldan and O. Shamir, "The power of depth for feedforward neural networks", in *Conference on learning theory*, PMLR, 2016, pp. 907–940.

[153] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units", in *International Conference on Learning Representations*, 2018.

[154] H. Xiong, L. Huang, M. Yu, L. Liu, F. Zhu, and L. Shao, "On the number of linear regions of convolutional neural networks", in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 10 514–10 523.

[155] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey", *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[156] Y. He and L. Xiao, "Structured pruning for deep convolutional neural networks: A survey", *IEEE transactions on pattern analysis and machine intelligence*, 2023.

[157] B. Hanin and D. Rolnick, "Complexity of linear regions in deep networks", in *International Conference on Machine Learning*, 2019, pp. 2596–2604.

[158] A. I. Humayun, R. Balestriero, G. Balakrishnan, and R. G. Baraniuk, "Splinecam: Exact visualization and characterization of deep network geometry and decision boundaries", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3789–3798.

[159] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images", 2009.

[160] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", in *International Conference on Learning Representations*, 2015.

[161] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013. DOI: 10.1109/TPAMI.2013.50.

[162] H. Narayanan and S. Mitter, "Sample complexity of testing the manifold hypothesis", *Advances in neural information processing systems*, vol. 23, 2010.

[163] E. Williams, A. H.-W. Ryoo, T. Jiralerspong, A. Payeur, M. G. Perich, L. Mazzucatto, and G. Lajoie, "Expressivity of neural networks with random weights and learned biases", in *High-dimensional Learning Dynamics 2024: The Emergence of Structure and Reasoning*, 2024.

[164] M. H. Stone, "The generalized weierstrass approximation theorem", *Mathematics Magazine*, vol. 21, no. 5, pp. 237–254, 1948.

[165] E. W. Cheney and W. A. Light, *A course in approximation theory*. American Mathematical Soc., 2009, vol. 101.

[166] P. J. Green and B. W. Silverman, *Nonparametric regression and generalized linear models: a roughness penalty approach*. Crc Press, 1993.

[167] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data", *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, 2003. DOI: 10.1073/pnas.1031596100.

[168] D. LeJeune, R. Balestriero, H. Javadi, and R. G. Baraniuk, "Implicit rugosity regularization via data augmentation", *arXiv preprint arXiv:1905.11639*, 2019.

[169] T. Yu, H. Long, and J. E. Hopcroft, "Curvature-based comparison of two neural networks", in *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, 2018, pp. 441–447.

[170] G. Hacohen, L. Choshen, and D. Weinshall, "Let's agree to agree: Neural networks share classification order on real datasets", in *International Conference on Machine Learning*, PMLR, 2020, pp. 3950–3960.

[171] I. Garg, D. Ravikumar, and K. Roy, "Memorization through the lens of curvature of loss function around samples", in *Forty-first International Conference on Machine Learning*, 2024.

[172] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: Analysis and efficient estimation", *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[173] V. Nagarajan and Z. Kolter, "Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience", in *International Conference on Learning Representations*, 2018.

[174] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks", *Advances in neural information processing systems*, vol. 30, 2017.

[175] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning", *Advances in neural information processing systems*, vol. 30, 2017.

[176] G. W. Brier, "Verification of forecasts expressed in terms of probability", *Monthly Weather Review*, vol. 78, no. 1, pp. 1–3, 1950.

[177] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks", in *International conference on machine learning*, PMLR, 2017, pp. 1321–1330.

[178] C. Ma and L. Ying, "On linear stability of sgd and input-smoothness of neural networks", *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 805–16 817, 2021.

[179] Y. Hosoe and T. Hagiwara, "On second-moment stability of discrete-time linear systems with general stochastic dynamics", *IEEE Transactions on Automatic Control*, vol. 67, no. 2, pp. 795–809, 2022. DOI: 10.1109/TAC.2021.3057994.

[180] L. Wu and C. Ma, "How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective", *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[181] S. P. Singh, A. Lucchi, T. Hofmann, and B. Schölkopf, "Phenomenology of double descent in finite-width neural networks", in *International Conference on Learning Representations*, 2022.

[182] T. Mori, L. Ziyin, K. Liu, and M. Ueda, "Power-law escape rate of sgd", in *International Conference on Machine Learning*, PMLR, 2022, pp. 15 959–15 975.

[183] K. Liu, L. Ziyin, and M. Ueda, "Noise and fluctuation of finite learning rate stochastic gradient descent", in *International Conference on Machine Learning*, PMLR, 2021, pp. 7045–7056.

[184] B. Dherin, M. Munn, M. Rosca, and D. Barrett, "Why neural networks find simple solutions: The many regularizers of geometric complexity", *Advances in Neural Information Processing Systems*, vol. 35, pp. 2333–2349, 2022.

[185] S. Arora, N. Cohen, and E. Hazan, "On the optimization of deep networks: Implicit acceleration by overparameterization", in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 244–253.

[186] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent", in *International conference on machine learning*, PMLR, 2016, pp. 1225–1234.

[187] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[188] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of dnns", *Advances in neural information processing systems*, vol. 31, 2018.

[189] L. Hodgkinson, C. van der Heide, R. Salomone, F. Roosta, and M. W. Mahoney, "The interpolating information criterion for overparameterized models", *arXiv preprint arXiv:2307.07785*, 2023.

[190] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data", in *Proceedings of the 24th International Conference on Machine Learning*, New York, 2007, 759–766.

[191] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao, "A survey on self-supervised learning: Algorithms, applications, and future trends", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[192] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models", *arXiv preprint arXiv:2302.13971*, 2023.

[193] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision", in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[194] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 619–15 629.

[195] T. Yerxa, Y. Kuang, E. Simoncelli, and S. Chung, "Learning efficient coding of natural images with maximum manifold capacity representations", *Advances in Neural Information Processing Systems*, vol. 36, pp. 24 103–24 128, 2023.

[196] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "Simclr: A simple framework for contrastive learning of visual representations", in *International Conference on Learning Representations*, vol. 2, 2020.

[197] A. Bardes, J. Ponce, and Y. LeCun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning", *arXiv preprint arXiv:2105.04906*, 2021.

[198] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction", in *International Conference on Machine Learning*, PMLR, 2021, pp. 12 310–12 320.

[199] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, *et al.*, "Bootstrap your own latent - a new approach to self-supervised learning", *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[200] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[201] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 113–123.

[202] J. Z. HaoChen, C. Wei, A. Gaidon, and T. Ma, "Provable guarantees for self-supervised deep learning with spectral contrastive loss", *Advances in Neural Information Processing Systems*, vol. 34, pp. 5000–5011, 2021.

[203] V. Cabannes, B. Kiani, R. Balestriero, Y. LeCun, and A. Bietti, "The ssl interplay: Augmentations, inductive bias, and generalization", in *International Conference on Machine Learning*, PMLR, 2023, pp. 3252–3298.

[204] R. Balestriero and Y. LeCun, "Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods", *Advances in Neural Information Processing Systems*, vol. 35, pp. 26 671–26 685, 2022.

[205] R. Zhai, B. Liu, A. Risteski, J. Z. Kolter, and P. K. Ravikumar, "Understanding augmentation-based self-supervised representation learning via rkhs approximation and regression", in *The Twelfth International Conference on Learning Representations*, 2024.

[206] J. B. Simon, M. Knutins, L. Ziyin, D. Geisz, A. J. Fetterman, and J. Albrecht, "On the stepwise nature of self-supervised learning", in *International Conference on Machine Learning*, PMLR, 2023, pp. 31 852–31 876.

[207] O. Roy and M. Vetterli, "The effective rank: A measure of effective dimensionality", in *2007 15th European signal processing conference*, IEEE, 2007, pp. 606–610.

[208] K. K. Agrawal, A. K. Mondal, A. Ghosh, and B. Richards, "α-req: Assessing representation quality in self-supervised learning by measuring eigenspectrum decay", *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 626–17 638, 2022.

[209] B. He and M. Ozay, "Exploring the gap between collapsed & whitened features in self-supervised learning", in *International Conference on Machine Learning*, PMLR, 2022, pp. 8613–8634.

[210] Q. Garrido, R. Balestriero, L. Najman, and Y. Lecun, "Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank", in *International Conference on Machine Learning*, PMLR, 2023, pp. 10 929–10 974.

[211] L. Jing, P. Vincent, Y. LeCun, and Y. Tian, "Understanding dimensional collapse in contrastive self-supervised learning", in *International Conference on Learning Representations*, 2022.

[212] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition", *IEEE transactions on electronic computers*, no. 3, pp. 326–334, 1965.

[213] S. Bubeck and M. Sellke, "A universal law of robustness via isoperimetry", *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[214] L. Fowl, M. Goldblum, P.-y. Chiang, J. Geiping, W. Czaja, and T. Goldstein, "Adversarial examples make strong poisons", *Advances in Neural Information Processing Systems*, vol. 34, pp. 30 339–30 351, 2021.

[215] H. Huang, X. Ma, S. M. Erfani, J. Bailey, and Y. Wang, "Unlearnable examples: Making personal data unexploitable", in *International Conference on Learning Representations*, 2021.

[216] A. Madry, A. Makelov, L. Scmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks", in *International Conference on Learning Representations*, 2018.

[217] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: From phenomena to black-box attacks using adversarial samples", *arXiv preprint arXiv:1605.07277*, 2016.

[218] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations", in *International Conference on Learning Representations*, 2018.

[219] Y. Bahri, E. Dyer, J. Kaplan, J. Lee, and U. Sharma, "Explaining neural scaling laws", *Proceedings of the National Academy of Sciences*, vol. 121, no. 27, e2311878121, 2024.

[220] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 104–12 113.

[221] A. Maloney, D. A. Roberts, and J. Sully, "A solvable model of neural scaling laws", *arXiv preprint arXiv:2210.16859*, 2022.

[222] C. Stephenson, A. Ganesh, Y. Hui, H. Tang, S. Chung, *et al.*, "On the geometry of generalization and memorization in deep neural networks", in *International Conference on Learning Representations*, 2020.

[223] E. Grigsby, K. Lindsey, and D. Rolnick, "Hidden symmetries of relu networks", in *International Conference on Machine Learning*, PMLR, 2023, pp. 11 734–11 760.

[224] D. Beaglehole, I. Mitliagkas, and A. Agarwala, "Gradient descent induces alignment between weights and the pre-activation tangents for deep nonlinear networks", in *High-dimensional Learning Dynamics 2024: The Emergence of Structure and Reasoning*, 2024.

Part II

Included Publications