Representational Capabilities of Feed-forward and Sequential Neural Architectures

Clayton Hendrick Sanford

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2024

Abstract

Representational Capabilities of Feed-forward and Sequential Neural Architectures

Clayton Hendrick Sanford

Despite the widespread empirical success of deep neural networks over the past decade, a comprehensive understanding of their mathematical properties remains elusive, which limits the abilities of practitioners to train neural networks in a principled manner. This dissertation provides a representational characterization of a variety of neural network architectures, including fully-connected feed-forward networks and sequential models like transformers. The representational capabilities of neural networks are most famously characterized by the universal approximation theorem, which states that sufficiently large neural networks can closely approximate any well-behaved target function. However, the universal approximation theorem applies exclusively to two-layer neural networks of unbounded size and fails to capture the comparative strengths and weaknesses of different architectures. The thesis addresses these limitations by quantifying the representational consequences of random features, weight regularization, and model depth on feed-forward architectures. It further investigates and contrasts the expressive powers of transformers and other sequential neural architectures. Taken together, these results apply a wide range of theoretical tools—including approximation theory, discrete dynamical systems, and communication complexity—to prove rigorous separations between different neural architectures and scaling regimes.

Table of Contents

List of Figures

List of Tables

Acknowledgements

A Ph.D. is rarely completed alone, and mine is no exception. This degree was made possible by the care, guidance, and support of numerous mentors, friends, and family members, and these few words cannot adequately express my gratitude to each of them.

Unlike previous stages of the academic journey, the Ph.D. is marked by ambiguity, with few clear milestones, little external validation, and no set class of "interesting problems" to solve. Throughout the Ph.D. process, I struggled with uncertainty about my personal and professional future, and the support of an entire community was crucial to my ability to persevere, determine my values, and complete the degree.

First and foremost, I am deeply indebted to my advisors, Professors Rocco Servedio and Daniel Hsu, whose academic mentorship, unwavering support, and willingness to explore new research areas have been instrumental to my growth as a researcher and a person. Rocco and Daniel are not only brilliant scientists but generous mentors. Their time investments in their students exceed any reasonable expectation, and the treatment of their students as valued collaborators creates a compassionate and intellectually vibrant academic community. Rocco and Daniel encourage us to pursue our interests while being unafraid to venture into the technical weeds with us. They model exceptional communication skills and have finely tuned senses of humor (Daniel's understated and Rocco's more boisterous). Both of my advisors were remarkably patient with my shifting research interests and my occasional bouts of existential dread and imposter syndrome. The academic freedom that they afforded me was a gift that allowed me to develop my own voice, lens, and research agenda. For all

of this and more, I am forever grateful.

I am also grateful to the members of my thesis committee, Professors Joan Bruna, Matus Telgarsky, and Christos Papadimitriou. I had the privilege of working with Joan and Matus on research projects, and I am thankful for their research insights, their thoughtful engagement with my work, and the inclusiveness of their academic communities. I served as a teaching assistant for Christos, and I benefited from his guidance and the freedom he gave me to experiment with my teaching style.

Throughout my graduate and undergraduate studies, I have been fortunate to have many faculty mentors whose teaching, research mentorship, and career guidance positively shaped my academic trajectory. I acknowledge the support of Professors Carly Klivans, Paul Valiant, Björn Sandstede, Eli Upfal, Anna Lysyanskaya, Shriram Krishnamurthy, and Tal Malkin. In particular, Carly's persistence in encouraging me to pursue a Ph.D. was instrumental in my decision to apply to graduate school.

My numerous research collaborators each individually fostered my growth as a researcher and made the Ph.D. experience much more enjoyable. I thank my colleagues, including but not limited to, Manolis Vlatakis-Gkaragkounis, Vaggos Chatziafratis, Navid Ardeshir, Ioannis Panageas, Stelios Stavroulakis, Min Jae Song, Alberto Bietti, Anna Kwa, Oliver Watt-Meyer, Christopher Bretherton, Cyril Zhang, Dylan Foster, Akshay Krishnamurthy, Jieming Mao, Jon Schnieder, Bahar Fatemi, Ethan Hall, Vahab Mirrokni, Berkan Ottlik, and Edward Ri. I would also like to extend my appreciation to those who provided me with career guidance and advice throughout the graduate process, from Ph.D. application to mid-Ph.D. crisis to the job market: Vaggos, Alberto, Cyril, Kiran Vodrahalli, Rajesh Jayaram, Cyrus Cousins, Yee Sian Ng, Jasper Lee, Kevin Yeo, Siddharth Karamcheti, Mitchell Wortsman, and Giannis Karamanolakis.

I am a proud product of the public school system, and I owe a debt of gratitude to the numerous teachers who challenged me to pursue my interests and sparked my curiosity. I would like to recognize the contributions of Ms. Joanne Roster, Mr. Todd Shaff, Ms. Anne

Cervantes, Ms. Dion O'Reilly, Ms. Gail Alaimo, Mr. Dan Siddens, Ms. Marissa Ferejohn-Swett, Ms. Robyn Miranda, and Ms. Branna Banks. I believe that my path to a Ph.D. would have been impossible without the tireless mentorship of all of these educators inside and outside the classroom and the specific opportunity to pursue more advanced mathematics in middle and high school due to the efforts of Mr. Shaff.

Graduate school would be a far lonelier and more dispiriting experience without the support of my friends and colleagues, especially given the effects of the COVID-19 pandemic. My childhood friends, including Justin Lang, Trevor Sanders, Steven Corona, Cameron Fanthorpe, Kate Osterhoudt, Scott Kauker, Juan Castillo, Eric Wells, Christina Dooka, and Amanda Favorite, have been a source of humor, joy, and grounding throughout my life, and I am grateful for their continued friendship (and tolerance of my D&D machinations). My college friends near and far, including Olivia Kelly (and the entire Kelly family), Vince Kubala, Kate Ferguson, Ethan Wright, JJ Ruth, Katie Scholl, and Lauren Montieth, have supported me through the Ph.D. program, inspired me with their accomplishments, and challenged me to think more deeply about my values and goals. My friends in New York City, including Priya Patel, Egor Shakhnovsky, Surbhi Madan, Arun Drelich, Natalie Tsvetkova, Samir Lavignia, Sarah McNeill, Pamela Mishkin, and Jimmy Lin, helped me explore the city, find joy away from Columbia, and provided a deeply supportive community, before, during, and after the pandemic. My Ph.D. co-adventurers and co-conspirators, including Tim Randolph, Shivam Nadimpalli, Navid Ardeshir, Sam Deng, Jingwen Liu, John Hui, Vikram Nitin, Dan Mitropolsky, Jason Milonis, Melanie Subbiah, Samir Gadre, Sachit Menon, Min Jae Song, Kathy Jang, Alessio Mazzetto, Adam Block, John Bostanci, Santiago Gonzalez, and Natalie Parham, form a resilient network of mutual support that has made the Ph.D. journey more intellectually vibrant, emotionally bearable, and fun. While the path to a Ph.D. is steep and stony, the positive memories far outweigh the negative ones due to the presence of these individuals.

The final year of my Ph.D. was made infinitely more tolerable by Shuai Tang, whose

presence has been a consistent source of support, laughter, and joy. In a tumultuous year, Shuai was a calming presence who helped me keep things in perspective. I appreciate how we push each other to explore the city and that I've managed to take him hiking, he brought me to *Hadestown*, and we both tried archery. I am grateful for Shuai's patience and irreverence, and I am excited to see what is next for us.

I am where I am today due to the many forms of love and support I have received from my family. I am grateful for the many cousins, aunts, and uncles who supported me throughout my life, including this most recent period, when I found homes away from home in Boston, Philadelphia, Seattle, the Bay Area, Los Angeles, San Diego, and San Antonio. As the second-youngest of many cousins, I have been inspired by the accomplishments of those around me and am fortunate to receive advice and guidance about life and careers from my relatives. I acknowledge in particular the support of my cousins Nate, Kelly, Eric, David, Kim, and Annie, who helped me find my way at critical points in the Ph.D. process; my Uncle Joe, whose dedication to his family inspires me and whose hospitality helped New York City feel like home; and my aunts and uncles, Ellen and Chet, Barbara and Jeff, Fred and Jolynn, Arthur and Michelle, Jane and Larry, and Brian and Linnea, who welcomed me into their households and generously shared their wisdom and love.

I am indebted to the love and support of my grandmother, Barbara Hugus, who passed away midway through my Ph.D. Grandma Barbara paved the path for all of us, earning a Ph.D. in biology in the 1960s while raising four children and having an illustrious career in cancer genomics research. She was an intellectual giant and a deeply caring grandmother with an endless supply of love, wisdom, and warm hugs. Her support was critical to the stability of my nuclear family and my ability to have a happy childhood and pursue my interests. Without my grandmother, this Ph.D. would have been a distant dream, and this thesis is dedicated to her memory.

My sister, Barbara Sanford (who was named after our grandmother), is also graduating this spring with a bachelor's degree, and I am immensely proud of her and grateful for our

relationship. Barbara is mature, intelligent, and caring. In the past five years, I have leaned on her for advice, support, and writing feedback. There are few stronger feelings of pride than those experienced by an older sibling who watches his younger sibling grow up, travel the world, find her place, and become someone he deeply admires. Barbara navigates her life with uncommon wisdom. She cares deeply about those around her, going extraordinary lengths to support her loved ones and make them feel valued. At the same time, her maniacal grin, inexplicable sense of humor, and vast inner world have persisted and flourished since childhood, and I treasure every glimpse I get. I hope she understands how much I appreciate her.

And finally, my greatest debt is to my mother, Paula Sanford. My mother is the wisest person I know. She is an empathetic listener, a curious mind, and a deeply moral individual. My mom challenges me to be more compassionate and more thoughtful, a better friend, brother, student, and partner. She did not choose to be a single parent, but circumstances forced her to raise my sister and me on her own (with the support of both grandmothers and our extended family) and she did so with extraordinary selflessness and unconditional love.

One of the most rewarding parts of the last five years has been watching her act on her passion for early childhood education, earn a formal education in the field, and find an infant care job that she loves. At one point in the pandemic, all three of us were writing papers simultaneously, and I appreciated that new familial connection, as we all explored our interests and planted seeds. I treasure the stories she tells me about the children she cares for; they are evidence of a deep-seated love for those around her and a reminder never to stop searching for what makes you happy. At one point in the Ph.D., when I was unsure about my future plans, she told me that she did not find a profession that she truly loved until very recently, after decades of different jobs. That conversation stuck with me and helped me relax a little, feel grateful, and find comfort in ambiguity. Thriving in the face of uncertainty is perhaps the most important skill for one to learn in a Ph.D. program, and I am deeply appreciative that I learned that lesson from my mother, in addition to so many

others.

Dedication

This thesis is dedicated to my grandmother, Barbara Hugus (1927 - 2021), whose intellect, kindness, and unconditional love made this work possible.

Chapter 1: Introduction

Neural networks have emerged as the dominant machine learning paradigm over the past decade, with applications in natural language processing, computer vision, protein folding, and many other areas. With the rise of the transformer model of Vaswani et al. (2017) for sequential learning tasks, this dominance has only further solidified, and the size of these models and the computational resources required to train these models have grown at an unprecedented rate. Designing a neural network to solve an ML task requires choosing an architecture and setting numerous hyperparameters. Due to the high computational and energy costs of training modern neural networks, the development of rigorous and empirically validated approaches for choosing the right neural architecture is of broad practical importance. Doing so in a principled way requires understanding the impact of the choice of architecture (e.g. multi-layer perceptron, recurrent neural network, transformer) and hyperparameters (e.g. depth, width, weight initialization) on the representational, optimization, and generalization properties of the corresponding network.

The field of *neural network theory* aims to provide a rigorous and empirically validated understanding of the capabilities and limitations of these models and to guide the development of new architectures and training algorithms. However, this theoretical understanding has not kept pace with rapid empirical progress of machine learning, in part due to its inability to capture the complexity of modern models. This dissertation aims to address this gap between the empirical challenges of architectural design and the theoretical understanding of neural networks by developing new mathematical tools and insights that are relevant to modern architectures and training algorithms. The focus of this dissertation is on the *representational* capabilities of neural networks, which quantifies the class of functions that a network can approximate. Although representational capabilities are only one aspect of the

performance of a neural network, they are a fundamental building block for understanding architectural properties.

At first glance, the representational capabilities of neural networks appear well-understood. The celebrated *universal approximation theorem* (UAT) of Cybenko (1989), Hornik, Stinchcombe, and White (1989), and Funahashi (1989) establishes the universality of two-layer neural networks by showing that any continuous multivariate function can be approximated by a two-layer network with a sufficiently large number of hidden units. However, these results offer no upper bound on the number of hidden units required, which limits the practical utility of the theorem to neural networks of constrained size. Moreover, the UAT does not provide insight into the comparative capabilities of neural networks with different architectures; while two layers suffice for universal approximation, practical networks often have many more layers and other architectural features that are not captured by the UAT. Furthermore, these results have limited applicability to sequential models like transformers, which operate on variable-length sequences (e.g. passages of text) as input, rather than fixed-length vectors. This thesis aims to venture beyond the unbounded-size framing of the UAT and to develop a more nuanced and fine-grained understanding of the representational capabilities of modern neural network architectures.

The contributions of this dissertation (which are discussed in greater detail in Section 1.3) are organized into two categories: (1) a more fine-grained understanding of the UAT for feedforward neural networks and (2) a study of the representational capabilities of the transformer model. The first category includes works that draw contrasts between the expressive powers of various classes of neural networks. These classes are defined based on whether weights are randomly sampled (Chapter 2), the depth (Chapter 3), and the boundedness of model weights (Chapter 4). The second category exhibits the unique capabilities of the transformer model compared with other sequential models by quantifying the abilities of multi-headed attention layers to draw associations between sequential inputs (Chapter 5) and of deep transformers to implement parallelizable algorithms (Chapter 6).

1.1 Historical context and background

Despite their recent prominence, biologically inspired neural networks are not a new idea, and the field has experienced several cycles of sensationalism and disillusionment. The *perceptron*, introduced by McCulloch and Pitts (1943) and implemented by Rosenblatt (1958) was the first widely studied neural network. A perceptron is a single-neuron model that computes an affine function of the inputs and applies a nonlinear threshold function σ to the result:

$$f(x) = \sigma(w^\mathsf{T} x + b) \tag{1.1}$$

where $x \in \mathbb{R}^d$ is the input, $w \in \mathbb{R}^d$ is a vector of weights, and $b \in \mathbb{R}$ is a bias term. The parameters of the perceptron can be learned from labeled training samples with the *perceptron learning algorithm*, which updates the weights and bias whenever the perceptron makes a mistake. Rosenblatt famously made outlandish claims about the promise of the perceptron model, deeming it a "machine capable of perceiving, recognizing, and identifying its surroundings without any human training or control" (Lefkowitz, 2019).

However, enthusiasm over the potential of the perceptron was short-lived due to a concrete description of its limitations by Minsky and Papert (1969). They showed that no perceptron could represent the *exclusive or* (XOR) function, which takes two binary inputs, outputting 1 if and only if exactly one of the two inputs is 1. This counterexample concisely demonstrated the representational limitations of the perceptron and crystallized the difficulty of realizing Rosenblatt's lofty ambitions. While this particular counterexample can be mitigated by utilizing feature transformations or multiple perceptrons, this negative result relegated neural networks to a niche subfield of artificial intelligence for several decades.

The study of neural networks experienced a resurgence in the 1980s and 1990s, driven by the development of new architectures and training algorithms. This era saw a shift in focus from the perceptron to more general feedforward neural networks or *multi-layer perceptrons* (MLPs), which are composed of multiple layers that consist of an array of neurons,

alongside other architectures such as the *long short-term memory* (LSTM; Hochreiter and Schmidhuber, 1997) and the *convolutional neural network* (CNN; Lecun et al., 1998). These architectural innovations were accompanied by the development of training algorithms such as *back-propagation* (Rumelhart, Hinton, and Williams, 1986) that made it possible to train networks with many layers and neurons, albeit without the formal guarantees of convergence or generalization that accompany other learning algorithms.

This second wave of neural network hype was accompanied by a body of theoretical research that investigated the capabilities and limitations of these models. Most famous among those works is the *universal approximation theorem* (UAT) of Cybenko (1989), Hornik, Stinchcombe, and White (1989), and Funahashi (1989), which proves that any multivariate function on a compact domain can be approximated by a sufficiently wide neural network of depth two. In contrast to the impact of the revelation of the perceptron's limitations by the XOR function, the UAT was widely cited as a theoretical justification for the practical application of multi-layer perceptrons.

Despite its status as a landmark theoretical result, the practical utility of the UAT was limited. While representational results like the UAT promise the existence of neural networks that fit any mapping of inputs to labels, they provide no insight into whether any learning algorithm can actually find these networks (optimization) or whether these networks will perform well on new inputs (generalization). Moreover, these representational results fail to quantify the size of neural networks necessary and sufficient to fit particular datasets and approximate certain target functions. Since these results focus on the universality of two-layer networks in the arbitrary-width regime, they neither pinpoint the comparative strengths and limitations of different neural architectures nor provide prescriptive insights to practitioners.

Neural networks would again fall from prominence in the machine learning research community during the late 1990s and 2000s. While neural network research persisted in the background during this period, neural networks were largely replaced in the literature by

approaches such as support vector machines, which were equipped with stronger optimization guarantees and were reliably trainable (Cortes and Vapnik, 1995). These alternatives were more amenable to mathematical analyses of convergence and generalization, making them more attractive to the machine learning theory community.

Neural networks experienced another resurgence in the early 2010s as neural networks established empirical dominance over other machine learning methods on a wide range of benchmark tasks. This progress was most prominently demonstrated by the AlexNet computer vision model (Krizhevsky, Sutskever, and Hinton, 2012a), which showcased the ability of a convolutional neural network to outperform all contemporary alternatives on standard computer vision tasks. Further refinements to the CNN architecture cemented this dominance, which has since remained unchallenged by conventional machine learning algorithms. Soon after, Sutskever, Vinyals, and Le (2014) and others demonstrated the dominance of neural networks such as the LSTM in natural language processing tasks. Landmark demonstrations of the powers of deep neural networks continued to emerge, perhaps most famously with the AlphaGo agent for playing Go (Silver et al., 2016).

This wave was made possible by immense increases in parallel computation power, which enabled researchers to design much larger models and train them on much larger datasets. Indeed, the moniker "deep learning" was coined due to the sharp increase in the depth of neural networks during this period. Past misgivings about the lack of algorithmic convergence guarantees of gradient descent were largely dispelled by the practical success of these models. In particular, these models were often so large as to be *over-parameterized* (having more trainable parameters than training samples) and experienced *benign overfitting*, in which the learning algorithm trains a neural model that perfectly fits the labeled training samples and still has favorable generalization properties (e.g. Belkin et al., 2018).

This boom saddled machine learning theorists with several unsettling contradictions between classical learning theory and modern empirical results. First, while the UAT demonstrates the capabilities of extremely wide two-layer models, the superior empirical perfor-

mance of much deeper models could not be rigorously explained using this representational lens alone. Second, despite the non-convexity of the deep networks' loss functions, which makes theoretical convergence difficult to prove, empirical evidence demonstrates the ability of gradient descent to produce networks that perfectly fit even samples with random labels (Zhang et al., 2017); practical networks exceeded their theoretical expectations. Finally, classical machine learning theory relies on establishing relationships between the expressivity of a concept class and its tendency to overfit. However, deep neural networks are an arbitrarily expressive class that perfectly fits training samples and nonetheless performs well on new samples.

In the late 2010s, theoreticians grappled with these contradictions and developed several new research areas in the study of neural networks. For instance, the subfield of *depth separation* (e.g. Telgarsky, 2016; Eldan and Shamir, 2016) emerged to justify the preference of practitioners for increasing depth by demonstrating particular target functions that can be approximated efficiently (i.e., with polynomial width) by deep models, but require exponential width to be approximated by shallow networks. Others studied the *neural tangent kernel* (NTK) to study a regime where neural networks behave similarly to kernel machines and provably converge due to convexity (Jacot, Gabriel, and Hongler, 2018). The generality and principles behind benign overfitting were theoretically exhibited in multiple alternative learning models, including least squares regression (Bartlett et al., 2019). These approaches ultimately struggled to provide practical insights for deep learning practitioners due to the relative simplicity of the neural architectures they considered and the need for intensive modeling assumptions to establish their results. However, these results encapsulated a new era of neural network theory that adapted classical insights to novel learning settings that capture certain aspects of empirical deep learning.

The deep learning boom of the 2010s has persisted in the early 2020s, with the number of model parameters, training samples, and research papers increasing at a furious pace. The rise of the *transformer* architecture (Vaswani et al., 2017) and subsequent progress in

large language models (LLMs) have been particularly emblematic of this era. Transformers, like recurrent neural networks, are sequential models, which take as input a variable-length series of tokens, such as a passage of text, and output a variable-length series of tokens, such as a translation of the input passage. Unlike RNNs, transformers make much more efficient usage of parallel computing infrastructure, which has enabled a dramatic increase in the sizes of textual inputs to neural networks—from at most 4096 tokens in GPT-3 (Brown et al., 2020) to 32,768 tokens in GPT-4 (OpenAI, 2023), and potentially up to 1,000,000 tokens in the latest Gemini models. In addition to their widespread dominance in NLP tasks, the capabilities of transformers have been demonstrated in a wide range of domains previously not tackled with language models, besting CNNs in computer vision tasks (Dosovitskiy et al., 2021), and alternative models in protein folding analysis (Jumper et al., 2021).

The recent proliferation of transformers across machine learning brings with it fundamental questions about the architecture. Is its parallelizable inference and training its sole advantage, or does the architecture carry further representational benefits? Can the successes of transformers be replicated by more computationally efficient alternatives? And are there new generalization principles that govern which kinds of tasks are natural to learn? These questions suggest the importance of novel theoretical approaches to draw inspiration from the XOR counterexample of Minsky and Papert (1969) to establish straightforward contrasts in neural architectures that crystallize their fundamental limitations and advantages.

1.2 Overview of neural architectures

Before we discuss the contributions of this dissertation, we provide a brief overview of the neural architectures that this work studies. As discussed in the previous section, the study of biologically inspired artificial neural networks began with the perceptron model of McCulloch and Pitts (1943), which was designed to model the behavior of a single biological neuron. Since then, a wide variety of neural architectures have been developed, each of which contains a collection of neurons that are connected in distinct ways and that are designed

for different tasks.

At its most general, a *neural network* is a parameterized function $f_\theta : \mathcal{X} \to \mathcal{Y}$ with model parameters $\theta \in \mathbb{R}^p$ that maps some input $x \in \mathcal{X}$ to an output $f_\theta(x) \in \mathcal{Y}$. In the case of the perceptron model in Equation (1.1), the input x is a real-valued vector (i.e. $\mathcal{X} = \mathbb{R}^d$), the output is a binary label (i.e. $\mathcal{Y} = \{0, 1\}$), and the model parameters $\theta = (w, b) \in \mathbb{R}^{d+1}$ consist of a weight vector $w \in \mathbb{R}^d$ and a bias term $b \in \mathbb{R}$. A *neural architecture* can be thought of as a family of neural networks $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$ over some set of parameters Θ.

Neural networks are *trained* on a finite training dataset $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ by finding parameters θ that minimize a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, which measures the difference between the predicted output $f_\theta(x_i)$ and the true output y_i for each training example (x_i, y_i). Popular examples of loss functions include the *squared loss* $\ell(\hat{y}, y) = (\hat{y} - y)^2$ for regression tasks and the *cross-entropy loss* $\ell(\hat{y}, y) = -y \log \hat{y} - (1-y) \log(1-\hat{y})$ for binary classification tasks. The parameters θ are retrieved by solving an optimization problem of the form

$$\min_{\theta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \ell(f_\theta(x_i), y_i) + \lambda R(\theta), \tag{1.2}$$

where $R(\theta)$ is a *regularization function* that penalizes overly complex parameter configurations and $\lambda > 0$ is a hyperparameter that controls the strength of the regularization.

Machine learning theorists study the properties of neural architectures by quantifying their representational, optimization, and generalization properties.

A neural architecture's *representational capabilities* measure the types of learning rules and datasets that can be fit by some neural network of that architecture. Concretely, for architecture $\mathcal{F}$ and some concept class $\mathcal{H}$ containing functions of the form $\mathcal{X} \to \mathcal{Y}$, we ask whether for every target $h \in \mathcal{H}$, there exists some neural network $f \in \mathcal{F}$ such that f has approximately identical outputs to h. This may be measured with L_∞ error, i.e.

$$\sup_{x \in \mathcal{X}} |f(x) - h(x)| \leq \epsilon,$$

or with L_2 error with respect to some measure μ over $\mathcal{X}$, i.e.

$$\mathop{\mathbb{E}}_{x\sim\mu}\left[\|f(x)-h(x)\|_2^2\right]\leq\epsilon.$$

For instance, the perceptron architecture with threshold activation $\sigma(t)=\mathbb{1}\{t\geq 0\}$ can perfectly approximate the family of linear threshold functions $\mathcal{H}=\{x\mapsto\sigma(v^{\mathsf{T}}x+c)\}$, but it cannot approximate the XOR target $x\mapsto x_1+x_2-2x_1x_2$ on $\mathcal{X}=\{0,1\}^d$.

The *optimization properties* of a neural architecture and a learning algorithm (such as stochastic gradient descent) measure how well the algorithm can recover the parameters θ that minimize the empirical risk in Equation (1.2). For the perceptron model, the perceptron learning algorithm provably converges to a solution that perfectly classifies the training data if the data is linearly separable, and the number of updates until convergence can be bounded in terms of the largest linear threshold margin. When the empirical risk is a convex function of the parameters θ, then the optimization problem is well-understood and can be solved efficiently. However, non-convex optimization problems are more challenging to solve, and most neural architectures have non-convex losses as a function of the parameters θ. This fact has motivated a large body of research into the "optimization landscape" of neural networks, which studies the geometry of the losses of parameterized networks as a function of θ and its relationship to how gradient-based optimization algorithms traverse this landscape.

The *generalization properties* of a neural architecture assess how well the model performs on unseen data. Most classical studies of generalization are concerned with generalization within the same distribution, where both training samples and test samples are drawn from the same distribution $\mathcal{D}$. These studies often rely on the *uniform convergence* framework, which bounds the gap between the empirical risk on training samples $\frac{1}{n}\sum_{i=1}^n\ell(f_\theta(x_i),y_i)$ and the expected risk on novel samples $\mathbb{E}_{x\sim\mathcal{D}}\left[\ell(f_\theta(x),y)\right]$ as a function of the complexity of the model class $\mathcal{F}$ (which can be quantified by measures such as the VC-dimension) and the number of training samples n. This framework suggests a tension between the

expressivity of an architecture (which allows empirical risk to be small) and its tendency to overfit (in which the generalization gap is nonetheless large). In the context of highly expressive neural networks, this theory presents an overly pessimistic view, where rigorous generalization bounds are difficult to obtain.

In the following sections, we present an overview of the feed-forward and sequential neural architectures whose representational properties are studied in later chapters of this dissertation.

1.2.1 Feed-forward neural networks

A *feed-forward neural network* or *multi-layer perceptron* (MLP) expands the perceptron model to consider an ensemble of neurons whose outputs are computed both in parallel and in series. Each neuron is parameterized similarly to a perceptron, with a nonlinear *activation function* σ applied to an affine transform of the input. The *width* of an MLP refers to the maximum number of neurons evaluated in a *layer* of parallel computation, and the *depth* is the number of such parallel layers. While $\sigma : \mathbb{R} \to \mathbb{R}$ can be defined as a wide range of nonlinear activations, we typically consider the case where σ is the *Rectified Linear Unit* (ReLU), i.e.

$$\sigma(t) = \mathrm{ReLU}(t) = \begin{cases} t & \text{if } t \geq 0 \\ 0 & \text{if } t \leq 0. \end{cases}$$

Unlike sequential architectures, we think of MLPs as a mapping from some fixed-dimensional input space ($\mathcal{X} = \mathbb{R}^d$) to some fixed-dimensional output space (which is typically single-dimensional, i.e. $\mathcal{Y} = \mathbb{R}$). The parameters of an MLP are the weights and biases of each neuron. While a wide range of variations on the standard MLP architecture exist, we focus on a simple version for this section, where there are no skip-layer connections and no batch normalization.

1.2.1.1 Two-layer neural networks

We first consider the case of *two-layer neural networks*, which consist of a single "hidden layer" of neurons between the input and output layers. The output of a two-layer ReLU neural network f_θ of width m is given by

$$f_\theta(x) = \sum_{i=1}^{m} u_i \mathrm{ReLU}(w_i^\mathsf{T} x + b_i) = u^\mathsf{T} \mathrm{ReLU}(Wx + b), \tag{1.3}$$

where $W \in \mathbb{R}^{m \times d}$ is a bottom-layer weight matrix with $W = (w_1, \ldots, w_m)$, $b \in \mathbb{R}^m$ is a bias vector, and $u \in \mathbb{R}^m$ is a top-layer weight vector. The network parameters are $\theta = (W, b, u) \in \mathbb{R}^{md+m+m}$. This neural network can be thought of as a linear combination of m "ReLU features" $\mathrm{ReLU}(w_i^\mathsf{T} x + b_i)$, where each feature is an affine transformation of the input x followed by a ReLU activation.

As discussed before, the universal approximation theorem of Cybenko (1989), Funahashi (1989), and Hornik, Stinchcombe, and White (1989) implies that the family of two-layer neural networks with ReLU activations (among other activation functions) can closely approximate any "nice" target function.

Theorem 1.1 (Universal approximation theorem)**.** *For any continuous target $h : \mathbb{R}^d \to \mathbb{R}$, any $\epsilon > 0$, and any compact set $K \subseteq \mathbb{R}^d$, there exists a two-layer ReLU neural network f_θ with finite width m such that*

$$\sup_{x \in K} |f_\theta(x) - h(x)| \leq \epsilon.$$

Unlike the perceptron model, two-layer neural networks have a universality property that allows them to approximate any function. Note that the nonlinear activation function is critical for this result; if the ReLU function were replaced by a linear activation, then the two-layer network would be equivalent to a linear model.

However, the universal approximation theorem has little to say about *efficient* approximation. This efficiency is typically measured by the minimum width m required to achieve

a given approximation error ϵ over some class of target functions $\mathcal{H}$. This framework can be used to study the limitations of the two-layer neural network architecture by providing certain target functions that can only be approximated by networks whose width m is exponentially large in the input dimension d.

In contrast, other researchers quantify efficient approximation by the norm of parameters θ required to achieve a given approximation error ϵ. While other chapters focus on the width of the network, Chapter 4 of this dissertation considers the task of fitting a target dataset with bounded weight norm $\|W\|_2 + \|u\|_2$.

When training a two-layer neural network, one typically minimizes the empirical risk as a function of all parameters $\theta = (W, b, u)$ with a gradient-based optimization algorithm. This "bilevel" optimization problem is non-convex, which makes optimization analysis challenging for even the simplest neural architectures.

1.2.1.2 Two-layer random feature networks

A *random feature network* is a two-layer neural network whose bottom-layer weights and biases are drawn from some probability distribution. That is, for some fixed distribution $\mathcal{P}$ over $\mathbb{R}^{d+1}$, the parameters of each bottom-layer neuron are drawn independently from $\mathcal{P}$, i.e. $(\mathbf{w}_i, \mathbf{b}_i) \sim \mathcal{P}$ for each $i \in [m]$. The top-layer weights $u \in \mathbb{R}^m$ are learnable by minimizing the empirical risk. An output of the neural network is given by

$$f_\theta(x) = \sum_{i=1}^{m} u_i \mathrm{ReLU}(\mathbf{w}_i^\mathsf{T} x + \mathbf{b}_i) = u^\mathsf{T} \mathrm{ReLU}(\mathbf{W}x + \mathbf{b}). \tag{1.4}$$

The expressiveness of this model is studied in detail in Chapter 2. Unlike standard two-layer neural networks, the random bottom-layer weights and biases are fixed during training. Since the resulting network is a linear combination of ReLU features, minimizing the empirical risk of a random feature network with a convex loss function ℓ is a well-understood convex optimization problem. Due to the connection between random feature

models and kernel methods, the generalization properties of random feature networks are well-understood as well (Neal, 1996; Rahimi and Recht, 2008; Cho and Saul, 2009).

1.2.1.3 Deeper neural networks

In the theoretical literature, any MLP with more than two layers is considered a *deep neural network*. We can define a deep neural network with L layers of width m recursively as

$$f_\theta^{(\ell)}(x) = \begin{cases} x & \text{if } \ell = 0, \\ \mathrm{ReLU}(W^{(\ell)} f_\theta^{(\ell-1)}(x) + b^{(\ell)}) & \text{if } \ell = 1, \dots, L-1, \\ u^\mathsf{T} f_\theta^{(L-1)}(x) & \text{if } \ell = L, \end{cases} \tag{1.5}$$

where $W^{(1)} \in \mathbb{R}^{d \times m}$ and $W^{(2)}, \dots, W^{(L)} \in \mathbb{R}^{m \times m}$ are weight matrices, $b^{(1)}, \dots, b^{(L)} \in \mathbb{R}^m$ are bias vectors, and $u \in \mathbb{R}^m$ is a top-layer weight vector. The parameters of this model are

$$\theta = (W^{(1)}, \dots, W^{(L)}, b^{(1)}, \dots, b^{(L)}, u) \in \mathbb{R}^{dm + m^2(L-1) + mL + m}.$$

Increasing the depth of the network allows for targets with a "hierarchical" structure to be approximated more efficiently. These relationships are made precise by Eldan and Shamir (2016) and Telgarsky (2016) and in Chapter 3 of this dissertation. Proving optimization and generalization properties of deep neural networks is even more challenging than for two-layer networks, which makes the study of deep network representational properties one of the highest potential directions for the study of deep models.

1.2.2 Sequential neural networks

In contrast to feed-forward neural networks, *sequential neural networks* are designed to process variable-length sequential data, including passages of text, audio recordings, and time-series data. The output of a sequential network may be another sequence or a fixed-dimensional output. In this dissertation, we model a sequential neural network as a function

$f_\theta : \mathcal{X}^N \to \mathcal{Y}^N$ that maps a sequence of N input vectors $X = (x_1, \ldots, x_N) \in \mathcal{X}^N$ to a sequence of N output vectors $Y = (y_1, \ldots, y_N) \in \mathcal{Y}^N$ with model parameters $\theta \in \mathbb{R}^p$. Unlike feed-forward neural networks, we consider model parameterizations that do not scale super-linearly with the sequence length N and therefore require some notion of parameter sharing between sequence elements.

This section introduces of two popular families of sequential neural networks: *recurrent neural networks* and *transformers*. Both architectures are designed to process sequential data, but their overall structures and computational properties differ greatly.

1.2.2.1 Recurrent neural networks

Recurrent neural networks (RNNs) or *state-space models* (SSMs) are a family of sequential neural networks that process a sequential input $X \in \mathcal{X}^N$ by "unrolling" a sequence of neural units g_θ. We define RNNs very loosely to capture a wide range of models, including the vanilla RNN, the long short-term memory (LSTM), and recent SSMs.

An RNN processes a sequence $X = (x_1, \ldots, x_N) \in \mathcal{X}^N$ iteratively updating a hidden state $z_0, \ldots, z_N \in \mathbb{R}^m$ and outputting a sequence $f_\theta(X) = Y = (y_1, \ldots, y_N) \in \mathcal{Y}^N$. For each $i \in [N]$, we compute

$$(y_i, z_i) = g_\theta(x_i, z_{i-1}),$$

where $g_\theta : \mathcal{X} \times \mathbb{R}^m \to \mathcal{Y} \times \mathbb{R}^m$ is a neural unit with parameters $\theta \in \mathbb{R}^p$ and z_0 is a fixed initial state. As a result, the number of parameters in f_θ is independent of the sequence length N.

This architecture is well-suited for processing temporal data, where the interactions between nearby elements in the sequence are most important. In the above formulation, each output y_i depends only on inputs $x_1, \ldots, x_i$ and the initial state z_0; "bidirectional" or "multi-pass" RNNs can alleviate this limitation if other dependencies are necessary for the sequence processing task. However, as is discussed in Chapters 5 and 6, the RNN architecture is limited by its hidden state, rendering it unable to capture long-range dependencies or pass large amounts of information between distant sequence elements.

While RNNs are structurally intuitive, they are known to be difficult to train due to the vanishing and exploding gradient problems, which are caused by the repeated application of the same neural unit g_θ. Furthermore, while the time complexity of an RNN scales linearly with the sequence length N, the iterative nature of the RNN makes its computation difficult to parallelize. The representational contrasts between these architectures are explored in Chapters 5 and 6.

1.2.2.2 Transformers

The *transformer* architecture of Vaswani et al. (2017) is an alternative to the RNN whose sequential mapping is determined by computed affinities between sequence elements. Rather than processing the sequence element-by-element, the transformer processes the entire sequence simultaneously by passing the input through a collection of *self-attention units*.

Each self-attention unit computes a N different weighted averages of the input sequence, where the weights are determined by a learned affinity matrix. Concretely, on input $X = (x_1, \ldots, x_N) \in \mathbb{R}^{N \times d}$, a self-attention unit computes *query*, *key*, and *value* sequences $XQ, XK, XV \in \mathbb{R}^{N \times m}$ for linear transformations $Q, K, V \in \mathbb{R}^{d \times m}$. The ith output of the self-attention unit is a weighted average of the value embeddings $V^\mathsf{T} x_1, \ldots, V^\mathsf{T} x_N \in \mathbb{R}^m$ with weights determined by the dot products of the queries and keys. That is, the output of the self-attention unit is given by

$$\text{softmax}(XQK^\mathsf{T}X^\mathsf{T})XV = \Big(\text{softmax}(x_i^\mathsf{T} QK^\mathsf{T} x_1, \ldots, x_i^\mathsf{T} QK^\mathsf{T} x_N)XV\Big)_{i=1,\ldots,N} \in \mathbb{R}^{N \times m},$$

where $\text{softmax} : \mathbb{R}^N \to \mathbb{R}^N$ is the softmax function applied element-wise. The softmax function ensures that the elements of the *attention matrix* $XQK^\mathsf{T} \in \mathbb{R}^{N \times N}$ sum to one and are large if the inner product of the corresponding query and key is large.

A *multi-headed self-attention layer* computes the output of H self-attention units in parallel and concatenates the results. A transformer model comprises a stack of L multi-

headed self-attention layers, each followed by an element-wise MLP, which applies a learnable nonlinear function to each element individually. The output $Y = f_\theta(X)$ of the transformer is the output of the final MLP layer. The parameters θ of the transformer include all query, key, and value matrices and the parameters of the MLPs.

Like an RNN, the parameterization of a transformer does not scale linearly with the sequence length N. The non-iterative nature of the transformer allows for more natural representations of long-range dependencies in the sequence than an RNN, enables more efficient parallelization, and avoids the vanishing and exploding gradient problems. However, implementing a self-attention unit requires computing an attention matrix of size $N \times N$, which causes a quadratic time complexity in the sequence length N. Several recent works have proposed more computationally efficient alternatives to the full transformer architecture, such as the Performer model of Choromanski et al. (2022).

1.3 Outline of results

This doctoral dissertation consists of a body of theoretical work that expands upon the universal approximation theorem and contrasts the expressivity of modern neural architectures. The contributions of this thesis fall into two broad categories: (1) those that expand the understanding of the representational capabilities of feedforward neural networks of various depths, initializations, and constraints; and (2) those that introduce novel theoretical lenses to the fundamental capabilities of transformers and other sequential architectures.

1.3.1 Parameter complexity and architectural trade-offs of feedforward networks

Chapters 2 to 4 build upon a large body of approximation theoretic work to draw sharp contrasts between the representational capabilities of feedforward (or multi-layer perceptron) architectures. While the universal approximation theorem guarantees the existence of neural network approximators of unbounded size, these chapters study the regime of *efficient approximation*, where the size of approximator networks grow polynomially in the input

dimension and relevant complexity measures of the target functions.

Each chapter establishes representational separations between architectural choices that reflect meaningful design decisions undertaken by deep learning practitioners. Chapter 2 characterizes the representational capabilities of bounded-width two-layer *random feature models*—in which bottom layer weights are fixed and randomly sampled from a distribution, and top layer weights can be chosen by some learning algorithm—and contrasts these abilities with standard two-layer networks with learnable weights on both layers. Chapter 3 contrasts the representational powers of networks as a function of depth by employing a novel connection to discrete dynamical systems. Chapter 4 considers two-layer networks with bounded weights and potentially unbounded widths that interpolate training samples to assess the representational impacts and intrinsic dimensionality of weight norm regularization. The contributions of each chapter are briefly described in the following sections.

Shallow random feature networks: dimensionality, smoothness, and width trade-offs (Chapter 2)

This chapter, which presents the work of Hsu, Sanford, Servedio, and Vlatakis-Gkaragkounis (2021), characterizes the minimum width of a two-layer random feature MLP that approximates target functions that satisfy certain smoothness conditions.

The *random feature model* is a simple and well-known class of neural networks, which takes the form of a linear combination of random features with a nonlinear activation function $\phi : \mathbb{R} \to \mathbb{R}$:

$$x \mapsto \sum_{i=1}^{m} u_i \sigma(\mathbf{w}_i^\mathsf{T} x + \mathbf{b}_i).$$

The bottom layer weights $\mathbf{w}_i \in \mathbb{R}^d$ and $\mathbf{b}_i \in \mathbb{R}$ are drawn independently from some fixed distribution, and the top layer weights $u \in \mathbb{R}^m$ can take on any value When structured as a learning problem, the bottom layer weights are fixed to their random initialization, and top layer weights are learned by minimizing a loss function over a training dataset. This model is a restriction of a more general family of two-layer neural networks with learnable weights

on both layers, and it serves as a bridge between kernel methods and neural networks, as discussed by Rahimi and Recht (2008).

In recent years, the random feature model has been of particular interest due to its relevance to the *neural tangent kernel* (NTK) regime of Jacot, Gabriel, and Hongler (2018). In the NTK regime, the neural network weights are initialized with variances that ensure that the bottom layer weights remain close to their random initialization, and the network's behavior is well-approximated by a random feature model. The tractability of this regime, its amenability to convex analysis, and the overall relationship to a well-known learning model are appealing to theoreticians. At the time of the work (Hsu et al., 2021), however, the practical relevance of the NTK regime to the capabilities of neural networks was not well understood. While works like Damian, Lee, and Soltanolkotabi (2022) would later demonstrate the generalization limitations of the NTK regime in illustrative theoretical settings (i.e. the inability of networks in the NTK regime to adapt to low-dimensional structure as well as networks in the alternative *mean-field* regime), the weaknesses of the NTK are also apparent in their representational limitations. We crystallize the capabilities and limitations of this regime by studying the minimum width of random feature models that approximate target functions of varying smoothness and dimensionality, and contrasting these results with the capabilities of two-layer neural networks with arbitrary weights.

At a high level, we demonstrate that efficient approximation by random feature models with the ReLU activation $\sigma(t) = \mathrm{ReLU}(t) = \max(0, t)$ is possible if and only if the target is either low-dimensional or highly smooth (e.g. 1-Lipschitz). We summarize the main results (Theorems 2.1 and 2.2) as follows.

Informal Theorem 1.2. *For any Lipschitz constant L and dimension d, the minimum-width m of a random feature model f that approximates any L-Lipschitz target function h*

over domain $[-1,1]^d$ *to constant accuracy satisfies*

$$\begin{aligned} m &\leq \text{poly}(L) && \text{if } d \text{ is constant,} \\ m &\leq \text{poly}(d) && \text{if } L \text{ is constant,} \\ m &\geq \exp(d) && \text{if } L = \Omega(\sqrt{d}). \end{aligned}$$

Moreover, the negative result indicated by the final inequality can be realized even by simple single-index target functions like $x \mapsto \sin(d \langle \theta, x \rangle)$ for $\|\theta\|_2 = 1$, which are "intrinsically one-dimensional." This target can be easily approximated by a two-layer MLP with polynomial width but requires exponential width for a high-accuracy random feature model. This result suggests that the NTK regime is ill-suited for approximating simple single-index targets. Subsequent work (e.g. Bietti et al., 2022) would further demonstrate the generalization benefits of the mean-field regime for these single-index targets.

Although universal approximability had previously been established for random feature models (Barron, 1993), the results of Hsu et al. (2021) are the first to establish an asymptotically tight characterization of the joint impact of dimension, smoothness, and accuracy on the minimum width of random feature models. Previous theoretical work introduced representational results that applied to either the constant smoothness regime (e.g. Yehudai and Shamir, 2019; Andoni et al., 2014a) or the constant dimension regime (e.g. Ji, Telgarsky, and Xian, 2019; Bach, 2017). In contrast, our results are distinguished by the applicability to a wide range of scaling regimes and their capture of all such trade-offs with the same construction and lower bound.

Our positive representational results apply a two-stage construction that first approximates the target function with a linear combination of low-frequency trigonometric functions and then approximates these functions individually with a random feature model. The negative results are based on an intuitive linear algebraic argument: The dimensionality of the span of the random features can be bounded by the number of features m, but it can be

shown that the dimension of the space of low-frequency trigonometric functions of bounded smoothness grows exponentially with the smoothness measurement and input dimension. These results are also extended to a different class of target functions that satisfy a notion of Sobolev smoothness, as opposed to Lipschitzness.

The results discussed in Chapter 2 serve as a point of contrast for subsequent chapters in the dissertation, which study the representational capabilities of neural networks with different architectural constraints and inductive biases. Chapter 3 proves negative results about a much broader class of bounded-depth neural networks. Chapter 4 considers the efficient approximation capabilities of two-layer neural networks, except with a definition of efficiency that requires bounded weight norms, rather than bounded width. The constructions used in the positive results therein utilize random feature models to establish the existence of neural network interpolants with the desired properties. Further, these chapters consider single-index targets, just as Chapter 2 does.

Powers of depth and the discrete dynamical systems lens (Chapter 3)

Chapter 3, which presents the contributions of Sanford and Chatziafratis (2022), studies the representational capabilities of feedforward neural networks of variable depth by drawing a sharp separation between target functions that are easy to approximate with polynomial-width two-layer networks and those that require much deeper models. In contrast to its predecessor, these results apply to all feedforward neural networks with polynomial width, rather than those with random features or bounded weights. Specifically, the results of Sanford and Chatziafratis (2022) establish a sharp threshold of network depth needed to approximate iteratively composed functions of the form $x \mapsto g^k(x) = (g \circ \cdots \circ g)(x)$, where $g : \mathbb{R} \to \mathbb{R}$ is some continuous univariate mapping. These results rely on a novel connection to bifurcation theory and discrete dynamical systems that establishes a phase transition between a "stable regime" where g^k can be approximated by a two-layer MLP, and a "chaotic regime" where g^k requires depth linear in k.

This work is motivated by a long-standing question posed by the empirical success of deep neural networks: What are the theoretical benefits of depth, and what are the depth-vs-width tradeoffs? This question gives rise to the subfield of neural network *depth-separation*, which characterizes the class of functions that are representable (or approximately representable) by a neural network of a certain depth, width, and activation. For instance, Eldan and Shamir, 2016 presents a family of "radial" functions in $\mathbb{R}^d$ that are easily expressible with three-layer feedforward neural nets of small width, but require any approximating two-layer network to have exponentially (in dimension d) many neurons. In other words, they formally show that depth—even if increased by 1—can be exponentially more valuable than width.

Towards this direction, one typically identifies a target function with a "measure of complexity" to demonstrate the benefits of increasing the depth of a network or making other architectural changes. For example, the seminal work by Telgarsky, 2016 relies on the number of oscillations of a narrow family of triangle mappings on $[0, 1]$ that can be expressed recursively with deep neural networks. Other relevant notions of complexity to the expressivity of neural networks include the VC dimension (Warren, 1968; Anthony and Bartlett, 1999), the number of linear regions (Montufar et al., 2014; Arora et al., 2016) or activation patterns (Hanin and Rolnick, 2019), the Fourier spectrum (Barron, 1993; Eldan and Shamir, 2016; Daniely, 2017a; Bresler and Nagaraj, 2020), fractals (Malach and Shalev-Shwartz, 2019), topological entropy (Bu, Zhang, and Luo, 2020), Lipschitzness (Savarese et al., 2019; Hsu et al., 2021), global curvature and trajectory length (Poole et al., 2016; Raghu et al., 2017).

We build new connections between deep learning theory and dynamical systems by applying results from discrete-time dynamical systems to obtain novel depth-width tradeoffs for the expressivity of neural networks. Studying the chaotic itineraries of unimodal mappings reveals subtle connections between expressivity and different types of periods. These itineraries shed new light on the benefits of depth in the form of enhanced width lower bounds and stronger approximation errors.

Concretely, we extend the work of Telgarsky (2016) and subsequent works by Chatziafratis et al. (2019) and Chatziafratis, Nagarajan, and Panageas (2020) that study the expressivity of neural networks as a function of the periodicity of the target function. Telgarsky (2016) proves a separation between the representational power of networks with depth $\Theta(\sqrt{k})$ and $\Theta(k)$ by considering the task of approximating the iterated univariate *tent map* $h = g^k$, where $g : [0, 1] \to [0, 1]$ satisfies

$$g(x) = \begin{cases} 2x & x \in [0, \frac{1}{2}], \\ 2(1-x) & x \in [\frac{1}{2}, 1]. \end{cases}$$

He proves this result by showing that $g^k(x)$ oscillates 2^k times as x increases from 0 to 1. Since the number of oscillations produced by any ReLU neural network of depth L and width m can be bounded as $\exp(O(L \log m))$, he concludes that no neural network of depth $O(\sqrt{k})$ and width $m = \exp(O(\sqrt{k}))$ can approximate g^k.

A natural next step to this result is to ask whether the tent map is an exceptional mapping, or whether there exists a broader family of discrete dynamical systems $g : [0, 1] \to [0, 1]$ that produce such a separation. This question was partially, but not systematically, answered by (Chatziafratis et al., 2019; Chatziafratis, Nagarajan, and Panageas, 2020), which studied expressivity from the lens of discrete-time dynamical systems and extended Telgarsky's results to targets recursively defined by mappings g other than the tent map. Specifically, they characterize the hardness of representing $h = g^k$ as a function of the *periodicity* of g. g has a higher-order fixed point or *periodic point* x if that $g^p(x) = x$ for some p. If g has some higher-order fixed point, then deeper neural networks can efficiently approximate h, but shallower nets require exponential width, with an exponential base dependent on the periodicity p of mapping g.

Our results establish that g's periodicity alone is not the end of the story and that the understanding of depth-width tradeoffs and connections between recurrent neural networks

and discrete dynamical systems is improved by considering the concept of *cyclic itineraries*. The analysis of these itineraries produces nearly-optimal tradeoffs for NNs. In particular, the oscillatory behavior of a large family of univariate mappings g^k can be precisely characterized. This leads to sharper and nearly-optimal lower bounds for the width of NNs that approximate g^k. The resulting lower bounds pertain to a stronger notion of approximation error than those obtained by periodicity alone by Chatziafratis et al. (2019) and Chatziafratis, Nagarajan, and Panageas (2020), which can be vacuous for particular choices of g. Finally, connections between periodicity and other function complexity measures like the VC dimension and the topological entropy (Alsedà, Llibre, and Misiurewicz, 2000) can be elucidated. These measures undergo a phase transition that exactly coincides with the emergence of the chaotic regime based on periods.

The principal results (Theorems 3.6 and 3.18) can be summarized as follows.

Informal Theorem 1.3. *For any continuous univariate mapping $g : [0, 1] \to [0, 1]$ and any $k \in \mathbb{N}$, the following are equivalent:*

1. *The only cyclic itineraries of g are "doubling cycles" of length 2^q for some $q \in \mathbb{N}$.*
2. *The number of oscaillations of g^k can be bounded by some* $\mathrm{poly}(k)$.
3. *g^k can be approximated by a two-layer ReLU network of width* $\mathrm{poly}(k)$.
4. *The VC dimension of $\{g^k : k \in \mathbb{N}\}$ is finite and bounded.*

These results provide a new measurement of the complexity of target functions and link the expressivity of neural networks to a rich literature on discrete dynamical systems and bifurcation theory. Taken together, the results of Chapters 2 and 3 provide a more nuanced understanding of the abilities of two-layer neural networks to represent oscillatory target functions. While the former demonstrates that learnable weights are necessary to represent $\mathrm{poly}(d)$-oscillatory functions, the latter illuminates the conditions of target functions that necessitate a scaling of depth linear in the number of oscillations. While repeatedly composed

target functions may seem like an artificial class of functions to approximate with feedforward networks, our techniques were later applied to a study of the importance of a properly scaled random initialization for deep recurrent neural networks (RNNs) (Chatziafratis, Panageas, Sanford, and Stavroulakis, 2022), where iterative targets are more natural. Furthermore, the targets described herein resemble sequential *compositionality* tasks that have been of interest in the study of transformers and other sequential models, as discussed in Chapter 6.

Intrinsic dimensionality of bounded-norm shallow neural network interpolants (Chapter 4)

Chapter 4, which reflects the work of Ardeshir, Hsu, and Sanford (2023), characterizes the properties of the inductive biases of two-layer neural networks with bounded weights and potentially unbounded widths that interpolate training samples.

The study of *inductive biases*, or the preferences of learning algorithms for certain classes of functions, is a central machine learning research topic due to its relevance to generalization. The generalization properties of over-parameterized neural networks trained with gradient descent have been of particular interest to researchers due to their exhibition of *benign overfitting*, in which the learning algorithm produces a model that perfectly fits the training sample and nonetheless has favorable generalization properties (Belkin et al., 2018). One approach for understanding the generalization properties of neural networks is to study the inductive biases of the learning algorithm and then relate these biases to the generalization properties of the learned model.

Furthermore, a focus on how the inductive biases of neural networks interact with the structure of the target function is of particular interest because real-world datasets are often very high-dimensional, yet very structured. For instance, the spaces of "natural images" can be modeled as a low-dimensional manifold embedded in a high-dimensional pixel space. This theory of intrinsic dimensionality is supported by the successes of dimensionality reduction techniques revealing low-dimensional structure in high-dimensional data (Roweis and Saul, 2000). While high-dimensional learning problems are often thought to be intractable due

to the curse of dimensionality, target functions with low-dimensional structures may be learnable by sample-efficient algorithms.

Low intrinsic dimensionality can be studied in the context of *multi-index models*, in which the target is a function of a k-dimensional projection of the input, for some $k \ll d$, i.e.

$$x \mapsto \phi(Ux),$$

for $U \in \mathbb{R}^{d \times k}$ and $\phi : \mathbb{R}^k \to \mathbb{R}$. Works like Damian, Lee, and Soltanolkotabi (2022) have demonstrated that neural networks can adapt in a sample-efficient manner to this multi-index structure in a way that kernel methods cannot. This conclusion is reinforced by Chapter 2, which demonstrates that random feature models (and by extension, the NTK regime) cannot even efficiently approximate simple single-index targets (having $k = 1$), which are "intrinsically one-dimensional."

The contents of Chapter 4 consider the inductive bias for two-layer neural networks implied by a variational norm called the $\mathcal{R}$-*norm* and characterize the interactions between the $\mathcal{R}$-norm-minimizing inductive bias and the intrinsic dimensionality of the target function. The chapter focuses on the *approximation* and *generalization* consequences of preferring networks with small $\mathcal{R}$-norm in the context of learning explicit target functions.

It is well-known that the size of the weights can play a critical role in generalization properties of neural networks (Bartlett, 1996), and weight-decay regularization is a common practice in gradient-based training (Hinton, 1987; Hanson and Pratt, 1988). The $\mathcal{R}$-norm was introduced by Savarese et al. (2019) and Ongie et al. (2019) to capture the functional effect of controlling the size of network weights. The $\mathcal{R}$-norm of a target function $f \colon \mathbb{R}^d \to \mathbb{R}$ (written as $\|f\|_{\mathcal{R}}$) can be informally defined as the minimum weight norm of a two-layer neural network that interpolates or perfectly fits f. Notably, the network that witnesses the $\mathcal{R}$-norm of the target is not required to have a bounded width. Consequently, the $\mathcal{R}$-norm can be regarded as a measure of complexity across the space of infinite-width, two-layer neural

networks. Due to this connection to weight-norm regularization, explaining the consequences of the $\mathcal{R}$-norm inductive bias can advance our understanding of generalization in practical settings.

To consider the question of minimum $\mathcal{R}$-norm interpolants, we formalize the *variational problem* of finding a neural network of minimum $\mathcal{R}$-norm that interpolates a given labeled dataset. While the solutions to this variational problem for one-dimensional datasets have been fully characterized by Debarre et al. (2022) and Hanin (2021), the solutions for multi-dimensional datasets remained largely unexplored. Given the importance of single-index models and other target functions with low-intrinsic dimensionality, it is natural to ask whether the $\mathcal{R}$-norm inductive bias prefers neural networks that are themselves low-dimensional when interpolating multi-dimensional datasets with low-intrinsic dimensionality.

The principal approximation results discussed in Chapter 4 show that, even in cases where the dataset can be perfectly fit by a single-index target function, the solutions f to the $\mathcal{R}$-norm-minimizing variational problem do not resemble the single-index interpolants described by Savarese et al. (2019) and Hanin (2021). Rather, the $\mathcal{R}$-norm is far better minimized by a *multi-directional*[1] neural network f that averages several ridge functions pointing in different directions, each of which approximates a small fraction of the data.

The theorems are given for a dataset supported on the Boolean hypercube $\{-1,1\}^d$ and labeled by the *parity* function $\chi : \{-1,1\}^d \to \{-1,1\}$ with $\chi(x) = \prod_{i=1}^d x_i$.

Informal Theorem 1.4 ($\mathcal{R}$-norm minimizers of the parity dataset are not ridge functions)**.** *Any $\mathcal{R}$-norm-minimizing neural network f that interpolates the parity dataset is not a single-index function, and any single-index interpolate has a suboptimal $\mathcal{R}$-norm by a factor of $\Omega(\sqrt{d})$.*

This result is significant because χ is typically thought of as a single-index model whose

[1]By a multi-directional function, we mean a function that does not *only* depend on a one-dimensional projection of its input—i.e., a function that is not a ridge function.

most "natural" interpolant is a single-index model that resembles a sawtooth. Rather, the most $\mathcal{R}$-norm-efficient way to interpolate the dataset with respect to the $\mathcal{R}$-norm is with an ensemble-based construction, which expresses χ as an average over random "partial sawtooth functions." These results are extensible to a broader family of periodic ridge functions. Further results in Chapter 4 capture the generalization properties of the $\mathcal{R}$-norm-minimizing neural networks, which are shown to have a suboptimal generalization error for the parity function.

This work offers an alternative perspective on neural network representation. In contrast to Chapters 2 and 3, which study neural networks that approximate target functions over a continuous domain, Chapter 4 studies neural networks that interpolate a discrete dataset. As such, the results are more relevant to the study of neural network optimization and generalization, while the results of Chapters 2 and 3 capture more fundamental limitations of particular architectures. These chapters also differ in their notion of efficient approximation: Chapter 2 studies the minimum width of random feature models that approximate target functions. In contrast, Chapter 4 studies the minimum $\mathcal{R}$-norm of neural networks that interpolate a dataset. Given the relevance of weight-norm regularization (implicit and explicit) to neural network generalization and the fact that bounded-weight infinite-width networks can be approximated themselves by random feature models, the $\mathcal{R}$-norm-minimizing approximation provides a useful alternative to the standard notion of efficient approximation.

1.3.2 Trade-offs and limitations of modern sequential architectures

The final two chapters of this thesis, Chapters 5 and 6, introduce novel theoretical lenses to the fundamental capabilities of transformers and other sequential architectures Sequential architectures process length-N input sequences and parameterize functions of the form $\mathcal{X}^N \to \mathcal{Y}^N$ for potentially large sequence lengths N. While the differences in representational capabilities among feedforward networks are well understood and have been studied for decades, basic questions about the variations in capabilities among sequential architec-

tures remain open. The answers to those representational questions are empirically relevant due to the rapid proliferation of novel architectures—such as the *transformer* of (Vaswani et al., 2017) and the *Mamba* state-space model of (Gu and Dao, 2023)—and the lack of principled guidance for evaluating their tradeoffs.

Associative capabilities of multi-headed attention layers (Chapter 5)

Chapter 5 presents the research of Sanford, Hsu, and Telgarsky (2023) into the representational capabilities of single-layer transformer models, which are composed of multi-headed self-attention units and element-wise multi-layer perceptrons. The chapter introduces a collection of tasks designed to elucidate the expressive powers of attention units as a function of their embedding dimension and to contrast transformers from other sequential models. The tasks—*sparse averaging*, *pairwise detection*, and *triple detection*—are designed to measure the abilities of attention units to compute functions that require identifying relationships between collections of sequence elements. The analyses of these tasks include both positive and negative results, which rely on a communication complexity framework.

In contrast to earlier chapters of this thesis, we assume that the sequence length N scales exceedingly rapidly. We deem a sequential model an *efficient* solution to a task if it can be computed with an embedding dimension and a number of heads that increase at a rate at most logarithmical in N. These assumptions are motivated by the previously discussed scaling of modern transformer models. By focusing on this scaling regime, the results presented in the chapter capture relevant expressivity trade-offs that cannot be understood by considering worst-case or fixed-size models.

While a recent collection of work proves other theoretical results about the fundamental limitations of transformer architectures, the results of Sanford, Hsu, and Telgarsky (2023) are distinguished by a realistic scaling regime, a correspondence between positive and negative results, and a reliance on communication complexity. For instance, Hahn (2020) established that transformers *of fixed size* cannot represent certain hierarchical functions (like recognizing

Dyck languages with nested parentheses and brackets) as the sequence length N grows arbitrarily large. In contrast, Dyck languages with a bounded nesting depth can be efficiently represented by transformers (Yao et al., 2021). While illuminating, these results presuppose that transformers are of fixed size and ought to be able to recognize arbitrarily long recursive functions; in contrast, our results provide a quantitative lower bound on the embedding dimension m in the regime where m can grow with sequence length N.

The *sparse averaging* task is used to characterize the expressive capabilities of individual attention units for variable embedding dimension m. The task takes as input a sequence of embeddings, each of which encodes a fixed vector and a set of q indices. Its output consists of all averages of the vectors corresponding to the indices. This task is designed based on the intuition that self-attention units act to associate each element of the input sequence with a small number of other elements. Since the associations are captured by an attention matrix obtained by applying the softmax function to the rank-m matrix $XQ^{\mathsf{T}}KX^{\mathsf{T}} \in \mathbb{R}^{N\times N}$, the sparse averaging task measures the expressivity of these association matrices as a function of m. The results of Theorems 5.4 and 5.6 establish an approximately linear relationship between the embedding dimension m and the number of indices q that can be efficiently averaged by a single attention unit.

Informal Theorem 1.5. *There exists a self-attention unit with embedding dimension $m = O(q \log N)$ that can approximate the q-sparse averaging task, and any self-attention unit that approximates the task must have embedding dimension $m = \Omega(q)$.*

Further theoretical results establish that these tasks *cannot* be efficiently solved by recurrent architectures or standard feedforward models without requiring a polynomial parameter scaling with N.

The *pairwise detection* and *triple detection* evaluate the abilities of multi-headed attention layers to identify relationships between different pairs and triples of elements in the input sequence. Concretely, the pairwise detection task takes as input a sequence of integers and returns as output a binary sequence that indicates whether each element sums to zero

with any other element. Similarly, the triple detection task outputs a binary sequence that indicates whether each element sums to zero with any two other elements[2]. These tasks are introduced to establish that self-attention units are natural candidates for solving problems (such as the co-reference resolution linguistic task) that require assessing whether pairs of elements are related, but that they cannot solve problems that pertain to triples. These results are expressed in Theorems 5.16 and 5.22 and summarized as follows.

Informal Theorem 1.6. *There exists a single self-attention unit that can efficiently compute the pairwise detection task. However, any one-layer transformer composed of an H-headed self-attention unit with embedding dimension m that computes the triple detection task requires having $mH \geq N^{\Omega(1)}$.*

While the scope of these results is limited to single-layer transformers, we conjecture that the results extend to deeper transformers. This work also introduces a "triple-wise" attention model that can efficiently solve the triple detection task by applying the softmax to a third-order tensor, which suggests that attention-based models are not necessarily limited to pair-wise associations.

Taken together, these tasks establish sharp trade-offs between different architectural choices and scaling regimes. The sharp tradeoffs presented resemble those established in Chapters 2 to 4, albeit with different modeling assumptions to capture modern sequential architectures and novel proof techniques. The most significant omission of Chapter 5 is the lack of a characterization of the representational capabilities of transformers with multiple layers, which provided the impetus for the subsequent work of Sanford, Hsu, and Telgarsky (2024).

Parallelizability of deep transformer networks (Chapter 6)

This chapter presents the work of Sanford, Hsu, and Telgarsky (2024), which extends the preceding work to deep transformers. This work refines and solidifies the communica-

[2]These tasks are sequential analogs of the well-known 2SUM and 3SUM problems.

tion complexity perspective on transformers by demonstrating that deep transformers can implement parallelizable algorithms with a small number of sequential steps and by relating transformers to the *massively parallel computation* (MPC) model of Karloff, Suri, and Vassilvitskii (2010). Its primary technical contribution is a representational equivalence between depth-L transformers and MPC algorithms that run in $O(L)$ sequential steps. As a consequence of that relationship, the work provides efficient constructions of algorithmic tasks that can be implemented by logarithmic-depth transformers and establishes the optimality of those constructions. Furthermore, they demonstrate that alternative architectures cannot simulate this efficient parallel algorithm and require a polynomial depth to solve the same task.

As in the previous section, our results apply to transformers with realistic parameter scalings, where the sequence length N scales rapidly and the embedding dimension m, the number of heads H, and the depth L are bounded as a function of N. While other theoretical works relate deeper transformer models to automata (e.g. Liu et al., 2022), formal language classes (e.g. Angluin, Chiang, and Yang, 2023), or circuit complexity classes (e.g. Merrill and Sabharwal, 2023b), the results of Sanford, Hsu, and Telgarsky (2024) are distinguished by their ability to capture non-constant depth and to provide a fine-grained characterization of the parallelizability of transformers with matching lower and upper bounds. Our work adds context to these results by examining how depth changes the tractability of various tasks. For instance, graph connectivity—which is shown by Merrill and Sabharwal (2023b) to be impossible to represent with constant-depth and polynomial-size transformers—can indeed be represented by logarithmic-depth transformers, and we show that logarithmic-depth transformers are optimal for this task.

The powers of transformers to implement parallelizable algorithms can be demonstrated with the *k-hop induction heads* task, which generalizes the induction heads task of (Elhage et al., 2021), which requires a transformer to perform a k-iteration "bigram-matching" tasks sequentially. For instance, 2-hop induction heads task is to predict c for the final token:

The k-hop induction heads task is a compositionality task similar to the LEGO task of Zhang et al. (2023), where a sequential model must make several sequential associations between tokens to solve the task.

While a naive solution to the k-hop induction heads task would require a transformer to perform k iterations of bigram matching, we prove the existence of a logarithmic-depth transformer that can solve the k-hop induction heads task by composing an iterated "pointer-doubling" operation. This result captures a unique capability of transformers to perform parallelizable algorithms, which is not shared by other sequential models like recurrent neural networks or finite automata. Theorem 6.18 and Corollaries 6.19 and 6.26 are summarized as follows.

Informal Theorem 1.7. *For any $k \geq 1$ and sequence length N, there exists a transformer of depth $L = O(\log k)$ and embedding dimension $m = O(N^{0.01})$ that can solve the k-hop induction heads task with high accuracy. In contrast, any transformer of depth $L = o(\log k)$ and any state-space model (including RNNs and Mamba models) of depth $L = o(k)$ requires embedding dimension $m = \Omega(\sqrt{k})$ to solve the k-hop induction heads task.*

The negative results also apply to certain sub-quadratic-time attention models (such as the Performer model of Choromanski et al. (2022)). We empirically verify this characterization of the powers of depth in transformers, which suggests that these pointer-passing algorithms are not only a theoretical curiosity, but also a practical advantage of transformers over other sequential models.

The communication complexity framework discussed in the previous is further developed in this chapter by a detailed analysis of the ability of transformers to simulate MPC protocols and vice versa. Rather than merely a technique for proving lower bounds, this chapter asserts that the communication lens is theoretically necessary and empirically relevant to

the understanding of how transformers process sequential data. These results stress the difficulty of replicating these parallel capabilities with alternative sequential architectures. The newly introduced tasks such as k-hop induction heads can therefore be regarded as a key benchmark for variations on the transformer architecture, which future novel architectures should aim to match.

Chapter 2: Shallow random feature networks: dimensionality, smoothness, and width trade-offs

This chapter considers the following question: how well can depth-two ReLU networks with randomly initialized bottom-level weights represent smooth functions? We give near-matching upper and lower bounds for L_2-approximation as a function of the Lipschitz constant, the desired accuracy, and the dimension of the problem, as well as similar results in terms of Sobolev norms. These bounds suggest that target functions that are either low-dimensional or "highly smooth" can be efficiently approximated by such random feature networks. Functions that lack these properties—including even seemingly simple *ridge* functions—are inapproximable without exponential width. Our positive results employ tools from harmonic analysis and ridgelet representation theory, while our lower bounds are based on (robust versions of) dimensionality arguments.

The research presented in this chapter reflects the work of Hsu, Sanford, Servedio, and Vlatakis-Gkaragkounis (2021).

2.1 Introduction

2.1.1 Background and motivation

Celebrated results of Cybenko (1989), Funahashi (1989), and Hornik, Stinchcombe, and White (1989) establish the universality of depth-2 neural networks by showing that any continuous function on $\mathbb{R}^d$ can be approximated by a neural network with a single hidden layer. However, these results offer no upper bound (e.g., in terms of d) on the width (number of bottom-level gates) required, leaving unanswered many natural questions about the approximation power of neural networks, including:

- Which functions can be approximated by two-layer neural networks of subexponential width?
- Can tradeoffs be achieved between depth and width for neural network function approximation?
- Given the practical importance of random weight initialization, what are the representational capabilities of neural networks with some randomly drawn weights (say, at the bottom level)?

The first two questions above have been studied intensely in the approximation-theoretic and depth-separation literature; this chapter focuses on the third question. Random weight initializations play an important role in training neural networks in practice, and are also of theoretical interest; as we discuss later in this introduction, they have been well studied as a way of understanding different aspects of approximation and generalization.

This chapter examines the representational ability of depth-2 random feature (RF) ReLU networks. Such a network is equivalent to a linear combination of rectified linear units (ReLUs), where the weight vector and bias of each ReLU are randomly and independently chosen from a fixed distribution. The top-level combining weights of the ReLUs are allowed to be arbitrary; we give precise definitions in Section 2.2.1. This particular setting is notable because, as discussed later, several papers have given approximation-theoretic results in this regime. The ReLU activation is employed due to its popularity in both theory and practice; we expect that the results of our paper could be generalized to a range of other activation functions.

Our main goal is to understand the abilities and limitations of depth-2 RF ReLU networks for approximating smooth functions of various types. We focus on smooth functions because they are a natural class of functions to consider, and because non-smooth functions are difficult to approximate by various kinds of neural networks. Indeed, several authors (e.g. Telgarsky, 2016; Daniely, 2017a) have established lower bounds on the width of neural

networks that approximate certain non-smooth functions by taking advantage of the fact that such functions can be highly oscillatory (have many "bumps") and can require many gates to approximate each "bump."

Our chief focus is on functions over the d-dimensional solid cube $[-1,1]^d$ whose smoothness is measured in two different ways. Our main results are about approximating functions on $[-1,1]^d$ with bounded *Lipschitz constants*; in Section 2.5, we also consider functions on $[-1,1]^d$ (satisfying certain periodicity conditions) with bounded *Sobolev norms*.

2.1.2 Our results

The main contribution of this work is to pose and answer the following question:

> *What is the minimum number of random ReLU features required so that (with high probability) there exists some linear combination of those features that closely approximates any sufficiently smooth function?*

This minimum number of random ReLU features is equivalent to the minimum width required for a depth-2 RF ReLU network to approximate the smooth function in question. We give full details about our setting in Section 2.2.1, and here only touch on some of the main aspects:

- *Random ReLU features* are functions from $\mathbb{R}^d$ to $\mathbb{R}$ that are drawn independently from some fixed distribution. These take the form $x \mapsto \mathrm{ReLU}(\langle \mathbf{w}, x\rangle + \mathbf{b})$ where $\mathrm{ReLU}(z) := \max(z, 0)$ and $\mathbf{w}$ and $\mathbf{b}$ are random variables taking values in $\mathbb{S}^{d-1}$ and $\mathbb{R}$ respectively.

- Our notion of *close approximation* refers to the L_2 distance between functions over the uniform distribution on the solid cube; we say that f is an ϵ-approximator for g if

$$\|f - g\|_{[-1,1]^d} := \sqrt{\mathop{\mathbb{E}}_{\mathbf{x}\sim \mathrm{Unif}([-1,1]^d)} [\|f(\mathbf{x}) - g(\mathbf{x})\|^2]} \leq \epsilon.$$

- As mentioned above, we chiefly measure the smoothness of a function by its Lipschitz constant. In Section 2.5, we extend our results to measure smoothness in terms of Sobolev norms.

The main results provide tight upper and lower bounds on the minimum width required for both Lipschitz and Sobolev smooth functions. The upper and lower bounds match up to polynomial factors (equivalently, up to constant factors in the exponent). The sharpest forms of our bounds involve the number of integer points in certain Euclidean balls; below, we present informal statements of our upper and lower bounds for Lipschitz functions with explicit asymptotics given for clarity:

Theorem 2.1 (Informal positive result for L-Lipschitz functions)**.** *For any $\epsilon, L > 0$ with $L/\epsilon \geq 2$, there exists a suitable distribution over ReLU features that satisfies the following property. For any L-Lipschitz function $h : [-1, 1]^d \to \mathbb{R}$ and some width m satisfying*

$$m = \min\left(d^{\tilde{O}\left(L^2/\epsilon^2\right)}, (L/\epsilon)^{\tilde{O}(d)}\right),$$

there exists a two-layer m-width random ReLU feature network f that satisfies

$$\|f - h\|_{[-1,1]^d} \leq \epsilon,$$

with probability 0.9 *over i.i.d. random ReLU features drawn from the distribution.*

Theorem 2.2 (Informal negative result for L-Lipschitz functions)**.** *Fix any $\epsilon, L > 0$. For some width m satisfying*

$$m = \min\left(d^{\tilde{\Omega}\left(L^2/\epsilon^2\right)}, (L/\epsilon)^{\tilde{\Omega}(d)}\right),$$

there exists an L-Lipschitz function $h : [-1, 1]^d \to \mathbb{R}$ such that with probability at least $\frac{1}{2}$ over a draw of m i.i.d. random ReLU features, $\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}$, every two-layer random ReLU

feature network h of width m with those features as bottom-layer weights has

$$\|f - h\|_{[-1,1]^d} > \epsilon.$$

Table 2.1 summarizes these results, as well as our analogs for functions in Sobolev balls.

Result	Smoothness	Minimum width m	Theorem
Postive	Lipschitz $\leq L$	$\min(d^{\tilde{O}(L^2/\epsilon^2)}, (L/\epsilon)^{\tilde{O}(d)})$	Thm. 2.1 / 2.9
Negative	Lipschitz $\leq L$	$\min(d^{\tilde{\Omega}(L^2/\epsilon^2)}, (L/\epsilon)^{\tilde{\Omega}(d)})$	Thm. 2.2 / 2.15
Positive	H^s norm $\leq \gamma$	$\min(d^{\tilde{O}(s\gamma^{2/s}/\epsilon^{2/s})}, (\gamma/\epsilon)^{\tilde{O}(d/s)})$	Thm. 2.25
Negative	H^s norm $\leq \gamma$	$\min(d^{\tilde{\Omega}(s\gamma^{2/s}/\epsilon^{2/s})}, (\gamma/\epsilon)^{\tilde{\Omega}(d/s)})$	Thm. 2.27

Table 2.1: Our upper and lower bounds on the minimum width needed for a random ReLU feature network to ϵ-approximate a function over $L_2([-1,1]^d)$ with either bounded Lipschitz constant L, or bounded order-s Sobolev norm γ (and periodic boundary conditions).

2.1.3 Discussion

Relationship between Lipschitzness and depth separation. Our results shed light on a question posed by Safran, Eldan, and Shamir, 2019 about the approximation power of unconstrained two-layer networks. They ask whether there exists a d-dimensional 1-Lipschitz function h that can be represented by a three-layer neural network with poly(d) neurons but requires width $\exp(\Omega(d))$ to be approximated by a two-layer network. As one of their main results, they answer this question in the negative for pointwise approximation when h is a radial function (depending only on $\|x\|_2$) over the unit ball, by showing that any such function can be efficiently approximated by a poly(d) width depth-2 network. Our results imply that the answer is also negative for L_2-approximation of *arbitrary* 1-Lipschitz functions (which need not be radial) over $[-1,1]^d$; this follows from our upper bounds for the case that $L = 1$ and ϵ is any constant, which establish the existence of approximators that are poly(d)-width, two-layer random ReLU feature networks. Our results do not answer their question outright, because showing that every 1-Lipschitz function can be approximated with respect

to the L_2 norm over $[-1,1]^d$ by a depth-2 network of poly(d) width does *not* imply that every 1-Lipschitz function is uniformly approximable by such a network.

Implications for learnability with gradient descent. Our upper bounds on the width that suffices to approximate Lipschitz functions are also useful for proving learnability hardness results for neural networks with more than two layers. Malach et al. (2021b) establish this connection between hardness of approximation and hardness of learning by showing that any function that cannot be weakly approximated by a network with three layers cannot be learned by gradient descent applied to a neural network of *any* depth, given certain assumptions about the random weight initialization and bounds on the number of units in the network and number of steps of gradient descent. Their result hinges on a technical lemma (their Lemma B.2), which shows that L-Lipschitz functions can be approximated by three-layer neural networks with bounded width. By replacing that lemma with our Theorem 2.9, their result can be strengthened to say that any function not weakly approximable by *two*-layer neural networks is not learnable by gradient descent for networks of any depth that obey their assumptions.

Neural tangent kernel and random features. The *neural tangent kernel* (NTK) regime of Jacot, Gabriel, and Hongler (2018) considers the behavior of neural networks in the infinite-width limit, where the dynamics of the network are governed by a kernel. Networks trained in this regime bear a close relationship to random feature models since bottom-layer weights in the NTK regime remain close to their initialization throughout training. Our results can be seen as modeling a finite-width version of the NTK regime, and the positive and negative results of this work can be interpreted as capturing the capabilities and limitations of the NTK regime. In particular, our negative results suggest that networks in the NTK scaling regime cannot efficiently approximate certain smooth functions, including some *single-index* targets (i.e., functions that depend only on a single coordinate of the input), such as the sinusoidal functions we use in our lower bound construction. In contrast, target functions

with low-intrinsic dimension can be approximated and provably learned efficiently by neural networks trained *outside* the NTK regime (e.g. Damian, Lee, and Soltanolkotabi, 2022; Bietti et al., 2022).

2.1.4 Techniques

In this section, we give a high-level overview of the ideas that underlie our upper and lower bounds.

2.1.4.1 Positive results

Our minimum-width upper bounds state that for any fixed function of the relevant sort, given sufficiently many independent random ReLU features, some linear combination of those features most likely approximates the function. These techniques bear similarity to the proof approach of Bresler and Nagaraj (2020), who show similar positive results for approximating smooth targets with two-layer neural networks using a random feature construction. We argue this in three steps. (Below, we only discuss the Lipschitzness smoothness measure, but the Sobolev case follows the same basic steps.)

1. The first step shows that for any L-Lipschitz target function h, there exists a low-degree trigonometric polynomial P that closely approximates h. We establish the existence of this trigonometric polynomial using the fact that any function in $L_2([-1,1]^d)$ can be expressed as a (potentially infinite) linear combination of sinusoidal functions, due to the existence of a Fourier representation for h. We use the Lipschitzness of h to show that high-frequency terms have negligibly small coefficients in the representation, which we drop to obtain a low-degree approximation P.

2. The second step expresses P as an infinite mixture of random ReLU features (à la Barron, 1993; Murata, 1996; Rubin, 1998; Candès, 1999). That is, for some distribution over biases $\mathbf{b}$ and weights $\mathbf{w}$ (which depends on L, ϵ, and d, but not h, and takes values

in $\mathbb{R} \times \mathbb{S}^{d-1}$), P can be written as

$$P(x) = \mathop{\mathbb{E}}_{\mathbf{b},\mathbf{w}} \left[h(\mathbf{b}, \mathbf{w}) \mathrm{ReLU} \left(\langle \mathbf{w}, x \rangle - \mathbf{b} \right) \right]$$

for some function $h(\mathbf{b}, \mathbf{w})$. Intuitively, this is possible because each sinusoidal component of P is a ridge function (a function that depends only on a one-dimensional projection of its input).

3. Finally, using a standard concentration argument, we show that the empirical average of sufficiently many random ReLUs gives a close approximation to P with high probability. Consequently, the overall weighted combination of random features closely approximates h.

2.1.4.2 Negative results

Our lower bounds are proved using a dimensionality argument, stemming from the simple observation that linear combinations of m features (functions) can span at most m dimensions in the function space $L_2([-1, 1]^d)$. The key is to give $N \gg m$ candidate functions $\varphi_1, \ldots, \varphi_N$ that are orthonormal in $L_2([-1, 1]^d)$. With such a set of functions in hand, any fixed outcome of a draw of m random features will be such that linear combinations of those m features cannot closely approximate more than a small fraction of the N functions because no m-dimensional subspace can be close to a large fraction of N orthonormal functions. (This kind of dimensionality argument has been used in several prior works, including Barron, 1993; Yehudai and Shamir, 2019; Kamath, Montasser, and Srebro, 2020 and elsewhere.)

Specializing to our context, to give a lower bound on the minimum width of random ReLU feature networks needed to approximate L-Lipschitz functions, it suffices to construct a large family of orthonormal L-Lipschitz functions. We do this with L-Lipschitz sinusoidal functions of the form $\sqrt{2} \sin(\pi \langle K, x \rangle)$ where $K \in \mathbb{Z}^d$. The quantity $\|K\|_2$ controls the Lipschitz constant of these functions, and as our analysis shows, the tradeoff between the

number of functions in the family (which increases with the allowed range of $\|K\|_2$ and controls our width bound m) and the Lipschitz constant L yields a lower bound that is quite close to our upper bound for L-Lipschitz functions.

The simple dimensionality argument sketched above establishes that some function among the N orthonormal functions is hard to approximate (in fact, that most of them are hard), but it does not yield an *explicit* hard function. By requiring the N orthonormal functions $\varphi_1, \ldots, \varphi_N$ to satisfy a natural symmetry property with respect to the random ReLU features, it is possible to get a lower bound for a single explicit function φ_1. Following this approach, we also give a quantitatively slightly weaker lower bound on the minimum width that random ReLU networks need to approximate an explicit function φ_1.

2.1.5 Related work

Since the pioneering universal approximation results for deterministic two-layer networks (Cybenko, 1989; Funahashi, 1989; Hornik, Stinchcombe, and White, 1989) mentioned in the introduction, many subsequent works have established quantitative bounds on the width that such networks require to approximate certain functions.[1] Random feature networks have also been the subject of considerable study owing to their connection to kernel methods (Neal, 1996; Rahimi and Recht, 2008; Cho and Saul, 2009) and, in particular, the Neural Tangent Kernel (NTK). Jacot, Gabriel, and Hongler (2018) argue that training neural networks with gradient descent with small step-sizes results in a learning rule similar to that obtained by a kernel method with the NTK. When the network weights are randomly initialized, then a finite-width NTK corresponds to a linear combination of random ReLUs. Both random ReLU feature networks and the finite-width NTK enjoy the same universal approximation property of deterministic networks (Sun, Gilbert, and Tewari, 2018; Ji, Telgarsky, and Xian, 2019), and hence quantitative bounds on the network width required to approximate families

[1] Our discussion here focuses on works that give non-asymptotic bounds. Pinkus (1999, Section 6) gives a review of asymptotic rates of approximation by neural networks of width r as $r \to \infty$ (regarding the dimension d as fixed).

of functions are of significant interest.

Positive results. A line of inquiry started by Barron (1993) (see also Klusowski and Barron, 2018) investigates upper bounds on the width of deterministic two-layer networks needed to approximate functions whose smoothness is measured in terms of their Fourier transforms. Although these results do not deal with random feature networks and hence are incomparable to ours, they do use randomization in the proof. Specifically, a target function is represented as a mixture of activation functions drawn from a target-specific distribution, and a finite-width depth-2 network approximating the function is obtained by sampling. Our results use a similar overall approach, but with the crucial difference that in our random feature setting, our distribution of ReLUs does not depend on the target function.

Perhaps the works on random ReLU feature networks that are most closely related to our own upper bounds are those of Andoni et al. (2014a), Yehudai and Shamir (2019), Bach (2017), and Ji, Telgarsky, and Xian (2019), all of which prove approximation-theoretic results by representing a target function as the expected value of weighted activation functions drawn from some distribution.

- Theorem 3.1 of Andoni et al. (2014a) shows how neural networks with complex-valued weights and exponential activation functions can approximate polynomials of bounded degree. Their bounds have an exponential dependence on that degree, which translates to an exponential dependence on the Lipschitz constant L even for constant dimension d; in contrast, our bounds are exponential in $\min\{d, L^2/\epsilon^2\}$, which can be much better if d is small.

- Yehudai and Shamir (2019) study two-layer random ReLU feature networks (as we do), but like Andoni et al. (2014a) focus on approximating polynomials of bounded degree. Since they consider a more stringent notion of L_∞-approximation (over the unit ball), their upper bounds on network width (see their Theorems 3.3 and 3.4) are

more pessimistic than ours and depend exponentially on the square of the polynomial degree.

- Proposition 3 of Bach (2017) and Theorem E.1 of Ji, Telgarsky, and Xian (2019) imply (or directly give) upper bounds on the width of two-layer random ReLU reature networks (or finite-width NTK) to approximate Lipschitz functions. Similar to Yehudai and Shamir (2019), they consider an L_∞ notion of approximation, so they obtain upper bounds that always are exponential in the dimension d.

Negative results. A number of recent and classical papers give width lower bounds for arbitrary (deterministic-weight) two-layer networks that approximate certain types of multivariate functions. Maiorov (1999) gives asymptotically tight upper and lower bounds on the error in approximating functions from a Sobolev class achieveable by any two-layer network of a given width. The asymptotic nature of Maiorov's results (and proof techniques) means that the results do not imply lower bounds on the network width required to achieve a given error rate ϵ unless ϵ is sufficiently small, possibly as a function of dimension. Our results differs from Maiorov's and other related results from the approximation theory literature by elucidating the interplay between the dimension and the error in both upper- and lower bounds.

More recently, Eldan and Shamir (2016) and Safran and Shamir (2017) give $\exp(d)$-type lower bounds on the width that depth-2 networks require to L_2-approximate certain simple functions under certain probability measures on $\mathbb{R}^d$. In Eldan and Shamir (2016) the function being approximated is not explicit, and in Safran and Shamir (2017) the lower bound is only for very high-accuracy approximation (to error at most $1/d^4$). In both works the relevant probability measures are rather involved. In contrast, our lower bounds hold only for two-layer random feature networks, but they are for simple explicit functions, for large (constant) values of the approximation parameter, and for L_2-approximation with respect to the uniform distribution over $[-1, 1]^d$. In other relevant work on two-layer lower bounds,

Martens et al. (2013) and Daniely (2017a) give $\exp(d)$-type (or better) width lower bounds for depth-2 networks approximating certain functions with large Lipschitz constants, but these lower bounds require a weight bound on the top-level combining gate. In contrast, our lower bonds for random feature networks have no restrictions on the weights of the top-level gate.

The work of Sonoda et al. (2020), which analyzes limitations on the approximation abilities of two-layer networks of random ReLU activation functions, is relevant to our lower bounds. Their lower bounds are independent of the width of the network; they give functions that cannot be approximated by random feature networks of *any* (potentially infinite) width. However, their lower bounds are for an extremely strong notion of approximation, namely L_2 approximation over all of $\mathbb{R}^d$ (without any weighting by a probability distribution).

Our lower bound idea of exploiting symmetry to obtain an *explicit* function that is difficult to approximate was inspired by Yehudai and Shamir (2019). Our approach for non-explicit lower bounds is quite similar to Theorem 19 of Kamath, Montasser, and Srebro (2020), which bounds the dimension of the space of all linear combinations of feature functions; similar to the lower bound of Kamath, Montasser, and Srebro (2020) (but unlike Yehudai and Shamir (2019)), our lower bounds hold regardless of the size of the weights used in the linear combination of the bottom-level random features.

Finally, we remark that while we do not consider networks of depth larger than two, our paper was in large part inspired by results from the literature on depth separation. Telgarsky (2016), Eldan and Shamir (2016), and Daniely (2017a) all prove lower bounds by constructing highly oscillatory functions and showing that shallow networks must be wide in order to approximate these functions. Safran, Eldan, and Shamir (2019) prove lower bounds on 1-Lipschitz functions that are non-oscillatory, such as $x \mapsto \max\{0, -\|x\| + 1\}$; however, these bounds only hold in the high-accuracy regime with small ϵ. These works motivated us to directly study the relationship between the Lipschitz constant of a target function and

the width needed to approximate it.

2.2 Preliminaries

For a positive integer $d \in \mathbb{Z}^+$, let $[d] := \{1, 2, \ldots, d\}$. The vectors $\vec{0} := (0, \ldots, 0) \in \mathbb{R}^d$ and $\vec{1} := (1, \ldots, 1) \in \mathbb{R}^d$ are, respectively, the all-zeros and all-ones vectors. Let $\mathbb{S}^{d-1} := \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$ denote the unit sphere in $\mathbb{R}^d$. Let $\|h\|_{\text{Lip}}$ denote the Lipschitz constant of $h \colon \mathbb{R}^d \to \mathbb{R}$ with respect to the Euclidean metric (i.e., the least L s.t. h is L-Lipschitz w.r.t. $\|\cdot\|_2$). We use bold font to denote random variables and write "$\mathbf{x} \sim \mathcal{D}$" to indicate that random variable $\mathbf{x}$ is distributed according to distribution $\mathcal{D}$.

We use the following notations for a multi-index $K \in \mathbb{N}^d$ (where $\mathbb{N} := \{z \in \mathbb{Z} : z \geq 0\}$). Let $|K| := \sum_{i=1}^d K_i$, $\|K\|_2 := (\sum_{i=1}^d K_i^2)^{1/2}$, and $K! := \prod_{i=1}^d (K_i!)$. Let $x^K := \prod_{i=1}^d x_i^{K_i}$ for $x \in \mathbb{R}^d$. Lastly, let $D^{(K)}h$ be the order-$|K|$ partial derivative of a function $h(x)$ with respect to x^K.

2.2.1 Random ReLU feature neural network approximation

Throughout the paper, we treat a two-layer random ReLU feature network as a random features model. The upper bounds in this paper demonstrate the representational powers of linear combinations of these random features, while the lower bounds demonstrate their limitations.

We define a family of distributions over the parameters of random ReLU activations. Note that our lower-bounds in Theorems 2.15, 2.23, 2.27, and 2.28 hold for *all* such distributions $\mathcal{D}$, while our upper-bounds in Theorems 2.9 and 2.25 hold for some fixed $\mathcal{D}$, which depends on an upper bound on the Lipschitz norm of the target function but not on the target function itself.

Definition 2.1 (Symmetric ReLU parameter distributions)**.** A product distribution $\mathcal{D} := \mathcal{D}_{\text{bias}} \times \mathcal{D}_{\text{weights}}$ over $\mathbb{R} \times \mathbb{S}^{d-1}$ is a *symmetric ReLU parameter distribution* if the coordinates

of $\mathcal{D}_{\text{weights}}$ are invariant to permutation. That is, $\mathcal{D}_{\text{weights}} = \pi \circ \mathcal{D}_{\text{weights}}$ for any permutation π of $[d]$.

Given a distribution over random ReLU parameters, we now introduce the full random ReLU features model. We define a notion of approximation and formalize the *minimum width* of the network (or the minimum number of random features to combine) needed to obtain a sufficiently accurate approximation with high probability.

Definition 2.2 (Minimum-width random ReLU feature network approximation)**.** Consider a symmetric ReLU parameter distribution $\mathcal{D}$, a measure μ over $\mathbb{R}^d$, and a network width $m \in \mathbb{Z}^+$. For each $i \in [m]$, we draw a *random network feature* $\mathbf{g}^{(i)} \in L_2(\mu)$ independently by drawing $(\mathbf{b}^{(i)}, \mathbf{w}^{(i)})$ from $\mathcal{D}$ and letting $\mathbf{g}^{(i)}(x) := \text{ReLU}(\langle \mathbf{w}^{(i)}, x \rangle - \mathbf{b}^{(i)})$.

Given $\epsilon, \delta > 0$ and a function $h : \mathbb{R}^d \to \mathbb{R}$ with bounded $\|h\|_\mu$, we define $\text{MinWidth}_{h,\epsilon,\delta,\mu,\mathcal{D}}$ to be the smallest $m \in \mathbb{Z}^+$ such that the following holds: With probability at least $1 - \delta$ over $\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}$,

$$\inf_{f \in \text{Span}\left(\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(r)}\right)} \|f - h\|_\mu \leq \epsilon.$$

2.2.2 Orthonormal basis for function space

We use $\langle \cdot, \cdot \rangle$ to denote the standard Euclidean inner product in $\mathbb{R}^d$ (and occasionally regard multi-indices $K \in \mathbb{N}^d$ as elements of $\mathbb{R}^d$). For a probability measure μ on $\mathbb{R}^d$, $L_2(\mu)$ denotes the space of square-integrable functions with inner product denoted by

$$\langle f, g \rangle_\mu := \mathop{\mathbb{E}}_{\mathbf{x} \sim \mu} [f(\mathbf{x}) g(\mathbf{x})] = \int_{\mathbb{R}^d} f(x) g(x) \mu(\mathrm{d}x).$$

Because many of the results of this chapter pertain to the uniform distribution over $[0, 1]^d$, we use the notations $L_2([-1, 1]^d)$ and $\langle \cdot, \cdot \rangle_{[-1,1]^d}$, and fix a particular orthonormal basis $\mathcal{T} = \{T_K : K \in \mathbb{Z}^d\}$ for $L_2([-1, 1]^d)$ based on trigonometric polynomials. We define this basis and prove properties related to its orthonormality (Section 2.2.2.2), derivatives (Section 2.2.2.3), and cardinality (Section 2.2.2.4) in the following sections.

Recall the definition of an orthonormal basis for the space $L_2(\mu)$:

Definition 2.3 (Orthonormal basis)**.** A countable set $\mathcal{G} \subset L_2(\mu)$ is an *orthonormal basis* for $L_2(\mu)$ if $\langle g, \tilde{g} \rangle_\mu = \mathbb{1}\{g = \tilde{g}\}$ for all $g, \tilde{g} \in \mathcal{G}$ and $\text{Span}(\mathcal{G}) = L_2(\mu)$.

We frequently apply the following standard facts about orthonormal bases:

Fact 2.3 (Facts about orthonormal bases)**.** *For some measure μ, let $\mathcal{G}$ be an orthonormal basis for $L_2(\mu)$. Any $f, \tilde{f} \in L_2(\mu)$ satisfy $f = \sum_{g \in \mathcal{G}} \alpha_g g$ and $\tilde{f} = \sum_{g \in \mathcal{G}} \tilde{\alpha}_g g$ for some real $(\alpha_g)_{g \in \mathcal{G}}$ and $(\tilde{\alpha}_g)_{g \in \mathcal{G}}$. Furthermore:*

- $\alpha_g = \langle f, g \rangle_\mu$*;*
- $\|f\|_\mu^2 = \sum_{g \in \mathcal{G}} \alpha_g^2$ *(Parseval); and*
- $\langle f, \tilde{f} \rangle_\mu = \sum_{g \in \mathcal{G}} \alpha_g \tilde{\alpha}_g$ *(Plancherel).*

2.2.2.1 Trigonometric polynomial basis definition

We define the basis of trigonometric polynomials $\mathcal{T}$ as

$$\mathcal{T} := \left\{ T_K : K \in \mathbb{Z}^d \right\},$$

where

$$T_K(x) := \begin{cases} 1 & K = \vec{0} \\ \sqrt{2} \sin\left(\pi \langle K, x \rangle\right) & K \in \mathcal{K}_{\sin} \\ \sqrt{2} \cos\left(\pi \langle K, x \rangle\right) & K \in \mathcal{K}_{\cos}, \end{cases} \tag{2.1}$$

and $\mathcal{K}_{\sin}$ and $\mathcal{K}_{\cos}$ form a partition of $\mathbb{Z}^d \setminus \{\vec{0}\}$[2] and are defined as

$$\mathcal{K}_{\sin} := \left\{K \in \mathbb{Z}^d \setminus \{\vec{0}\} : K_i > 0, \text{ where } i = \min\{j \in [d] : x_j \neq 0\}\right\},$$
$$\mathcal{K}_{\cos} := \left\{K \in \mathbb{Z}^d \setminus \{\vec{0}\} : K_i < 0, \text{ where } i = \min\{j \in [d] : x_j \neq 0\}\right\}.$$

The set $\mathcal{T}$ is a useful family of functions for both our upper and our lower bounds on the minimum-width random ReLU feature network needed to approximate Lipschitz functions. The fact that $\mathcal{T}$ is an orthonormal basis for $L_2([-1,1]^d)$ (Fact 2.5) permits us to express other functions in $L_2([-1,1]^d)$ as a linear combination of the elements of $\mathcal{T}$. As we show in Fact 2.6, those orthogonality properties of the elements of $\mathcal{T}$ are maintained even after taking partial derivatives. In addition, every function in $\mathcal{T}$ is a ridge function (that is, $T_K(x) = \phi_K(\langle K, x\rangle)$ for some $\phi_K : \mathbb{R} \to \mathbb{R}$), which, as we will see later, means (very usefully for us) that T_K is easily approximated by linear combinations of shifted ReLUs. Finally, the Lipschitz constant of all functions in $\mathcal{T}$ is bounded: $\|T_K\|_{\text{Lip}} \leq \sqrt{2}\pi \|K\|_2$.

2.2.2.2 Orthonormality of $\mathcal{T}$

To prove that $\mathcal{T}$ is orthogonal, we rely on the following fact from integral calculus.

Fact 2.4 (Integrals of multivariate sinusoids)**.** *For each $K \in \mathbb{Z}^d$,*

$$\int_{[-1,1]^d} \cos(\pi \langle K, x\rangle)\, \mathrm{d}x = 2^d \cdot \mathbb{1}\{K = \vec{0}\} \quad \& \quad \int_{[-1,1]^d} \sin(\pi \langle K, x\rangle)\, \mathrm{d}x = 0.$$

Proof. We use a simple inductive argument on d to evaluate the first integral. The base case $d = 1$ is straightforward, so assume $d > 1$ and define $x_{-1} = (x_2, \ldots, x_d) \in \mathbb{R}^{d-1}$ for any $x \in \mathbb{R}^d$. Assume inductively that

$$\int_{[-1,1]^{d-1}} \cos(\pi \langle K_{-1}, x_{-1}\rangle)\, \mathrm{d}x_{-1} = 2^{d-1}\mathbb{1}\{K_{-1} = \vec{0}\}.$$

[2]Note that this partition of $\mathbb{Z}^d - \{\vec{0}\}$ is an arbitrary one. The only property this partition is designed to satisfy is that if K corresponds to $\sin(\pi\langle K, x\rangle)$, then $-K$ must correspond to $\cos(-\pi\langle K, x\rangle)$ (and vice versa).

By the cosine addition formula, we have that:

$$\begin{aligned}
&\int_{[-1,1]^d} \cos\left(\pi\langle K, x\rangle\right) \mathrm{d}x \\
&= \int_{[-1,1]^d} \left[\cos\left(\pi K_1 x_1\right) \cos\left(\pi\langle K_{-1}, x_{-1}\rangle\right) - \sin\left(\pi K_1 x_1\right) \sin\left(\pi\langle K_{-1}, x_{-1}\rangle\right)\right] \mathrm{d}x \\
&= \left[\int_{-1}^{1} \cos\left(\pi K_1 x_1\right) \mathrm{d}x_1\right] \left[\int_{[-1,1]^{d-1}} \cos\left(\pi\left\langle K_{-1}, x_{-1}\right\rangle\right) \mathrm{d}x_{-1}\right] \\
&\quad - \left[\int_{-1}^{1} \sin\left(\pi K_1 x_1\right) \mathrm{d}x_1\right] \left[\int_{[-1,1]^{d-1}} \sin\left(\pi\left\langle K_{-1}, x_{-1}\right\rangle\right) \mathrm{d}x_{-1}\right] \\
&= 2 \cdot \mathbb{1}\{K_1 = 0\} \left[\int_{[-1,1]^{d-1}} \cos\left(\pi\left\langle K_{-1}, x_{-1}\right\rangle\right) \mathrm{d}x_{-1}\right] = 2^d \cdot \mathbb{1}\{K = \vec{0}\}.
\end{aligned}$$

The second claim follows by a nearly identical inductive argument, which we omit. □

Fact 2.5. *$\mathcal{T}$ is an orthonormal basis for $L_2([-1,1]^d)$.*

Proof. First, we make use of the well-known fact that the constant 1 function, along with $z \mapsto \sqrt{2}\sin(\pi k z)$ and $z \mapsto \sqrt{2}\cos(\pi k z)$ for all $k \in \mathbb{Z}^+$, collectively form an orthonormal basis for $L_2([-1,1])$. (For details, see Dym and McKean, 1972.) Thus, the d-fold Cartesian product of this collection is an orthonormal basis for $L_2([-1,1]^d)$.[3] Each function in this basis is a product of d functions—one per variable, and each being either a constant, sine, or cosine as above—and can be rewritten as a linear combination of functions from $\mathcal{T}$ using basic product-to-sum trigonometric identities. Thus, $\mathrm{Span}\left(\mathcal{T}\right) = L_2([-1,1]^d)$.

To complete our proof, it remains to show that all elements of $\mathcal{T}$ are orthogonal and have unit norm. It suffices to show that $\langle T_K, T_{K'}\rangle_{[-1,1]^d} = \mathbb{1}\{K = K'\}$ for all $K, K' \in \mathbb{Z}^d$. There are six possible scenarios for this claim depending on which partitioning subsets of $\mathbb{Z}^d$ contain K and K': *(1)* $K, K' \in \mathcal{K}_{\cos}$; *(2)* $K, K' \in \mathcal{K}_{\sin}$; *(3)* $K = K' = \vec{0}$; *(4)* $K \in \mathcal{K}_{\cos}, K' = \vec{0}$ or $K = \vec{0}, K' \in \mathcal{K}_{\cos}$; *(5)* $K \in \mathcal{K}_{\sin}, K' = \vec{0}$ or $K = \vec{0}, K' \in \mathcal{K}_{\sin}$; and *(6)* $K \in \mathcal{K}_{\sin}, K' \in \mathcal{K}_{\cos}$ or $K \in \mathcal{K}_{\cos}, K' \in \mathcal{K}_{\sin}$. For the sake of simplicity, we only explicitly prove the claim for scenario *(1)*. The other cases can be proved with similar trigonometric arguments, all of

[3]This is also an orthonormal basis, so we could similarly represent functions in $L_2([-1,1]^d)$ as linear combinations of the elements of this basis and apply the properties of Fact 2.3. However, this representation is unhelpful for our analysis because its elements have large Lipschitz constants and are not ridge functions.

which involve applying Fact 2.4. For scenario *(1)*, we observe that

$$\begin{aligned}
\langle T_K, T_{K'} \rangle_{[-1,1]^d} &= \frac{1}{2^d} \int_{[-1,1]^d} 2\cos\left(\pi \langle K, x \rangle\right) \cos\left(\pi \langle K', x \rangle\right) \mathrm{d}x \\
&= \frac{1}{2^d} \int_{[-1,1]^d} \left[\cos\left(\pi \langle K - K', x \rangle\right) - \cos\left(\pi \langle K + K', x \rangle\right)\right] \mathrm{d}x \\
&= \frac{1}{2^d} \left[2^d \mathbb{1}\left\{K - K' = 0\right\} - 2^d \mathbb{1}\left\{K + K' = 0\right\}\right] \\
&= \mathbb{1}\left\{K = K'\right\}.
\end{aligned}$$

The last equality holds because if $K + K' = 0$, then either K or K' must belong to $\mathcal{K}_{\sin}$ by the definitions of $\mathcal{K}_{\sin}$ and $\mathcal{K}_{\cos}$. □

We additionally derive the following useful fact about the partial derivatives of elements of the trigonometric basis $\mathcal{T}$.

Fact 2.6 (Orthogonality of derivatives of $\mathcal{T}$)**.** *For all $M \in \mathbb{N}^d$ and for all $K, K' \in \mathbb{Z}^d$,*

$$\left\langle D^{(M)} T_K, D^{(M)} T_{K'} \right\rangle_{[-1,1]^d} = \mathbb{1}\left\{K = K'\right\} \pi^{2|M|} K^{2M}.$$

Proof. The partial derivatives of T_K for every $\mathcal{K} \in \mathbb{Z}^d$ can be exactly characterized by inductively taking derivatives of sin and cos functions:

$$D^{(M)} T_K(x) = \begin{cases}
\pi^{|M|} T_K(x) K^M & |M| \equiv 0 \pmod 4 \\
\pi^{|M|} T_{-K}(x) K^M & |M| \equiv 1 \pmod 4 \ \&\ K \in \mathcal{K}_{\sin} \\
-\pi^{|M|} T_{-K}(x) K^M & |M| \equiv 1 \pmod 4 \ \&\ K \in \mathcal{K}_{\cos} \cup \{\vec{0}\} \\
-\pi^{|M|} T_K(x) K^M & |M| \equiv 2 \pmod 4 \\
-\pi^{|M|} T_{-K}(x) K^M & |M| \equiv 3 \pmod 4 \ \&\ K \in \mathcal{K}_{\sin} \\
\pi^{|M|} T_{-K}(x) K^M & |M| \equiv 3 \pmod 4 \ \&\ K \in \mathcal{K}_{\cos} \cup \{\vec{0}\}.
\end{cases} \tag{2.2}$$

The conclusion follows by applying the orthonormality of trigonometric basis elements from

Fact 2.5 to Equation (2.2). □

2.2.2.3 Derivatives and boundary conditions of $\mathcal{T}$

To prove that a function $h \in L_2([-1,1]^d)$ can be represented by a linear combination of sufficiently many random ReLUs, we first show that h can be approximated by a low-degree trigonometric polynomial. To do so, we upper-bound the higher-order coefficients of the trigonometric expansion of h. Obtaining these bounds requires taking partial derivatives of h by differentiating term-by-term the trigonometric expansion of h. However, this is not always possible; for instance, if $h(x) = x_1$, the terms of the trigonometric expansion of $\partial h/\partial x_1$ do not correspond to the term-by-term derivatives of the expansion of h.[4] We define a notion of *boundary periodicity* that lets us perform term-by-term differentiation:

Definition 2.4 (Periodic boundary conditions)**.** $h \in L_2([-1,1]^d)$ satisfies the *periodic boundary conditions* if for all $i \in [d]$ and for all $x \in [-1,1]^d$

$$h(x_1, \ldots, x_{i-1}, -1, x_{i+1}, \ldots, x_d) = h(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_d).$$

Note that all basis elements in $\mathcal{T}$ satisfy the periodic boundary conditions. The next lemma gives sufficient conditions for term-by-term differentiation of a function's trigonometric representation.

Lemma 2.7 (Term-by-term differentiation of trigonometric basis representations)**.** *Consider some $h \in L_2([-1,1]^d)$ and $i \in [d]$ such that h satisfies the periodic boundary conditions, h is differentiable with respect to x_i, and $\partial h/\partial x_i \in L_2([-1,1]^d)$. Then, h and $\partial h/\partial x_i$ have*

[4]Because $\partial h/\partial x_1 = 1$, its trigonometric expansion $\partial h/\partial x_1 = \sum_{K\in\mathbb{Z}^d} \beta_K T_K$ will have $\beta_K = \mathbb{1}\{K = \vec{0}\}$. Because $h = \sum_{K\in\mathbb{Z}^d} \alpha_K T_K$ will have $\alpha_K \neq 0$ for some $K \neq \vec{0}$, $\beta_K \neq 0$ if term-by-term differentiation were possible. Since this contradicts the expansion of $\partial f/\partial x_1$, term-by-term differentiation is impossible in this case.

trigonometric expansions of the form

$$h = \sum_{K \in \mathbb{Z}^d} \alpha_K T_K \qquad \& \qquad \frac{\partial h}{\partial x_i} = \sum_{K \in \mathbb{Z}^d} \beta_K T_K,$$

where their coefficients $(\alpha_K)_{K \in \mathbb{Z}^d}, (\beta_K)_{K \in \mathbb{Z}^d}$ *are related as follows:*

$$\beta_K = \begin{cases} \pi K_i \alpha_{-K} & K \in \mathcal{K}_{\cos} \\ -\pi K_i \alpha_{-K} & K \in \mathcal{K}_{\sin} \\ 0 & K = \vec{0}. \end{cases} \tag{2.3}$$

Therefore,

$$\frac{\partial h}{\partial x_i} = \sum_{K \in \mathbb{Z}^d} \alpha_K \frac{\partial T_K}{\partial x_i}.$$

Proof. Without loss of generality, let $i = 1$. Because each of h and $\partial h / \partial x_1$ is in $L_2([-1,1]^d)$, there exist α and β by Fact 2.5 such that h and $\partial h / \partial x_1$ are exactly represented by the expansions given in the lemma statement. It remains to show that (2.3) holds. We fix any $K \in \mathcal{K}_{\cos}$, where $T_K(x) = \sqrt{2}\cos(\pi \langle K, x \rangle)$ and $\partial T_K(x)/\partial x_1 = -\sqrt{2}\pi K_1 \sin(\pi \langle K, x \rangle)$. By Fact 2.3, each coefficient of the representation is an inner-product: $\alpha_K = \langle h, T_K \rangle_{[-1,1]^d}$ and $\beta_K = \langle \partial h / \partial x_1, T_K \rangle_{[-1,1]^d}$. Moreover, β_K is related to α_{-K}, as shown in the following:

$$\begin{aligned}
\beta_K &= \left\langle \frac{\partial h}{\partial x_1}, T_K \right\rangle_{[-1,1]^d} \\
&= \frac{\sqrt{2}}{2^d} \int_{[-1,1]^d} \frac{\partial h(x)}{\partial x_1} \cos(\pi \langle K, x \rangle) \, \mathrm{d}x \\
&= \frac{\sqrt{2}}{2^d} \int_{[-1,1]^{d-1}} \int_{-1}^{1} \frac{\partial h(x)}{\partial x_1} \cos(\pi \langle K, x \rangle) \, \mathrm{d}x_1 \, \mathrm{d}x_{-1} \\
&= \frac{\sqrt{2}}{2^d} \int_{[-1,1]^{d-1}} \left[h(x) \cos(\pi \langle K, x \rangle) \Big|_{-1}^{1} + \int_{-1}^{1} h(x) \pi K_1 \sin(\pi \langle K, x \rangle) \, \mathrm{d}x_1 \right] \mathrm{d}x_{-1} \qquad (2.4) \\
&= \frac{\sqrt{2}}{2^d} \int_{[-1,1]^d} h(x) \pi K_1 \sin(\pi \langle K, x \rangle) \, \mathrm{d}x \qquad (2.5) \\
&= \pi K_1 \langle f, T_{-K} \rangle_{[-1,1]^d} = \pi K_1 \alpha_{-K}.
\end{aligned}$$

We integrate by parts for Equation (2.4) and take advantage of the periodic boundary conditions of h and T_K for Equation (2.5). A symmetric argument proves the claim for $K \in \mathcal{K}_{\sin}$. When $K = \vec{0}$, we repeat the above argument, and the periodic boundary conditions of h imply that $\beta_{\vec{0}} = 0$. □

2.2.2.4 Cardinality of subsets of $\mathcal{T}$

We also consider certain finite-dimensional subspaces of $L_2([-1,1]^d)$ which are spanned by a set of functions indexed by $\mathcal{K}_{k,d} := \{K \in \mathbb{Z}^d : \|K\|_2 \leq k\}$. These subspaces of $L_2([-1,1]^d)$ of primary interest in our analysis are spanned by a set of orthonormal functions that are indexed by the integer lattice points contained in given Euclidean balls. The next fact places upper and lower bounds the number of such points (and hence the dimension of such a subspace $Q_{k,d} := |\mathcal{K}_{k,d}|$).

Fact 2.8. *For all $d \in \mathbb{Z}^+$ and $k \geq 1$,*

$$Q_{k,d} = \exp\left(\Theta\left(\min\left(d\log\left(\frac{k^2}{d}+2\right), k^2\log\left(\frac{d}{k^2}+2\right)\right)\right)\right).$$

Proof. For the upper bound, we use the fact that $\|K\|_1 \leq \|K\|_2^2$ for all $K \in \mathbb{Z}^d$:

$$\begin{aligned} Q_{k,d} &= \left|\left\{K \in \mathbb{Z}^d : \|K\|_2 \leq k\right\}\right| \leq \left|\left\{K \in \mathbb{Z}^d : \|K\|_1 \leq k^2\right\}\right| \\ &\leq \left|\left\{K \in \mathbb{N}^{2d} : \|K\|_1 \leq k^2\right\}\right| \qquad (2.6) \\ &\leq \binom{\lceil k^2\rceil + 2d - 1}{\lceil k^2\rceil}. \qquad (2.7) \end{aligned}$$

Inequality (2.6) holds because we replace each integer in K from the previous line with two natural numbers (there would be equality if we forced one of each pair of natural numbers to equal zero). Line (2.7) follows from a standard stars-and-bars counting argument. Note that

$$\binom{\lceil k^2\rceil + 2d - 1}{\lceil k^2\rceil} = \binom{\lceil k^2\rceil + 2d - 1}{2d-1}.$$

We show two separate upper bounds on that quantity, which together prove the claim:

$$Q_{k,d} \le \binom{\lceil k^2 \rceil + 2d - 1}{2d-1} \le \left(\frac{e\lceil k^2 \rceil}{2d-1} + e\right)^{2d-1} \le \exp\left(\Theta\left(d \log\left(\frac{k^2}{d} + 2\right)\right)\right);$$
$$Q_{k,d} \le \binom{\lceil k^2 \rceil + 2d - 1}{\lceil k^2 \rceil} \le \left(\frac{2ed}{\lceil k^2 \rceil} + e\right)^{\lceil k^2 \rceil} \le \exp\left(\Theta\left(k^2 \log\left(\frac{d}{k^2} + 2\right)\right)\right).$$

For the lower bound, we observe that

$$\min\left(d \log\left(\frac{k^2}{d} + 2\right), k^2 \log\left(\frac{d}{k^2} + 2\right)\right) = \begin{cases} d \log\left(\frac{k^2}{d} + 2\right) & \text{if } k^2 \ge d, \\ k^2 \log\left(\frac{d}{k^2} + 2\right) & \text{if } k^2 < d. \end{cases}$$

We will lower-bound $Q_{k,d}$ by the appropriate term in each of the two cases, $k^2 \ge d$ and $k^2 < d$.

For the case $k^2 < d$, we lower-bound $Q_{k,d}$ by a sum of binomial coefficients:

$$\begin{aligned} Q_{k,d} &= \left|\left\{K \in \mathbb{Z}^d : \sum_{i=1}^{d} K_i^2 \le k^2\right\}\right| \\ &\ge \left|\left\{K \in \{0,1\}^d : \sum_{i=1}^{d} K_i \le k^2\right\}\right| \\ &= \binom{d}{0} + \binom{d}{1} + \cdots + \binom{d}{\lfloor k^2 \rfloor}. \end{aligned}$$

If $\lfloor k^2 \rfloor \le d/2$, then the sum of binomial coefficients is at least the last one, which we bound using

$$\binom{d}{\lfloor k^2 \rfloor} \ge \exp\left(\left\lfloor k^2 \right\rfloor \ln \frac{d}{\lfloor k^2 \rfloor}\right) \ge \exp\left(\frac{\lfloor k^2 \rfloor}{2} \ln\left(\frac{d}{\lfloor k^2 \rfloor} + 2\right)\right) = \exp\left(\Theta\left(k^2 \ln\left(\frac{d}{k^2} + 2\right)\right)\right).$$

Otherwise, if $d/2 < \lfloor k^2 \rfloor < d$, the sum of binomial coefficients is at least $2^{\lfloor k^2 \rfloor}$, and

$$2^{\lfloor k^2 \rfloor} = \exp\left((\ln 2)\left\lfloor k^2 \right\rfloor\right) \ge \exp\left(\frac{\ln 2}{\ln 4}\left\lfloor k^2 \right\rfloor \ln\left(\frac{d}{\lfloor k^2 \rfloor} + 2\right)\right) = \exp\left(\Theta\left(k^2 \ln\left(\frac{d}{k^2} + 2\right)\right)\right).$$

When $k^2 \geq d$, we show that $Q_{k,d}$ grows at a rate similar to that of the volume of a d-dimensional ball of sufficiently large radius $\Theta(k)$. To do so, we regard each $K \in \mathcal{K}_{k,d}$ as an element of $\mathbb{R}^d$, and define

$$A_{k,d} := \left\{ x \in \mathbb{R}^d : \min_{K \in \mathcal{K}_{k,d}} \|x - K\|_\infty \leq \frac{1}{2} \right\}.$$

This is the Minkowski sum of $\mathcal{K}_{k,d}$ and the ℓ_∞ ball of radius $1/2$ in $\mathbb{R}^d$. Note that $A_{k,d}$ has Lebesgue measure $\mathrm{vol}(A_{k,d}) = |\mathcal{K}_{k,d}| = Q_{k,d}$. Let $B_2^d(r) := \{x \in \mathbb{R}^d : \|x\|_2 \leq r\}$ be the d-dimensional Euclidean ball of radius r. We claim that $B_2^d(k - \sqrt{d}/2) \subset A_{k,d}$, which in turn implies

$$Q_{k,d} \geq \mathrm{vol}\left(B_2^d\left(k - \sqrt{d}/2\right)\right).$$

To see why this claim holds, consider any $x \in B_2^d(k - \sqrt{d}/2)$. We'll show that $x \in A_{k,d}$. Indeed, there exists some $y \in \mathbb{Z}^d$ such that $\|x - y\|_\infty \leq 1/2$, and hence this y also satisfies $\|x - y\|_2 \leq \sqrt{d}/2$. By the triangle inequality,

$$\begin{aligned} \|y\|_2 &\leq \|x\|_2 + \|x - y\|_2 \\ &\leq \left(k - \frac{\sqrt{d}}{2}\right) + \frac{\sqrt{d}}{2} = k. \end{aligned}$$

Thus, $y \in \mathcal{K}_{k,d}$, which implies $x \in A_{k,d}$.

To complete our lower bound on $Q_{k,d}$, we observe that

$$\begin{aligned} Q_{k,d} &\geq \mathrm{vol}\left(B_d\left(k - \frac{1}{2}\sqrt{d}\right)\right) \geq \mathrm{vol}\left(B_d\left(\frac{k}{2}\right)\right) \\ &= \frac{\pi^{d/2}(k/2)^d}{\Gamma\left(\frac{d}{2} + 1\right)} \geq \left(\frac{\pi k^2}{2d+4}\right)^{d/2} \geq \exp\left(\Theta\left(d \log\left(\frac{k^2}{d} + 2\right)\right)\right), \end{aligned}$$

where Γ is the gamma function and we have used a standard bound on the volume of the d-dimensional Euclidean ball. $\square$

2.3 Positive results for Lipschitz targets

Our upper bounds on the minimum-width random ReLU feature network that approximates a Lipschitz function are dominated by the quantity $Q_{k,d}$, which represents the number of integer points contained in a d-dimensional ball of radius k (see Section 2.2.2.4).

Theorem 2.9 (Formal version of Theorem 2.1: Upper-bound for L-Lipschitz functions)**.** *Fix some $\delta \in (0, \frac{1}{2}]$ and $\epsilon, L > 0$ with $\frac{L}{\epsilon} \geq 2$. Then, there exists some symmetric ReLU parameter distribution $\mathcal{D}$ such that for any $h \in L_2([-1,1]^d)$ with $\|h\|_{\text{Lip}} \leq L$ and $|\mathbb{E}_{\mathbf{x}}[h(\mathbf{x})]| \leq L$,*

$$\text{MinWidth}_{h,\epsilon,\delta,[-1,1]^d,\mathcal{D}} \leq O\left(\frac{L^6 d^2}{\epsilon^6} \ln\left(\frac{1}{\delta}\right) Q^2_{2L/\epsilon,d}\right).$$

Applying the asymptotics of $Q_{k,d}$ from Fact 2.8 reveals that the minimum width can also be bounded by the term in Theorem 2.1. That expression shows that the minimum width is polynomial in $\frac{L}{\epsilon}$ when d is a fixed constant, and polynomial in d when $\frac{L}{\epsilon}$ is a fixed constant.

This section contains the proof of Theorem 2.1. In Section 2.3.1, we give an high-level overview of the argument and state Lemmas 2.10 and 2.12, which are sufficient to prove the claim. We prove these lemmas in Sections 2.3.2 and 2.3.3.

2.3.1 Proof outline for Theorem 2.9

To prove Theorem 2.9, we break the process of approximating a Lipschitz function h with an random ReLU feature network into two steps. We first approximate h with a bounded-degree trigonometric polynomial P in Lemma 2.10 and then approximate P with an random ReLU feature network in Lemma 2.12. We state the lemmas and discuss their proofs in Sections 2.3.1.1 and 2.3.1.2 respectively. Section 2.3.1.3 gives a formal proof of Theorem 2.9.

In Section 2.5.1, we present and prove Theorem 2.25, a parallel result to Theorem 2.9 that instead considers the approximation of some function h that has a bounded Sobolev norm and which (along with its derivatives) satisfies periodic boundary conditions. The proof of Theorem 2.25 only differs from that of Theorem 2.9 by obtaining a trigonometric

polynomial approximation for f from Lemma 2.26 (stated and proved in Section 2.5.1) rather than Lemma 2.10.

2.3.1.1 Approximating Lipschitz functions with bounded-degree trigonometric polynomials

Lemma 2.10. *Fix some $L, \epsilon > 0$ with $\frac{L}{\epsilon} \geq 1$ and consider any function $h \in L^2([-1,1]^d)$ with $\|h\|_{\text{Lip}} \leq L$ and $|\mathbb{E}_{\mathbf{x}}[h(\mathbf{x})]| \leq L$. Then, taking $k = \frac{L}{\epsilon}$, there exists a bounded-degree trigonometric polynomial*

$$P(x) = \sum_{K \in \mathcal{K}_{k,d}} \beta_K T_K\left(\frac{x}{2}\right)$$

such that $\|h - P\|_{[-1,1]^d} \leq \epsilon$. Moreover, $|\beta_K| \leq L$ for all K.

We formally prove this lemma in Section 2.3.2. Here we highlight a central part of the argument (used in the full proof) by stating and proving a special case of the lemma which additionally requires that h satisfy periodic boundary conditions.

Lemma 2.11 (Approximating Lipschitz functions with periodic boundary conditions)**.** *Fix some $L, \epsilon > 0$ with $\frac{L}{\epsilon} \geq 2$. Consider any function $h \in L^2([-1,1]^d)$ such that h satisfies periodic boundary conditions, $\|h\|_{\text{Lip}} \leq L$, and $|\mathbb{E}_{\mathbf{x}}[h(\mathbf{x})]| \leq \frac{L}{2}$. Then, taking $k = \frac{L}{2\epsilon}$, there exists a bounded-degree trigonometric polynomial*

$$P(x) = \sum_{K \in \mathcal{K}_{k,d}} \beta_K T_K(x)$$

such that $\|h - P\|_{[-1,1]^d} \leq \epsilon$. Moreover, $|\beta_K| \leq \frac{L}{2}$ for all K.

To prove Lemma 2.11, we consider the representation of h as an infinite linear combination of trigonometric basis elements from $\mathcal{T}$. We show that h can only be L-Lipschitz if all high-degree terms of this representation have vanishingly small coefficients. This requires the term-by-term differentiation of the trigonometric representation of h, which is possible due to its periodic boundary conditions (see Lemma 2.7 of Section 2.2.2).

Proof. By appealing to a standard approximation argument (e.g., Folland, 1999, Proposition 8.17), we may assume that f is differentiable. Because $\mathcal{T}$ is an orthonormal basis over $L_2([-1,1]^d)$, we can express h as

$$h(x) = \sum_{K \in \mathbb{Z}^d} \alpha_K T_K(x).$$

The condition $\|h\|_{\text{Lip}} \leq L$ implies that $\|\nabla h(x)\|_2 \leq L$ for all $x \in [-1,1]^d$. Because h has periodic boundary conditions, h is differentiable, and $\partial h(x)/\partial x_i \in L_2([-1,1]^d)$ for all i, Lemma 2.7 can be applied to relate L to the coefficients $(\alpha_K)_{K \in \mathbb{Z}^d}$:

$$\begin{aligned}
L^2 &\geq \mathop{\mathbb{E}}_{\mathbf{x} \sim [-1,1]^d} \left[\|\nabla h(\mathbf{x})\|_2^2 \right] = \sum_{i=1}^d \mathop{\mathbb{E}}_{\mathbf{x}} \left[\left(\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}_i} \right)^2 \right] \\
&= \sum_{i=1}^d \mathop{\mathbb{E}}_{\mathbf{x}} \left[\left(\sum_{K \in \mathbb{Z}^d} \alpha_K \frac{\partial T_K(\mathbf{x})}{\partial \mathbf{x}_i} \right)^2 \right] \qquad (2.8) \\
&= \sum_{i=1}^d \sum_{K \in \mathbb{Z}^d} \alpha_K^2 \left\| \frac{\partial T_K}{\partial x_i} \right\|_{[-1,1]^d}^2 + 2 \sum_{i=1}^d \sum_{K \in \mathbb{Z}^d} \sum_{K' \neq K} \alpha_K \alpha_{K'} \left\langle \frac{\partial T_K}{\partial x_i}, \frac{\partial T_{K'}}{\partial x_i} \right\rangle_{[-1,1]^d} \\
&= \sum_{i=1}^d \sum_{K \in \mathbb{Z}^d} \alpha_K^2 \pi^2 K_i^2 = \pi^2 \sum_K \alpha_K^2 \|K\|_2^2 . \qquad (2.9)
\end{aligned}$$

Equations (2.8) and (2.9) follow from Lemma 2.7 and Fact 2.6 respectively. An immediate consequence of the above inequality is that $|\alpha_K| \leq L/\pi \leq L/2$ as long as $K \neq \vec{0}$. Because $|\mathbb{E}_{\mathbf{x}}[h(\mathbf{x})]| \leq L/2$, $|\alpha_{\vec{0}}| \leq L/2$ as well. We define the trigonometric polynomial $P = \sum_{K \in \mathcal{K}_{k,d}} \beta_K T_K$ by letting $\beta_K := \alpha_K$ for all K with $\|K\|_2 \leq k$. Parseval's identity (Fact 2.3) and the inequality ending on line (2.9) guarantee that

$$\begin{aligned}
\|h - P\|_{[-1,1]^d}^2 &= \sum_{K \in \mathbb{Z}^d \setminus \mathcal{K}_{k,d}} \alpha_K^2 \leq \sum_{K \in \mathbb{Z}^d \setminus \mathcal{K}_{k,d}} \alpha_K^2 \cdot \frac{\|K\|_2^2}{k^2} \leq \frac{1}{k^2} \sum_{K \in \mathbb{Z}^d} \alpha_K^2 \|K\|_2^2 \\
&\leq \frac{L^2}{\pi^2 k^2} \leq \frac{L^2}{2^2 k^2} = \epsilon^2 .
\end{aligned}$$

□

The proof of Lemma 2.10 is a reduction to Lemma 2.11. Instead of approximating

h with a low-degree trigonometric polynomial, we approximate $\tilde{h}$, a scaled, shifted, and reflected version of h that has periodic boundary conditions and thus can be differentiated term-by-term. The bulk of the proof involves transforming h into $\tilde{h}$ and transforming $\tilde{P}$ (the trigonometric polynomial obtained by applying Lemma 2.11 to $\tilde{h}$) back into P. This scaling and reflection argument is why we approximate h with combinations of trigonometric polynomials of the form $T_K(x/2)$, rather than $T_K(x)$.

2.3.1.2 Approximating bounded-degree trigonometric polynomials with random ReLU feature nets

Lemma 2.12. *Fix some $\delta \in (0, 1/2]$, $\epsilon > 0$, $\rho \in (0, 1]$, $k \geq 1$, and $d \in \mathbb{Z}^+$. Then, there exists some symmetric ReLU parameter distribution $\mathcal{D}_k$ such that for any trigonometric polynomial*

$$P(x) = \sum_{K \in \mathcal{K}_{k,d}} \beta_K T_K(\rho x)$$

with $|\beta_K| \leq \beta_{\max}$ for all $K \in \mathcal{K}_{k,d}$,

$$\text{MinWidth}_{P,\epsilon,\delta,[-1,1]^d,\mathcal{D}_k} \leq O\left(\frac{\beta_{\max}^2 d^2 k^4}{\epsilon^2} Q_{k,d}^2 \ln\left(\frac{1}{\delta}\right)\right).$$

We restate and prove this lemma in Section 2.3.3. We take advantage of the fact that every low-degree trigonometric polynomial can be expressed as a linear combination of ridge functions. As shown in Lemma 2.13, each of those ridge functions can in turn be represented as an infinite mixture of ReLUs. We then represent the entire trigonometric polynomial as an expectation over weighted random ReLU features with parameters drawn from a symmetric ReLU parameter distribution $\mathcal{D}_k$ (Definition 2.5). By bounding the maximum norm of every random ReLU drawn from $\mathcal{D}_k$, a concentration bound (Lemma 2.14) can show that this expectation can be closely approximated with a sufficiently large finite linear combination of randomly sampled ReLUs.

2.3.1.3 Proof of Theorem 2.9

Consider any $h \in L_2([-1,1]^d)$ with $\|h\|_{\text{Lip}} \le L$ and $|\mathbb{E}_{\mathbf{x}}[h(\mathbf{x})]| \le L$. By Lemma 2.10, there exists a bounded-degree trigonometric polynomial $P(x) = \sum_{K \in \mathcal{K}_{k,d}} \beta_K T_K(x/2)$ with $k := 2L/\epsilon$ and $|\beta_K| \le L$ for all $K \in \mathcal{K}_{k,d}$, such that $\|h - P\|_{[-1,1]^d} \le \epsilon/2$. By applying Lemma 2.12 to P with $\rho = 1/2$,

$$\text{MinWidth}_{P,\epsilon/2,\delta,[-1,1]^d,\mathcal{D}_k} \le O\left(\frac{\beta_{\max}^2 d^2 k^4}{\epsilon^2} Q_{k,d}^2 \ln\left(\frac{1}{\delta}\right)\right) \le O\left(\frac{d^2 L^6}{\epsilon^6} Q_{2L/\epsilon,d}^2 \ln\left(\frac{1}{\delta}\right)\right).$$

Thus (see Definition 2.2) there exists an random ReLU feature network f of width

$$m = \text{MinWidth}_{P,\epsilon/2,\delta,[-1,1]^d,\mathcal{D}_k}$$

such that $\|P - f\|_{[-1,1]^d} \le \epsilon/2$. By the triangle inequality, $\|f - h\|_{[-1,1]^d} \le \epsilon$. We conclude that

$$\text{MinWidth}_{f,\epsilon,\delta,[-1,1]^d,\mathcal{D}_k} = O\left(\frac{d^2 L^6}{\epsilon^6} Q_{2L/\epsilon,d}^2 \ln\left(\frac{1}{\delta}\right)\right).$$

2.3.2 Proof of Lemma 2.10

We restate and prove Lemma 2.10 by modifying the proof of Lemma 2.11.

Lemma 2.10. *Fix some $L, \epsilon > 0$ with $\frac{L}{\epsilon} \ge 1$ and consider any function $h \in L^2([-1,1]^d)$ with $\|h\|_{\text{Lip}} \le L$ and $|\mathbb{E}_{\mathbf{x}}[h(\mathbf{x})]| \le L$. Then, taking $k = \frac{L}{\epsilon}$, there exists a bounded-degree trigonometric polynomial*

$$P(x) = \sum_{K \in \mathcal{K}_{k,d}} \beta_K T_K\left(\frac{x}{2}\right)$$

such that $\|h - P\|_{[-1,1]^d} \le \epsilon$. Moreover, $|\beta_K| \le L$ for all K.

Proof. To give a low-degree trigonometric polynomial approximation for h, we transform h into a function $\tilde{h}$ that satisfies periodic boundary conditions, apply Lemma 2.11 to approx-

imate $\tilde{h}$ with trigonometric polynomial $\tilde{P}$, and obtain P from $\tilde{P}$. Roughly, the argument proceeds as follows:

1. We define $\bar{h} : [0,1]^d \to \mathbb{R}$ to be a rescaling and shift of h so that its domain is the cube $[0,1]^d$. That is, for $x \in [-1,1]^d$ and $y \in [0,1]^d$, $\bar{h}(y) = h(2y - \vec{1})$ and $h(x) = \bar{h}((x+\vec{1})/2)$. Then it holds that $\|\bar{h}\|_{\text{Lip}} \leq 2L$ and $|\mathbb{E}_{\mathbf{y}\sim[0,1]^d}[\bar{h}(\mathbf{y})]| = |\mathbb{E}_{\mathbf{x}\sim[-1,1]^d}[h(\mathbf{x})]| \leq L$.

2. We define $\tilde{h} : [-1,1]^d \to \mathbb{R}$ by reflecting $\bar{h}$ across orthants as follows: $\tilde{h}(x) = \bar{h}(\text{sign}(x) \odot x)$, where $\text{sign}(x) := (\text{sign}(x_1), \ldots, \text{sign}(x_d))$ and $\odot$ represents element-wise multiplication. The function $\tilde{h}$ is $2L$-Lipschitz, satisfies the periodic boundary conditions, and has
$$\left| \mathop{\mathbb{E}}_{\mathbf{x}\sim[-1,1]^d} \left[\tilde{h}(\mathbf{x})\right] \right| = \left| \mathop{\mathbb{E}}_{\mathbf{y}\sim[0,1]^d} \left[\bar{h}(\mathbf{y})\right] \right| \leq L.$$

3. We find a low-degree trigonometric polynomial $\tilde{P}$ that ϵ-approximates $\tilde{h}$ over $[-1,1]^d$.

4. Such a $\tilde{P}$ must ϵ-approximate $\tilde{h}$ in at least one of the 2^d unit cubes contained in the orthants of $[-1,1]^d$. Therefore, there exists some sign vector $\nu \in \{-1,1\}^d$ such that $\bar{h}(y)$ is approximated by $\tilde{P}(\nu \odot y)$ on $[0,1]^d$.

5. By shifting and rescaling $\tilde{P}(\nu \odot y)$, we obtain a trigonometric polynomial P that ϵ-approximates h on $[-1,1]^d$ as desired.

Steps (1) and (2) are immediate.

Step (3) is a consequence of Lemma 2.11. Because $\tilde{h}$ is $2L$-Lipschitz, $\tilde{h}$ satisfies the periodic boundary conditions, $|\mathbb{E}_{\mathbf{x}\sim[-1,1]^d}[\tilde{h}(\mathbf{x})]| \leq L$, and $2L/\epsilon \geq 2$, Lemma 2.11 guarantees the existence of some trigonometric polynomial

$$\tilde{P}(x) = \sum_{K \in \mathcal{K}_{k,d}} \tilde{\beta}_K T_K(x)$$

such that $\|\tilde{h} - \tilde{P}\|_{[-1,1]^d} \leq \epsilon$ and $|\tilde{\beta}_K| \leq L$ for all K.

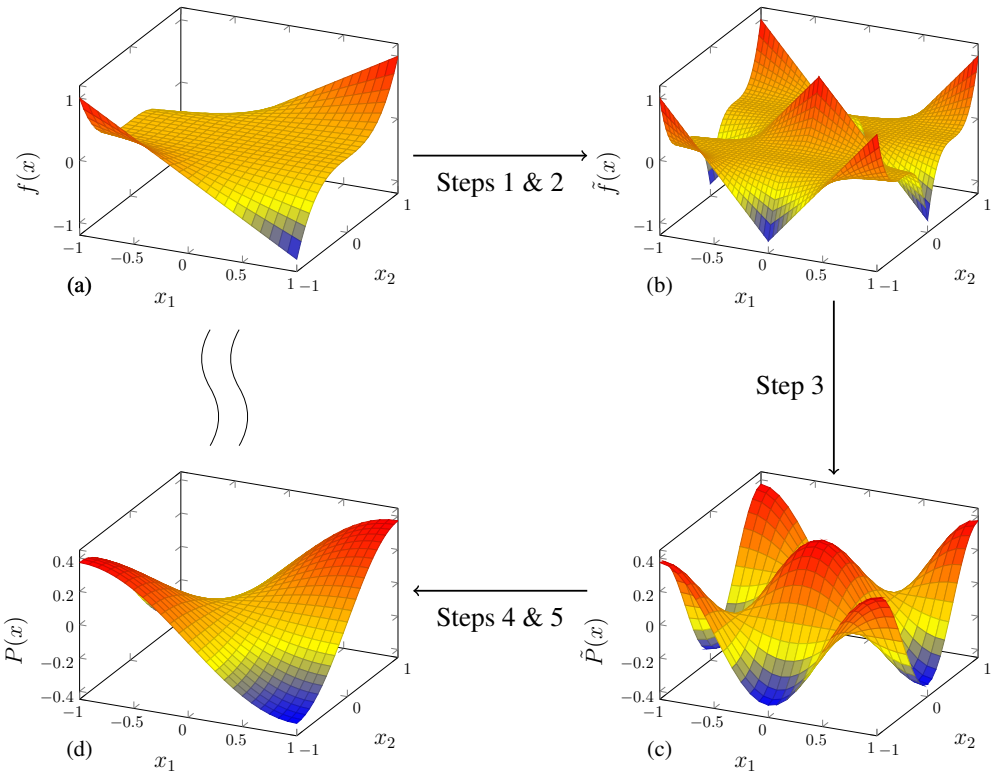


Figure 2.1: A depiction of the function transformations used to give an approximation of f in Lemma 2.10. The original function f is in (a), which is scaled and reflected to yield a function $\tilde{f}$ with periodic boundary conditions in (b), which is given a trigonometric polynomial approximation $\tilde{P}$ in (c), which is in turn scaled and shifted to obtain P approximating the original f in (d).

For step (4), if $\tilde{P}$ is an ϵ-approximator for $\tilde{h}$ over $L_2([-1,1]^d)$, then there must exist a unit cube in some orthant corresponding to some $\nu \in \{-1,1\}^d$ where $\tilde{P}$ also ϵ-approximates $\tilde{h}$. That is,

$$\mathop{\mathbb{E}}_{\mathbf{y}\sim[0,1]^d}\left[\left(\tilde{P}(\nu \odot \mathbf{y}) - \bar{h}(\mathbf{y})\right)^2\right] \leq \epsilon^2.$$

For step (5), by translating the distribution from $[-1,1]^d$ to $[0,1]^d$ and taking $P(x) := \tilde{P}(\nu \odot (x + \vec{1})/2)$, we obtain

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim[-1,1]^d}\left[(P(\mathbf{x}) - h(\mathbf{x}))^2\right] = \mathop{\mathbb{E}}_{\mathbf{y}\sim[0,1]^d}\left[\left(\tilde{P}(\nu \odot \mathbf{y}) - \bar{h}(\mathbf{y})\right)^2\right]$$

It remains to show that we can represent P as a proper trigonometric polynomial with halved frequencies and bounded coefficients. We do so by examining each term of the

expansion of $\tilde{P}$. Fix any $K \in \mathbb{Z}^d$ with $\|K\|_2 \leq k$ and $K \in \mathcal{K}_{\sin}$. Then, $T_K(y) = \sqrt{2}\sin(\pi\langle K, y\rangle)$. Consider the term corresponding to K of $P(x)$ represented as an expansion of $\tilde{P}$, $\tilde{\beta}_K T_K(\nu \odot (x + \vec{1})/2)$. By rearranging its inner product and applying sum-of-angles trigonometric identities, we obtain the following identity:

$$T_K\left(\frac{1}{2}\nu \odot (x+\vec{1})\right) = \sqrt{2}\sin\left(\frac{\pi}{2}\langle \nu \odot K, x\rangle + \frac{\pi}{2}\langle \nu \odot K, \vec{1}\rangle\right)$$
$$= \begin{cases} \sqrt{2}\sin\left(\frac{\pi}{2}\langle \nu \odot K, x\rangle\right) & \langle \nu \odot K, \vec{1}\rangle \equiv 0 \pmod 4 \\ \sqrt{2}\cos\left(\frac{\pi}{2}\langle \nu \odot K, x\rangle\right) & \langle \nu \odot K, \vec{1}\rangle \equiv 1 \pmod 4 \\ -\sqrt{2}\sin\left(\frac{\pi}{2}\langle \nu \odot K, x\rangle\right) & \langle \nu \odot K, \vec{1}\rangle \equiv 2 \pmod 4 \\ -\sqrt{2}\cos\left(\frac{\pi}{2}\langle \nu \odot K, x\rangle\right) & \langle \nu \odot K, \vec{1}\rangle \equiv 3 \pmod 4. \end{cases}$$

This yields the final representation for T_K functions:

$$T_K\left(\frac{1}{2}\nu \odot (x+\vec{1})\right) = \begin{cases} T_{\nu\odot K}(\frac{x}{2}) & \langle \nu \odot K, \vec{1}\rangle \equiv 0 \pmod 4 \\ T_{-\nu\odot K}(\frac{x}{2}) & \langle \nu \odot K, \vec{1}\rangle \equiv 1 \pmod 4 \\ -T_{\nu\odot K}(\frac{x}{2}) & \langle \nu \odot K, \vec{1}\rangle \equiv 2 \pmod 4 \\ -T_{-\nu\odot K}(\frac{x}{2}) & \langle \nu \odot K, \vec{1}\rangle \equiv 3 \pmod 4. \end{cases}$$

Similarly,

$$T_{-K}\left(\frac{1}{2}\nu \odot (x+\vec{1})\right) = \begin{cases} T_{-\nu\odot K}(\frac{x}{2}) & \langle \nu \odot K, \vec{1}\rangle \equiv 0 \pmod 4 \\ -T_{\nu\odot K}(\frac{x}{2}) & \langle \nu \odot K, \vec{1}\rangle \equiv 1 \pmod 4 \\ -T_{-\nu\odot K}(\frac{x}{2}) & \langle \nu \odot K, \vec{1}\rangle \equiv 2 \pmod 4 \\ T_{\nu\odot K}(\frac{x}{2}) & \langle \nu \odot K, \vec{1}\rangle \equiv 3 \pmod 4. \end{cases}$$

Using these identities, we can rewrite P as its own trigonometric polynomial with coefficients β_K for all $K \in \mathbb{Z}^d$ such that $\beta_K \in \{\tilde{\beta}_{\nu\odot K}, -\tilde{\beta}_{\nu\odot K}\}$ if $\langle \nu \odot K, \vec{1}\rangle \equiv 0 \pmod 2$, and

$\beta_K \in \{\tilde{\beta}_{-\nu\odot K}, -\tilde{\beta}_{-\nu\odot K}\}$ otherwise. Due to the existence of such β_K coefficients, the following trigonometric polynomial approximates h over $[-1,1]^d$:

$$P(x) = \sum_{K\in\mathcal{K}_{k,d}} \tilde{\beta}_K T_K\left(\frac{1}{2}\nu \odot (x+\vec{1})\right) = \sum_{K\in\mathcal{K}_{k,d}} \beta_K T_K\left(\frac{x}{2}\right). \qquad \square$$

2.3.3 Proof of Lemma 2.12

In this section, we prove Lemma 2.12 by employing general purpose lemma that bounds the width needed to approximate trigonometric polynomials of bounded degree.

We first define the specific symmetric ReLU parameter distribution $\mathcal{D}_k$ used in the proof, which can be shown to meet the symmetry criteria spelled out in Definition 2.1. (As a result, the lower-bounds on the minimum width in Theorems 2.15 and 2.23 hold for $\mathcal{D}_k$.)

Definition 2.5 (Symmetric ReLU parameter distribution $\mathcal{D}_k$ for $[-1,1]^d$ upper-bounds)**.** Define $\mathcal{D}_k := \mathcal{D}_{\text{bias}} \times \mathcal{D}_{\text{weights},k}$ as a product distribution with the following components:

- $\mathcal{D}_{\text{bias}}$ is the uniform distribution over $[-2\sqrt{d}, 2\sqrt{d}]$; and
- $\mathcal{D}_{\text{weights},k}$ is a distribution over weights $\mathbf{w}$ taking value in $\mathbb{S}^{d-1}$. To draw $\mathbf{w}$ from $\mathcal{D}_{\text{weights},k}$, draw $\mathbf{K}$ uniformly at random from $\mathcal{K}_{k,d}$ and let $\mathbf{w} := \mathbf{K}/\left\|K\right\|_2$. (If $\mathbf{K} = \vec{0}$, let $\mathbf{w} := \vec{1}/\sqrt{d}$.)

We also introduce notation to represent the set of vectors contained in $\mathcal{K}_{k,d}$ that generate each $w \in \text{supp}(\mathcal{D}_{\text{weights},k}) \subset \mathbb{S}^{d-1}$:

$$\mathcal{K}_{k,d,w} := \begin{cases} \{K \in \mathcal{K}_{k,d} : K = \eta w, \eta \geq 0\} & w = \frac{1}{\sqrt{d}}\vec{1} \\ \{K \in \mathcal{K}_{k,d} : K = \eta w, \eta > 0\} & \text{otherwise.} \end{cases}$$

Note that every $w \in \text{supp}(\mathcal{D}_{\text{weights},k})$ is drawn with probability $|\mathcal{K}_{k,d,w}|/Q_{k,d}$, which is at least $1/Q_{k,d}$ and at most $(k+1)/Q_{k,d}$.

To prove Lemma 2.12, we represent P as an expectation over random ReLU features with parameters drawn from $\mathcal{D}_k$. We first express each trigonometric basis element T_K as an expectation over random ReLUs. We leverage the fact that each individual T_K is a ridge function (that is, $T_K(x) = \phi(\langle K, x\rangle)$ for some ϕ). In the following lemma, we show that every ridge function on $[-1,1]^d$ can be represented as a mixture of ReLUs with random bias terms $\mathbf{b}$ drawn from $\mathcal{D}_{\text{bias}}$.

Lemma 2.13 (Representing ridge functions as a mixture of ReLUs)**.** *Let $\phi : [-\sqrt{d}, \sqrt{d}] \to \mathbb{R}$ be twice differentiable and let $f : [-1,1]^d \to \mathbb{R}$ be $f(x) = \phi(\langle v, x\rangle)$ for some $v \in \mathbb{S}^{d-1}$. Then, for all $x \in [-1,1]^d$,*

$$f(x) = \mathop{\mathbb{E}}_{\mathbf{b}\sim\mathcal{D}_{\text{bias}}} \left[\psi(\mathbf{b})\text{ReLU}\left(\langle v, x\rangle - \mathbf{b}\right)\right],$$

where

$$\psi(b) := \begin{cases} 4\sqrt{d}a_0 := \frac{16}{\sqrt{d}}\phi(-\sqrt{d}) - 4\phi'(-\sqrt{d}) & b \in [-2\sqrt{d}, -\frac{3}{2}\sqrt{d}) \\ 4\sqrt{d}a_1 := -\frac{16}{\sqrt{d}}\phi(-\sqrt{d}) + 12\phi'(-\sqrt{d}) & b \in [-\frac{3}{2}\sqrt{d}, -\sqrt{d}) \\ 4\sqrt{d}\phi''(b) & b \in [-\sqrt{d}, \sqrt{d}] \\ 0 & b \in (\sqrt{d}, 2\sqrt{d}]. \end{cases}$$

Proof. We expand the expectation over $\mathbf{b}$. For $x \in [-1,1]^d$, let $z := \langle v, x\rangle \in [-\sqrt{d}, \sqrt{d}]$. We

have the following:

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{b}\sim\mathcal{D}_{\text{bias}}}\left[\phi(\mathbf{b})\text{ReLU}\left(\langle v, x\rangle - \mathbf{b}\right)\right] \\
&= a_0 \int_{-2\sqrt{d}}^{-\frac{3}{2}\sqrt{d}} \text{ReLU}(z-b)\,\mathrm{d}b + a_1 \int_{-\frac{3}{2}\sqrt{d}}^{-\sqrt{d}} \text{ReLU}(z-b)\,\mathrm{d}b + \int_{-\sqrt{d}}^{\sqrt{d}} \phi''(b)\text{ReLU}(z-b)\,\mathrm{d}b \\
&= a_0\left(zb - \frac{1}{2}b^2\right)\Bigg|_{-2\sqrt{d}}^{-\frac{3}{2}\sqrt{d}} + a_1\left(zb - \frac{1}{2}b^2\right)\Bigg|_{-\frac{3}{2}\sqrt{d}}^{-\sqrt{d}} + \int_{-\sqrt{d}}^{z} \phi''(b)(z-b)\,\mathrm{d}b \\
&= \frac{\sqrt{d}}{2}z\left(a_0 + a_1\right) + \frac{d}{8}\left(7a_0 + 5a_1\right) + \left(\phi'(b)(z-b)\right)\Big|_{-\sqrt{d}}^{z} - \int_{-\sqrt{d}}^{z} \phi'(b)\cdot(-1)\,\mathrm{d}b \\
&= z\phi'(-\sqrt{d}) + \phi(-\sqrt{d}) + \sqrt{d}\phi'(-\sqrt{d}) - \phi'(-\sqrt{d})(z+\sqrt{d}) + \phi(z) - \phi(-\sqrt{d}) \\
&= \phi(z) = f(x). \qquad \square
\end{aligned}
$$

Once P is represented as an expectation over random ReLUs with parameters drawn from $\mathcal{D}_k$, we conclude the proof by arguing that this expectation can be closely approximated with high probability by a linear combination of sufficiently many randomly sampled ReLUs. We do so by applying a concentration bound due to Yurinskii, 1976 for sums of independent random variables taking values in a Hilbert space. We use a convenient version of the bound from Rahimi and Recht (2009, Lemma 4):

Lemma 2.14 (Concentration inequality for Hilbert spaces)**.** *Let $\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}$ be independent random variables that take values in a Hilbert space with norm $\|\cdot\|$ such that $\|\mathbf{g}^{(i)}\| \leq M$ for all i. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$
\left\| \frac{1}{m}\sum_{i=1}^{m} \mathbf{g}^{(i)} - \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} \mathbf{g}^{(i)}\right] \right\| \leq \frac{M}{\sqrt{m}}\left(1 + \sqrt{2\log\left(\frac{1}{\delta}\right)}\right).
$$

We are now prepared to formally prove Lemma 2.12.

Lemma 2.12. *Fix some $\delta \in (0, 1/2]$, $\epsilon > 0$, $\rho \in (0,1]$, $k \geq 1$, and $d \in \mathbb{Z}^+$. Then, there exists*

some symmetric ReLU parameter distribution $\mathcal{D}_k$ such that for any trigonometric polynomial

$$P(x) = \sum_{K \in \mathcal{K}_{k,d}} \beta_K T_K(\rho x)$$

with $|\beta_K| \leq \beta_{\max}$ for all $K \in \mathcal{K}_{k,d}$,

$$\text{MinWidth}_{P,\epsilon,\delta,[-1,1]^d,\mathcal{D}_k} \leq O\left(\frac{\beta_{\max}^2 d^2 k^4}{\epsilon^2} Q_{k,d}^2 \ln\left(\frac{1}{\delta}\right)\right).$$

Proof. We first represent any trigonometric monomial T_K as an expected value over weighted ReLUs of the form $\text{ReLU}(\langle K/\left\|K\right\|_2, x\rangle + \mathbf{b})$ for $\mathbf{b} \sim \mathcal{D}_{\text{bias}}$. For each K, we have $T_K(\rho x) = \phi_K(\langle K/\left\|K\right\|_2, x\rangle)$, where

$$\phi_K(z) = \begin{cases} \sqrt{2}\cos(\pi\rho\left\|K\right\|_2 z) & K \in \mathcal{K}_{\cos} \\ \sqrt{2}\sin(\pi\rho\left\|K\right\|_2 z) & K \in \mathcal{K}_{\sin} \\ 1 & K = \vec{0}. \end{cases}$$

By Lemma 2.13,

$$T_K(\rho x) = \mathop{\mathbb{E}}_{\mathbf{b}\sim\mathcal{D}_{\text{bias}}}\left[\psi_K(b)\text{ReLU}\left(\frac{1}{\left\|K\right\|_2}\langle K, x\rangle - \mathbf{b}\right)\right],$$

where ψ_K is the function defined in Lemma 2.13 for ϕ_K. Because $|\phi_K(z)| \leq \sqrt{2}$, $|\phi_K'(z)| \leq \sqrt{2}\pi\rho\left\|K\right\|_2$, and $|\phi_K''(z)| \leq \sqrt{2}\pi^2\rho^2\left\|K\right\|_2^2$ for all z, we can bound ψ_K:

$$|\psi_K(z)| \leq \max\left\{\frac{16}{\sqrt{d}} \cdot \sqrt{2} + 12 \cdot \sqrt{2}\pi\rho\left\|K\right\|_2, 4\sqrt{d}\sqrt{2}\pi^2\rho^2\left\|K\right\|_2^2\right\} \leq 60\sqrt{d}\left(\left\|K\right\|_2^2 + 1\right).$$

Because any sinusoidal basis element T_K can be expressed as an expectation of random ReLUs and because P is a linear combination of a finite number of those basis elements, we

can also represent P as an expectation over ReLUs. We define $g : \mathbb{R} \times \mathbb{S}^{d-1} \to \mathbb{R}$ as

$$g(b, w) = \frac{Q_{k,d}}{|\mathcal{K}_{k,d,w}|} \sum_{K \in \mathcal{K}_{k,d,w}} \beta_K \psi_K(b) = \frac{1}{\Pr_{\mathbf{w} \sim \mathcal{D}_{\text{weights},k}}[\mathbf{w} = w]} \sum_{K \in \mathcal{K}_{k,d,w}} \beta_K \psi_K(b),$$

and represent $P(x)$ as an infinite mixture of ReLU functions weighted by g over all $x \in [-1, 1]^d$.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{b},\mathbf{w}} \left[g(\mathbf{b}, \mathbf{w}) \text{ReLU}\left(\langle \mathbf{w}, x \rangle - \mathbf{b} \right) \right] \\
&= \sum_{w \in \text{supp}(\mathcal{D}_{\text{weights},k})} \mathop{\mathbb{E}}_{\mathbf{b} \sim \mathcal{D}_{\text{bias}}} \left[\sum_{K \in \mathcal{K}_{k,d,w}} \beta_K \psi_K(\mathbf{b}) \text{ReLU}\left(\langle w, x \rangle - \mathbf{b} \right) \right] \\
&= \sum_{w \in \text{supp}(\mathcal{D}_{\text{weights},k})} \sum_{K \in \mathcal{K}_{k,d,w}} \beta_K \mathop{\mathbb{E}}_{\mathbf{b} \sim \mathcal{D}_{\text{bias}}} \left[\psi_K(\mathbf{b}) \text{ReLU}\left(\frac{1}{\|K\|_2} \langle K, x \rangle - \mathbf{b} \right) \right] \\
&= \sum_{K \in \mathcal{K}_{k,d}} \beta_K \mathop{\mathbb{E}}_{\mathbf{b} \sim \mathcal{D}_{\text{bias}}} \left[\psi_K(\mathbf{b}) \text{ReLU}\left(\frac{1}{\|K\|_2} \langle K, x \rangle - \mathbf{b} \right) \right] \\
&= \sum_{K \in \mathcal{K}_{k,d}} \beta_K T_K(\rho x) \\
&= P(x).
\end{aligned}$$

To conclude the proof, let $(\mathbf{w}^{(1)}, \mathbf{b}^{(1)}), \ldots, (\mathbf{b}^{(r)}, \mathbf{w}^{(m)})$ be independent copies of $(\mathbf{w}, \mathbf{b})$, and define $\mathbf{g}^{(i)} \in L_2([-1, 1]^d)$ for $i = 1, \ldots, m$ by

$$\mathbf{g}^{(i)}(x) := g(\mathbf{w}^{(i)}, \mathbf{b}^{(i)}) \text{ReLU}(\langle \mathbf{w}^{(i)}, x \rangle - \mathbf{b}^{(i)}).$$

Now we apply Lemma 2.14 to the random variables $\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}$. Note that

$$\mathop{\mathbb{E}}_{\mathbf{b}^{(i)},\mathbf{w}^{(i)}} \left[\mathbf{g}^{(i)}(x) \right] = P(x).$$

To apply the lemma, we first bound $\|\mathbf{g}^{(i)}\|_{[-1,1]^d}$:

$$
\begin{aligned}
\left\|\mathbf{g}^{(i)}\right\|_{[-1,1]^d} &\leq \max_{b\in[-2\sqrt{d},2\sqrt{d}],w\in\mathbb{S}^{d-1},x\in[-1,1]^d} \left|g(b,w)\mathrm{ReLU}\left(\langle w,x\rangle - b\right)\right| \\
&\leq \left(\max_{b,w,x}\left|\mathrm{ReLU}\left(\langle w,x\rangle - b\right)\right|\right)\left(\max_{b,w}\left|g(b,w)\right|\right) \\
&= \left(\max_{w,x}\|w\|_2\|x\|_2 + \max_b|b|\right)\left(\max_{b,w}\frac{Q_{k,d}}{|\mathcal{K}_{k,d,w}|}\left|\sum_{K\in\mathcal{K}_{k,d,w}}\beta_K\psi_K(b)\right|\right) \\
&\leq 3\sqrt{d}Q_{k,d}\max_w\frac{1}{|\mathcal{K}_{k,d,w}|}\sum_{K\in\mathcal{K}_{k,d,w}}|\beta_K|\cdot 60\sqrt{d}\left(\|K\|_2^2+1\right) \\
&\leq 360dQ_{k,d}\beta_{\max}k^2.
\end{aligned}
$$

Therefore, with probability $1-\delta$ over the choice of $\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(m)}$, we have

$$
\begin{aligned}
\inf_{f\in\mathrm{Span}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(r)}\right)}\|f-P\|_{[-1,1]^d} &\leq \left\|\frac{1}{m}\sum_{i=1}^m\mathbf{g}^{(i)} - \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^m\mathbf{g}^{(i)}\right]\right\|_{[-1,1]^d} \\
&\leq \frac{360d\beta_{\max}k^2Q_{k,d}}{\sqrt{m}}\left(1+\sqrt{2\ln\frac{1}{\delta}}\right)\leq\epsilon,
\end{aligned}
$$

which holds as long as we choose m with

$$
m \geq \frac{360^2d^2\beta_{\max}^2k^4Q_{k,d}^2}{\epsilon^2}\left(1+\sqrt{2\ln\frac{1}{\delta}}\right)^2.
$$

Based on Definiton 2.2, this gives the desired upper bound on MinWidth. □

2.4 Negative results for Lipschitz targets

This section consists of the statement and proof of two lower bounds on the minimum width necessary for two-layer random ReLU feature networks to approximate certain families of functions. Section 2.4.1 provides a formalization of the primary negative result of the chapter (Theorem 2.2) and provides an overview of its proof strategy. Sections 2.4.2 to 2.4.4 contain the proofs of the key building blocks of this theorem. Finally, Section 2.4.5 states and

proves a modification of Theorem 2.2, which gives an *explicit* target function that random ReLU feature networks cannot efficiently approximate, as opposed to proving the existence on some target.

2.4.1 Proof outline for Theorem 2.15

We give lower-bounds on the minimum width needed to ϵ-approximate L-Lipschitz functions using two-layer random ReLU feature networks. Below we present a formal statement of Theorem 2.2, which shows that a particular family of "simple" functions must contain some hard-to-approximate function. Like the upper-bounds in Section 2.3, the minimum width is polynomial (in fact linear) in the quantity $Q_{k,d}$, where $k = \Theta(L/\epsilon)$.

Theorem 2.15. *[Formal version of Theorem 2.2: Lower-bound for L-Lipschitz functions] Fix any $\epsilon, L > 0$ and fix any symmetric ReLU parameter distribution $\mathcal{D}$. Then, there exists some multi-index $K \in \mathbb{N}^d$ with $\|K\|_2 \leq L/18\epsilon$ such that the function $h(x) := 4\epsilon T_K$ (recall that $T_K \in \mathcal{T}$) satisfies $\|h\|_{\text{Lip}} \leq L$ and*

$$\text{MinWidth}_{h,\epsilon,\frac{1}{2},[-1,1]^d,\mathcal{D}} \geq \frac{1}{4} Q_{L/18\epsilon,d}.$$

The informal version, Theorem 2.2, follows by applying Fact 2.8 to lower-bound $Q_{k,d}$. We note that the function f used in the lower-bound aligns nicely with the approximation techniques from Section 2.3 because h is *(i)* a ridge function and *(ii)* a scalar multiple of a sinusoidal function from the trigonometric basis $\mathcal{T}$.

We prove Theorem 2.15 in stages by proving a sequence of claims which are successively more closely tailored to our random ReLU feature model.

2.4.1.1 Negative results for generic random feature models

In Section 2.4.2, we state and prove Theorem 2.18, which gives a general result about the limitations of linear combinations of m random features. This theorem states that

a large fraction of any set of N orthonormal functions must be inapproximable by linear combinations of m random features when $N \gg m$. We state a simplified version of the theorem below.

Theorem 2.16. *Let $\Phi = \{\varphi_1, \ldots, \varphi_N\} \subset L_2(\mu)$ be a family of N functions such that $\langle \varphi_i, \varphi_{i'} \rangle_\mu = \mathbb{1}\{i = i'\}$. Let $\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}$ be i.i.d. copies of an $L_2(\mu)$-valued random variable. Then, there exists some $\varphi_i \in \Phi$ such that*

$$\mathop{\mathbb{E}}_{\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(m)}} \left[\inf_{f \in \mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m} \|f - \varphi_i\|_\mu^2 \right] \geq 1 - \frac{m}{N}.$$

The proof hinges on an intuitive linear algebraic fact generalized to function spaces: N orthogonal vectors cannot all be close to the span of m vectors when $N \gg m$. It does so by applying the Hilbert Projection Theorem (Fact 2.19). The full generality of Theorem 2.18 also includes function families Φ that are "nearly orthonormal" rather than strictly orthonormal. It also proves the inapproximability of some explicit function φ_1 when the family Φ satisfies a suitable notion of symmetry relative to $\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}$.

2.4.1.2 Lower bounds on minimum widths of random ReLU feature networks

Lemma 2.21 of Section 2.4.3 adapts Theorem 2.18 to our random ReLU features by giving a lower-bound on the minimum-width random ReLU feature network needed to ϵ-approximate some function for any $\epsilon > 0$. Below is a simplified version of the lemma that is restricted to orthonormal function families, considers only the uniform measure over $[-1, 1]^d$, and omits the special "symmetric case" for Φ.

Lemma 2.17. *Let $\mathcal{D}$ be a symmetric ReLU parameter distribution. Fix any $\Phi = \{\varphi_1, \ldots, \varphi_N\} \subset L_2([-1, 1]^d)$ such that $\langle \varphi_i, \varphi_{i'} \rangle_{[-1,1]^d} = \mathbb{1}\{i = i'\}$. Then, for any $\epsilon > 0$, there exists some $\varphi_i \in \Phi$ such that* $\mathrm{MinWidth}_{4\epsilon\varphi_i,\epsilon,1/2,[-1,1]^d,\mathcal{D}} \geq N/4$.

The proof combines a scaling argument with the definition of MinWidth to provide lower-bounds for any choice of the error parameter ϵ.

2.4.1.3 Conclusion

We conclude the proof of Theorem 2.15 in Section 2.4.4. Lemma 2.22 shows the existence of a low-degree element of the sinusoidal basis $\mathcal{T}$ that cannot be approximated over $[-1,1]^d$ by an random ReLU feature network of small width. It does so by defining the orthonormal family of functions to be $\Phi := \{T_K \in \mathcal{T} : K \in \mathcal{K}_{k,d}\}$ and invoking Lemma 2.21. The proof of Theorem 2.15 only requires applying Lemma 2.22 for some $k = \Theta(L/\epsilon)$ and showing that all $T_K \in \Phi$ have $\|T_K\|_{\text{Lip}} \leq L$.

Lemma 2.22 also yields an immediate proof of Theorem 2.27, the Sobolev analogue of Theorem 2.15, in Section 2.5.2. Theorem 2.27 uses the same function family Φ, but must bound the Sobolev norm of all functions in Φ rather than the Lipschitz constant.

2.4.2 Negative results for generic random feature models (Theorem 2.18)

In Theorem 2.18, we give the most general form of our lower-bound. In this setting, we consider linear combinations of features drawn independently from some distribution over functions (which are not required to be ReLUs or even ridge functions). We argue that the span of any m such random functions in $L_2(\mu)$ cannot cover more than m dimensions of that function space and that we therefore cannot approximate most of the members of a family of N orthonormal functions if $N \gg m$.

If the family of N functions satisfies a suitable notion of symmetry with respect to the random features, then we can additionally argue that each function in that family is equally likely to be inapproximable. This makes it possible to construct a single explicit function that cannot be approximated with high probability by linear combinations of random features. We give the relevant notion of symmetry below:

Definition 2.6 (Symmetry of random functions)**.** Let $\mathbf{g}$ be an $L_2(\mu)$-valued random variable for some measure μ. We say $\mathbf{g}$ is *symmetric* with respect to the set of functions $\Phi = \{\varphi_1, \ldots, \varphi_N\} \subset L_2(\mu)$ if the distribution of $\langle \mathbf{g}, \varphi_i \rangle_\mu$ is the same for all $i = 1, \ldots, N$.

In fact, strict orthonormality of the hard functions is not needed for our approach; we introduce a notion of "average coherence,"

which allows us to quantify how far the family is from being orthogonal and prove lower-bounds that depend on this quantity.

Definition 2.7 (Average coherence)**.** For any set of functions $\Phi = \{\varphi_1, \ldots, \varphi_N\} \subset L_2(\mu)$ with $\|\varphi_i\|_\mu = 1$ for all $i = 1, \ldots, N$, its *(average) coherence* is $\kappa(\Phi) := \sqrt{\sum_{i \neq j} \langle \varphi_i, \varphi_j \rangle_\mu^2}$.

We are particularly interested in large collections of functions with low coherence. Note that a collection of orthogonal functions has zero coherence. Our main approximation lower bounds in Theorems 2.15 and 2.23 are achieved using an orthogonal collection. However, our general lower bound (Theorem 2.18) extends to the case where the family of functions has small (but nonzero) coherence.

The following general lower bound works for any distribution over random features that meets the above symmetry condition and for any set of "nearly-orthonormal" functions that have a bounded average coherence κ. It is akin to Theorem 19 of Kamath, Montasser, and Srebro, 2020 although that result does not involve a symmetry notion (and hence does not yield an explicit hard function).

Theorem 2.18 (Lower-bound for linear combinations of random features)**.** *Fix a family of functions $\Phi = \{\varphi_1, \ldots, \varphi_N\} \subset L_2(\mu)$ with $\|\varphi_i\|_\mu^2 = 1$ for all $i = 1, \ldots, N$. Let $\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}$ be i.i.d. copies of an $L_2(\mu)$-valued random variable. Then, there exists some $\varphi_i \in \Phi$ such that*

$$\mathop{\mathbb{E}}_{\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}} \left[\inf_{g \in \mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m} \|g - \varphi_i\|_\mu^2 \right] \geq 1 - \frac{m\,(1+\kappa(\Phi))}{N}. \tag{2.10}$$

In particular, for any $\alpha \in [0, 1]$,

$$\Pr_{\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}} \left[\inf_{g \in \mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m} \|g - \varphi_i\|_\mu^2 \geq \alpha \left(1 - \frac{m\,(1+\kappa(\Phi))}{N} \right) \right] \geq (1-\alpha) \left(1 - \frac{m\,(1+\kappa(\Phi))}{N} \right). \tag{2.11}$$

Moreover, if $\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}$ are symmetric with respect to Φ, then (2.10) *and* (2.11) *hold for*

$i = 1$.

We recall two tools that will be used in the proof of Theorem 2.18, namely the Hilbert projection theorem and the Boas-Bellman inequality.

Fact 2.19 (Hilbert projection theorem (Rudin, 1987))**.** *For some measure μ and $g^{(1)}, \dots, g^{(m)} \in L_2(\mu)$, consider the subspace $W = \mathrm{Span}(g^{(1)}, \dots, g^{(m)})$ of $L_2(\mu)$. For any $h \in L_2(\mu)$, it holds that*

$$\inf_{f \in W} \|f - h\|_\mu^2 = \|\Pi_W h - h\|_\mu^2 = \|h\|_\mu^2 - \|\Pi_W h\|_\mu^2 \,, \tag{2.12}$$

where $\Pi_W \colon L_2(\mu) \to W$ is the orthogonal projection operator for W. Moreover, the orthogonal projection $\Pi_W h$ depends on h only through $(\langle g^{(1)}, h\rangle_\mu, \dots, \langle g^{(m)}, h\rangle_\mu)$.

The following is a generalization of Bessel's inequality due to Boas, 1941 and Bellman, 1944, specialized to our present context.

Fact 2.20 (Boas-Bellman inequality)**.** *For any $h, \varphi_1, \dots, \varphi_N \in L_2(\mu)$,*

$$\sum_{i=1}^{N} \langle h, \varphi_i\rangle_\mu^2 \le \|h\|_\mu^2 \left(\max_{1 \le i \le N} \|\varphi_i\|_\mu^2 + \kappa(\{\varphi_1, \dots, \varphi_N\}) \right). \tag{2.13}$$

Proof of Theorem 2.18. By the Hilbert projection theorem (Fact 2.19), for all $i \in [N]$ we have that

$$\mathop{\mathbb{E}}_{\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(m)}} \left[\inf_{f \in \mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m} \|f - \varphi_i\|_\mu^2 \right] = 1 - \mathop{\mathbb{E}}_{\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(m)}} \left[\left\| \Pi_{\mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m} \varphi_i \right\|_\mu^2 \right].$$

We now upper-bound the sum of the expected norms of the projections of each function in Φ onto $\mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m$. Let $\mathbf{u}_1, \dots, \mathbf{u_d}$ be an orthonormal basis for $\mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m$, where

$\mathbf{d} := \dim \operatorname{Span}(\mathbf{g}^{(j)})_{j=1}^m$. Then

$$\begin{aligned}
\sum_{i=1}^N \left\| \Pi_{\operatorname{Span}(\mathbf{g}^{(j)})_{j=1}^m} \varphi_i \right\|_\mu^2 &= \sum_{i=1}^N \sum_{k=1}^{\mathbf{d}} \langle \mathbf{u}_k, \varphi_i \rangle_\mu^2 = \sum_{k=1}^{\mathbf{d}} \sum_{i=1}^N \langle \mathbf{u}_k, \varphi_i \rangle_\mu^2 && \text{(Plancherel's identity, Fact 2.3)} \\
&\leq \sum_{k=1}^{\mathbf{d}} (1 + \kappa(\Phi)) = \mathbf{d} \cdot (1 + \kappa(\Phi)) && \text{(Fact 2.20)} \\
&\leq m \cdot (1 + \kappa(\Phi)) && (\dim \operatorname{Span}(\mathbf{g}^{(j)})_{j=1}^m \leq m).
\end{aligned}$$

Hence, we conclude by linearity of expectation that

$$\frac{1}{N} \sum_{i=1}^N \mathop{\mathbb{E}}_{\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}} \left[\inf_{f \in \operatorname{Span}(\mathbf{g}^{(j)})_{j=1}^m} \|f - \varphi_i\|_\mu^2 \right] \geq 1 - \frac{m \cdot (1 + \kappa(\Phi))}{N}. \tag{2.14}$$

Therefore, there exists some $i \in [N]$ such that

$$\mathop{\mathbb{E}}_{\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(r)}} \left[\inf_{g \in \operatorname{Span}(\mathbf{g}^{(j)})_{j=1}^m} \|g - \varphi_i\|_\mu^2 \right] \geq 1 - \frac{r \cdot (1 + \kappa(\Phi))}{N},$$

which gives us inequality (2.10). Inequality (2.11) follows by an application of Markov's inequality to the random variable $1 - \inf_{f \in \operatorname{Span}(\mathbf{g}^{(j)})_{j=1}^m} \|f - \varphi_i\|_\mu^2$ (which is easily seen to be non-negative), which by the first part of the theorem has expected value at most $m \cdot (1 + \kappa(\Phi))/N$.

We conclude by proving the stronger version of the theorem, where we additionally assume that the random features are symmetric. Suppose $\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}$ are symmetric with respect to Φ. As mentioned in Fact 2.19, the orthogonal projection $\Pi_{\operatorname{Span}(\mathbf{g}^{(j)})_{j=1}^m} \varphi_1$ depends on φ_1 only through the (random) vector $(\langle \mathbf{g}^{(1)}, \varphi_1 \rangle_\mu, \ldots, \langle \mathbf{g}^{(m)}, \varphi_1 \rangle_\mu)$. Therefore, by the symmetry assumption on the distribution of each $\mathbf{g}^{(i)}$, the orthogonal projection $\Pi_{\operatorname{Span}(\mathbf{g}^{(j)})_{j=1}^m} \varphi_1$ has the same distribution as $\Pi_{\operatorname{Span}(\mathbf{g}^{(j)})_{j=1}^m} \varphi_i$ for all $i \in [N]$. Then

$$\mathop{\mathbb{E}}_{\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}} \left[\left\| \Pi_{\operatorname{Span}(\mathbf{g}^{(j)})_{j=1}^m} \varphi_1 \right\|_\mu^2 \right] = \frac{1}{N} \sum_{i=1}^N \mathop{\mathbb{E}}_{\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(m)}} \left[\left\| \Pi_{\operatorname{Span}(\mathbf{g}^{(j)})_{j=1}^m} \varphi_i \right\|_\mu^2 \right]. \tag{2.15}$$

Plugging Equation (2.15) into Inequality (2.14) proves that Inequalities (2.10) and (2.11) hold for $i = 1$.

□

2.4.3 Lower bounds for minimum widths of random ReLU feature networks (Lemma 2.21)

Here, we specialize Theorem 2.18 to the case of ReLU networks, which prepares us to prove the specific lower-bounds that will be given in the subsequent sections.

Lemma 2.21. *Let $\mathcal{D}$ be a symmetric ReLU parameter distribution and μ be some measure over $\mathbb{R}^d$. Fix any $\Phi = \{\varphi_1, \dots, \varphi_N\} \subset L_2(\mu)$ such that $\|\varphi_i\|_\mu^2 = 1$ for all $i \in [N]$. Then, for any $\epsilon > 0$, there exists some $\varphi_i \in \Phi$ such that*

$$\mathrm{MinWidth}_{4\epsilon\varphi_i, \epsilon, \frac{1}{2}, \mu, \mathcal{D}} \geq \frac{N}{4 + 4\kappa(\Phi)}. \tag{2.16}$$

Additionally, suppose that the functions in Φ are symmetric up to some permutation of variables and μ is invariant to permutation of variables. That is, for all $i, i' \in [N]$ there exists a permutation $\pi_{i,i'}$ over $[d]$ such that $\varphi_i \circ \pi_{i,i'} = \varphi_{i'}$. Then, Inequality (2.16) always holds for $i = 1$.

Proof. By applying Theorem 2.18 for any $m \leq N/(4 + 4\kappa(\Phi))$ and for $\alpha = 1/3$, there exists some $i \in [N]$ such that

$$\Pr_{\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(m)}} \left[\inf_{f \in \mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m} \|\varphi_i - f\|_\mu < \frac{1}{4} \right] < \frac{1}{2}.$$

Note that for all h, there exists $f \in \mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m$ with $\|f - h\|_\mu < \epsilon$ if and only if there exists $f' \in \mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m$ with $\|h/4\epsilon - f'\|_\mu < 1/4$. Thus, we conclude the following:

$$\Pr_{\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(m)}} \left[\inf_{f \in \mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m} \|4\epsilon\varphi_i - f\|_\mu < \epsilon \right] = \Pr_{\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(m)}} \left[\inf_{f' \in \mathrm{Span}(\mathbf{g}^{(j)})_{j=1}^m} \|\varphi_i - f'\|_\mu < \frac{1}{4} \right] < \frac{1}{2}.$$

To prove the stronger version of the theorem that assumes permutation symmetry for Φ,

we apply the stronger version of Theorem 2.18. To do so, we must show that each $\mathbf{g}^{(i)}$ is symmetric with respect to Φ.

Because the ReLU feature parameters $\mathbf{b}^{(i)}$ are chosen independently $\mathbf{w}^{(i)}$ and the distribution of $\mathbf{w}^{(i)}$ is invariant to variable permutation, each $\mathbf{g}^{(i)}$ is drawn from a distribution that is also invariant to permutation. We prove the symmetry property by showing that the inner product distributions are identical for $\mathbf{g}^{(1)}$, without loss of generality. Because each function in $\varphi_1, \ldots, \varphi_N$ is symmetric to a permutation of variables, there exists some permutation $\pi_{i,i'}$ such that for all $x \in \mu$, $\varphi_i(x) = \varphi_{i'}(\pi_{i,i'}(x))$. To show that the two inner products induce the same distribution, consider any $z \in \mathbb{R}$. Then:

$$
\begin{aligned}
&\Pr_{\mathbf{g}^{(1)}}[\langle \mathbf{g}^{(1)}, \varphi_i \rangle_\mu \geq z] \\
&= \Pr_{\mathbf{g}^{(1)}}\left[\mathop{\mathbb{E}}_{\mathbf{x}\sim\mu}[\mathbf{g}^{(1)}(\mathbf{x})\varphi_i(\mathbf{x})] \geq z\right] \\
&= \Pr_{\mathbf{g}^{(1)}}\left[\mathop{\mathbb{E}}_{\mathbf{x}\sim\mu}[\mathbf{g}^{(1)}(\mathbf{x})\varphi_j(\pi_{i,i'}(\mathbf{x}))] \geq z\right] && \text{(Existence of } \pi_{i,i'}\text{)} \\
&= \Pr_{\mathbf{g}^{(1)}}\left[\mathop{\mathbb{E}}_{\mathbf{x}\sim\mu}[\mathbf{g}^{(1)}(\pi_{i,i'}(\mathbf{x}))\varphi_j(\pi_{i,i'}(\mathbf{x}))] \geq z\right] && \text{(Symmetry of } \mathbf{g}^{(1)}\text{'s distribution)} \\
&= \Pr_{\mathbf{g}^{(1)}}\left[\mathop{\mathbb{E}}_{\mathbf{x}\sim\mu}[\mathbf{g}^{(1)}(\mathbf{x})\varphi_{i'}(\mathbf{x})] \geq z\right] && \text{(Symmetry of } \mu\text{)} \\
&= \Pr_{\mathbf{g}^{(1)}}[\langle \mathbf{g}^{(1)}, \varphi_{i'} \rangle_\mu \geq z]
\end{aligned}
$$

Hence, recalling Definition 2.6, $\mathbf{g}^{(1)}$ is symmetric with respect to $\varphi_1, \ldots, \varphi_N$. By invoking Theorem 2.18 with the additional symmetry assumption, inequality (2.16) holds when $i = 1$. □

2.4.4 Proof of Theorem 2.15

To finalize the proof of Theorem 2.15, we first show that some low-degree trigonometric polynomial cannot be approximated by a combination of random ReLU features.[5]

[5]We prove Lemma 2.22 separately from Theorem 2.15 since we also make use of Lemma 2.22 in Section 2.5.2 when proving lower-bounds based on the Sobolev norm of a function, rather than its Lipschitz constant.

Lemma 2.22. *For any $k > 0$, any $\epsilon > 0$, and any symmetric ReLU parameter distribution $\mathcal{D}$, there exists some $K \in \mathbb{N}^d$ with $\|K\|_2 \leq k$ such that*

$$\text{MinWidth}_{4\epsilon T_K, \epsilon, \frac{1}{2}, [-1,1]^d, \mathcal{D}} \geq \frac{1}{4} Q_{k,d}.$$

Proof. Let $\mathcal{T}_k := \{T_K \in \mathcal{T} : K \in \mathcal{K}_{k,d}\}$ be a subset of trigonometric basis elements with bounded degree. Because $\mathcal{T}$ is an orthonormal family of functions, $\mathcal{T}_k$ is as well, and $\kappa(\mathcal{T}_k) = 0$. Then, Lemma 2.21 implies the existence of some $T_K \in \mathcal{T}_k$ such that

$$\text{MinWidth}_{4\epsilon T_K, \epsilon, \frac{1}{2}, [-1,1]^d, \mathcal{D}} \geq \frac{|\mathcal{T}_k|}{4} = \frac{1}{4} Q_{k,d}. \qquad \square$$

We prove Theorem 2.15 by applying Lemma 2.22 and bounding the Lipschitz constant of the inapproximable function.

Theorem 2.15. *[Formal version of Theorem 2.2: Lower-bound for L-Lipschitz functions] Fix any $\epsilon, L > 0$ and fix any symmetric ReLU parameter distribution $\mathcal{D}$. Then, there exists some multi-index $K \in \mathbb{N}^d$ with $\|K\|_2 \leq L/18\epsilon$ such that the function $h(x) := 4\epsilon T_K$ (recall that $T_K \in \mathcal{T}$) satisfies $\|h\|_{\text{Lip}} \leq L$ and*

$$\text{MinWidth}_{h, \epsilon, \frac{1}{2}, [-1,1]^d, \mathcal{D}} \geq \frac{1}{4} Q_{L/18\epsilon, d}.$$

Proof of Theorem 2.15. Consider any $T_K \in \mathcal{T}$ with $\|K\|_2 \leq k$. Then, for all $x, x' \in [-1,1]^d$,

$$|T_K(x) - T_K(x')| \leq \sqrt{2}\pi \langle K, x - x' \rangle \leq \sqrt{2}\pi \|K\|_2 \|x - x'\|_2 \leq \sqrt{2}\pi k \|x - x'\|_2.$$

Thus, $\|T_K\|_{\text{Lip}} \leq \sqrt{2}\pi k$ and $\|f\|_{\text{Lip}} \leq 4\sqrt{2}\pi k\epsilon \leq 18k\epsilon$. By applying Lemma 2.22 with $k := L/18\epsilon$, there exists a satisfactory h such that $\|h\|_{\text{Lip}} \leq L$. $\square$

2.4.5 Negative result for an *explicit* L-Lipschitz target

The lower-bound established in Theorem 2.15 is non-explicit; it guarantees the existence of some inapproximable function in $\mathcal{T}$, but does not by itself let us deduce the specific identity of a hard function. Since it is desirable to have a lower-bound for a fully explicit function, we also give a variant that achieves this goal at only a small cost in the resulting quantitative lower-bound:

Theorem 2.23. *For some $\epsilon, L > 0$, let $\ell := \min(\lceil d/2 \rceil, \lfloor L^2/32\pi^2\epsilon^2 \rfloor)$. Fix any symmetric ReLU parameter distribution $\mathcal{D}$. Then the function $h(x) := 4\sqrt{2}\epsilon \sin(\pi \sum_{i=1}^{\ell} x_i)$ satisfies $\|h\|_{\mathrm{Lip}} \leq L$ and*

$$\mathrm{MinWidth}_{h,\epsilon,\frac{1}{2},[-1,1]^d,\mathcal{D}} \geq \frac{1}{4}\binom{d}{\ell} \geq \exp\left(\Omega\left(\min\left(\frac{L^2}{\epsilon^2}\log\left(\frac{d\epsilon^2}{L^2}+2\right), d\right)\right)\right).$$

Comparing the quantitative lower bounds of Theorem 2.15 and Theorem 2.23, we see that the latter is weaker only by a logarithmic factor in the exponent.

The only difference between the proofs of Theorems 2.15 and 2.23 is in the last step. Theorem 2.23 relies on Lemma 2.24, an analogue of Lemma 2.22, which invokes Lemma 2.21 with a different family Φ of trigonometric polynomials that are symmetric up to a permutation of variables. That is, for every $T_K, T_{K'} \in \Phi$, there exists some permutation π over $[d]$ such that $T_K = T_{K'} \circ \pi$. (Roughly speaking, the larger family of orthonormal functions used in the proof of Theorem 2.15 consists of functions of the form $\sin(\pi \langle K, x \rangle)$ where $K \in \mathbb{N}^d$ is only constrained by having $\|K\|$ satisfy some bound, whereas the smaller family of orthonormal functions used in the proof of Lemma 2.24 consists of functions of the form $\sin(\pi \langle K, x \rangle)$ where K is restricted to be a 0/1 vector of some specific Hamming weight. The latter family is easily seen to satisfy symmetry with respect to any permutation π of the d coordinates, whereas the former family does not satisfy such a symmetry condition.) This symmetry condition makes it easy to argue that all functions in the symmetric family Φ are "equally hard," from which a lower bound follows straightforwardly.

Finally, we mention that Lemma 2.24 also supports a proof of the inapproximability of an explicit function with bounded Sobolev norm; this is established in Theorem 2.28 of Section 2.5.2.

As in the previous section, we prove Lemma 2.24 by applying Lemma 2.21 to a family of orthonormal functions. In order to obtain an explicit function f that is hard to approximate, we invoke the stronger version of Lemma 2.21, which requires showing that that the family of functions exhibits symmetry up to a permutation of variables.

Lemma 2.24. *For any $\ell \in \mathbb{Z}^+$ with $\ell \leq d$, any $\epsilon > 0$, and any symmetric ReLU parameter distribution $\mathcal{D}$, define $h : \mathbb{R}^d \to \mathbb{R}$ to be the function $h(x) := 4\sqrt{2}\epsilon \sin(\pi \sum_{i=1}^{\ell} x_i)$. Then,*

$$\text{MinWidth}_{h,\epsilon,\frac{1}{2},[-1,1]^d,\mathcal{D}} \geq \frac{1}{4}\binom{d}{\ell}.$$

Proof. We prove the claim by constructing a family of functions Φ_ℓ with $\frac{1}{4\epsilon}h \in \Phi_\ell$ and applying Lemma 2.21. We define a family of functions

$$\Phi_\ell := \left\{ \varphi_S : x \mapsto \sqrt{2}\sin\left(\pi \sum_{i \in S} x_i\right) \;\mid\; S \subseteq [d], |S| = \ell \right\}.$$

Note that $|\Phi_\ell| = \binom{d}{\ell}$ and that $\varphi_1 := \frac{1}{4\epsilon}h = \varphi_{[\ell]} \in \Phi_\ell$. Because $\Phi_\ell \subseteq \mathcal{T}$ and $\mathcal{T}$ is an orthonormal basis for $L_2([-1,1]^d)$ (Fact 2.5), the functions in Φ_ℓ are orthonormal and $\kappa(\Phi_\ell) = 0$. Thus, because the Φ_ℓ satisfies the symmetry conditions for the special case of Lemma 2.21,

$$\text{MinWidth}_{h,\epsilon,\frac{1}{2},[-1,1]^d,\mathcal{D}} \geq \frac{1}{4}\binom{d}{\ell}. \qquad \square$$

Proof of Theorem 2.23. This is immediate from Lemma 2.24 and from the fact that $\|h\|_{\text{Lip}} = 4\pi\epsilon\sqrt{2\ell} \leq L$. The right-hand side of the bound follows by lower-bounding $\binom{d}{\ell}$ for our choice of ℓ.

If $\ell = \lceil d/2 \rceil$ and $d \geq 2$,[6] then

$$\binom{d}{\ell} \geq \left(\frac{d}{\lceil d/2 \rceil}\right)^{\lceil d/2 \rceil} \geq \left(\frac{3}{2}\right)^{d/2} \geq \exp\left(\Theta(d)\right).$$

Otherwise, $\ell < d/2$ and

$$\binom{d}{\ell} \geq \left(\frac{d}{\ell}\right)^{\ell} \geq \exp\left(\Theta\left(\ell \log\left(\frac{d}{\ell} + 2\right)\right)\right) = \exp\left(\Theta\left(\frac{L^2}{\epsilon^2} \log\left(\frac{d\epsilon^2}{L^2} + 2\right)\right)\right). \qquad \square$$

This matches the exponent asymptotically up to logarithmic factors of the corresponding Lipschitz upper-bound, Theorem 2.9.

2.5 Positive and negative results for Sobolev targets

In this section, we present upper and lower bounds on the width required for depth-2 RBL ReLU approximation of functions in a larger family of smooth functions, namely the order-s Sobolev functions. Sobolev spaces are normed function spaces arising in the study of partial differential equations, and their norms quantify the effective "bumpiness" of their constituent functions in terms of their weak derivatives. Let μ denote the uniform probability measure on an open subset of $\mathbb{R}^d$. Following Leoni (2017), we denote the *order-s Sobolev space of functions in $L_2(\mu)$* for $s \in \mathbb{N}$ by[7]

$$H^s(\mu) := \left\{h : \mathbb{R}^d \to \mathbb{R} : \ D^{(M)}h \in L_2(\mu), \ \forall M \in \mathbb{N}^d \text{ s.t. } |M| \leq s\right\}.$$

The norm on this space is

$$\|h\|_{H^s(\mu)} := \sqrt{\sum_{|M| \leq s} \|D^{(M)}h\|_\mu^2}.$$

[6]There is no need to consider the $d = 1$ case, because then $\text{MinWidth}_{f,\epsilon,\frac{1}{2},[-1,1]^d,\mathcal{D}} \geq \frac{1}{4} = \exp(\Theta(1))$, which satisfies the claim.

[7]Technically, $D^{(M)}h$ is interpreted as the M-th weak partial derivative of h. However, it satisfies the integration-by-parts formulas that appear in the proof of Lemma 2.7, which is all we require.

(We do not consider Sobolev spaces in $L_p(\mu)$ for $p \neq 2$ since we rely on Hilbert space structure.)

We focus on the classical spaces $H^s(\mu)$ in $L_2(\mu)$, where μ is the uniform product probability measure on the torus $\mathbb{T}^d$ and $\mathbb{T} = \mathbb{R}/(2\mathbb{Z})$. As a short-hand, we refer to this space as $H^s(\mathbb{T}^d)$ in $L_2(\mathbb{T}^d)$. Recall that $\mathbb{T}$ is obtained by identifying points in $\mathbb{R}$ that differ by $2z$ for some $z \in \mathbb{Z}$. Functions on $\mathbb{T}^d$ can be regarded as functions on $[-1,1]^d$, which, along with their derivatives, satisfy the periodic boundary conditions. Note that $\mathcal{T}$ is also an orthonormal basis for $\mathbb{T}^d$, because all of the trigonometric polynomials in $\mathcal{T}$ and all their derivatives have periodic boundary conditions and because the probability density of the uniform distribution on $\mathbb{T}^d$ is the same as the density over the uniform distribution on $[-1,1]^d$.

2.5.1 Upper-bounds for functions in $H^s(\mathbb{T}^d)$

We prove an analogue to Theorem 2.9 that places an upper-bound on the minimum-width random ReLU feature network that approximates a function with bounded order-s Sobolev norm.

Theorem 2.25. *Fix some $\delta \in (0, 1/2]$, $\epsilon, \gamma > 0$, and $s \in \mathbb{Z}^+$. Let $k := \sqrt{s}\gamma^{1/s}/\epsilon^{1/s}$. Then, there exists some ReLU parameter distribution $\mathcal{D}$ such that for any fixed $h \in H^s(\mathbb{T}^d)$ that satisfies $\|h\|_{H^s(\mathbb{T}^d)} \leq \gamma$, we have*

$$\text{MinWidth}_{h,\epsilon,\delta,\mathbb{T}^d,\mathcal{D}} \leq O\left(\frac{s^2\gamma^{2+4/s}d^2}{\epsilon^{2+4/s}} Q_{k,d}^2 \ln\left(\frac{1}{\delta}\right)\right).$$

Remark 2.1. *When $s = 1$,*

$$\text{MinWidth}_{h,\epsilon,\delta,\mathbb{T}^d,\mathcal{D}} \leq O\left(\frac{\gamma^6 d^2}{\epsilon^6} Q_{\gamma/\epsilon,d}^2 \ln\left(\frac{1}{\delta}\right)\right),$$

which is a near-perfect match to the upper-bound for Lipschitz functions in Theorem 2.9. This is unsurprising, because all L-Lipschitz functions h with $|\mathbb{E}[h]| \leq L$ have a squared

1-order Sobolev norm with the following bound:

$$\|h\|_{H^s(\mathbb{T}^d)}^2 = \|h\|_{\mathbb{T}}^2 + \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{T}^d}\left[\|\nabla h(\mathbf{x})\|^2\right] \leq O(L^2).$$

Thus, the two theorems give nearly identical upper-bounds for L-Lipschitz functions that satisfy periodic boundary conditions.

Remark 2.2. *Applying Fact 2.8 to Theorem 2.25 implies that*

$$\text{MinWidth}_{h,\epsilon,\delta,\mathbb{T}^d,\mathcal{D}} \leq \ln\left(\frac{1}{\delta}\right)\exp\left(O\left(\min\left(d\log\left(\frac{s\gamma^{2/s}}{d\epsilon^{2/s}}+2\right), \frac{s\gamma^{2/s}}{\epsilon^{2/s}}\log\left(\frac{d\epsilon^{2/s}}{s\gamma^{2/s}}+2\right)\right)\right)\right).$$

Like the proof of Theorem 2.9, we first show that every function in $H^s(\mathbb{T}^d)$ can be approximated by low-degree trigonometric polynomial in Lemma 2.26, which is a parallel result to Lemma 2.10. Unlike Theorem 2.9, however, we require that h and its first s derivatives satisfy the periodic boundary conditions, which is assured by the fact that $h \in H^s(\mathbb{T}^d)$. Thanks to this assumption, we eliminate the need for the "reflection" trick from Lemma 2.10, which simplifies the proof.

Lemma 2.26 (Approximating Sobolev functions with low-degree trigonometric polynomials)**.** *Fix any values $\gamma, \epsilon > 0$ and $s \in \mathbb{Z}^+$. Consider any $h \in H^s(\mathbb{T}^d)$ with $\|h\|_{H^s(\mathbb{T}^d)} \leq \gamma$. Let $k := \sqrt{s}\gamma^{1/s}/(2\epsilon)^{1/s}$. Then, there exists a trigonometric polynomial*

$$P(x) = \sum_{K\in\mathcal{K}_{k,d}} \beta_K T_K(x)$$

such that $\|h - P\|_{\mathbb{T}^d} \leq \epsilon$. Moreover, $|\beta_K| \leq \|h\|_{\mathbb{T}^d} \leq \gamma$ for all $K \in \mathcal{K}_{k,d}$.

Proof. Because $\mathcal{T}$ is an orthonormal basis over $\mathbb{T}^d$, we express h as the expansion

$$h = \sum_{K\in\mathbb{Z}^d} \alpha_K T_K.$$

Since h can be regarded as a function on $[-1,1]^d$ whose first s partial derivatives satisfy

boundary conditions, Lemma 2.7 implies that this expansion of h can be differentiated term-by-term. By taking term-by-term partial derivatives of h, applying Parseval's identity (Fact 2.3), and using the known norms of partial derivatives of T_K (Fact 2.6), we obtain the following closed-form $L_2(\mathbb{T}^d)$ norm for $D^{(M)}h$ for all $M \in \mathbb{N}^d$ with $|M| \leq s$:

$$\left\|D^{(M)}h\right\|_{\mathbb{T}^d}^2 = \sum_{K\in\mathbb{Z}^d} \alpha_K^2 (\pi K)^{2M}.$$

Therefore, the squared $H^s(\mathbb{T}^d)$-norm of h can be written as

$$\|h\|_{H^s(\mathbb{T}^d)}^2 = \sum_{|M|\leq s} \left\|D^{(M)}h\right\|_{\mathbb{T}^d}^2 = \sum_{|M|\leq s} \sum_{K\in\mathbb{Z}^d} \alpha_K^2 (\pi K)^{2M} = \sum_{K\in\mathbb{Z}^d} \alpha_K^2 c_{K,s}, \tag{2.17}$$

where

$$c_{K,s} := \sum_{|M|\leq s} (\pi K)^{2M}.$$

We lower-bound $c_{K,s}$ in terms of s and $\|K\|_2$ with the multinomial theorem:

$$c_{K,s} \geq \sum_{|M|=s} (\pi K)^{2M} \geq \frac{\pi^{2s}}{s!} \sum_{|M|=s} \frac{s!}{M!} K^{2M} = \frac{\pi^{2s}}{s!} \|K\|_2^{2s} \geq \left(\frac{\pi^2 \|K\|_2^2}{s}\right)^s.$$

We define $\beta_K := \alpha_K$ for all $K \in \mathcal{K}_{k,d}$ and $\beta_K := 0$ for all other $K \in \mathbb{Z}^d$. Note that if $K \in \mathbb{Z}^d$ has $\|K\|_2 > k \geq \sqrt{s}\gamma^{1/s}/\pi\epsilon^{1/s}$, then $c_{K,s} > \gamma^2/\epsilon^2$. By Parseval's identity, we have $\beta_K^2 \leq \|h\|_{\mathbb{T}^d}^2$. Moreover,

$$\|h - P\|_{\mathbb{T}^d}^2 = \sum_{K\in\mathbb{Z}^d\setminus\mathcal{K}_{k,d}} \alpha_K^2 \leq \sum_{\substack{K\in\mathbb{Z}^d: \\ c_{K,s}>\gamma^2/\epsilon^2}} \alpha_K^2 \leq \sum_{\substack{K\in\mathbb{Z}^d: \\ c_{K,s}>\gamma^2/\epsilon^2}} \alpha_K^2 \cdot \frac{c_{K,S}}{\gamma^2/\epsilon^2} \leq \frac{\epsilon^2}{\gamma^2} \sum_{K\in\mathbb{Z}^d} \alpha_K^2 c_{K,s} \leq \epsilon^2.$$

Above, the first equality uses Parseval's identity, and the final equality uses Equation (2.17).

□

Proof of Theorem 2.25. This proof is identical to the proof of Theorem 2.9 in Section 2.3.1.3, except that we make use of Lemma 2.26 instead of Lemma 2.10, and instead set $k :=$

$\sqrt{s}\gamma^{1/s}/\epsilon^{1/s}$ and $\rho := 1$. □

2.5.2 Lower-bounds for functions in $H^s([-1,1]^d)$

Similar to Section 2.4, we give lower-bounds on the width of random ReLU feature neural networks required to approximate certain functions (now ones with bounded s-order Sobolev norm). As before, we present two variants of the lower-bound, one non-explicit tight bound and one looser explicit bound.

- Theorem 2.27 is analogous to Theorem 2.15. It shows the existence of some sinusoidal function with bounded Sobolev norm which matches the upper-bound Theorem 2.25 by depending on the same combinatorial term.
- Theorem 2.28, like Theorem 2.23, offers an explicit sinusoidal function with bounded Sobolev norm whose minimum width can be bounded by a term that differs from the asymptotics of the exponent of the upper-bound by a logarithmic factor.

These results follow from proofs that directly apply Lemmas 2.22 and 2.24 respectively and bound the s-order Sobolev norms of the resulting functions.

2.5.2.1 A tight lower-bound

We give a bound on the minimum width two-layer random ReLU feature network needed to approximate some function with bounded Sobolev norm, which is a scaled version of some function in $\mathcal{T}$. The family of functions is identical to that of Theorem 2.27; the only difference is that we parameterize the bounds by the s-order Sobolev norm of the function, rather than its Lipschitz constant.

Theorem 2.27. *Fix some $\epsilon, \gamma > 0$ and $s \in \mathbb{Z}_+$ with $\gamma^2/\epsilon^2 \geq 16(s+1)$. Let*

$$k := \frac{1}{\pi} \cdot \left(\frac{\gamma}{4\epsilon\sqrt{s+1}} \right)^{1/s}.$$

Then, there exists some $K \in \mathcal{K}_{k,d}$ such that for $h := 4\epsilon T_K$ and for any symmetric ReLU parameter distribution $\mathcal{D}$,

$$\text{MinWidth}_{h,\epsilon,\frac{1}{2},\mathbb{T}^d,\mathcal{D}} \geq \frac{1}{4} Q_{k,d},$$

and $\|h\|_{H^s(\mathbb{T}^d)} \leq \gamma$.

Remark 2.3. *By invoking Fact 2.8, we have*

$$\text{MinWidth}_{h,\epsilon,1/2,\mathbb{T}^d,\mathcal{D}} \geq \exp\left(\Omega\left(\min\left(d\log\left(\frac{\gamma^{2/s}}{d\epsilon^{2/s}}+2\right), \frac{\gamma^{2/s}}{\epsilon^{2/s}}\log\left(\frac{d\epsilon^{2/s}}{\gamma^{2/s}}+2\right)\right)\right)\right).$$

Note that we can drop $(s+1)^{1/s}$ terms from the asymptotics of the exponent, because $(s+1)^{1/s} \in (1,2]$ for all $s \in \mathbb{Z}^+$. The asymptotics of the exponents match the upper-bound on the minimum width presented in Remark 2.2, when $\delta = 1/2$ and s is regarded as a small constant.

Proof. To prove the existence of h, we need only invoke Lemma 2.22 for our choice of k. It remains to bound the s-order Sobolev norm of h. We do so by expanding the squared Sobolev norm of h and applying Fact 2.6 to obtain an exact representation of the norms of derivatives of the basis elements $T_K \in \mathcal{T}$.

$$\begin{aligned}
\|h\|^2_{H^s(\mathbb{T}^d)} &= \sum_{M:|M|\leq s} \left\|D^{(M)}h\right\|^2_{\mathbb{T}^d} = 16\epsilon^2 \sum_{M:|M|\leq s} \left\|D^{(M)}T_K\right\|^2_{\mathbb{T}^d} \\
&= 16\epsilon^2 \sum_{M:|M|\leq s} \pi^{2|M|}K^{2M} = 16\epsilon^2 \sum_{m=0}^{s} \pi^{2m} \sum_{|M|=m} K^{2M} \\
&\leq 16\epsilon^2 \sum_{m=0}^{s} \pi^{2m} \sum_{|M|=m} \frac{m!}{K!} K^{2M} = 16\epsilon^2 \sum_{m=0}^{s} \pi^{2m} \|K\|_2^{2m} \\
&\leq 16\epsilon^2 \sum_{m=0}^{s} \left(\pi^2 k^2\right)^m = 16\epsilon^2 \sum_{m=0}^{s} \left(\frac{\gamma^{2/s}}{16^{1/s}\epsilon^{2/s}(s+1)^{1/s}}\right)^m
\end{aligned}$$

Because of our assumed lower-bound on γ^2/ϵ^2, the final term of the sum cannot be smaller than any preceding terms. Therefore, we conclude with the following trivial bound on the

sum.

$$\|h\|_{H^s(\mathbb{T}^d)}^2 \leq 16\epsilon^2 \sum_{m=0}^{s} \left(\frac{\gamma^{2/s}}{16^{1/s}\epsilon^{2/s}(s+1)^{1/s}} \right)^m \leq 16\epsilon^2(s+1) \left(\frac{\gamma^{2/s}}{16^{1/s}\epsilon^{2/s}(s+1)^{1/s}} \right)^s = \gamma^2.$$

□

2.5.2.2 A lower-bound for an explicit sinusoidal function

We give an explicit lower-bound that bounds the Sobolev norm of the function f used in Lemma 2.24. In that way, it is nearly identical to Theorem 2.23.

Theorem 2.28. *Fix some $\epsilon, \gamma > 0$ and $s \in \mathbb{Z}_+$ with $\gamma^2/\epsilon^2 \geq 16(s+1)$. Let*

$$\ell := \min\left(\left\lceil \frac{d}{2} \right\rceil, \left\lfloor \frac{\gamma^{2/s}}{\pi^2 16^{1/s}\epsilon^{2/s}(s+1)^{1/s}} \right\rfloor \right).$$

Fix any symmetric ReLU parameter distribution $\mathcal{D}$. Then, the function

$$h(x) := 4\sqrt{2}\epsilon \sin\left(\pi \sum_{i=1}^{\ell} x_i \right)$$

satisfies $\|h\|_{H^s(\mathbb{T}^d)} \leq \gamma$ and

$$\text{MinWidth}_{h,\epsilon,\frac{1}{2},\mathbb{T}^d,\mathcal{D}} \geq \frac{1}{4}\binom{d}{\ell} \geq \exp\left(\Omega\left(\min\left(\frac{\gamma^{2/s}}{\epsilon^{2/s}} \log\left(\frac{d\epsilon^{2/s}}{\gamma^{2/s}} + 2 \right), d \right) \right) \right).$$

Proof. The width bound is immediate from Lemma 2.24 and from the lower-bounds on $\binom{d}{\ell}$ shown in the proof of Theorem 2.23. Note that h can be written as $h = 4\epsilon T_K$ for some K with

$$\|K\|_2 = \sqrt{\ell} \leq \frac{\gamma^{1/s}}{\pi 4^{1/s}\epsilon^{1/s}(s+1)^{1/2s}}.$$

Thus, we conclude that $\|h\|_{H^s(\mathbb{T}^d)} \leq \gamma$ by applying the same chain of inequalities from Theorem 2.27, making use of our lower-bound on γ^2/ϵ^2. □

2.6 Conclusion

This chapter discusses the results of Hsu, Sanford, Servedio, and Vlatakis-Gkaragkounis (2021), which investigates the representational capabilities and limitations of two-layer neural networks with random ReLU features. We show that the expressivity of these models can be characterized by a trade-off between the input dimensionality, the smoothness of the target function, and the width of the network. If either the input dimensionality is small or the target function is smooth, then the width of the random feature model can be polynomially bounded. This asymptotically tight characterization relies on a relationship between the dimensionality of the functional space spanned by the random features and an orthogonal decomposition of the target function.

These results serve as an apt first chapter to the dissertation, as they capture the key themes of the thesis. This chapter produces a tight separation between the expressivity of different neural architectures—in this case, two-layer random feature models and general neural networks of variable width—that depends on some complexity measure of the target function. Later chapters of the thesis will similarly explore separations between shallow and deep models (Chapter 3), between bounded-weight networks and unrestricted networks (Chapter 4), and between different sequential architectures (Chapters 5 and 6). Akin to this result, each of these results constructs a particular target function that certifies the separation between the models and captures the intuitive limitations of the simpler model. In this case, that target function—a single-index sinusoidal function—exhibits an exponential gap in expressivity between the random feature model and unrestricted neural networks.

More personally, this chapter was the first work completed by the author in graduate school, the first theoretical work that the author led, and the first in machine learning theory. While future papers written by the author (and subsequent chapters of this thesis) would study more intricate neural architectures and apply different mathematical tools, the research project presented in this chapter had a significant impact in the author's lens on

machine learning theory and his taste in research problems.

Chapter 3: Powers of depth and the discrete dynamical systems lens

This chapter investigates the core representational question of whether a shallow neural network can approximate a given target function f. Previous works (e.g. Chatziafratis et al., 2019; Chatziafratis, Nagarajan, and Panageas, 2020) examined this basic question on neural network *expressivity* from the lens of dynamical systems and provide novel "depth-vs-width" trade-offs for a large family of functions f. They suggested that such trade-offs are governed by the existence of *periodic* points or *cycles* in f. This chapter, by further deploying dynamical systems concepts, illuminates a more subtle connection between periodicity and expressivity: we prove that periodic points alone lead to suboptimal depth-width trade-offs and we improve upon them by demonstrating that certain "chaotic itineraries" give stronger exponential trade-offs, even in regimes where previous analyses only imply polynomial gaps. Contrary to prior works, these bounds are nearly optimal, tighten as the period increases, and handle strong notions of inapproximability (e.g., constant L_1 error). More broadly, we identify a phase transition to the *chaotic regime* that exactly coincides with an abrupt shift in other notions of function complexity, including VC-dimension and topological entropy.

The research presented in this chapter reflects the work of Sanford and Chatziafratis (2022).

3.1 Introduction

Whether a neural network (NN) succeeds or fails at a given task crucially depends on whether or not its architecture (depth, width, types of activation units, etc.) is suitable for the task at hand. For example, a "size-inflation" phenomenon has occurred in recent

years, in which neural networks tend to have more layers and parameters. Recall that in 2012, AlexNet had 8 layers. In 2015, ResNet won the ImageNet competition with 152 layers (Krizhevsky, Sutskever, and Hinton, 2012b; He et al., 2016). This trend continues, with modern models using billions (or possibly trillions) of parameters (Brown et al., 2020). The empirical success of deep neural networks motivates researchers to ask: What are the theoretical benefits of depth, and what are the depth-vs-width tradeoffs?

This question gives rise to the study of neural network *expressivity*, which characterizes the class of functions that are representable (or approximately representable) by a neural network of certain depth, width, and activation. For instance, Eldan and Shamir (2016) propose a family of "radial" functions in $\mathbb{R}^d$ that are easily expressible with 3-layered feed-forward neural nets of small width but require any approximating 2-layer network to have exponentially (in d) many neurons. In other words, they formally show that depth—even if increased by 1—can be exponentially more valuable than width.

Not surprisingly, understanding the expressivity of NNs was an early question asked in 1969 when Minsky and Papert showed that the Perceptron can only learn linearly separable data and fails on simple XOR functions (Minsky and Papert, 1969). The natural question of which functions can be expressed by two-layer ensembles of Perceptrons (i.e., multilayer feed-forward NN) was addressed later by Cybenko (1989) and Hornik, Stinchcombe, and White (1989) in the so-called *universal approximation* theorem. This states, roughly, that just one hidden layer of standard activation units (e.g., sigmoids, ReLUs, etc.) suffices to approximate any continuous function arbitrarily well. Taken at face value, any continuous function is a two-layer (i.e., containing one hidden layer) network in disguise, and hence, there is no reason to consider deeper networks. However, the width required can grow arbitrarily, and many works in the following decades quantify those depth-vs-width tradeoffs.

Towards this direction, one typically identifies a function with a "measure of complexity" to demonstrate the benefits of depth. For example, the seminal work by Telgarsky (2015) and Telgarsky (2016) relies on the number of oscillations of a narrow family of triangle mappings

on $[0, 1]$ that can be expressed recursively with deep neural networks. Other relevant notions of complexity to the expressivity of NNs include the VC dimension (Warren, 1968; Anthony and Bartlett, 1999; Schmitt, 2000), the number of linear regions (Montufar et al., 2014; Arora et al., 2016) or activation patterns (Hanin and Rolnick, 2019), the dimension of algebraic varieties (Kileel, Trager, and Bruna, 2019), the Fourier spectrum (Barron, 1993; Eldan and Shamir, 2016; Daniely, 2017a; Lee et al., 2017; Bresler and Nagaraj, 2020), fractals (Malach and Shalev-Shwartz, 2019), topological entropy (Bu, Zhang, and Luo, 2020), Lipschitzness (Safran, Eldan, and Shamir, 2019; Hsu et al., 2021), global curvature and trajectory length (Poole et al., 2016; Raghu et al., 2017).

This work builds upon recent papers (Chatziafratis et al., 2019; Chatziafratis, Nagarajan, and Panageas, 2020), which study expressivity from the lens of discrete-time dynamical systems and extend Telgarsky's results beyond triangle (tent) maps. At a high level, their idea is the following: if the initial layers of a neural network output a real-valued function f, then concatenating the *same* layers k times one after the other outputs $f^k := f \circ f \circ \ldots \circ f$, i.e., the composition of f with itself k times. By associating each discrete timestep k to the output of the corresponding layer in the network, one can study expressivity via the underlying properties of f's trajectories. Indeed, if f contains higher-order fixed points, called *periodic* points, then deeper NNs can efficiently approximate f^k, but shallower nets would require exponential width, governed by f's periodicity.

Inspired by these novel connections to discrete dynamical systems, we pose the following natural question:

> *Apart from periodicity, are there other properties of f's trajectories that govern the expressivity tradeoffs?*

We prove that f's periodicity alone is not the end of the story, and we improve on the known depth-width tradeoffs from several perspectives. We exhibit functions of the same period with very different behaviors (see Section 3.1.4) that can be distinguished by the concept of "chaotic itineraries." We analyze these here to achieve nearly optimal tradeoffs for NNs.

Our work highlights why previous works that examine periodicity alone only obtain loose bounds. More specifically:

- We accurately quantify the oscillatory behavior of a large family of functions f. This leads to sharper and nearly optimal lower bounds for the width of NNs that approximate f^k.
- Our lower bounds cover a stronger notion of approximation error, i.e., *constant* separations between NNs, instead of bounds that become small depending heavily on f and its periodicity.
- At a conceptual level, we introduce and study certain chaotic itineraries, which supersede Sharkovsky's theorem (see Section 3.1.2).
- We elucidate connections between periodicity and other function complexity measures like the VC-dimension and the topological entropy (Alsedà, Llibre, and Misiurewicz, 2000). We show that all of these measures undergo a phase transition that coincides with the emergence of the chaotic regime based on periods.

To the best of our knowledge, we are the first to incorporate the notion of chaotic itineraries from discrete dynamical systems into the study of NN expressivity. Before stating and interpreting our results, we provide some basic definitions.

3.1.1 Function Approximation and NNs

This chapter employs three notions of approximation to compare functions $f, g : [0,1] \to [0,1]$.

- $L_1(f,g) = \|f-g\|_1 = \int_0^1 |f(x)-g(x)|\,dx$.
- $L_\infty(f,g) = \|f-g\|_\infty = \sup_{x\in[0,1]} |f(x)-g(x)|$.

- Classification error $\mathcal{R}_{S,t}$: For some sample $S = \{x_1, \ldots, x_n\} \subseteq [0,1]$ and threshold $t \in [0,1]$, let $\mathcal{R}_{S,t}(f,g)$ be the fraction of samples that classifiers derived by thresholding f and g disagree on. That is, $\mathcal{R}_{S,t}(f,g) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{[[f(x_i)]]_t \neq [[g(x_i)]]_t\}$ for $[[x]]_t = \mathbb{1}\{x \geq t\}$.

While L_1 and L_∞ directly measure the ability of a hypothesis to approximate a fixed function, $\mathcal{R}_{S,t}$ measures the difference between functions by framing the question as a classification problem.

For what follows, let $\mathcal{N}(u,\ell)$ be the family of feedforward NNs of depth ℓ and width at most u per layer with ReLU activation functions.[1] All our results also hold for the more general family of semialgebraic activations (Telgarsky, 2016).

3.1.2 Discrete Dynamical Systems

To construct families of functions that yield depth-separation results, we rely on a standard notion of *unimodal* functions from dynamical systems (Metropolis, Stein, and Stein, 1973).

Definition 3.1. Let $f : [0,1] \to [0,1]$ be a continuous and piece-wise differentiable function. We say f is a *unimodal mapping* if:

1. $f(0) = f(1) = 0$, and $f(x) > 0$ for all $x \in (0,1)$.
2. There exists a unique maximizer $x' \in (0,1)$ of f, i.e., f is strictly increasing on the interval $[0,x')$ and strictly decreasing on $(x',1]$.

Our constructions rely on unimodal functions that are concave and also symmetric around $\frac{1}{2}$ (i.e., $f(x) = f(1-x)$ for all $x \in [0,1]$)[2]. We note that the resulting function family is fairly general, already capturing the triangle waves of Telgarsky (2016) and the logistic map used in previous depth-separation results (Schmitt, 2000). Moreover, the study of one-dimensional

[1] Recall $\mathrm{ReLU}(x) = \max(x,0)$.

[2] Throughout, *symmetric* f refers to such functions that are symmetric around $\frac{1}{2}$.

discrete dynamical systems by applied mathematicians explicitly identifies unimodal mappings as important objects of study (Metropolis, Stein, and Stein, 1973; Alsedà, Llibre, and Misiurewicz, 2000).

Recall that a fixed point x^* of f is a point where $f(x^*) = x^*$. A more general notion of higher-order fixed points is that of *periodicity*.

Definition 3.2. For some $p \in \mathbb{N}$, we say that $x_1 \in [0,1]$ is a *point of period p* if $f^p(x_1) = x_1$ and $f^k(x_1) \neq x_1$[3] for all $k \in [p-1]$.[4] The sequence $x_1, f(x_1), \ldots, f^{p-1}(x_1)$ is called a *p-cycle*, and f has *periodicity p* if such a cycle exists.

For example, the identity map $f(x) = x$ has a fixed point (or a point of period 1) at any $x \in [0,1]$. Likewise, $f(x) = 1 - x$ has a fixed point at $x = \frac{1}{2}$ and a point of period 2 at any other choice of x. The triangle map $f(x) = \min(2x, 2(1-x))$ has a fixed point at $x = \frac{2}{3}$; a 2-cycle with $x_1 = \frac{2}{5}$ and $x_2 = \frac{4}{5}$; and a 3-cycle with $x_1 = \frac{2}{9}$, $x_2 = \frac{4}{9}$ and $x_3 = \frac{8}{9}$ (among other cycles of higher periodicity).

Does the existence of some p-cycle in f have any implications about the existence of other cycles? These relations between the periods of f are of fundamental importance to dynamical systems analysis. In particular, Li and Yorke (1975) proved that "period 3 implies chaos" in their celebrated work, which also introduced the term "chaos" to mathematics and later spurred the development of chaos theory. Interestingly, an even more general result was already obtained a decade earlier in Eastern Europe, by Sharkovsky (1964) and Sharkovsky (1965):

Theorem 3.1 (Sharkovsky's Theorem)**.** *Let $f : [0,1] \to [0,1]$ be continuous. If f contains period p and $p \triangleright p'$, then f also contains period p', where the symbol "$\triangleright$" is defined based on the following (decreasing) ordering:*

$$3 \triangleright 5 \triangleright 7 \triangleright \ldots \triangleright 2 \cdot 3 \triangleright 2 \cdot 5 \triangleright 2 \cdot 7 \triangleright \ldots$$

[3] Throughout the chapter, f^k means composition of f with itself k times, or $f^k = \underbrace{f \circ f \circ \cdots \circ f}_{k}$.

[4] As is common, $[m] = \{1, 2, \ldots, m\}$.

$$\ldots \triangleright 2^2 \cdot 3 \triangleright 2^2 \cdot 5 \triangleright 2^2 \cdot 7 \triangleright \ldots \triangleright 2^3 \triangleright 2^2 \triangleright 2 \triangleright 1.$$

This total ordering, called *Sharkovsky's ordering*, sorts all natural numbers by defining $l \triangleright r$ whenever l is to the left of r. The maximum number in this ordering is 3; if f contains period 3, then it also has all other periods, which is also known as *Li-Yorke chaos*. Chatziafratis et al. (2019) and Chatziafratis, Nagarajan, and Panageas (2020) apply this theorem to obtain depth-width tradeoffs based on periods and obtain their most powerful results when $p = 3$. We go beyond Sharkovsky's theorem and prove that tradeoffs are determined by the "itineraries" of periods.[5]

Definition 3.3 (Itineraries)**.** For a p-cycle $x_1, \ldots, x_p$, suppose that $x_{a_1} < \cdots < x_{a_p}$ for $a_j \in [p]$. The *itinerary* of the cycle is the cyclic permutation of $x_{a_1}, \ldots, x_{a_p}$ induced by f, which we represent by the string $\mathbf{a} = a_1 \ldots a_p$. Because cyclic permutations are invariant to rotation, we assume (without loss of generality) that $a_1 = 1$.

Definition 3.4 (Chaotic Itineraries)**.** A p-cycle is a *chaotic itinerary* or an *increasing cycle* if its itinerary is $12 \ldots p$. That is, $x_1 < \cdots < x_p$.

Examining chaotic itineraries circumvents the limitations of prior works based on periods and yields sharper exponential depth-width tradeoffs. For example, there are two distinct itineraries of 4-cycles on unimodal maps: $\mathbf{a} = 1234$ and $\mathbf{a} = 1324$. The former is chaotic, and repeatedly applying the function yields a complex function that is hard to approximate; the latter does not guarantee hardness of approximation, and there exist easily approximable functions f^k derived recursively from mappings f that have the 1324 itinerary. We discuss this case more thoroughly in Section 3.1.4 and explore other examples of chaotic itineraries in Section 3.4.1. Unlike other function complexity properties, the existence of a chaotic itinerary is easily verifiable (see Section 3.4.3).

[5]These are called "patterns" by Alsedà, Llibre, and Misiurewicz (2000).

3.1.3 Our Main Contributions

Our principal goal is to use knowledge about f's itineraries to quantify the number of oscillations of f^k more accurately. The number of oscillations (equivalent to the monotone pieces of sufficient size, formally defined in Definitions 3.5 and 3.6) is the relevant function complexity measure for our depth-width trade-offs, which we relate to other complexity measures. Section 3.2 produces sharper and more robust NN approximability tradeoffs than prior works by leveraging chaotic itineraries and unimodality. Section 3.3 shows how a phase transition in VC-dimension and topological entropy of f occurs exactly when the growth rate of oscillations shifts from polynomial to exponential.

While previous works count oscillations too, they either construct too narrow a range of functions[6], obtain loose depth-width tradeoffs[7], or have unsatisfactory approximation error. [8] In Section 3.2, we improve along these three directions by taking advantage of the unimodality and itineraries of f. The *unimodality* of f allows us to quantify both the number of piecewise monotone pieces of f^k (i.e., oscillations) and the corresponding height between the highest and lowest values of f^k's oscillations. This improvement on the height enables stronger notions of function approximation (e.g., constant error rates with no dependence on f or its period p). *Chaotic itineraries* allow an improved analysis of the number of oscillations in f^k. The existence of these itineraries provide sharper exponential lower bounds on the width of any shallow net g approximating f^k.

We say that our results are *nearly optimal* because we exhibit a broad family of functions f that are inapproximable by shallow networks of width $O(\rho^k)$ for ρ arbitrarily close to 2. Because no unimodal function f can induce more than 2^k oscillations in f^k, we cannot aspire to tighter exponent bases in this setting.[9] On the other hand, none of the bounds

[6] e.g., Telgarsky (2016) analyze only a restricted family of surjective triangle mappings constructed from neural networks with semi-algebraic gates.

[7] e.g., Chatziafratis et al. (2019) and Chatziafratis, Nagarajan, and Panageas (2020) have a suboptimal dependence on p under stringent Lipschitz assumptions.

[8] e.g., Chatziafratis et al. (2019), Chatziafratis, Nagarajan, and Panageas (2020), and Bu, Zhang, and Luo (2020) do not obtain constant error rates.

[9] Our results also transfer to non-unimodal functions via the observation that for bimodal g, there is some

from previous works (except the narrow bounds of Telgarsky (2016)) produce width bounds of more than $\Omega(\phi^k)$, where $\phi \approx 1.618$ is the Golden Ratio. To demonstrate our sharper tradeoffs, we state a special case of our results for the L_∞ error.

Theorem 3.2. *For $p \geq 3$ and $k \in \mathbb{N}$, consider any symmetric, concave unimodal mapping f with an increasing p-cycle and any $g \in \mathcal{N}(u, \ell)$ with width*

$$u \leq \frac{1}{8}\left(\max\left(2 - \frac{4}{2^p}, \phi\right)\right)^{k/\ell}$$

Then, $L_\infty(f^k, g) = \Omega(1)$, independent of f, p, k.

When g is shallow with depth $\ell = O(k^{1-\epsilon})$ (e.g., $\ell = k^{0.99}$), then its width must be exponentially large to approximate f^k closely. This exponential separation in k is sharper than prior works (Chatziafratis et al., 2019; Chatziafratis, Nagarajan, and Panageas, 2020), and quickly becomes sharper (tending to 2) with larger values of p. This is counterintuitive as Sharkovsky's ordering implies that period 3 is the most chaotic and prior works recover a suboptimal rate of at most $\phi \approx 1.618$ (see Table 3.1).

Our approximation error is constant independent of all other parameters f, k, p. Previous results (Chatziafratis et al., 2019; Chatziafratis, Nagarajan, and Panageas, 2020; Bu, Zhang, and Luo, 2020) obtain a gap that depends on f, p and may be arbitrarily small. Moreover, we have required nothing of the Lipschitz constant of f, unlike the strict assumptions on the Lipschitz constant L of f by Chatziafratis, Nagarajan, and Panageas, 2020 (e.g., they require $L = \phi$ for period $p = 3$). Indeed, Propositions 3.12 and 3.13 in the Section 3.2.4 illustrate how their lower bounds break down for large L and how their L_∞ bounds can shrink, becoming arbitrarily weak for certain 3-periodic f.

We also present analogous results for the classification error and L_1 error in Theorems 3.6 and 3.7. Furthermore, Theorems 3.10 and 3.11 offer an improvement on the results of Chatziafratis, Nagarajan, and Panageas (2020) by giving constant-accuracy L_∞ lower bounds

unimodal f such that the number of oscillations of g is at most twice those of f.

without needing a chaotic itinerary.

In addition, Section 3.3 relates our chaotic itineraries to standard notions of function complexity like the VC dimension and the topological entropy (for precise definitions, see Sec. 3.3). The types of periodic itineraries of f give rise to two regimes: the *doubling* regime and the *chaotic* regime. In the former, we have a polynomial number of oscillations, while the latter is characterized by an exponential number of oscillations. Here we show the following correspondence:

Informal Theorem 3.3. *The transition between these two regimes coincides with a sharp transition in the VC-dimension of the iterated mappings f^k for fixed f (from bounded to infinite) and the topological entropy (from zero to positive).*

Our Techniques To quantify the oscillations of f^k, we use its chaotic itineraries to decompose the $[0, 1]$ interval into several subintervals $\{I_j\}_{j=1}^{j=p-1}$. We count the number of times f^k "visits" each I_j, by identifying a suitable matrix A whose spectral radius is a lower bound on the growth rate of oscillations. The associated characteristic polynomial of A is $\lambda^p - 2\lambda^{p-1} + 1$ and has a larger spectral radius than that of prior works for *all* periods. Moreover, the corresponding oscillations of at least one of the subintervals I_j do not shrink in size, giving a bound on the total number of oscillations of a *sufficient* size. This provides a lower bound on the height between the peak and the bottom of these oscillations that later provides *constant* approximation errors for small shallow NNs.

More broadly, our work builds on the efforts to characterize large families of functions that give depth separations and addresses questions raised by Eldan and Shamir (2016), Telgarsky (2016), Poole et al. (2016), and Malach and Shalev-Shwartz (2019) about the properties of hard-to-represent functions. Similar to periods, the concept of chaotic itineraries can serve as a certificate of complexity, which is also easy to verify for unimodal f (see Proposition 3.35).

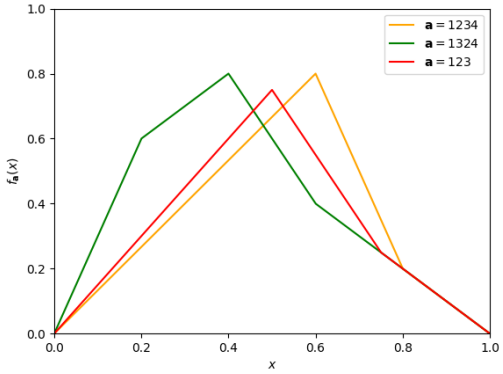


Figure 3.1: Plots of unimodal mappings with different itineraries f_{1234}, f_{1324}, and f_{123}. Despite their similarities, f_{1234} leads to the most oscillations and sharpest depth-width tradeoffs (see Fig. 3.2).

3.1.4 Warm-up examples

This section presents illustrative examples and instantiates our results for some simple cases. These highlight the limitations of exclusively considering periodicity of cycles alone—and not itineraries—when developing accurate oscillation/crossing bounds (see also Def. 3.5, 3.6) and sharp expressivity tradeoffs.

Consider the three unimodal mappings in Figure 3.1, $f_{\mathbf{a}}$ with itineraries

$$\mathbf{a} \in \{1324, 1234, 123\}.$$

Observe that f_{1234} has the cycle $(\frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5})$, f_{1324} has $(\frac{1}{5}, \frac{3}{5}, \frac{2}{5}, \frac{4}{5})$, and f_{123} has $(\frac{1}{4}, \frac{1}{2}, \frac{3}{4})$. Despite their similarities, they give rise to significantly different behaviors in $f_{\mathbf{a}}^k$.

What do prior works based on NN approximation with respect to periods and Sharkovsky's theorem alone tell us? Chatziafratis et al. (2019) and Chatziafratis, Nagarajan, and Panageas (2020) show that the 3-cycle of f_{123} ensures that f^k has $\Omega(\phi^k)$ oscillations, where $\phi \approx 1.618$ is the golden ratio. However, their theorems do not imply anything for f_{1324} and f_{1234}, since 4 is a power of 2, and they require odd periods.

As it turns out, f_{1234} leads to exponential oscillations and f_{1324} leads only to polynomial

oscillations:

- A mapping with a 1324-itinerary is guaranteed to have no cycles besides the 2-cycle and a fixed point (Metropolis, Stein, and Stein, 1973). Sharkovsky's theorem and Chatziafratis et al. (2019) predict this outcome, since 4 is the third-right-most element of the Sharkovsky ordering, and its existence alone promises nothing more. The ordering of itineraries introduced by Metropolis, Stein, and Stein (1973) (see Table 3.3) indicates that the particular 1324-itinerary only implies periods 2 and 1, and confirms this intuition. We categorize this itinerary as part of the *doubling regime* and prove in Theorem 3.18 that any f^k with a *maximal* 1324-itinerary (that is, there is no 8-cycle) cannot exhibit sharp depth-width tradeoffs: for any $\epsilon > 0$, there exists a two-layer ReLU neural network g of width $O(\frac{k^3}{\epsilon})$ such that $L_\infty(f^k_{1324}, g) \leq \epsilon$.

- Beyond Sharkovsky's theorem, a mapping with a 1234-itinerary—even though it is of period 4—is guaranteed to contain a 3-cycle (see Table 3.3). Hence, "itinerary-1234 implies period-3, implies chaos," and f^k_{1234} has at least $\Omega(\phi^k)$ oscillations and is hard to approximate by small shallow NNs. Moreover, Theorem 3.6 and Table 3.1 show that f^k_{1234} actually has $\Omega(\rho^k)$ oscillations for $\rho \approx 1.839 > \phi$. A corollary is that any NN g of depth $\sqrt{k}$ and width $O(1.839^{\sqrt{k}})$ has $L_\infty(f^k_{1234}, g) = \Omega(1)$, which is a stronger separation (*constant* error) than the ones given by Chatziafratis et al. (2019) and Chatziafratis, Nagarajan, and Panageas (2020).

The reverse is not true: Sharkovsky's Theorem guarantees that a period-3 cycle implies the existence of a period-4 cycle. However, the respective 4-cycle is the non-chaotic 1324-itinerary, which was already shown to lead to minimal function complexity.

Furthermore, as p increases, the existence of a chaotic itinerary $12 \ldots p$ on f ensures that f^k has $\Omega(\rho^k)$ oscillations for $\rho \to 2$.[10] Figure 3.2 demonstrates these differences in oscillations

[10]Similarly to Telgarsky (2016), the optimal achievable rate is $\rho \leq 2$ if we start with a unimodal f (e.g., tent map). If one used multimodal functions as a building block (e.g., starting with $f' = f^2$ or $f' = f^3$), we could achieve larger rates (e.g., 4 or 8 respectively).

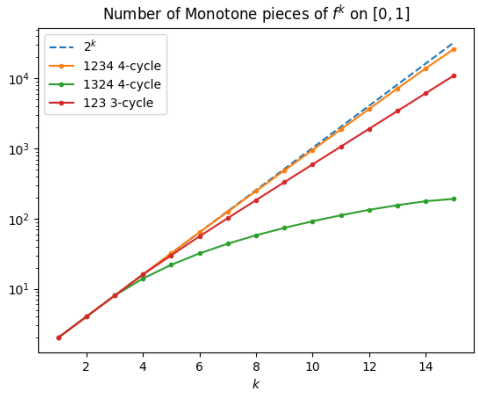


Figure 3.2: The chaotic itinerary f_{1234} has more oscillations than f_{123} even though $3 \triangleright 4$ by Sharkovsky's Theorem. Itineraries f_{1234} and f_{1324} (both of period 4) differ dramatically in oscillation count, showing why periodicity alone fails to capture the optimal tradeoffs. The 1234 itinerary produces a more "complex" function with more monotone pieces than 123, despite the Sharkovsky analysis from Chatziafratis et al., 2019 arguing that 3-cycles are the most powerful when determining iteration counts. Moreover, the number of monotone pieces in the 1234 and 123 itineraries increases exponentially, while that of the 1324 itineraries does not.

(by counting the number of monotone pieces in functions $f_{\mathbf{a}}^k$ with a maximal itinerary-$\mathbf{a}$). As indicated theoretically, the number of oscillations of f_{1324} is polynomially bounded, while the others grow exponentially fast, with f_{1234} being closer to 2^k.

We further visualize two types of chaotic itineraries, Figures 3.3 and 3.4 demonstrate two emblematic cases with piecewise-linear and logistic unimodal mappings respectively where the differences in function complexity of f_{123}, f_{1234}, and f_{1324} are most evident. Both figures provide a function for each $f_{\mathbf{a}}$ that has a maximal itinerary of $\mathbf{a}$. (That is, there is no "higher-ranked" itinerary from Table 3.3 present in $f_{\mathbf{a}}$; all other cycles are induced by the existence of a cycle with itinerary $\mathbf{a}$.)

1. Figures 3.2 and 3.3 visualize the sensitive dependence of the oscillation patterns of compositions of piecewise-linear unimodal mappings on the periodicity of the mappings. The plots visualize a simple case where the elements of the cycles are evenly spaced ($\frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ for f_{123}; $\frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}$ for f_{1234}, f_{1324}). Even though f_{1234} and f_{1324} have the same maximum value, they exhibit substantially different fractal-like patterns, which

produce exponentially more oscillations for f_{1234}.

2. Figures 3.4 and 3.5 instead considers logistic maps of the form $f_{\log,r}(x) = 4rx(1-x)$ for the values of r where itinerary **a** is *super-stable*, or when nearby iterates converge to the cycle exponentially fast. These functions are concave, symmetric, and unimodal. Here, complexity strictly increases with the maximum value of $f_{\log,r}$. Indeed, f_{1234}, f_{123} and f_{1324} ordered by height is the order by which they exhibit most to least chaotic behavior.

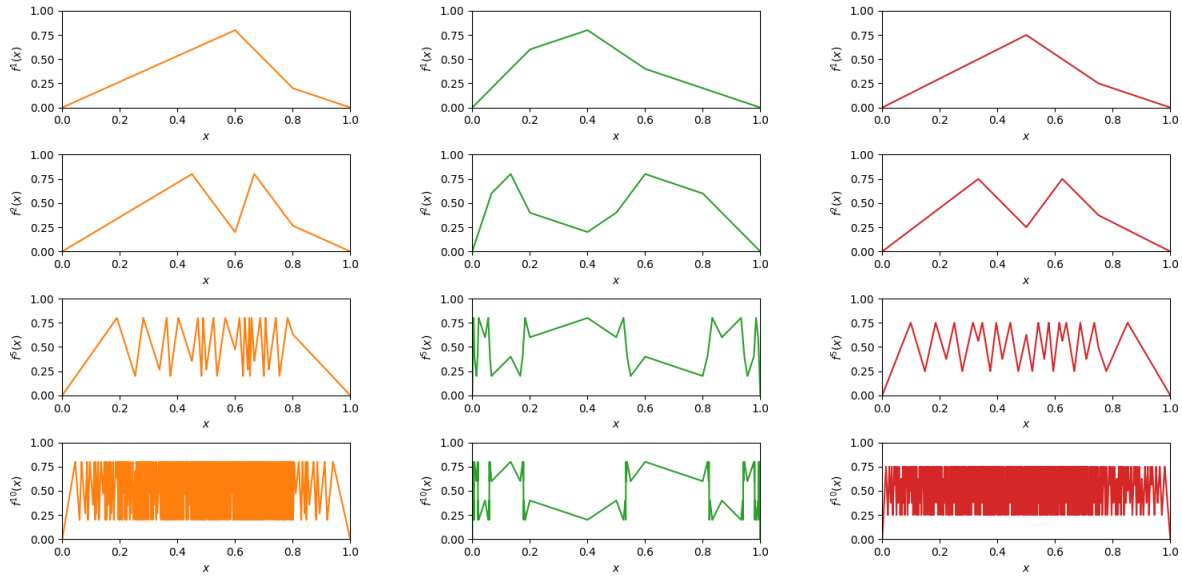


Figure 3.3: A comparison of the function complexity (as measured by the number of monotone pieces) in f^k for unimodal mappings f having cycles with different itineraries. The left shows f, f^2, f^5, and f^{10} for a function with a 1234 4-cycle. The center has a 1324 4-cycle. The bottom has a 123 3-cycle. Figure 3.2 shows how the number of monotone pieces in f^k increases with k for each mapping.

Generally, prior constructions where the oscillation count of f^k increases at a rate faster than ϕ^k were too narrow (including only the triangle map). Because f_{1234} breaks the barrier, we abstract away the details and point to chaotic itineraries as the main source of complexity, leading to sharper depth-width tradeoffs.

While periodicity tells a compelling story about why f_{123}^k is difficult to approximate, it fails to explain why f_{1234}^k is even more complex. The exponential-vs-polynomial gap in the

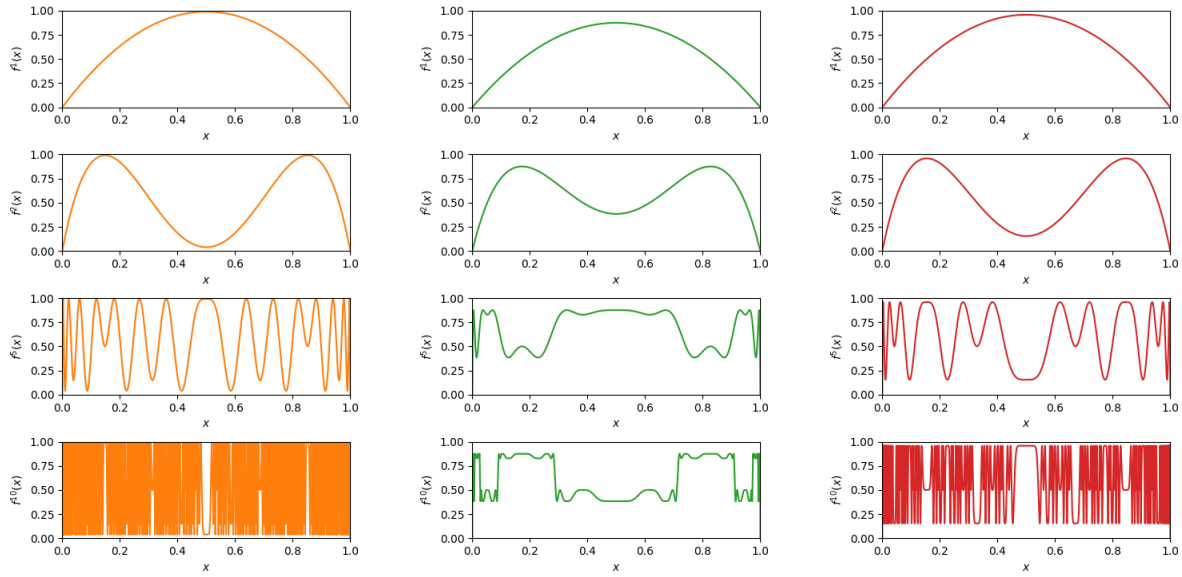


Figure 3.4: Demonstrates the same ideas as Figure 3.3, except instead of using asymmetric and non-concave piecewise functions, we use the scaled logistic map, $f_{\log,r}$. Using Table 1 of Metropolis, Stein, and Stein (1973), we set the parameter r to 3.96, 3.50, and 3.83 respectively to ensure that a super-stable 1234, 1324, and 123 cycle exists.

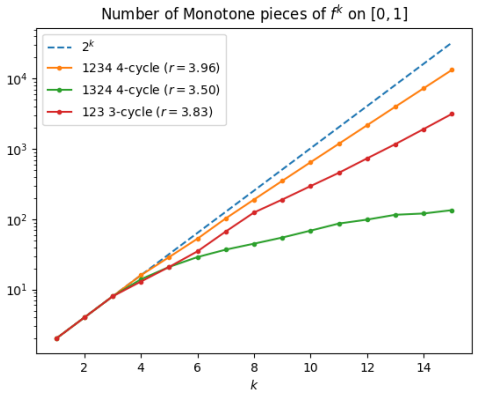


Figure 3.5: Like Figure 3.2, visualizes the differences in the number of monotone pieces for the logistic mappings described in Figure 3.4.

function complexity of f_{1234} and f_{1324} depends solely on the order of the elements of the cycle and distinguishes functions that NNs can easily approximate from those they cannot.

The remainder of the chapter addresses the question introduced here—when does the itinerary tell us much more than the length of the period—in a general context that explores a "hierarchy" of such chaotic itineraries, strengthens a host of NN inapproximability bounds (Sec. 3.2), and reveals tight connections with other complexity notions, like the VC-dimension and topological entropy (Sec. 3.3).

3.2 Depth-width tradeoffs via chaotic itineraries

We give our main hardness results on the inapproximability of functions generated by repeated compositions of f to itself when f has certain cyclic behavior. We define the relevant notions of monotonicity and present basic preliminaries in Section 3.2.1. Section 3.2.2 applies insights about chaotic itineraries to prove constant L_∞ and L_1 lower bounds on the accuracy of approximating f^k when f has an increasing cycle. Section 3.2.3 strengthens previous bounds on the number of oscillations when f has an odd cycle, which is not necessarily increasing.

Subsequent sections contextualize and prove these results. Section 3.2.4 justifies the assumptions that f be symmetrical and concave by showing that shallow networks can approximate f^k when these assumptions are violated. Section 3.2.5 presents Table 3.2, which exhaustively compares our separation results to prior work. Section 3.2.6 contains the proofs of the section's main results.

3.2.1 Preliminaries and notation

To measure the function complexity of f^k, we count the number of times f^k oscillates. We employ two notions of oscillation counts. The first is relatively weak and counts every interval on which f is either increasing or decreasing, regardless of its size.

Definition 3.5. Let $f : [0,1] \to [0,1]$. $M(f)$ represents the *number of monotone pieces* of f. That is, it is the minimum m such that there exists $x_0 = 0 < x_1 < \cdots < x_{m-1} < x_m = 1$ where f is monotone on $[x_{j-1}, x_j]$ for all $j \in [m]$.

The second instead counts the number of times a fixed interval of size $b - a$ is crossed:

Definition 3.6. Let $f : [0,1] \to [0,1]$ and $[a,b] \subseteq [0,1]$. $C_{a,b}(f)$ represents the *number of crossings* of f on the interval $[a,b]$. That is, it is the maximum c such that there exist

$$0 \leq x_1 < x_1' \leq x_2 < x_2' \leq \cdots \leq x_c < x_c' \leq 1$$

where for all $j \in [c]$, $f([x_j, x_j']) \subset [a,b]$ and either $f(x_j) = a$ and $f(x_j') = b$ or vice versa.

Characteristic Polynomials The base of the exponent of our width bounds is shown to equal the largest root of one of two polynomials:

$$P_{\text{inc},p}(\lambda) = \lambda^p - 2\lambda^{p-1} + 1,$$
$$P_{\text{odd},p}(\lambda) = \lambda^p - 2\lambda^{p-2} - 1.$$

Let $\rho_{\text{inc},p}$ and $\rho_{\text{odd},p}$ be the largest roots of $P_{\text{inc},p}$ and $P_{\text{odd},p}$ respectively. Table 3.1 illustrates that as p grows, $\rho_{\text{inc},p}$ increases to 2, while $\rho_{\text{odd},p}$ drops to $\sqrt{2}$. Note that $\rho_{\text{odd},p} \in (\sqrt{2}, \sqrt{2 + 2/2^{p/2}})$ (Alsedà, Llibre, and Misiurewicz, 2000). We bound the growth rate of $\rho_{\text{inc},p}$ with the following:

Fact 3.4. $\rho_{\text{inc},p} \in [\max(2 - \frac{4}{2^p}, \phi), 2)$, *where* $\phi = \frac{1+\sqrt{5}}{2}$ *is the Golden Ratio.*

We prove Fact 3.4 in Section 3.2.6.2.

3.2.2 Inapproximability of Iterated Functions with Increasing Cycles

Our inapproximability results that govern the size of neural network g necessary to adequately approximate f^k when f has an increasing cycle (like Theorem 3.2) rely on a key

Table 3.1: Approximate values of $\rho_{\text{inc},p}$, the lower bound on $\rho_{\text{inc},p}$ in Fact 3.4, and $\rho_{\text{odd},p}$ (for odd p).

p	$\rho_{\text{inc},p}$	Fact 3.4	$\rho_{\text{odd},p}$
3	1.618	1.618	1.618
4	1.839	1.75	n/a
5	1.928	1.875	1.513
6	1.966	1.938	n/a
7	1.984	1.969	1.466
8	1.992	1.984	n/a
9	1.996	1.992	1.441
10	1.999	1.996	n/a

lemma that bounds the number of constant-size oscillations of f^k.

Lemma 3.5 (Oscillation Bound for Increasing Cycles)**.** *Suppose f is a symmetric, concave unimodal mapping with an increasing p-cycle for some $p \geq 3$. Then, there exists $[a, b] \subset [0, 1]$ with $b - a \geq \frac{1}{18}$ such that $C_{a,b}(f^k) \geq \frac{1}{2}\rho_{\text{inc},p}^k$ for all $k \in \mathbb{N}$.*

We prove Lemma 3.5 in Section 3.2.6.1. For an increasing p-cycle $x_1, \ldots, x_p$, we lower-bound $M(f^k)$ (the total number of monotone pieces, regardless of size) by relating the number of times f^k crosses each interval $[x_j, x_{j+1}]$ to the number of crossings of f^{k-1}. Doing so entails analyzing the largest eigenvalues of a transition matrix, which gives rise to the polynomial $P_{\text{inc},p}$. We prove that the intervals crossed must be sufficiently large due to the symmetry, concavity, and unimodality of f.

Remark 3.1. *If one does not wish to assume that f is unimodal, symmetric, or concave, then the proof can be modified to show that $C_{a,b}(f^k) = \Omega(\rho^k)$ for the same ρ, but for $b - a$ dependent on f. These results are similar in flavor to those of Chatziafratis et al. (2019), Chatziafratis, Nagarajan, and Panageas (2020), and Bu, Zhang, and Luo (2020), and they suffer from the same drawback: potentially vacuous approximation bounds when a and b are close. Section 3.2.4 shows natural functions that are either not symmetric or not concave, whose oscillations shrink in size arbitrarily.*

3.2.2.1 L_∞ Approximation and Classification

Our first result is a restatement of Theorem 3.2 that quantifies inapproximability in terms of both L_∞ and classification error, which are comparable to the respective results of Bu, Zhang, and Luo, 2020 and Chatziafratis et al., 2019.

Theorem 3.6. *Suppose f is a symmetric concave unimodal mapping with an increasing p-cycle for some $p \geq 3$. Then, any $k \in \mathbb{N}$ and $g \in \mathcal{N}(u, \ell)$ with $u \leq \frac{1}{8}\rho_{\text{inc},p}^{k/\ell}$ have $\|f^k - g\|_\infty = \Omega(1)$.*

Moreover, there exists S with $|S| = \frac{1}{2}\lfloor \rho_{\text{inc},p}^{k/\ell} \rfloor$ and $t \in (0,1)$ such that $\mathcal{R}_{S,t}(f^k, g) \geq \frac{1}{4}$.

The proof follows from our main Lemma 3.5 above and Theorem 3.15/Corollary 3.16 in Section 3.2.6.3 (two previous inapproximability bounds based on oscillations).

Despite relying on unimodality assumptions and the existence of increasing cycles, Theorem 3.6 obtains much stronger bounds than its previous counterparts:

- The assumption that f has an increasing cycle causes a much larger exponent base for the width bound. Chatziafratis et al. (2019) and Chatziafratis, Nagarajan, and Panageas (2020) only prove that the existence of 3-cycle mandates a width of $\Omega(\phi^{k/\ell})$. We exactly match that bound for $p = 3$, and improve upon it when $p > 3$. As illustrated by Table 3.1, increasing p pushes the base $\rho_{\text{inc},p}$ rapidly to 2, which is the maximum exponent base for the increase of oscillations of any unimodal map. (And the maximal topological entropy of a unimodal map.) This also approximately matches the bases from Bu, Zhang, and Luo, 2020, which scale with the topological entropy of f.

- As illustrated in Section 3.2.4, the inaccuracy of neural networks with respect to the L_∞ approximation in Chatziafratis et al., 2019; Chatziafratis, Nagarajan, and Panageas, 2020; Bu, Zhang, and Luo, 2020 may be arbitrarily small for certain choices of f. Our unimodality assumptions ensure that the oscillations of f^k are large and hence, that the inaccuracy of g is constant.

3.2.2.2 L_1 Approximation

We also strengthen the bound on L_1-inapproximability given by Chatziafratis, Nagarajan, and Panageas (2020) by again introducing a stronger exponent and applying unimodality to yield a constant-accuracy bound.

Theorem 3.7. *Consider any L-Lipschitz $f : [0,1] \to [0,1]$ with an increasing p-cycle for some $p \geq 3$. If $L = \rho_{\text{inc},p}$, then for any $k \in \mathbb{N}$, any $g \in \mathcal{N}(u,\ell)$ with $u \leq \frac{1}{16}\rho_{\text{inc},p}^{k/\ell}$ has $\|f^k - g\|_1 = \Omega(1)$.*

The proof follows again from Lemma 3.5 and is a consequence of Theorem 3.17.

We make Theorem 3.7 more explicit by showing that many tent maps meet the Lipschitzness condition. Let $f_{\text{tent},r} = 2r\min(x, 1-x)$ be the tent map, parameterized by $r \in (0,1)$. Our result improves upon Chatziafratis, Nagarajan, and Panageas, 2020, by obtaining constant approximation error and using the larger $\rho_{\text{inc},p}$ rather than $\rho_{\text{odd},p}$.

Corollary 3.8. *For any $p \geq 3$ and $k \in \mathbb{N}$, any $g \in \mathcal{N}(u,\ell)$ with $u \leq \frac{1}{16}\rho_{\text{inc},p}^{k/\ell}$ has $\|f_{tent,\rho_{\text{inc},p}}^k - g\|_1 = \Omega(1)$.*

We prove Corollary 3.8 in Section 3.2.6.4. The only non-trivial part of the proof involves proving the existence of an increasing p-cycle that causes f^k to have $\Omega(\rho_{\text{inc},p}^k)$ oscillations.

3.2.3 Improved Bounds for Odd Periods

While Theorems 3.6 and 3.7 give stricter bounds on the width of neural networks needed to approximate iterated functions f^k than Chatziafratis et al. (2019) and Chatziafratis, Nagarajan, and Panageas (2020), they also require extra assumptions about the cycles—namely, that the cycles are increasing. However, more powerful inapproximability results with constant error are still possible even without additional assumptions. Specifically, we leverage unimodality to improve the desired inaccuracy to a constant without compromising width.

As before, the results hinge on a key technical lemma that bounds the number of interval crossings.

Lemma 3.9. *For some odd $p \geq 3$, suppose f is a symmetric concave unimodal mapping with an odd p-cycle. Then, there exists $[a, b] \subset [0, 1]$ with $b - a \geq 0.07$ such that $C_{a,b}(f^k) = \rho_{\text{odd},p}^{k-p}$ for any $k \in \mathbb{N}$.*

We prove Lemma 3.9 in Section 3.2.6.5. The challenging part is to find a lower bound on the length of the intervals crossed.

Like before, we provide lower-bounds on approximation up to a constant degree.

Theorem 3.10. *For some odd $p \geq 3$, suppose f is a symmetric, concave unimodal mapping with any p-cycle. Then, any $k \in \mathbb{N}$ and any $g \in \mathcal{N}(u, \ell)$ with $u \leq \frac{1}{8}\rho_{\text{odd},p}^{(k-p)/\ell}$ have $\|f^k - g\|_\infty = \Omega(1)$.*

Moreover, there exists S with $|S| = \frac{1}{2}\lfloor \rho_{\text{odd},p}^k \rfloor$ and $t \in (0, 1)$ such that $\mathcal{R}_{S,t}(f^k, g) \geq \frac{1}{4}$.

The proof is immediate from Lemma 3.9, Theorem 3.15, and Corollary 3.16.

An analogous result for the L_1 error can be obtained as follows.

Theorem 3.11. *Consider any L-Lipschitz $f : [0, 1] \to [0, 1]$ with a p-cycle for some odd $p \geq 3$. If $L = \rho_{\text{odd},p}$, then, any $k \in \mathbb{N}$ and $g \in \mathcal{N}(u, \ell)$ with $u \leq \frac{1}{16}\rho_{\text{odd},p}^{(k-p)/\ell}$ have $\|f^k - g\|_1 = \Omega(1)$.*

We impose strict conditions on the Lipschitz constant because the bounds are vacuous or impossible for functions with other Lipschitz constants. By Lemma 3.1 of Chatziafratis, Nagarajan, and Panageas, 2020, there are no L-Lipschitz interval mappings f whose iterates f^k have $\Omega(\rho_{\text{odd},p})^k$ oscillations when $L < \rho_{\text{odd},p}$. On the other hand, if $L > \rho_{\text{odd},p}$, then our proofs would yield vacuous lower bounds because they depend on $(\frac{\rho_{\text{odd},p}}{L})^k$, which is arbitrarily small for large k. See Section 3.1 of Chatziafratis, Nagarajan, and Panageas, 2020 for a more thorough treatment of this issue.

The proof is immediate from Lemma 3.5 and Theorem 3.17.

3.2.4 Necessity of Symmetry and Concavity Assumptions in Theorems 3.6 and 3.7

We demonstrate the weakness of the bounds promised by Chatziafratis et al. (2019), Chatziafratis, Nagarajan, and Panageas (2020), and Bu, Zhang, and Luo (2020) and argue that our assumptions of symmetry and concavity are necessary in order to avoid such non-vacuous bounds. To do so, we exhibit two families of functions in Propositions 3.12 and 3.13 which contain functions with increasing p-cycles for every p that produce large numbers of oscillations, yet are trivial to approximate because their oscillations can be made arbitrarily small. The functions considered in both cases are unimodal and lack symmetry and concavity respectively.

These expose a fundamental shortcoming of other approaches to the hardness of neural network approximation in the aforementioned works because they all rely on showing that for every mapping f meeting some condition (e.g. odd period, positive topological entropy), there exists some $[a, b] \in [0, 1]$ where $C_{a,b}$ is exponentially large, and hence no poly-size shallow neural network g can obtain $L_\infty(f^k, g) \leq P(b-a)$ for some polynomial P. However, because $[a, b]$ depends on f, their difference can potentially be arbitrarily small. The propositions show that this concern is significant and that $[a, b]$ indeed becomes arbitrarily narrow for simple 3-periodic functions. While Chatziafratis et al., 2019 avoid addressing this issue head-on by focusing on classification error over L_∞ error, their classification lower-bounds rely on misclassification of points whose actual distance can be shrinking (see for example Figure 3.6).

The implications of these propositions contrast with the more robust hardness results we present in Theorems 3.6, 3.7, 3.10, and 3.11, which leverage unimodality, symmetry, and concavity to ensure that the accuracy of approximation can be no better than some constant (independent on f, p) when the neural network g is too small. We show here that those assumptions are necessary by exhibiting functions that satisfy all but one, and become easy to L_∞-approximate with small depth-2 ReLU networks.

Proposition 3.12. *For $p \geq 3$ and for sufficiently small $\epsilon > 0$, there exists a concave unimodal mapping f with a chaotic p-cycle such that for any k, there exists $g \in \mathcal{N}(3,2)$ with*

$$L_\infty(f^k, g) \leq \epsilon.$$

Proof. For all $j \in [p]$, let $x_j = 1 - \frac{p-j+1}{p}\epsilon$. Define f to be a piecewise-linear function with $p+1$ pieces chosen with boundaries that satisfy

$$f(0) = 0, f(x_1) = x_2, f(x_2) = x_3, \ldots, f(x_{p-1}) = x_p, f(x_p) = x_1, f(1) = 0.$$

We visualize f for $p = 3$ in Figure 3.6. f is unimodal because it increases on $[0, x_{p-1}]$ and decreases on $[x_{p-1}, 1]$. It is concave because $f'(x)$ does not increase as x grows, since

$$f'(x) = \begin{cases} \frac{1-\frac{p-1}{p}\epsilon}{1-\epsilon} > 1 & x \in [0, x_1) \\ 1 & x \in (x_1, x_{p-1}) \\ -p+1 & x \in (x_{p-1}, x_p) \\ -\frac{1-\epsilon}{\epsilon} & x \in (x_p, 1], \end{cases}$$

as long as $\frac{1-\epsilon}{\epsilon} > p-1$.

We show inductively that for all k, there exists $a_k < b_k$ such that $f^k(a_k) = f^k(b_k) = 1-\epsilon$, $f^k([a_k, b_k]) \in [1-\epsilon, 1]$, and f^k has exactly one linear piece for each of the intervals $[0, a_k]$ and $[b_k, 1]$.

These are true for the base case $k = 1$ for $a_1 \in (0, x_1)$ and $b_1 = x_p$.

If the claim holds for k, then there is some $a_{k+1} \in (0, a_k)$ and $b_{k+1} \in (b_k, 1)$ such that $f(a_{k+1}) = f(b_{k+1}) = a_k$. Then, $f^{k+1}(a_{k+1}) = f^{k+1}(b_{k+1}) = 1-\epsilon$ and $f^{k+1}([0, a_{k+1}]) = f^{k+1}([b_{k+1}, 1]) = [0, 1-\epsilon]$. For all $x \in [0, a_{k+1}]$, $f^j(x) \leq 1-\epsilon$ for all $j \leq k+1$. Hence, f^{k+1} is linear on $[0, a_{k+1}]$ (and also $[b_{k+1}, 1]$). Because $f([x_1, x_p]) = [x_1, x_p]$, $f^{k+1}([a_{k+1}, b_{k+1}]) \subseteq [x_1, x_p] \subseteq [1-\epsilon, 1]$. The claim then holds for $k+1$.

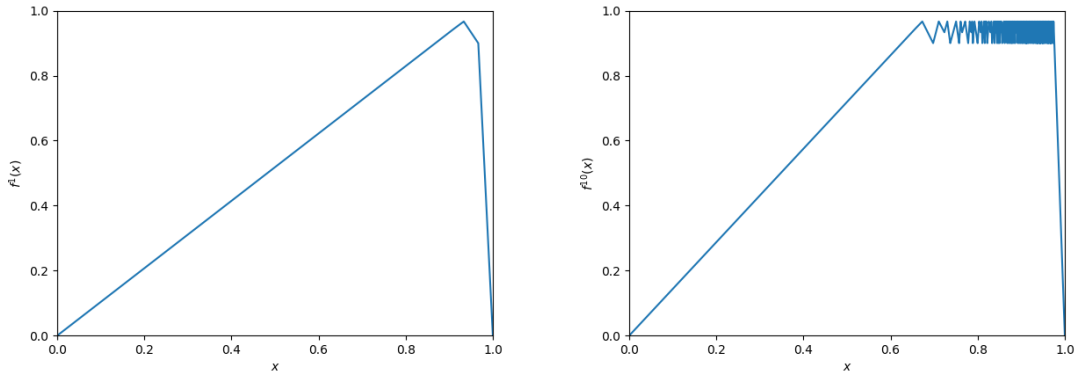


Figure 3.6: Plots the asymmetric function with a p-cycle referenced in Proposition 3.12 for $p = 3$ and $\epsilon = 0.1$. While f oscillates frequently, f can be trivially 0.1-approximated by three ReLUs. As $\epsilon \to 0$, the L_∞ approximation hardness guarantees implied by Chatziafratis et al., 2019 become vacuous because the oscillations, even though they are exponentially many, they shrink in size.

Thus, the piecewise linear mapping g with boundaries $g(0) = 0$, $g(a_k) = 1 - \epsilon$, $g(b_k) = 1 - \epsilon$, and $g(1) = 0$ is an ϵ-approximation of f. Because g has three pieces and contains the origin, it can be exactly represented by a linear combination of four ReLUs, and hence as a depth-2 neural network of width 3. □

Proposition 3.13. *For $p \geq 3$ and for sufficiently small $\epsilon > 0$, there exists a symmetric unimodal mapping f with a chaotic p-cycle such that for any k, there exists $g \in \mathcal{N}(3, 2)$ with*

$$L_\infty(f^k, g) \leq \epsilon.$$

Proof. Let $x_j = \frac{1}{2} - \frac{p-1-j}{2(p-1)}\epsilon$ for all $j \in [p-1]$ and $x_p = \frac{1}{2} + \frac{\epsilon}{2}$. Let f be a piecewise-linear function with boundaries

$$f(0) = 0,\ f\left(\frac{1}{2} - \frac{\epsilon}{2}\right) = \frac{1}{2} - \frac{p-2}{p-1} \cdot \frac{\epsilon}{2},\ f\left(\frac{1}{2} - \frac{\epsilon}{2(p-1)}\right) = \frac{1}{2},\ f\left(\frac{1}{2}\right) = \frac{1}{2} + \frac{\epsilon}{2},$$
$$f\left(\frac{1}{2} + \frac{\epsilon}{2(p-1)}\right) = \frac{1}{2},\ f\left(\frac{1}{2} + \frac{\epsilon}{2}\right) = \frac{1}{2} - \frac{p-2}{p-1} \cdot \frac{\epsilon}{2},\ f(1) = 0.$$

We visualize f for $p = 3$ in Figure 3.7. Note that f is symmetric and unimodal and has

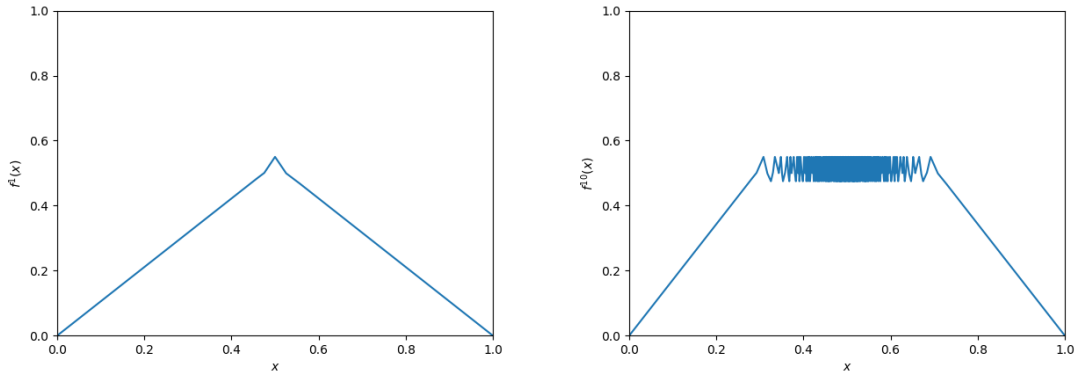


Figure 3.7: Another example of a function with a 3-cycle that can be ϵ-approximated for arbitrarily small ϵ. (Here, $\epsilon = 0.1$.) This function corresponds to the one in Proposition 3.13 and the Chatziafratis et al., 2019 bounds are again vacuous for small ϵ. Unlike Figure 3.6, this function is symmetric, but not concave.

an increasing p-cycle $x_1 < \cdots < x_p$. It is *not* concave because $f'(x) = 1$ for $x \in [x_1, x_{p-2}]$ and $f'(x) = 2(p-1)$ for $x \in [x_{p-2}, x_{p-1}]$.

Using a very similar argument to argument from the proof of Proposition 3.12, for all k, there exists $a_k < b_k$ such that f^k is linear on $[0, a_k]$ and $[b_k, 1]$ and $f^k([a_k, b_k]) \in [\frac{1}{2} - \epsilon, \frac{1}{2} + \epsilon]$. As before, there exists a piecewise linear function with three pieces (which can be thought of as a depth-2 neural network of width 3) that ϵ-approximates f. □

3.2.5 Comparison with prior works

Given the large number of results presented in this paper and the many axes of comparison one can draw between these results and their predecessors in Telgarsky, 2016; Chatziafratis et al., 2019; Chatziafratis, Nagarajan, and Panageas, 2020, we provide Table 3.2 to illuminate these comparisons. It reinforces our key contributions, namely that (1) the presence of increasing cycles makes a function more difficult to approximate than a 3-cycle alone; (2) requiring that f satisfy unimodality constraints gives lower-bounds to constant accuracy that cannot be made vacuous by adversarial choices of f; and (3) the key distinction between "hard" and "easy" functions is the existence of non-primary power-of-two cycles.

We provide context for each column to clarify what its cells mean and how to compare

	Condition	Appx.	Uni?	Conc?	Sym?	$L \leq \rho$?	Acc.	Exp.	Hard?	Source
1	**Maximal PO2**	L_∞	**Yes**	**No**	**Yes**	**No**	$\Omega(1)$	**Any**	**No**	**Thm 3.18**
2	$h_{\text{top}}(f) \geq \rho$	L_∞	No	No	No	No	$\epsilon(f)$	ρ	Yes	BZL Thm 16
3	**Non-primary**	**Cls.**	**No**	**No**	**No**	**No**	$\frac{1}{4}$	$(1, \phi]$	**Yes**	**CNPW Thm 1.6, Remark 3.2**
4	**Non-primary**	L_∞	**No**	**No**	**No**	**No**	$\epsilon(f)$	$(1, \phi]$	**Yes**	**CNPW Thm 1.6, Remark 3.2, BZL Thm 16**
5	Non-PO2	Cls.	No	No	No	No	$\frac{1}{4}$	$(1, \phi]$	Yes	CNPW Thm 1.6
6	Non-PO2	L_∞	No	No	No	No	$\epsilon(f)$	$(1, \phi]$	Yes	CNPW Thm 1.6, BZL Thm 16
7	Odd cycle	Cls.	No	No	No	No	$\frac{1}{4}$	$(\sqrt{2}, \phi]$	Yes	CNP Thm 1.1
8	Odd cycle	L_∞	No	No	No	No	$\epsilon(f)$	$(\sqrt{2}, \phi]$	Yes	CNP Thm 1.1, BZL Thm 16
9	**Odd cycle**	L_∞	**Yes**	**Yes**	**Yes**	**No**	$\Omega(1)$	$(\sqrt{1}, \phi]$	**Yes**	**Thm 3.10**
10	Odd cycle	L_1	No	No	No	Yes	$\epsilon(f)$	$(\sqrt{2}, \phi]$	Yes	CNP Thm 1.2
11	$f_{\text{tent},\rho_p/2}$	L_1	Implied	Implied	Implied	Implied	$\Omega(1)$	$(\sqrt{2}, \phi]$	Yes	CNP Lemma 3.6
12	**Odd cycle**	L_1	**Yes**	**Yes**	**Yes**	**Yes**	$\Omega(1)$	$(\sqrt{2}, \phi]$	**Yes**	**Thm 3.11**
13	**Inc. Cycle**	**Cls.**	**No**	**No**	**No**	**No**	$\frac{1}{4}$	$[\phi, 2)$	**Yes**	**Thm 3.6, Remark 3.1**
14	**Inc. Cycle**	L_∞	**No**	**No**	**No**	**No**	$\epsilon(f)$	$[\phi, 2)$	**Yes**	**Thm 3.6, Remark 3.1**
15	**Inc. Cycle**	L_∞	**Yes**	**Yes**	**No**	**No**	$\Omega(1)$	$[\phi, 2)$	**No**	**Prop 3.12**
16	**Inc. Cycle**	L_∞	**Yes**	**No**	**Yes**	**No**	$\Omega(1)$	$[\phi, 2)$	**No**	**Prop 3.13**
17	**Inc. Cycle**	L_∞	**Yes**	**Yes**	**Yes**	**No**	$\Omega(1)$	$[\phi, 2)$	**Yes**	**Thm 3.6**
18	**Inc. Cycle**	L_1	**No**	**No**	**No**	**Yes**	$\epsilon(f)$	$[\phi, 2)$	**Yes**	**Thm 3.7, CNP Thm 1.2**
19	**Inc. Cycle**	L_1	**Yes**	**Yes**	**Yes**	**Yes**	$\Omega(1)$	$[\phi, 2)$	**Yes**	**Thm 3.7**
20	$f_{\mathbf{tent},\rho_p/2}$	L_1	**Implied**	**Implied**	**Implied**	**Implied**	$\Omega(1)$	$[\phi, 2)$	**Yes**	**Cor 3.8**
21	$f_{\text{tent},1}$	L_1	Implied	Implied	Implied	Implied	$\Omega(1)$	2	Yes	Telgarsky

Table 3.2: Compares the conditions and limitations of the theoretical results presented in this paper and its predecessors. New results are bolded.

their values.

- **Condition** specifies what must be true of the complexity of f in order for the relevant bounds to occur. All but the latter two conditions describe a very broad array of functions, while the last two focus only on a restricted subset of tent mappings.
  - "Maximal PO2" means that the maximal cycle of f is a primary[11] p-cycle where p is a power of two. This means that f lies in the doubling regime described in Theorem 3.18.

[11] See Section 3.4.2.

- "$h_{\text{top}}(f) \geq \rho$" considers any f with a lower-bound on its topological entropy for some $\rho > 1$. Notably, all conditions other than "Maximal PO2" satisfy this for some ρ.
- "Non-primary" means that any non-primary cycle exists in f. That is, if f is known to have a non-primary power-of-two cycle, then the results apply.
- "Non-PO2" refers to any f that has a p-cycle where p is not a power of two.
- "Odd cycle" includes any f that has a p-cycle where p is odd.
- "Inc. cycle" means that f has an increasing p-cycle for some p, i.e. a cycle with itinerary $12 \ldots p$.
- $f_{\text{tent},\rho_p/2}$ refers to families of tent maps scaled by ρ_p solving the polynomials from Chatziafratis, Nagarajan, and Panageas, 2020 Lemma 3.6 (for odd periods) and Corollary 3.8 (for increasing cycles).
- The last row refers exclusively to the tent map of height 1 and slope 2.

- **Appx.** refers to how difference between neural network g and iterated map f^k is measured. The options are L_1, L_∞, and classification error. It's easier to show that g can L_1-approximate f^k than it is to show that g can L_∞-approximate f; conversely, it's most impressive to show lower bound results with respect to the L_1 error than it is for the L_∞ error.

  Chatziafratis et al., 2019; Chatziafratis, Nagarajan, and Panageas, 2020 consider classification error, Bu, Zhang, and Luo, 2020 focus on L_∞ approximation, and Chatziafratis, Nagarajan, and Panageas, 2020 also consider L_1 approximation. We routinely translate classification errors to L_∞ errors using Corollary 3.16, which draws on Theorem 16 of Bu, Zhang, and Luo, 2020.

- **Uni?**, **Conc?**, and **Sym?** indicate whether f satisfying the respective precondition is necessary for the results to hold. We mark "Implied" if the value of **Condition** already ensures that the property is satisfied and the requirement need not be enforced.

- "$L \leq \rho$?" is "Yes" if the results only hold if f is chosen with a Lipschitz constant less than the rate of growth of its oscillations. This is a very restrictive condition met by very few functions (including no logistic maps with cycles).

- **Acc.** specifies the desired accuracy of the hardness result. "$\Omega(1)$" means that there exists some constant ϵ such that for any choice of f in the category, any neural network g will be unable to approximate f up to accuracy ϵ. "$\epsilon(f)$" means that the degree of approximation may depend on the chosen function f (and the period p) that belongs to the category; these bounds may be vacuous by an adversarial choice of f. As a result, hardness results with "$\Omega(1)$" are more impressive.

- **Exp.** refers to the base of the exponent of the lower-bound on the width necessary to approximate f^k using a shallow network g. Larger values indicate stronger bounds.

- **Hard?** denotes whether every f satisfying the conditions to the left cannot be approximated up to the specified accuracy by any neural network g.

- **Source** denotes where to find the result. Some of the less interesting results are not given their own theorems and rather are immediate implications of several theorems across this body of literature. For the sake of space, we use "CNPW" to refer to (Chatziafratis et al., 2019); "CNP" for (Chatziafratis et al., 2019); "BZL" for (Bu, Zhang, and Luo, 2020); and "Telgarsky" for (Telgarsky, 2016).

3.2.6 Proofs of depth-width trade-offs

3.2.6.1 Proof of Lemma 3.5

We restate and prove the lemma. This is the main technical lemma that we use to obtain the sharper depth-width tradeoffs and the improved notion of *constant* approximation.

Lemma 3.5 (Oscillation Bound for Increasing Cycles)**.** *Suppose f is a symmetric, concave unimodal mapping with an increasing p-cycle for some $p \geq 3$. Then, there exists $[a, b] \subset [0, 1]$*

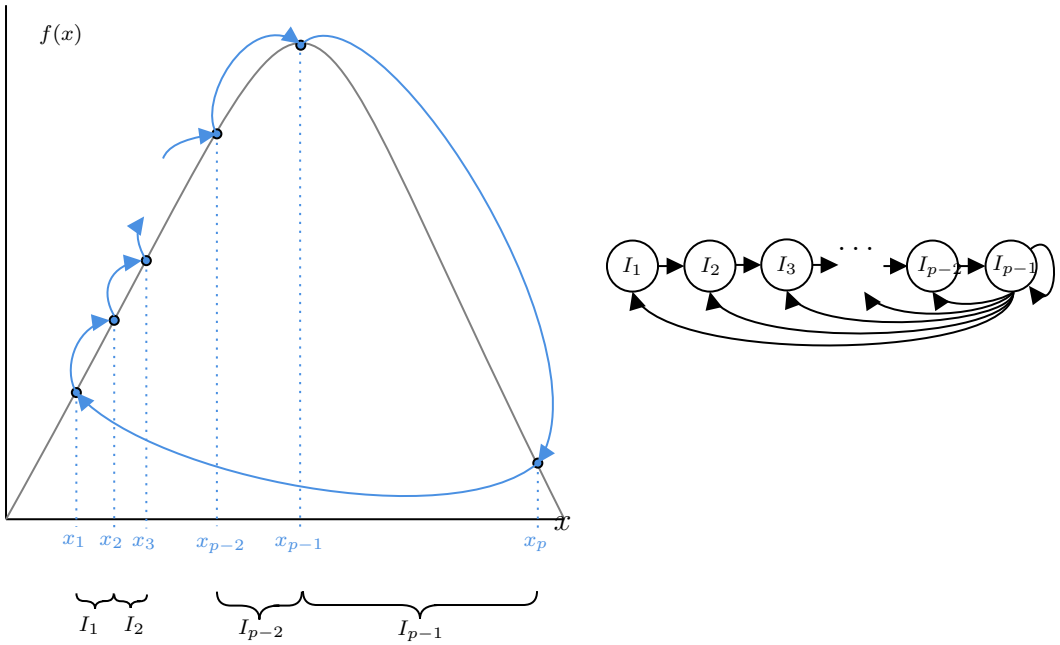


Figure 3.8: Visualizes the intervals $I_1, \ldots, I_{p-1}$ defined in the proof of Lemma 3.5 and which intervals f maps to one another when f has an increasing p-cycle.

with $b - a \geq \frac{1}{18}$ *such that* $C_{a,b}(f^k) \geq \frac{1}{2}\rho^k_{\text{inc},p}$ *for all* $k \in \mathbb{N}$.

Proof. We first lower-bound the total number of oscillations that will appear an increasing p-cycle is present. Later, we show that the size of the oscillations is large as well.

Because we have an increasing cycle of itinerary $12 \ldots p$, we assume (wlog) that the cycle is $(x_1, \ldots, x_p)$ with $x_1 < x_2 < \cdots < x_p$. Define intervals $I_j := [x_j, x_{j+1}]$ for $j \in \{1, \ldots, p-1\}$. Because f is continuous, we conclude that $I_{j+1} \subset f(I_j)$ for all $j < p$ and $I_j \subset f(I_{p-1})$ for all j. Figure 3.8 visualizes these relationships.

Using the methods of Chatziafratis et al. (2019), we define $y^{(k)} \in \mathbb{N}^{p-1}$ such that $y^{(k)}_j$ is a lower bound on the number of times f^k passes through interval I_j, or

$$C_{x_j, x_{j+1}}(f^k) \geq y^{(k)}_j.$$

We can then encode the interval relationships above with $y^{(k+1)} = A_p y^{(k)}$ where $y^{(0)}$ is a

vector of all ones and and $A_p \in \{0,1\}^{(p-1)\times(p-1)}$ with $(A_p)_{i,j} = \mathbb{1}\{j = p-1 \text{ or } i = j+1\}$. We get the following adjacency matrix for the intervals, capturing the mapping relationships (under f) between them:

$$A_p = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 1 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{pmatrix}.$$

We find the characteristic polynomial of A_p and lower-bound $y^{(k+1)}$ with the spectral radius of A_p. We show by induction on $p \geq 3$ that

$$\det(A_p - \lambda I) = (-1)^{p-1}\left(\lambda^{p-1} - \sum_{i=0}^{p-2} \lambda^i\right).$$

For the base case $p = 3$, we have:

$$\det(A_3 - \lambda I) = \begin{vmatrix} -\lambda & 1 \\ 1 & 1-\lambda \end{vmatrix} = \lambda^2 - \lambda - 1,$$

which satisfies the desired form.

Now, we show the inductive step by expanding the determinant of $A_p - \lambda I$.

$$\det(A_p - \lambda I) = -\lambda \begin{vmatrix} -\lambda & 0 & \cdots & 0 & 1 \\ 1 & -\lambda & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -\lambda & 1 \\ 0 & 0 & \cdots & 1 & 1-\lambda \end{vmatrix} - \begin{vmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & -\lambda & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -\lambda & 1 \\ 0 & 0 & \cdots & 1 & 1-\lambda \end{vmatrix}.$$

The left determinant exactly equals $\det(A_{p-1} - \lambda I)$, which we can expand using the inductive hypothesis. The second equals $(-1)^{p-2}$, because $p-2$ row swaps (which are elementary row operations) can be used to move the first row to the bottom and make the matrix upper-triangular with diagonals of one. We conclude the inductive step below.

$$\begin{aligned} \det(A_p - \lambda I) &= -\lambda \det(A_{p-1} - \lambda I) - (-1)^{p-2} \\ &= -\lambda(-1)^{p-2}\left(\lambda^{p-2} - \sum_{i=0}^{p-3} \lambda^i\right) + (-1)^{p-1} = (-1)^{p-1}\left(\lambda^{p-1} - \sum_{i=0}^{p-2} \lambda^i\right). \end{aligned}$$

We find the eigenvalues of A_p by finding the roots of the polynomial

$$P(x) = \lambda^{p-1} - \sum_{i=0}^{p-2} \lambda^i = 0.$$

Observe that there must be a root greater than 1 because $P(1) = 2-p < 0$ and $P(2) = 1 > 0$. Equivalently, if $\lambda \neq 1$,

$$P(x) = \lambda^{p-1} - \frac{1-\lambda^{p-1}}{1-\lambda} = \frac{\lambda^p - 2\lambda^{p-1} + 1}{\lambda - 1} = 0.$$

Hence, finding the largest root of P is equivalent to finding the largest root of $\lambda^p - 2\lambda^{p-1} + 1$, which is $\rho_{\text{inc},p}$ by definition.

This implies that the spectral radius of A_p, $\text{sp}(A_p) = \rho_{\text{inc},p} > 1$, and hence, we also have $\text{sp}(A_p^k) = \text{sp}(A_p)^k = \rho_{\text{inc},p}^k$. Since all the elements in A_p and in A_p^k are non-negative, then the infinity norm of A_p^k is by definition the maximum among its row sums. Since the last column of A_p is the all 1's vector, the largest row sum in A_p^k appears at its last row:

$$||A_p^k||_\infty = \sum_{j=1}^{p-1} (A_p^k)_{p-1,j}$$

We can now use the fact that the infinity norm of a matrix is larger than its spectral norm:

$$||A_p^k||_\infty \geq \rho_{\text{inc},p}^k$$

We conclude that there exists at least one interval I_{j^*} (e.g., the interval I_{p-1}) which is crossed at least $\rho_{\text{inc},p}^k$ times by f^k, so $C_{x_{j^*},x_{j^*+1}}(f^k) \geq \rho_{\text{inc},p}^k$.

Thus, for some a', b' we get $C_{a',b'}(f^k) \geq \rho_{\text{inc},p}^k$. But can we find a', b' with large difference $b' - a'$?

Now, we show that the intervals traversed are sufficiently large, in order to lower-bound $C_{a,b}(f^k)$ with $b - a \geq \frac{1}{18}$. By Lemma 3.14, there exists some j with $x_{j+1} - x_j \geq \frac{1}{18}$. It suffices to show that f^k traverses the interval I_j sufficiently many times.

From earlier in the proof, there exists some j^* such that f crosses I_{j^*} at least $N := \rho_{\text{inc},p}^k$ times. We conclude by showing that every other interval is traversed at least half as often as this most popular interval, which suggests that $C_{x_j,x_{j+1}}(f) \geq \frac{N}{2}$.

For $A \in \mathbb{R}^{(p-1)\times(p-1)}$ as defined earlier in the section and for $y^{(k)} := A^k \vec{1}$, we argue inductively that the elements of $y^{(k)}$ are non-decreasing and that $y_{p-1}^{(k)} \leq 2y_1^{(k)}$. For the base case, this is trivially true for $k = 0$.

Suppose it holds for k. By construction, we have $y_1^{(k+1)} = y_{p-1}^{(k)}$ and $y_j^{(k+1)} = y_{j-1}^{(k)} + y_{p-1}^{(k)}$ for all $j > 1$. By the inductive hypotheses,

$$y_1^{(k+1)} \leq y_2^{(k+1)} \leq \cdots \leq y_{p-1}^{(k+1)} \leq 2y_1^{(k+1)}.$$

Therefore, f^k crosses interval I_j at least $\frac{N}{2}$ times, and I_j has width at least $\frac{1}{18}$. The claim immediately follows. $\square$

Lemma 3.14. *For some $p \geq 3$, consider a symmetric concave unimodal function f with an increasing p-cycle of $x_1 < \cdots < x_p$. Then, there exists $j \in [p-1]$ such that $x_{j+1} - x_j \geq \frac{1}{18}$.*

Proof. By the continuity of f, note that $[x_1, x_p] \subset f^3([x_{p-3}, x_{p-2}])$. There then exists some $y_1 \in [x_{p-3}, x_{p-2}]$ such that $f^3(y_1) = y_1$, $y_2 := f(y_1) \in [x_{p-2}, x_{p-1}]$, and $y_3 := f(y_2) \in [x_{p-1}, x_p]$. Thus, if f has a maximal p-cycle, then f also has a 3-cycle corresponding to $x_{p-3} < y_1 < y_2 < y_3 < x_p$.

We now show that $y_3 - y_1$ must be sufficiently large by concavity. For f to be concave, the following inequality must hold:

$$\frac{f(y_1) - f(0)}{y_1 - 0} \geq \frac{f(y_2) - f(y_1)}{y_2 - y_1} > 0 > \frac{f(y_3) - f(y_2)}{y_3 - y_2} \geq \frac{f(1) - f(y_3)}{1 - y_3},$$

or equivalently,

$$\frac{y_2}{y_1} \geq \frac{y_3 - y_2}{y_2 - y_1} > 0 > -\frac{y_3 - x_1}{y_3 - y_2} \geq -\frac{y_1}{1 - y_3}.$$

In addition, note that $y_1 < \frac{1}{2}$ and $y_3 > \frac{1}{2}$. If the former were false, then $f(y_2) \leq f(y_1)$ (by unimodality), which contradicts $y_3 > y_2$. If the latter were false, then $f(y_3) > f(y_2)$, which contradicts $y_1 < y_3$.

We consider two cases and show that either way, the interval must have width at least $\frac{1}{6}$.

- If $y_2 - y_1 \leq \frac{2}{5}(y_3 - y_1)$, then $\frac{y_3 - y_2}{y_2 - y_1} \geq \frac{3}{2}$, which mandates that $y_1 \leq \frac{2y_2}{3}$ to ensure concavity. Thus,
$$y_3 - y_1 \geq y_3 - \frac{2y_2}{3} \geq \frac{y_3}{3} \geq \frac{1}{6}.$$

- If $y_2 - y_1 \geq \frac{2}{5}(y_3 - y_1)$, then $\frac{y_3 - y_1}{y_3 - y_2} \geq \frac{5}{3}$, and thus $y_1 \geq \frac{5}{3}(1 - y_3)$ and $y_3 \geq 1 - \frac{3y_1}{5}$. Then,
$$y_3 - y_1 \geq 1 - \frac{3y_1}{5} - y_1 = 1 - \frac{8y_1}{5} \geq \frac{1}{5}.$$

Thus, we must have

$$\max\{x_{p-2} - x_{p-3}, x_{p-2} - x_{p-1}, x_p - x_{p-1}\} \geq \frac{1}{18}. \qquad \square$$

3.2.6.2 Proof of Fact 3.4

Fact 3.4. *$\rho_{\text{inc},p} \in [\max(2 - \frac{4}{2^p}, \phi), 2)$, where $\phi = \frac{1+\sqrt{5}}{2}$ is the Golden Ratio.*

Proof. Let $P_{\text{inc},p}(\lambda) = \lambda^p - 2\lambda^{p-1} + 1$.

First, observe that $\rho_{\text{inc},p} < 2$, because $P_{\text{inc},p}(\lambda) > 0$ whenever $\lambda \geq 2$. We lower-bound $\rho_{\text{inc},p}$ by finding some λ for each p such that $P_{\text{inc},p}(\lambda) \leq 0$ or equivalently $\lambda^{p-1}(2-\lambda) \geq 1$ for all $p \geq 3$, which bounds $\rho_{\text{inc},p}$ by the Intermediate Value Theorem.

Consider $\lambda = 2 - \frac{4}{2^p}$. Then,

$$\begin{aligned}
\lambda^{p-1}(2-\lambda) &= \left(2 - \frac{4}{2^p}\right)^{p-1} \cdot \frac{4}{2^p} = 2\left(1 - \frac{2}{2^p}\right)^{p-1} \\
&\geq 2\left(1 - \frac{2(p-1)}{2^p}\right) = 2 - 2 \cdot \frac{p-1}{2^{p-1}} \\
&\geq 2 - 2 \cdot \frac{1}{2} = 1.
\end{aligned}$$

$\square$

3.2.6.3 Prior Results about Hardness of Approximating Oscillatory Functions

We rely on prior results from Chatziafratis et al. (2019) and Chatziafratis, Nagarajan, and Panageas (2020) to show that an iterated function f^k is inapproximable by neural networks. These results hold if f^k has sufficiently many crossings of some interval. We apply these results later with improved bounds on both the number and the size of crossings.

Chatziafratis et al. (2019) show that the classification error of f^k can be bounded if there are enough oscillations.

Theorem 3.15 (Chatziafratis et al. (2019), Section 4)**.** *Consider any continuous $f : [0,1] \to [0,1]$ and any $g \in \mathcal{N}(u, \ell)$. Suppose there exists $a < b$ such that $C_{a,b}(f) = \Omega(\rho^t)$ and suppose*

$u \leq \frac{1}{8}\rho^{k/\ell}$. *Then, for* $t = \frac{a+b}{2}$, *there exists* S *with* $|S| = \frac{1}{2}\left\lfloor \rho^k \right\rfloor$ *samples such that*

$$\mathcal{R}_{S,t}(f^k, g) \geq \frac{1}{2} - \frac{(2u)^\ell}{n}.$$

We adapt that claim to lower-bound the L_∞ approximation of f^k by g.

Corollary 3.16. *Consider any continuous* $f : [0,1] \to [0,1]$ *and any* $g \in \mathcal{N}(u, \ell)$. *Suppose there exists* $a < b$ *such that* $C_{a,b}(f) = \Omega(\rho^t)$ *and suppose* $u \leq \frac{1}{8}\rho^{k/\ell}$. *Then,*

$$\left\| f^k - g \right\|_\infty \geq \frac{b-a}{2}.$$

Proof. By Theorem 3.15, there exists some $x \in [0,1]$ such that (wlog) $f^k(x) \leq a$ and $g(x) \geq \frac{a+b}{2}$. The conclusion for the L_∞ error is immediate by definition. □

Chatziafratis, Nagarajan, and Panageas (2020) give a lower bound on the ability of a neural network g to L_1-approximate f^k, provided a correspondence between the Lipschitz constant of f and the rate of oscillations ρ.

Theorem 3.17 (Chatziafratis, Nagarajan, and Panageas, 2020 Theorem 3.2)**.** *Consider any L-Lipschitz* $f : [0,1] \to [0,1]$ *and any* $g \in \mathcal{N}(u, \ell)$. *Suppose there exists* $a < b$ *such that* $C_{a,b}(f) = \Omega(\rho^t)$. *If* $L \leq \rho$ *and* $u \leq \frac{1}{16}\rho^{k/\ell}$, *then*

$$\left\| f^k - g \right\|_1 = \Omega((b-a)^2).$$

The Lipschitzness assumption is extremely strict, especially because they show in their Lemma 3.1 that $L \geq \rho$ whenever f has a period of odd length.

3.2.6.4 Proof of Corollary 3.8

Corollary 3.8. *For any* $p \geq 3$ *and* $k \in \mathbb{N}$, *any* $g \in \mathcal{N}(u, \ell)$ *with* $u \leq \frac{1}{16}\rho_{\text{inc},p}^{k/\ell}$ *has* $\|f^k_{tent,\rho_{\text{inc},p}} - g\|_1 = \Omega(1)$.

Proof. This theorem follows from Theorem 3.7 and Lemma 3.5. Because $f_{\text{tent},\rho_p/2}$ is ρ_p-Lipschitz, it remains only to prove that there exists an increasing p-cycle. We show that

$$\frac{1}{2}, f\left(\frac{1}{2}\right), \ldots, f^{p-1}\left(\frac{1}{2}\right)$$

is such a cycle.

By definition of the tent map, $f(\frac{1}{2}) = \frac{\rho_{\text{inc},p}}{2}$ and $f^2(\frac{1}{2}) = \rho_{\text{inc},p}(1 - \frac{\rho_{\text{inc},p}}{2})$. If we assume for now that $f^j(\frac{1}{2}) \leq \frac{1}{2}$ for all $j \in \{2, \ldots, p-1\}$, then

$$f^p\left(\frac{1}{2}\right) = \rho_{\text{inc},p}^{p-1}\left(1 - \frac{\rho_{\text{inc},p}}{2}\right) = -\frac{1}{2}\left(\rho_{\text{inc},p}^{p} - 2\rho_{\text{inc},p}^{p-1} + 1\right) + \frac{1}{2} = 0 + \frac{1}{2}.$$

Because $f^p(\frac{1}{2}) = \frac{1}{2}$ and we assumed that $f^{j+1}(\frac{1}{2}) = \rho_{\text{inc},p} f^j(\frac{1}{2})$ for $j \geq 2$ and $\rho > 1$, it must be the case that $f^j(\frac{1}{2}) \leq \frac{1}{2}$ for all $j \in \{2, \ldots, p-1\}$.

Lemma 3.5 thus implies that f^k has $\Omega(\rho_{\text{inc},p}^k)$ crossings, which enables us to complete the proof by invoking Theorem 3.7, since the Lipschitzness condition is met. □

3.2.6.5 Proof of Lemma 3.9

Lemma 3.9. *For some odd $p \geq 3$, suppose f is a symmetric concave unimodal mapping with an odd p-cycle. Then, there exists $[a, b] \subset [0, 1]$ with $b - a \geq 0.07$ such that $C_{a,b}(f^k) = \rho_{\text{odd},p}^{k-p}$ for any $k \in \mathbb{N}$.*

Proof. By Theorems 2.94 and 3.11.1 of Alsedà, Llibre, and Misiurewicz (2000), there exists a p-cycle of the form

$$x_p < x_{p-2} < \cdots < x_3 < x_1 < x_2 < x_4 < \cdots < x_{p-1},$$

which is known as a *Stefan cycle*. The analysis of Section 3.2 of Chatziafratis, Nagarajan, and Panageas (2020) shows that $C_{[x_1,x_2]}(f^k) \geq \rho_{\text{odd},p}^k$. Their exploitation of the relationships between intervals is visualized in Figure 3.9. By the continuity of f, applying f an additional

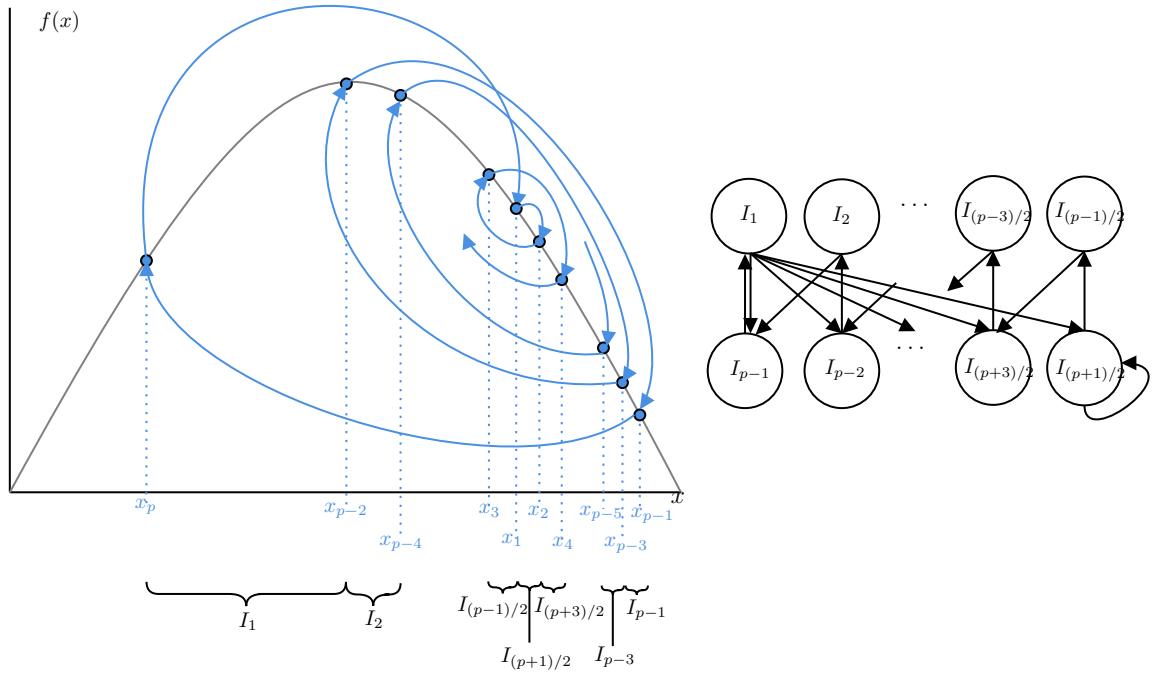


Figure 3.9: Gives an example of a Stefan p-cycle (which is relied upon in Lemma 3.9 and demonstrates the interval relationships). Analogous to Figure 3.8.

$p-1$ times gives $C_{[x_p,x_1]}(f^{k+p-1}) \geq \rho^k_{\text{odd},p}$. Because $[x_{p-2}, x_1] \subset [x_p, x_1]$, applying f one more time gives $C_{[x_2,x_{p-1}]}(f^{k+p}) \geq \rho^k_{\text{odd},p}$.

Hence, by redefining k, we have

$$\max\{C_{[x_1,x_2]}(f^k), C_{[x_2,x_{p-1}]}(f^k), C_{[x_p,x_1]}(f^k)\} \geq \rho^{k-p}_{\text{odd},p}.$$

Since $[x_p, x_{p-1}]$ is the disjoint union of $[x_1, x_2]$, $[x_2, x_{p-1}]$, and $[x_p, x_1]$, there exists $[a, b] \subset [x_p, x_{p-1}]$ with $b - a \geq \frac{1}{3}(x_{p-1} - x_p)$ such that $C_{[a,b]}(f^k) \geq \rho^{k-p}_{\text{odd},p}$.

The problem reduces to placing a lower bound on $x_{p-1} - x_p$. To do so, we derive contradictions on the concavity and symmetry of f. Let $r = f(\frac{1}{2}) \in (x_p, 1)$ be the the largest outcome of f, and let

$$a = \sup_{x,x' \in [1-r,r]} \left| \frac{f(x) - f(x')}{x - x'} \right|$$

be the maximum absolute slope of f on $[1-r, r]$. a must be finite by the concavity and continuity of f, and if f is differentiable, $a = f'(1-r) = -f'(r)$. Thus, f is a-Lipschitz on that interval.

Because $f([x_p, x_{p-1}]) \subseteq [x_p, r] \subset [1-r, r]$, it follows that $|f^2(x) - f^2(x')| \leq a^2 |x - x'|$. Thus, $x_2 - x_p \leq a^2(x_{p-2} - x_p)$ and $x_2 - x_p \leq x_4 - x_p \leq a^2(x_2 - x_{p-2})$. Averaging the two together, we have $x_2 - x_p \leq \frac{a^2}{2}(x_2 - x_p)$, which means $a \geq \sqrt{2}$.

To satisfy concavity, the following must be true:

$$\frac{f(1-r) - f(0)}{1 - r - 0} = \frac{f(r)}{1-r} \geq a \geq \sqrt{2}.$$

We rearrange the inequality and apply properties of monotonicity to lower-bound r away from $\frac{1}{2}$:

$$r \geq 1 - \frac{f(r)}{\sqrt{2}} \geq 1 - \frac{f(x_{p-1})}{\sqrt{2}} = 1 - \frac{x_p}{\sqrt{2}} > 1 - \frac{1}{2\sqrt{2}}.$$

It also must be the case for any $x \in [\frac{1}{2}, 1]$, that:

$$\left| \frac{f(x) - f\left(\frac{1}{2}\right)}{x - \frac{1}{2}} \right| \leq 2.$$

Otherwise, the concavity of f would force $f(\frac{1}{2}) > 1$.

We finally assemble the pieces to lower-bound the gap between x_{p-1} and x_p:

$$\begin{aligned} x_{p-1} - x_p \geq x_{p-1} - \frac{1}{2} \geq -\frac{1}{2}\left(f(x_{p-1}) - f\left(\frac{1}{2}\right)\right) = \frac{r}{2} - \frac{x_p}{2} \\ > \frac{1}{2} - \frac{1}{4\sqrt{2}} - \frac{1}{4} = \frac{1}{4} - \frac{1}{4\sqrt{2}} > 0.07. \end{aligned}$$

□

3.3 Periods, phase transitions, and function complexity

We formalize the correspondence between different notions of function complexity in dynamical systems and learning theory: neural network approximation, oscillation count, cycle itinerary, topological entropy, and VC-dimension. We make Informal Theorem 3.3 rigorous

by presenting two regimes into which unimodal mappings can be classified—the *doubling regime* and the *chaotic regime*—and show that all of these measurements of complexity hinge on which regime a function belongs to.[12]

The following two theorems split most of the space of unimodal mappings into one of two regimes and show that the doubling regime (so called because all cycles have power-of-two lengths and their itineraries are not chaotic) is intrinsically simpler from an approximation theoretic and a function complexity standpoint than the chaotic regime (where there exist chaotic itineraries). The pair of theorems combined known facts about approximation and topological entropy with new ideas about VC dimension. They support the claim that the phase transition that separates mappings with chaotic itineraries from those without is meaningful, because it also separates functions f^k that cannot be tractably approximated from those that can and separates highly expressive iterates f^k from those that cannot express complex data patterns.

Some components of the claims regarding the topological entropy are the immediate consequences of other results; however, we include them to give a complete picture of the gap between the two regimes. We believe the upper bound on monotone pieces of f^k in the doubling regime and both VC-dimension bounds below to be novel.

We define VC-dimension and introduce topological entropy in Section 3.3.2, along with the proofs of both theorems. For the VC-dimension, we consider the hypothesis class $\mathcal{H}f, t := \{[[f^k]]_t : k \in \mathbb{N}\}$, which corresponds to the class of iterated fixed maps.

Theorem 3.18. *[Doubling Regime] Suppose f is a symmetric unimodal mapping whose maximal cycle is a primary cycle of length $p = 2^q$. That is, there exists a p-cycle but no $2p$-cycles (and thus, no cycles with lengths non-powers-of-two). Then, the following are true:*

1. *For any $k \in \mathbb{N}$, $M(f^k) = O((4k)^{q+1})$.*

[12]These two regimes correspond to different settings of the parameters r in the bifurcation diagram of Figure 3.16. The doubling regime is the left-hand-side, where the stable periods routinely split in two before the first chaos is encountered. The chaotic regime is to the right-hand-side, which is characterized by chaos punctuated by intermittent stability.

2. *For any $k \in \mathbb{N}$, there exists $g \in \mathcal{N}(u, 2)$ with $u = O((4k)^{q+1}/\epsilon)$ such that $\left\|g - f^k\right\|_\infty \leq \epsilon$. Moreover, if $f = f_{tent,r}$, then there exists $g \in \mathcal{N}(u, 2)$ with $u = O((4k)^{q+1})$ and $g = f^k$.*

3. *$h_{\text{top}}(f) = 0$.*

4. *For any $t \in (0, 1)$, $VC(\mathcal{H}f, t) \leq 18p^2$.*

The proof of Theorem 3.18 relies on a recursive characterization of f^k whenever f has a maximum cycle length of 2^q. To prove the first claim, we use this recursive structure to bound the number of monotone regions by relating the number of monotone regions of f^k to some g^{2k}, where g has a maximum cycle length no more than 2^{q-1}. The second and third claims are implications of the first. The fourth claim relies on a different recursive argument which shows that the family of iterated maps f^k for fixed f are unable to shatter certain subsets of points.

Theorem 3.19. *[Chaotic Regime] Suppose f is a unimodal mapping that has a p-cycle where p is not a power-of-two. Then, the following are true:*

1. *There exists some $\rho \in (1, 2]$ such that for any $k \in \mathbb{N}$, $M(f^k) = \Omega(\rho^k)$.*

2. *For any $k \in \mathbb{N}$ and any $g \in \mathcal{N}(u, \ell)$ with $\ell \leq k$ and $u \leq \frac{1}{8}\rho^{k/\ell}$, there exist samples S with $|S| = \frac{1}{2}\left\lfloor \rho^k \right\rfloor$ such that $\mathcal{R}_{S,1/2}(f^k, g) \geq \frac{1}{4}$.*

3. *$h_{\text{top}}(f) \geq \rho > 0$.*

4. *There exists a $t \in (0, 1)$ such that $VC(\mathcal{H}f, t) = \infty$.*

Remark 3.2. *As discussed in Section 3.4, any non-primary cycle implies the existence of a cycle whose length is not a power of two. Thus, these results also apply if there exists any non-primary power-of-two cycle, such as the 1234-itinerary 4-cycle.*

The first three claims are implications of the proofs from previous sections of paper and previous works. The fourth claim relies on applying Sharkovsky's theorem to prove the

existence of an infinitely large number of cycles with coprime lengths. Then, by considering a set of points each contained in a cycle of different coprime lengths, we show that a large number of iterates k is sufficient to "shatter" the points by realizing every possible labeling.

3.3.1 Preliminaries and notation

Before reintroducing and proving the theorems about the doubling and chaotic regime, we introduce topological entropy and define VC-dimension.

3.3.1.1 Topological Entropy

Topological entropy is a well-known measure of function complexity in dynamical systems that measures the "bumpiness" of a mapping. Like we do with chaotic itineraries, Bu, Zhang, and Luo (2020) draw analogies between the neural network approximability of f^k and the topological entropy of f. We do not give a rigorous definition of topological entropy, but we include a well known result connecting topological entropy to the number of monotone pieces (not constant-sized crossings), which is stated as Lemma 3 of the aforementioned work.

Lemma 3.20. *[Misiurewicz and Szlenk, 1980; Young, 1981] If $f : [0, 1] \to [0, 1]$ is continuous and piece-wise monotone, then the topological entropy of f satisfies the following:*

$$h_{\text{top}}(f) = \lim_{k\to\infty} \frac{1}{k} \log M(f^k).$$

3.3.1.2 VC-Dimension

We capture the complexity of the mappings produced by repeated application of f, by measuring the capability of a family of iterates to fit arbitrarily-labeled samples with the VC-dimension. For some threshold parameter $t \in (0, 1)$, we first define a hypothesis class that we use to cast this family of iterated functions as Boolean-valued.

Definition 3.7. For some unimodal $f : [0,1] \to [0,1]$ and threshold $t \in (0,1)$, let

$$\mathcal{H}_{f,t} := \{[[f^k]]_t : k \in \mathbb{N}\}$$

be the Boolean-valued hypothesis class of classifiers of composed functions.

The following is the standard definition of the VC-dimension:

Definition 3.8 (Vapnik and Chervonenkis, 1968)**.** For some hypothesis class $\mathcal{H}$ containing functions $[0,1] \to \{0,1\}$, we say that $\mathcal{H}$ *shatters* samples $x_1, \ldots, x_d \in [0,1]$ if for every labeling of the samples $\sigma_1, \ldots, \sigma_d \in \{0,1\}$, there exists some $h \in \mathcal{H}$ such that $h(x_i) = \sigma_i$ for all $i \in [d]$. The *VC-dimension* of $\mathcal{H}$, $\mathrm{VC}(\mathcal{H})$ is the maximum d such that there exists $x_1, \ldots, x_d \in [0,1]$ that $\mathcal{H}$ shatters.

$\mathrm{VC}(\mathcal{H}_{f,t})$ will be a useful measurement of complexity of the mapping f, which as we show is tighly connected with the notion of periodicity and oscillations. Notably, this is a measurement of the complexity of iterated maps and is *not* a typical formulation of VC-dimension for neural networks, since those typically would consider a fixed depth and a fixed width, but variable values for the weights, rather than fixed f and variable k.

3.3.2 Proofs of Theorems 3.18 and 3.19

Theorem 3.18. *[Doubling Regime] Suppose f is a symmetric unimodal mapping whose maximal cycle is a primary cycle of length $p = 2^q$. That is, there exists a p-cycle but no $2p$-cycles (and thus, no cycles with lengths non-powers-of-two). Then, the following are true:*

1. *For any $k \in \mathbb{N}$, $M(f^k) = O((4k)^{q+1})$.*
2. *For any $k \in \mathbb{N}$, there exists $g \in \mathcal{N}(u,2)$ with $u = O((4k)^{q+1}/\epsilon)$ such that $\left\|g - f^k\right\|_\infty \leq \epsilon$. Moreover, if $f = f_{tent,r}$, then there exists $g \in \mathcal{N}(u,2)$ with $u = O((4k)^{q+1})$ and $g = f^k$.*
3. *$h_{\mathrm{top}}(f) = 0$.*

4. *For any* $t \in (0,1)$, $VC(\mathcal{H}f, t) \leq 18p^2$.

Proof. Claim 1 follows from a somewhat involved argument in Section 3.3.2.1 that uses an inductive argument to compare the behavior of a mapping with a maximal p-cycle to one with a maximal $\frac{p}{2}$-cycle. By categorizing intervals of $[0,1]$ based on how f^k behaves on that interval, we analyze how f^{k+1} in turn behaves, which leads to a bound on the monotone pieces $M(f^k)$.

Claim 2 is a simple consequence of Claim 1, by using the fact that a ReLU network can piecewise approximate each monotone piece of f^k. This argument appears in Section 3.3.2.2.

Claim 3 follows easily from Claim 1 and Lemma 3.20. We note that this derivation about the topological entropy and the periodicity of f is a known fact in the dynamical systems community.

Claim 4 relies on another recursive argument that frames VC-dimension in terms of the possible trajectories of $f^k(x)$ for fixed x and changing k. We characterize these trajectories by making use of Regular Expressions and by bounding the corresponding VC dimension in Section 3.3.2.3. □

Theorem 3.19. *[Chaotic Regime] Suppose f is a unimodal mapping that has a p-cycle where p is not a power-of-two. Then, the following are true:*

1. *There exists some $\rho \in (1,2]$ such that for any $k \in \mathbb{N}$, $M(f^k) = \Omega(\rho^k)$.*
2. *For any $k \in \mathbb{N}$ and any $g \in \mathcal{N}(u,\ell)$ with $\ell \leq k$ and $u \leq \frac{1}{8}\rho^{k/\ell}$, there exist samples S with $|S| = \frac{1}{2}\left\lfloor \rho^k \right\rfloor$ such that $\mathcal{R}_{S,1/2}(f^k, g) \geq \frac{1}{4}$.*
3. *$h_{\text{top}}(f) \geq \rho > 0$.*
4. *There exists a $t \in (0,1)$ such that $VC(\mathcal{H}f, t) = \infty$.*

Proof. Claims 1 and 2 are immediate implications Theorems 1.5 and 1.6 of Chatziafratis et al. (2019). Claim 3 follows by applying Lemma 3.20 to Claim 1 (again this derivation about the topological entropy is basic in the literature on dynamical systems).

The most interesting part of the theorem is the last claim. We prove Claim 4 in Section 3.3.2.4 by showing that the VC-dimension of the class is at least d for all $d \in \mathbb{N}$. The argument relies on the existence of an infinite number of cycles of other lengths, as guaranteed by Sharkovsky's Theorem. □

3.3.2.1 Proof of Theorem 3.18, Claim 1

We restate Claim 1 of the theorem as the following proposition and prove it.

Proposition 3.21 (Claim 1 of Theorem 3.18)**.** *Suppose f is a symmetric unimodal mapping whose maximal cycle is of length $p = 2^q$. Then, for any $k \in \mathbb{N}$, $M(f^k) = O((4k)^{q+1})$.*

In order to bound the number of times f oscillates based on its power-of-two periods, we categorize f by its cyclic behavior and the bound the number of local maxima and minima f has based on its characterization.

Definition 3.9 (Category)**.** For $q \geq 0$ and $z \in \{0, 1\}$, let $\mathcal{F}_{q,z}$ contain the set of all symmetric unimodal functions f such that (1) f has a 2^q-cycle, (2) f does not have a 2^{q+1}-cycle, and (3) $[[f^{2^q}(\frac{1}{2})]]_{1/2} = z$.

We abuse notation to let $M(\mathcal{F}^k_{q,z}) = \max_{f \in \mathcal{F}_{q,z}} M(f^k)$. Thus, for f given in the theorem statement with a 2^q-cycle, but not a 2^{q+1}-cycle, our final bound is obtained by

$$M(f^m) \leq \max\{M(\mathcal{F}^m_{q,0}), M(\mathcal{F}^m_{q,1})\}.$$

We let $M(f, a, b)$ represent the number of monotone pieces of f on the sub-interval $[a, b] \subset [0, 1]$.

We build a large-scale inductive argument by first bounding base cases $M(\mathcal{F}^k_{0,0})$ and $M(\mathcal{F}^k_{0,1})$. Then, we relate $M(\mathcal{F}^k_{q,z})$ to $M(\mathcal{F}^k_{q-1,1-z})$ to get the desired outcome.

Before beginning the proof, we state a slight refinement of the part of the theorem, which takes into account the newly-introduced categories, from which the claim follows.

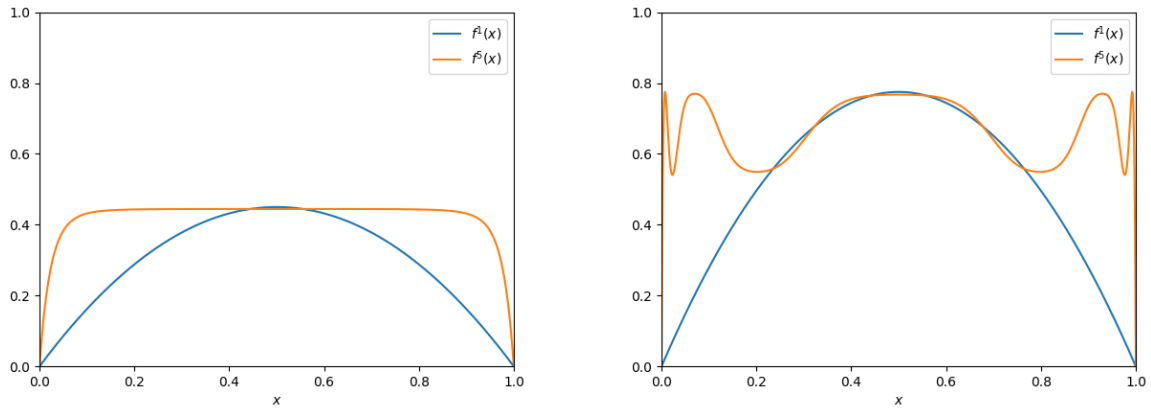


Figure 3.10: The base case results of Proposition 3.22 demonstrate the number of oscillations of f^k increases when f moves from $\mathcal{F}_{0,0}$ to $\mathcal{F}_{0,1}$. The plots show f and f^5 for $f \in \mathcal{F}_{0,0}$ ($f = f_{\log,0.45}$) on the left and $f \in \mathcal{F}_{0,1}$ ($f = f_{\log,0.775}$) on the right.

Proposition 3.22. *For any $k \in \mathbb{N}$, $q \geq 0$, and $z \in \{0, 1\}$,*

$$M(\mathcal{F}^k_{q,z}) \leq \begin{cases} 2(3q)^k & q \text{ is even, } z = 0, \text{ or } q \text{ is odd, } z = 1 \\ 2(3q)^{k+1} & q \text{ is even, } z = 1, \text{ or } q \text{ is odd, } z = 0. \end{cases}$$

Thus, proving Proposition 3.22 completes the proof of Proposition 3.21. The remainder of the section proves Proposition 3.22.

Proof of Proposition 3.22 if $q = 1$ We show that $M(\mathcal{F}^k_{0,0}) = 2$ and $M(\mathcal{F}^k_{0,1}) = 2k$.

For f_r as defined above, we characterize the number of oscillations that are added by increasing r past $\frac{1}{2}$, where super-stability of a fixed point exists. Figure 3.10 illustrates those results.

To analyze the oscillation patterns of f^k, we define several "building blocks," which represent disjoint pieces of f^k. That is, the interval $[0, 1]$ can be partitioned into several sub-intervals, each of which has f^k follow certain simple behavior that we categorize. We argue that any iterate can be decomposed into those pieces and then show how applying f to f^k modifies the pieces in order to analyze f^{k+1}. Here are the function pieces that we analyze, which map interval $[a, b] \subseteq [0, 1]$ to $[0, 1]$:

Definition 3.10. For any $f : [0,1] \to [0,1]$ and for any $[a,b] \subseteq [0,1]$, f is referred to on interval $[a,b]$ as:

- a **increasing crossing piece** Ic if f is strictly increasing on $[a,b]$ and has $f(a) = 0$, $f(b) > \frac{1}{2}$, and $f'(b) > 0$;
- a **decreasing crossing piece** Dc if f is strictly decreasing on $[a,b]$ and has $f(a) > \frac{1}{2}$, $f(b) = 0$, and $f(a) < 0$;
- a **up peak** Up if there exists some $c \in (a,b)$ that maximizes f on $[a,b]$, f is strictly increasing on $[a,c)$, f is strictly decreasing on $(c,b]$, and $f(x) > \frac{1}{2}$ for all $x \in [a,b]$;
- a **up valley** Uv if there exists some $c \in (a,b)$ that minimizes f on $[a,b]$, f is strictly decreasing on $[a,c)$, f is strictly increasing on $(c,b]$, and $f(x) > \frac{1}{2}$ for all $x \in [a,b]$; and
- a **down peak** Dp if there exists some $c \in (a,b)$ that maximizes f on $[a,b]$, f is strictly increasing on $[a,c)$, f is strictly decreasing on $(c,b]$, and $f(x) \leq \frac{1}{2}$ for all $x \in [a,b]$.

If there exists a sequence of intervals $J_1, \ldots, J_m$ such that f is piece η_i on J_i, then we represented f with the string $\eta_1 \ldots \eta_m$.

We specify an invariant for each part of the theorem, such that proving the invariant is sufficient to prove the proposition:

1. If $f \in \mathcal{F}_{0,0}$, then f^k is a down peak on $[0,1]$ for all k, and f^k has two monotone pieces.
2. If $f \in \mathcal{F}_{0,1}$, f is represented by $\mathsf{Ic}(\mathsf{UpUv})^{k-1}\mathsf{UpDc}$. That is, $[0,1]$ can be partitioned into $2k+1$ subsequent intervals $J_1, \ldots J_{2k+1}$ such that f^k is an increasing crossing piece on J_1, a decreasing crossing piece on J_{2k+1} (if $k \neq 0$), an up peak on J_{2j} for $j \in \{1, \ldots, k\}$, and a up valley on J_{2j+1} for $j \in \{1, \ldots, k-1\}$. Hence, f^k has k distinct maxima and $2k$ monotone pieces. Figure 3.11 illustrates this invariant.

Base Case:

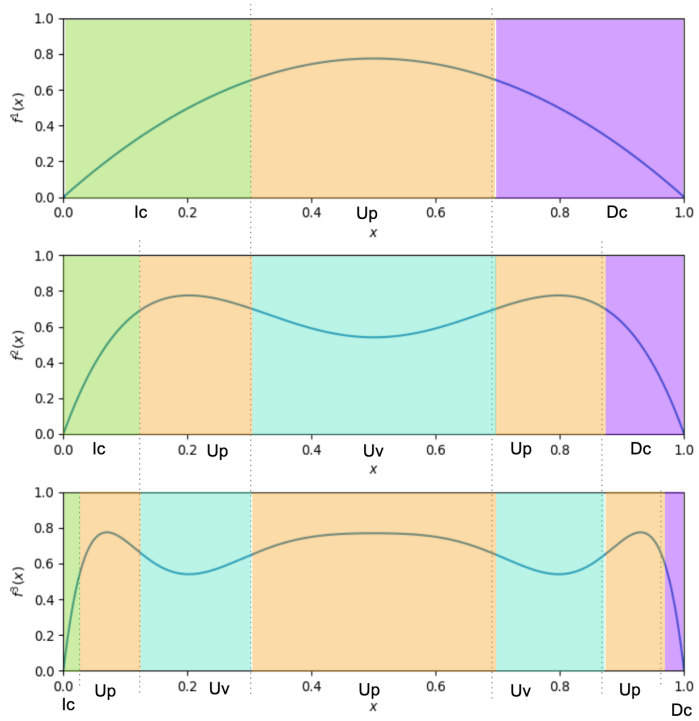


Figure 3.11: For $f \in \mathcal{F}_{0,1}$ ($f = f_{\log,0.775}$), visualizes the decomposition of f, f^2, and f^3 into IcUpDc, IcUpUvUpDc, and Ic$(\text{UpUv})^2$UpDc respectively.

1. For $f \in \mathcal{F}_{0,0}$, $f^1 = f$ is trivially a down peak on $[0, 1]$ by the definition of $\mathcal{F}_{0,0}$, since $\frac{1}{2}$ maximizes f.

2. For $f \in \mathcal{F}_{0,1}$, f can be represented by IcUpDc. That is, $[0, 1]$ can be decomposed into intervals I_1, I_2, and I_3, on which f_r is an increasing crossing piece, an up peak, and a decreasing crossing piece respectively.

Inductive Step:

We examine what happens to each function piece when f is applied to it. We can use the following analysis, along with the inductive hypothesis to show that f^{k+1} can be decomposed as we expect it to be.

1. Examining the **down peak** proves first invariant for the case when $f \in \mathcal{F}_{0,0}$. Because f strictly increases on $[0, \frac{1}{2}]$ and because $f([0, 1]) \subseteq [0, \frac{1}{2}]$ if f^k is a down peak, $f \circ f^k$ also supports a down peak on $[0, 1]$.

Because we inductively assume that f^k is a low peak on $[0,1]$, it then follows that f^{k+1} is also a down peak on $[0,1]$.

2. We first prove a claim, which implies that f has no down peaks for $f \in \mathcal{F}_{0,1}$. Let $x_{\max} = f(\frac{1}{2})$,

   Claim 3.23. *If $f \in \mathcal{F}_{0,1}$, then $f((\frac{1}{2}, x_{\max}]) \subseteq (\frac{1}{2}, x_{\max}]$.*

   Proof. Because $\frac{1}{2}$ maximizes f, $f(x) \leq x_{\max}$ for all $x \in [\frac{1}{2}, x_{\max}]$. Since f monotonically decreases, on $[\frac{1}{2}, x_{\max}]$, the claim can only be false if $f(x_{\max}) < \frac{1}{2}$. We show by contradiction that this is impossible.

   Because f is continuous and monotonically increases on $[0, \frac{1}{2}]$ and ranges from 0 to $x_{\max} \geq \frac{1}{2}$, there exists some $x' \leq \frac{1}{2}$ such that $f(x') = \frac{1}{2}$ and $f^2(x') = x_{\max}$.

   Let $g(x) = f^2(x) - x$. By assumption, $g(\frac{1}{2}) = f(x_{\max}) - \frac{1}{2} < 0$. By definition of x', $g(x') = \frac{1}{2} - x' \geq 0$. Because g is continuous, the Intermediate Value Theorem implies the existence of $x'' \in [x', \frac{1}{2})$ such that $g(x'') = 0$ and $f^2(x'') = x''$. Since f has no two-cycles, it must be the cause that $f(x'') = x''$ and $x'' = \frac{1}{2}$. However, this contradicts our finding that $x'' < \frac{1}{2}$, which means that $f(x_{\max}) \geq \frac{1}{2}$ and the claim holds. □

   Now, we proceed with analyzing each of the function pieces on some interval $[a,b] \subseteq [0,1]$ when $f \in \mathcal{F}_{0,1}$. The transformations are visualized in Figure 3.11.

   - **Increasing crossing piece**: If f^k has an lc on $[a,b]$, then f^{k+1} can be represented by lcUp on $[a,b]$.

     There exist c and d such that $a < d < c < \frac{1}{2} < b$, $f^k(c) = \frac{1}{2}$, and $f^k(d) = c$. Then, $[a, \frac{1}{2}(c+d)]$ supports an increasing crossing piece on $f \circ f^k$—because $f(f^k(a)) = 0$, $f(f^k(\frac{1}{2}(c+d))) > \frac{1}{2}$, and $f \circ f^k$ is strictly increasing on that interval since f is increasing before reaching $\frac{1}{2}$. $[0.5(c+d), b]$ supports a high peak—because c is a local maxima on $f \circ f^k$, and $f \circ f^k$ is strictly increasing before c and strictly decreasing after c.

- **Decreasing crossing piece**: For the same arguments, f^{k+1} can represented by UpDc on $[a, b]$ if f^k is represented by Dc on $[a, b]$.
- **Up peak**: Because f strictly decreases for $x > \frac{1}{2}$ and because $f^k([a, b]) \subseteq (\frac{1}{2}, x_{\max}]$ if Up represents f^k on $[a, b]$, c becomes a local minimum for $f \circ f^k$, and f^{k+1} is a high valley Uv on $[a, b]$.
- **Up valley**: Because f strictly decreases for $x > \frac{1}{2}$ and because $f^k([a, b]) \subseteq (\frac{1}{2}, x_{\max}]$ if Uv represents f^k on $[a, b]$, c becomes a local maximum for $f \circ f^k$, and f^{k+1} is a high peak Up on $[a, b]$.

Now, consider the inductive hypothesis. Because f^k can be represented by

$$\mathsf{Ic}(\mathsf{UpUv})^{k-1}\mathsf{UpDc},$$

applying the above transformations to each piece implies that f^{k+1} can be represented by

$$\mathsf{Ic}(\mathsf{UpUv})^{k}\mathsf{UpDc}.$$

Hence, the inductive argument goes through.

Proof of Proposition 3.22 generally The argument proceeds inductively. We show that if we have some $f \in \mathcal{F}_{q,k}$, then we can find some other function $h \in \mathcal{F}_{q-1,1-k}$ and characterize the behavior of f in terms of the behavior of h.

Since we assume that $q \geq 1$, there will always exist some $x^* > \frac{1}{2}$ that is a fixed point of f.[13] By symmetry, $f(1 - x^*) = x^*$. Let $\phi : [0, 1] \to [1 - x^*, x^*]$ be a decreasing isomorphism with $\phi(x) = x^* - x(2x^* - 1)$, and let

$$h = \phi^{-1} \circ f^2 \circ \phi.$$

[13]Sharkovsky's Theorem yields this by showing that the existence of a 2^q-cycle implies the existence of any 2^j-cycle, for all $j \in \{0, \ldots, q-1\}$. $x^* > \frac{1}{2}$ by our assumption that a 2-cycle $x_1 < x_2$ exists. It must be true that $x_2 > \frac{1}{2}$; otherwise, $f(x_2) > x_2 > x_1$, which breaks the cycle. Because $f(\frac{1}{2}) > \frac{1}{2}$ and $f(x_2) < x_2$, there exists $x^* \in (\frac{1}{2}, x_2)$ such that $f(x^*) = x^*$ by the Intermediate Value Theorem.

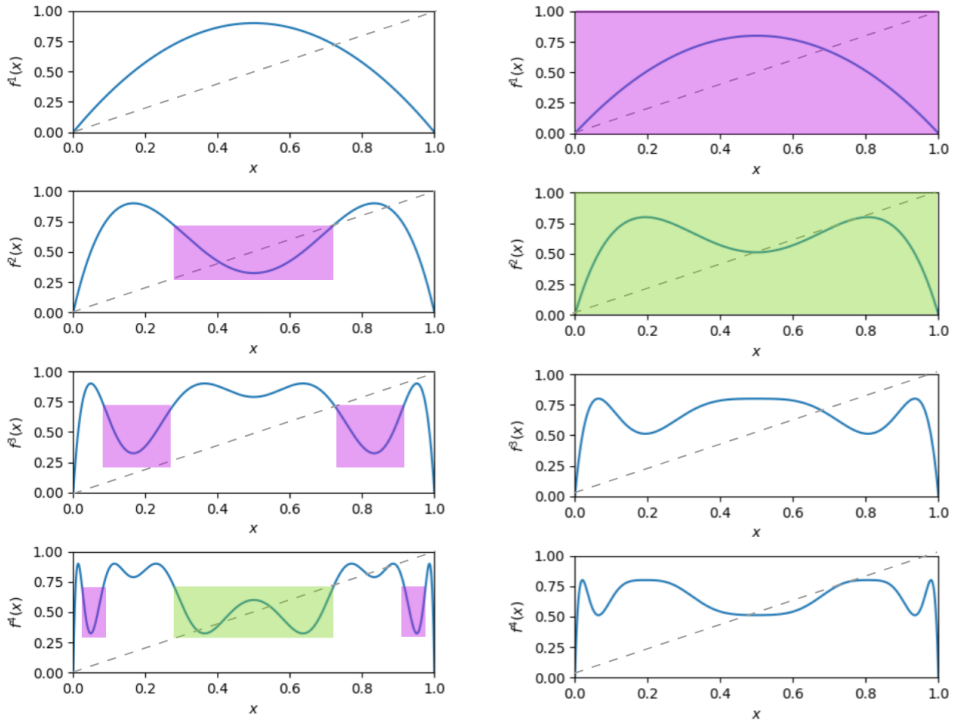


Figure 3.12: Visualizes the analogy between mappings in $\mathcal{F}_{q,z}$ and $\mathcal{F}_{q-1,1-z}$. The left plots the first 4 iterates of $f = f_{\log,0.9} \in \mathcal{F}_{4,1}$ (has a maximal 4-cycle with $f^4(\frac{1}{2}) > \frac{1}{2}$), while the right plots those of $f = f_{\log,0.85} \in \mathcal{F}_{2,0}$ (has a maximal 2-cycle with $f^2(\frac{1}{2}) < \frac{1}{2}$. The purple highlighted regions on the left behave qualitatively similar to $f_{\log,0.85}$, while the green regions are similar to $f^2_{\log,0.85}$.

h is a useful construct, because its behavior resembles simpler versions of f, with fewer cycles and oscillations. We use properties of h to relate pieces of f^k to those of $h^{k/2}$. We illustrate this recursive and fractal-like behavior in Figure 3.12.

Note that $h^k = \phi^{-1} \circ f^{2k} \circ \phi$.

Lemma 3.24. *h is a symmetric unimodal mapping with $h \in \mathcal{F}_{q-1,1-z}$.*

Proof. We verify the conditions for f to be unimodal mapping.

1. h is continuous and piece-wise differentiable on $[0,1]$ because f^2 is, and h is merely a linear transformation of f^2.

2. $h(0) = h(1) = 0$. $h((0,1))$ is strictly positive because $f((1-x^*, x^*)) = (x^*, x_{\max})$, $f^2((1-x^*, x^*)) = (f(x_{\max}), x^*)$, and $f(x_{\max}) < f(x^*) = x^*$ by f being decreasing on

$[\frac{1}{2}, 1]$.

3. h is uniquely maximized by $\frac{1}{2}$ because $\frac{1}{2}$ minimizes f^2 on the interval $[1 - x^*, x^*]$. f maps both $[1 - x^*, \frac{1}{2}]$ and $[\frac{1}{2}, x^*]$ onto $[x^*, x_{\max}]$ and is increasing and decreasing on the respective intervals. Because f maps $[x^*, x_{\max}]$ onto $[f(x_{\max}), x^*]$ and $f(x_{\max}) < x^*$ and is decreasing on $[x^*, x_{\max}]$, f^2 is increasing on $[1 - x^*, \frac{1}{2}]$ and decreasing on $[\frac{1}{2}, x^*]$. Thus, h is maximized by $\frac{1}{2}$, increases before $\frac{1}{2}$, and decreases after $\frac{1}{2}$.

4. We must also show that h is well-defined, which entails proving that $h(x) \leq 1$ for all $x \in [0, 1]$. Suppose that were not the case. Then, $h(\frac{1}{2}) > 1$, and there exists some $x' \leq \frac{1}{2}$ with $h(x') = 1$. There also exists some $x^{**} \in [1 - x', 1]$ with $h(x^{**}) = x^{**}$ by the Intermediate Value Theorem.

   Let $g(x) = h^3(x) - x$ and note that g is continuous on $[0, x']$. Observe that $g(1 - x^{**}) = 2x^{**} - 1 > 0$ and $g(x') = -x' < 0$. Thus, there exists $x'' \in [1 - x^{**}, x']$ with $g(x'') = 0$. Because h is increasing on $[0, x']$ and $x' > 1 - x^{**}$, it must be the case that $h(x') > x^{**} > x'$. Thus, x^{**} is not a fixed point and must be on a 3-cycle in h.

   However, if x^{**} is on a 3-cycle in h, then $\phi(x^{**})$ must be part of a 6-cycle in f. This contradicts the assumption that f cannot have a 2^{q+1}-cycle, because Sharkovsky's Theorem states that a 6-cycle implies a 2^{q+1}-cycle.

We show that h is symmetric.

$$h(x) = \phi^{-1}(f^2(\phi(x))) = \phi^{-1}(f^2(1 - \phi(x))) = \phi^{-1}(f^2(1 - x^* + x(2x^* - 1)))$$
$$= \phi^{-1}(f^2(x^* - (1 - x)(2x^* - 1))) = \phi^{-1}(f^2(\phi(1 - x))) = h(1 - x).$$

If $f^{2^q}(\frac{1}{2}) \geq \frac{1}{2}$, then $h^{2^{q-1}}(\frac{1}{2}) \leq \frac{1}{2}$, and if $f^{2^q}(\frac{1}{2}) \leq \frac{1}{2}$, then $h^{2^{q-1}}(\frac{1}{2}) \geq \frac{1}{2}$. Thus, $[[h^{2^{q-1}}(\frac{1}{2})]]_{1/2} = [[f^{2^q}(\frac{1}{2})]]_{1/2}$. By Lemma 3.25, h has a 2^{q-1}-cycle and does not have a 2^q-cycle. Thus, $h \in \mathcal{F}_{q-1,1-z}$. □

Lemma 3.25. *For $p \in \mathbb{Z}_+$, h has a p-cycle if and only if f has a $2p$-cycle.*

Proof. Suppose $x_1, \ldots, x_p$ is a p-cycle for h. Then, $\phi(x_1), \ldots, \phi(x_p)$ is a p-cycle for f^2. If $x_1, \ldots, x_p$ are distinct, then so must be $\phi(x_1), \ldots, \phi(x_p)$, since ϕ is an isomorphism. Thus,

$$\phi(x_1), f(\phi(x_1)), \ldots, \phi(x_p), f(\phi(x_p))$$

is a $2p$-cycle for f.

Conversely, if $x_1, \ldots, x_{2p}$ is a $2p$-cycle for f, then $x_1, x_3, \ldots, x_{2p-1}$ is a p-cycle for f^2 and

$$\phi^{-1}(x_1), \ldots, \phi^{-1}(x_{2p})$$

is a p-cycle for h. □

We proceed with a proof similar in structure to the one in the last section, where we divide each f^k into intervals and monitor the evolution of each as k increases. We define the classes of the pieces of some 1-dimensional map f^k on interval $[a, b]$ below. We visualize these classes in Figure 3.13.

- f^k is an **approach** A on $[a, b]$ if f is strictly increasing, $f^k(a) = 0$, and $f^k(b) = 1 - x^*$.
- Similarly, f^k is a **departure** D on $[a, b]$ if f^k is strictly decreasing, $f^k(a) = 1 - x^*$, and $f^k(b) = 0$.
- f^k is an i**-Left Valley** Lv_i on $[a, b]$ if $f^k : [a, b] \to [f(x_{\max}), x^*]$ and if there exists some strictly increasing and bijective $\sigma : [a, b] \to [1 - x^*, x^*]$ such that $f^k = \phi \circ h^i \circ \phi^{-1} \circ \sigma$ on $[a, b]$. Note that $f^k(a) = f^k(b) = x^*$—unless $i = 0$, in which case $f^k(a) = 1 - x^*$ and $f_r^k(b) = x^*$.
- f^k is analogously a i**-Right Valley** Rv_i if the same condition holds, except that σ is strictly decreasing.
- f^k is an i**-Left Peak** Lp_i on $[a, b]$ if f^{k-1} is Lv_{i-1} on $[a, b]$. It follows that $f^k : [a, b] \to [x^*, x_{\max}]$, that there exists some $c \in [a, b]$ such that $f^k(c) = x_{\max}$ (because

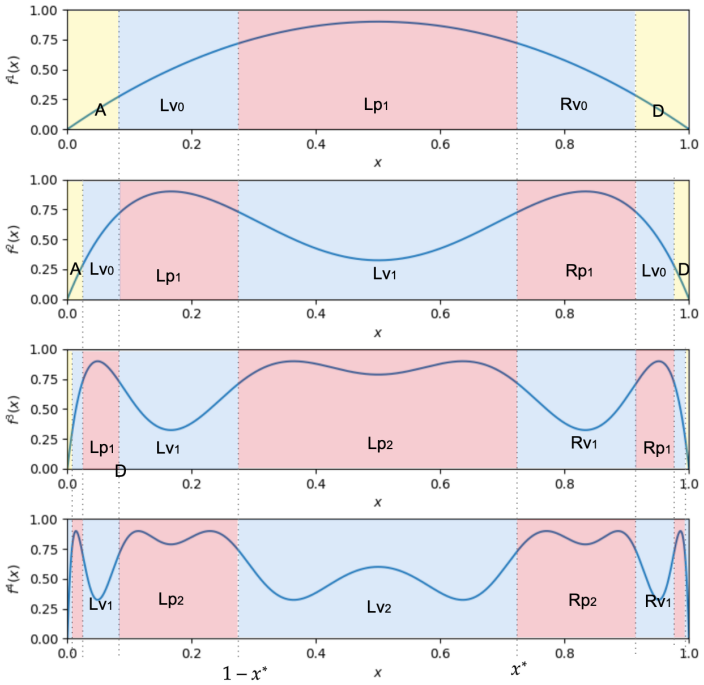


Figure 3.13: Similar to Figure 3.11, visualizes the classifications of f, f^2, f^3, f^4 for $f = f_{\log,0.9} \in \mathcal{F}_{4,1}$, and demonstrates that the decompositions are $\mathsf{ALv}_0\mathsf{Lp}_0\mathsf{Rv}_0\mathsf{D}$, $\mathsf{ALv}_0\mathsf{Lp}_1\mathsf{Lv}_1\mathsf{Rp}_1\mathsf{Rv}_0\mathsf{D}$, $\mathsf{ALv}_0\mathsf{Lp}_1\mathsf{Lv}_1\mathsf{Lp}_2\mathsf{Rv}_1\mathsf{Rp}_1\mathsf{Rv}_0\mathsf{D}$, and $\mathsf{ALv}_0\mathsf{Lp}_1\mathsf{Lv}_1\mathsf{Lp}_2\mathsf{Lv}_2\mathsf{Rp}_2\mathsf{Rv}_1\mathsf{Rp}_1\mathsf{Rv}_0\mathsf{D}$ respectively.

$\frac{1}{2} \in [f(x_{\max}), x^*])$, and that $f^k(a) = f^k(b) = x^*$.

- f^k is an ***i*-Right Peak** Rp_i on $[a, b]$ if f^{k-1} is Rv_{i-1} on $[a, b]$. The same claims hold as Lp_i.

Now, the proof of the number of oscillations proceeds in two steps. (1) We analyze how each of the above pieces evolves with each application of f. (2) We show how many maxima and minima each translates to.

Lemma 3.26. *When $f \in \mathcal{F}_{q,z}$ for $q \geq 1$ and for all $k \in \mathbb{Z}_+$, f^k can be decomposed into*

$2k+3$ *pieces* $\eta_1, \ldots, \eta_{2k+3}$ *such that*

$$\eta_i \text{ is } \begin{cases} \mathsf{A} \text{ if } i = 1 \\ \mathsf{Lv}_j \text{ if } i = 2j+2 \text{ for } j \in \{0, 1, \ldots, \lfloor k/2 \rfloor\} \\ \mathsf{Lp}_j \text{ if } i = 2j+1 \text{ for } j \in \{1, \ldots, \lfloor (k+1)/2 \rfloor\} \\ \mathsf{Rv}_j \text{ if } i = 2k-2j+2 \text{ for } j \in \{0, 1, \ldots, \lfloor (k-1)/2 \rfloor\} \\ \mathsf{Rp}_j \text{ if } i = 2k-2j+3 \text{ for } j \in \{1, \ldots, \lfloor k/2 \rfloor\} \\ \mathsf{D} \text{ if } i = 2k+3 \end{cases}$$

That is, if k *is even, then* f *can be represented by*

$$\mathsf{ALv}_0\mathsf{Lp}_1\mathsf{Lv}_1 \ldots \mathsf{Lv}_{k/2-1}\mathsf{Lp}_{k/2}\mathsf{Lv}_{k/2}\mathsf{Rp}_{k/2}\mathsf{Rv}_{k/2-1} \ldots \mathsf{Rv}_1\mathsf{Rp}_2\mathsf{Rv}_0\mathsf{D}.$$

If k *is odd, then* f *is represented by*

$$\mathsf{ALv}_0\mathsf{Lp}_1\mathsf{Lv}_1 \ldots \mathsf{Lp}_{(k-1)/2}\mathsf{Lv}_{(k-1)/2}\mathsf{Lp}_{(k+1)/2}\mathsf{Rv}_{(k-1)/2}\mathsf{Rp}_{(k-1)/2} \ldots \mathsf{Rv}_1\mathsf{Rp}_2\mathsf{Rv}_0\mathsf{D}.$$

Proof. This lemma is proved inductively. f can be decomposed into the pieces $\mathsf{ALv}_0\mathsf{Lp}_1\mathsf{Rv}_0\mathsf{D}$.

- By unimodality and symmetry, f is strictly increasing on $[0, \frac{1}{2})$ and strictly decreasing on $(\frac{1}{2}, 1]$. There exists some x_1 such that $[0, x_1]$ is strictly increasing and $f(x_1) = 1 - x^*$ (because $1 - x^* < x^* < x_{\max}$). Thus, f is A on $[0, x_1]$. Similarly, $[1 - x_1, 1]$ is strictly decreasing and $f(1 - x_1) = 1 - x^*$, which implies that f is D on $[1 - x_1, 1]$.

- Note that $x_1 < 1 - x^* < x^* < 1 - x_1$, and f is increasing on $[x_1, 1 - x^*]$ and decreasing on $[x^*, 1 - x_1]$.

  Because $[x_1, 1-x^*]$ is monotone, there exists continuous and increasing $\sigma : [x_1, 1-x^*] \to [1 - x^*, x^*]$ such that $f(x) = \sigma(x)$. Since h^0 is the identity map, it trivially also holds that $f(x) = \phi(h^0(\phi^{-1}(\sigma(x))))$. Because $f(x_1) = 1 - x^*$ and $f(x^*) = x^*$, it follows that

f is Lv_0 on $[x_1, 1-x^*]$.

By a similar argument, f is Rv_0 on $[x^*, 1-x_1]$, with the only difference being that σ needs to be strictly decreasing for it to hold.

- $[1-x^*, x^*]$ is Lp_1 because $[1-x^*, x^*]$ is Lv_0 on the identity map f^0. This trivially holds using the identity σ map.

Now, we prove the inductive step, which can be summed up by the following line:

$$\mathsf{A} \to \mathsf{ALv}_0; \mathsf{D} \to \mathsf{Rv}_0\mathsf{D}; \mathsf{Lv}_j \to \mathsf{Lp}_{j+1}; \mathsf{Lp}_j \to \mathsf{Lv}_j; \mathsf{Rv}_j \to \mathsf{Rp}_{j+1}; \mathsf{Rp}_j \to \mathsf{Rv}_j.$$

We show each part of the relationship as follows:

- If f^k is A on $[0, b]$, then there exists some $c \in (0, b)$ such that $f^{k+1}(c) = 1 - x^*$ because f^k is an isomorphism between $[0, b]$ and $[0, x^*]$.

  It follows that f^{k+1} is A on $[0, c]$ because f^{k+1} is strictly increasing on the interval from 0 to $1 - x^*$.

  $[c, b]$ is Lv_0 because there must exist some increasing σ such that $f^{k+1}(x) = \sigma(x)$ on that interval. Thus, it follows that $f^{k+1} = \phi \circ h^0 \circ \phi^{-1} \circ \sigma$ on $[0, b]$.

- The same argument holds for D. If f^k is D on $[a, 1]$, then there exists $c \in (a, 1)$ such that $[a, c]$ is Rv_0 and $[c, 1]$ is D.

- If f^k is Lv_j on $[a, b]$, then f^{k+1} is Lp_{j+1} on the same interval by the definition of Lp_{j+1}.

- Similarly, if f^k is Rv_j on $[a, b]$, then f^{k+1} is Rp_{j+1} on the same interval by the definition of Rp_{j+1}.

- If f^k is Lp_j on $[a, b]$, then f^{k-1} is Lv_{j-1} and hence f^{k-1} maps to $[f(x_{\max}), x^*]$ on the interval. Therefore, there exists σ such that $f^{k-1} = \phi \circ h^{j-1} \circ \phi^{-1} \circ \sigma$ on the interval.

We use the properties of h to show that f^{k+1} is Lv_j on $[a, b]$. Note that $f^2 = \phi \circ h \circ \phi^{-1}$ on $[f(x_{\max}), x^*]$.

$$f^{k+1} = f^2 \circ f^{k-1} = \phi \circ h \circ \phi^{-1} \circ \phi \circ h^{j-1} \circ \phi^{-1} \circ \sigma = \phi \circ h^j \circ \phi^{-1} \circ \sigma$$

Thus, f^{k+1} satisfies the condition to be Lv_j.

- By a identical argument, if f^k is Rp_j on $[a, b]$, then f^{k+1} is Rv_j.

The remainder of this argument follows by applying the above transition rules for each piece to the inductive hypothesis about the ordering of pieces in f^k to obtain the ordering for f^{k+1}. □

Now, we determine how many local maxima and minima are contained in each type of piece. Let maxima(f) and minima(f) represent the number of local maxima and minima respectively on mapping f on interval $[0, 1]$. We bound the total number of monotone pieces with these bounds by using $M(f) = 2\text{maxima}(f)$. We similarly abuse notation to bound the number of maxima and minima in a category with maxima($\mathcal{F}^k_{q,z}$) and minima($\mathcal{F}^k_{q,z}$), and in the interval $[a, b]$ with maxima(f, a, b) and minima(f, a, b).

By the base case in the previous section maxima($\mathcal{F}^k_{0,0}$) = 1, minima($\mathcal{F}^k_{0,0}$) = 2, maxima($\mathcal{F}^k_{0,1}$) = k, and minima($\mathcal{F}^k_{0,1}$) = $k + 1$. We obtain recurrences to represent maxima($\mathcal{F}^k_{q,z}$) and minima($\mathcal{F}^k_{q,z}$).

For each part, we rely on the following facts: If σ is a strictly increasing bijection, then maxima($f \circ \sigma, a, b$) = maxima(f, a, b). If σ is strictly decreasing, then minima($f \circ \sigma, a, b$) = maxima(f, a, b). (The reverse are true for minima of f.)

We analyze each type of piece individually, considering what happens when f has some kind of piece on interval $[a, b]$.

- Because A and D segments are strictly increasing or decreasing, maxima(f, a, b) = 0 when f has either piece on $[a, b]$. minima(f, a, b) = 1 because segments that support

A contain 0 and segments with D have 1, each of which f maps to 0.

- Because each Lv_i segment of f^k on $[a, b]$ can be represented as $\phi \circ h^i \circ \phi^{-1} \circ \sigma$, and because ϕ is strictly decreasing, $\text{maxima}(f_r^k, a, b) = \text{minima}(h^i)$ and $\text{minima}(f_r^k, a, b) = \text{maxima}(h^i)$. By Lemma 3.24, $h \in \mathcal{F}_{q-1,1-z}$, $\text{maxima}(f_r^k, a, b) \leq \text{minima}(\mathcal{F}^i_{q-1,1-z})$ and $\text{minima}(f_r^k, a, b) \leq \text{maxima}(\mathcal{F}^i_{q-1,1-z})$.

  The same analysis holds for each Rv_i segment.

- Consider an Lp_i segment of f^k on $[a, b]$, which has output spanning the interval $[x^*, x_{\max}]$. Because $x^* > \frac{1}{2}$, f is strictly decreasing on the domain $[x^*, x_{\max}]$. Thus, f^{k+1} must satisfy $\text{maxima}(f_r^{k+1}, a, b) = \text{minima}(f_r^k, a, b)$ and $\text{minima}(f_r^{k+1}, a, b) = \text{maxima}(f_r^k, a, b)$.

  Note by the definition of Lp_i that $[a, b]$ must also support an Lv_{i-1} segment on f^{k-1} and an Lv_i segment on f^{k+1}. From the previous bullet, the Lv_i segment must have at most $\text{minima}(\mathcal{F}^i_{q-1,1-z})$ maxima and $\text{maxima}(\mathcal{F}^i_{q-1,1-z})$ minima. Because there must be a one-to-one correspondence between minima of f^{k+1} and maxima of f^k on the interval and vice versa, the Lp_i segment has $\text{maxima}(f_r^k, a, b) \leq \text{maxima}(\mathcal{F}^i_{q-1,1-z})$ and $\text{minima}(f_r^k, a, b) \leq \text{minima}(\mathcal{F}^i_{q-1,1-z})$.

  The same analysis hold for each Rp_i segment.

Therefore, we can construct a recurrence relationship for the number of maxima and minima for f_r^k based on the sequences found in Lemma 3.26.

$$\begin{aligned}
\operatorname{maxima}(\mathcal{F}^k_{q,z}) \leq{}& \underbrace{\sum_{i=0}^{\lfloor k/2 \rfloor} \operatorname{minima}(\mathcal{F}^k_{q-1,1-z})}_{\mathsf{Lv}_i} + \underbrace{\sum_{i=0}^{\lfloor (k-1)/2 \rfloor} \operatorname{minima}(\mathcal{F}^k_{q-1,1-z})}_{\mathsf{Rv}_i} \\
&+ \underbrace{\sum_{i=1}^{\lfloor (k+1)/2 \rfloor} \operatorname{maxima}(\mathcal{F}^k_{q-1,1-z})}_{\mathsf{Lp}_i} + \underbrace{\sum_{i=1}^{\lfloor k/2 \rfloor} \operatorname{maxima}(\mathcal{F}^k_{q-1,1-z})}_{\mathsf{Rp}_i} \\
\operatorname{minima}(\mathcal{F}^k_{q,z}) ={}& \underbrace{2}_{\mathsf{A\&D}} + \underbrace{\sum_{i=0}^{\lfloor k/2 \rfloor} \operatorname{maxima}(\mathcal{F}^k_{q-1,1-z})}_{\mathsf{Lv}_i} + \underbrace{\sum_{i=0}^{\lfloor (k-1)/2 \rfloor} \operatorname{maxima}(\mathcal{F}^k_{q-1,1-z})}_{\mathsf{Rv}_i} \\
&+ \underbrace{\sum_{i=1}^{\lfloor (k+1)/2 \rfloor} \operatorname{minima}(\mathcal{F}^k_{q-1,1-z})}_{\mathsf{Lp}_i} + \underbrace{\sum_{i=1}^{\lfloor k/2 \rfloor} \operatorname{minima}(\mathcal{F}^k_{q-1,1-z})}_{\mathsf{Rp}_i}
\end{aligned}$$

We bound $\operatorname{maxima}(\mathcal{F}^k_{q,z})$ and $\operatorname{minima}(\mathcal{F}^k_{q,z})$ by induction to prove Proposition 3.22. We use the following inductive assumption over all k, q, and z, which suffices to prove the claim:

$$\operatorname{maxima}(\mathcal{F}^k_{q,z}), \operatorname{minima}(\mathcal{F}^k_{q,z}) \leq \begin{cases} (4q)^k & q \text{ is even, } z = 0, \text{ or } q \text{ is odd, } z = 1 \\ (4q)^{k+1} & q \text{ is even, } z = 1, \text{ or } q \text{ is odd, } z = 0. \end{cases}$$

By the previous section, the claim holds for $q = 0$ and all k and z, which gives the base case.

Moving forward, we assume that the claim holds for all values of q' with $q' \leq q$ and any k and z. We prove that it holds for $q + 1$ with any choices of k and z.

We show that the bound holds for $\operatorname{minima}(\mathcal{F}^k_{q+1,z})$ when $q+1$ is even and $z = 1$, or $q+1$ is odd and $z = 0$. The other cases are nearly identical. Since the bounds are trivial for $k = 1$, we prove them below for $k \geq 2$.

$$\begin{aligned}\text{minima}(\mathcal{F}^k_{q+1,z}) &\le 2 + \sum_{i=0}^{\lfloor k/2 \rfloor} (4i)^{q+1} + \sum_{i=0}^{\lfloor (k-1)/2 \rfloor} (4i)^{q+1} + \sum_{i=0}^{\lfloor (k+1)/2 \rfloor} (4i)^{q+1} + \sum_{i=0}^{\lfloor k/2 \rfloor} (4i)^{q+1} \\ &\le 4 \cdot \frac{k}{2} \cdot (2k)^{q+1} + (2(k+1))^{q+1} \le (2k)^{q+2} + (3k)^{q+1} \le (4k)^{q+2}.\end{aligned}$$

3.3.2.2 Proof of Theorem 3.18, Claim 2

We restate the claim:

Proposition 3.27 (Claim 2 of Theorem 3.18)**.** *Suppose f is a symmetric unimodal mapping whose maximal cycle is of length $p = 2^q$. For any $k \in \mathbb{N}$, there exists $g \in \mathcal{N}(u, 2)$ with width $u = O((4k)^{q+1}/\epsilon)$ such that $L_\infty(f^k, g) \le \epsilon$. Moreover, if $f = f_{tent,r}$, then there exists g of width $O((4k)^{q+1})$ with $g = f^k$.*

Proof. This part follows the bound on monotone pieces of f^k given in Proposition 3.21 and a simple neural network approximation bound.

Lemma 3.28. *Consider some continuous $f : [0, 1] \to [0, 1]$ with $M(f) \le m$. For any $\epsilon \in (0, 1)$, there exists $g \in \mathcal{N}(u, 2)$ of width $u = O(\frac{m}{\epsilon})$ such that $L_\infty(f, g) \le \epsilon$.*

Proof. A monotone function mapping to $[0, 1]$ can be ϵ-approximated by a piecewise-linear function with $O(\frac{1}{\epsilon})$ pieces, and hence, a 2-layer ReLU network of width $O(\frac{1}{\epsilon})$.

Every monotone piece can be approximated as such, which means that g has width $O(\frac{m}{\epsilon})$. □

For the case where $f = f_{\text{tent},r}$ for some r, it is always true that $\left|\frac{d}{dx} f^k_{\text{tent},r}(x)\right| = (2r)^k$, except when x is a local maximum or minimum. Thus, every monotone piece of f^k is linear, and f can be exactly expressed with a piecewise linear function with $O((4q)^{k+1})$ pieces, and also a ReLU neural network of width $((4q)^{k+1})$. □

3.3.2.3 Proof of Theorem 3.18, Claim 4

Recall that for unimodal $f : [0,1] \to [0,1]$ and threshold $t \in (0,1)$,

$$\mathcal{H}f, t := \{[[f^k]]_t : k \in \mathbb{N}\}$$

is the hypothesis class under consideration.

Proposition 3.29 (Claim 4 of Theorem 3.18)**.** *Suppose f is a symmetric unimodal mapping whose maximal cycle is of length $p = 2^q$. For any $t \in (0,1)$, $VC(\mathcal{H}f, t) \leq 18p^2$.*

This proof is involved and requires new notations and concepts.

Proof notations and preliminaries. Let $\{0,1\}^{\mathbb{N}}$ represent all countable infinite sequences of Boolean values, and let $\{0,1\}^*$ represent all finite sequences (including the empty sequence).

For $y \in \{0,1\}^{\mathbb{N}}$, let $y_{i:j} = (y_i, \ldots, y_j) \in \{0,1\}^{j-i+1}$ and $y_{i:} = (y_i, y_{i+1}, \ldots) \in \{0,1\}^{\mathbb{N}}$. For $w \in \{0,1\}^n, w' \in \{0,1\}^{n'}$, let $ww' = w \circ w' \in \{0,1\}^{n+n'}$ be their concatenation. Let $w^j = w \circ w \circ \cdots \circ w \in \{0,1\}^{jn}$.

Before we give the main result, we give a way to upper-bound the VC-dimension of countably infinite hypothesis classes $\mathcal{H} = \{h_1, h_2, \ldots, \} \subseteq ([0,1] \to \{0,1\})$. For some $x \in \mathcal{X}$, define $s_{\mathcal{H}} : [0,1] \to \{0,1\}^{\mathbb{N}}$ as $s_{\mathcal{H}}(x) = (h_i(x))_{i \in \mathbb{N}}$. We denote all patterns expressed by elements of the concept class $\mathcal{H}$ over all choices of $x \in [0,1]$:

$$\mathcal{S}_{\mathcal{H}} = \{s_{\mathcal{H}}(x) : \ x \in [0,1]\} \subset \{0,1\}^{\mathbb{N}}.$$

With this notation, $\mathcal{H}$ shatters d points if and only if there exist $y^{(1)}, \ldots, y^{(d)} \in \mathcal{S}_{\mathcal{H}}$ such that $|\{(y_j^{(1)}, \ldots, y_j^{(n)}) : j \in \mathbb{N}\}| = 2^d$. We equivalently say that $y^{(1)}, \ldots, y^{(d)}$ are shattered.

Here's where the idea of Regular Expressions (Regexes) comes in. If we can show all elements in $\mathcal{S}_{\mathcal{H}}$ are represented by some infinite-length Regex, then we can upper-bound

the number of points $\mathcal{H}$ can shatter, which is necessary to bound the expressive capacity of unimodal functions with recursive properties.

To that end, we first introduce a different notion of shattering. Then, we'll give an upper-bound for the VC-dimension of $\mathcal{H}$ when we have a Regex for $\mathcal{S}_{\mathcal{H}}$.

Definition 3.11. We say that $\mathcal{H}$ (or $\mathcal{S}_{\mathcal{H}}$) **weakly shatters** d points if there exist $w^{(1)} \circ y^{(1)}, \ldots, w^{(d)} \circ y^{(d)} \in \mathcal{S}_{\mathcal{H}}$ for $w^{(1)}, \ldots, w^{(d)} \in \{0,1\}^*$ such that $y^{(1)}, \ldots, y^{(d)}$ are shattered. Let the **weak VC-dimension** of $\mathcal{H}$ represent the maximum number of points $\mathcal{H}$ can weakly shatter and denote it $\mathrm{VC}_{\mathrm{weak}}(\mathcal{H}) = \mathrm{VC}_{\mathrm{weak}}(\mathcal{S}_{\mathcal{H}})$.

Using this notation, we can extend our notion of weak VC-dimension to any subset of $\{0,1\}^{\mathbb{N}}$, whether or not it corresponds to a hypothesis class. If $\mathcal{H} \subset S \subset \{0,1\}^{\mathbb{N}}$, then $\mathrm{VC}_{\mathrm{weak}}(\mathcal{H}) \leq \mathrm{VC}_{\mathrm{weak}}(S)$.

Note that if $\mathcal{H}$ shatters d points, then it also trivially weakly shatters d points. We can get this by taking $w_1 = \ldots, w_d$ to be the empty strings. Thus, the $\mathrm{VC}(\mathcal{H}) \leq \mathrm{VC}_{\mathrm{weak}}(\mathcal{H})$.

A Regex is a recursively defined subset of $\{0,1\}^{\mathbb{N}}$ that can be represented by a string. We describe how a Regex $R \subseteq \{0,1\}^{\mathbb{N}}$ can be defined below.

- One way to define a Regex is with a repeating sequence w^{∞} for $w \in \{0,1\}^n$. That is,

$$w^{\infty} = \{y \in \{0,1\}^{\mathbb{N}} : y_{in+1:(i+1)n} = w, \forall i \in \mathbb{N}\}.$$

  For instance, $(011)^{\infty} = \{(0,1,1,0,1,1,0,1,1,\ldots)\}$.

- For $w \in \{0,1\}^n$, if R is a Regex, then wR is also a Regex. This means satisfying sequences must start with w and then the remainder of the bits must satisfy R.

$$wR = \{y \in \{0,1\}^{\mathbb{N}} : y_{1:n} = w, y_{n+1:} \in R\}.$$

- w^*R is also a Regex, where w^* represents any number of recurrences of the finite

sequence s. That is,

$$w^* R = \cup_{j=0}^{\infty} w^j R.$$

- If R' is also a Regex, then so is $R \cup R'$.
- If R' is also a Regex, then so is $R \oplus R'$, where the odd entries of sequences in $R \oplus R'$ concatenated together must be in R and the even entries must be in R'.

$$R \oplus R' = \{y \in \{0,1\}^{\mathbb{N}} : y_{1,3,5,\ldots} \in R, y_{2,4,6,\ldots} \in R'\}.$$

Now, we can create a recursive upper-bound on the number of points $\mathcal{H}$ can weakly shatter. To do so, we assume that $\mathcal{H} \subseteq R$ for some Regex R and bound the weak VC dimension of R.

Lemma 3.30. *Consider infinite-length Regexes R, R', R'' and $w \in \{0,1\}^n$.*

1. *If $R = w^\infty$, then $VC_{weak}(R) \leq \log_2 n$.*
2. *If $R = wR'$, then $VC_{weak}(R) \leq VC_{weak}(R') + \log_2 n + 1$.*
3. *If $R = w^* R'$, then $VC_{weak}(R) \leq VC_{weak}(R') + \log_2 n + 1$.*
4. *If $R = R' \cup R''$, then $VC_{weak}(R) \leq VC_{weak}(R') + VC_{weak}(R'')$.*
5. *If $R = R' \oplus R''$, then $VC_{weak}(R) \leq 4 \max(VC_{weak}(R'), VC_{weak}(R'')) + 2$.*

Proof. 1. If $R = w^\infty$, then the set $Y = \{y : w \circ y \in w^\infty, w \in \{0,1\}^*\}$ contains at most n elements. Hence,

$$\left|\{(y_j^{(1)}, \ldots, y_j^{(d)}) : j \in \mathbb{N}\}\right| \leq n$$

for any fixed $y^{(1)}, \ldots, y^{(d)} \in Y$, and no more than $d = \log_2 n$ points can be weakly shattered.

2. Suppose R weakly shatters d points, so $y^{(1)}, \ldots, y^{(d)}$ are shattered for some $w^{(1)} \circ y^{(1)}, \ldots, w^{(d)} \circ y^{(d)} \in R$. If $Y = \{(y_j^{(1)}, \ldots, y_j^{(d)}) : j \in \mathbb{N}\}$ and $Y_n = \{(y_j^{(1)}, \ldots, y_j^{(d)}) : j \leq n\}$, then $|Y| = 2^d$ and $|Y_n| \leq n$. There exists some $v \in \{0,1\}^{1+\log_2 n}$ such that $v \circ \sigma \in Y \setminus Y_n$ for all $\sigma \in \{0,1\}^{d-1-\log_2 n}$. Therefore,

$$|\{(y_j^{(2+\log_2 n)}, \ldots, y_j^{(d)}) : j > n\}| = 2^{d-1-\log_2 n},$$

and there exist $d - 1 - \log_2 n$ points that can be weakly shattered by R', since none of the labelings with w are necessary.

3. Once again, suppose R weakly shatters d points, $y^{(1)}, \ldots, y^{(d)}$ for $w^{(1)} \circ y^{(1)}, \ldots, w^{(d)} \circ y^{(d)} \in R$. Because each $w^{(i)} \circ y^{(i)} \in w^* R'$, there exists an index ℓ_i such that $(w^{(i)} \circ y^{(i)})_{1:\ell_i} = w^{\ell_i/n}$ and $(w^{(i)} \circ y^{(i)})_{\ell_i+1:} \in R'$. Without loss of generality, assume $y^{(1)}, \ldots, y^{(d)}$ are ordered such that $\ell_i - \left|w^{(i)}\right|$ decreases. That is, the first $1 + \log_2 n$ sequences are the ones that "leave w^* last." Let $\ell^* := \ell_{1+\log_2 n} - \left|w^{(1+\log_2 n)}\right|$. Define Y and Y_{ℓ^*} analogously to the previous part and note that $|Y| = 2^d$. Because Y_{ℓ^*} corresponds only to labelings where the first $1 + \log_2 n$ elements come from subsets of w^∞, there exists some $v \in \{0,1\}^{1+\log_2 n}$ such that $v \circ \sigma \in Y \setminus Y_{\ell^*}$ for all $\sigma \in \{0,1\}^{d-1-\log_2 n}$. As before, there exist $d - 1 - \log_2 n$ points that can be weakly shattered by R'

4. There is no set of $\mathrm{VC}_{\mathrm{weak}}(R') + 1$ and $\mathrm{VC}_{\mathrm{weak}}(R'') + 1$ points that can be weakly shattered by R' and R'' respectively. Any $\mathrm{VC}_{\mathrm{weak}}(R') + \mathrm{VC}_{\mathrm{weak}}(R'') + 1$ points in R must have at either $\mathrm{VC}_{\mathrm{weak}}(R') + 1$ points in R' or $\mathrm{VC}_{\mathrm{weak}}(R'') + 1$ points in R''. Thus, at least one subset cannot be shattered.

5. Suppose without loss of generality that $d := \mathrm{VC}_{\mathrm{weak}}(R') \geq \mathrm{VC}_{\mathrm{weak}}(R'')$. Consider any $w^{(1)} \circ y^{(1)}, \ldots, w^{(d)} \circ y^{(4d+3)} \in R$. WLOG, assume that $\left|w^{(1)}\right|, \ldots, \left|w^{(2d+2)}\right|$ are even, which implies that $w_{\mathrm{odd}}^{(1)} \circ y_{\mathrm{odd}}^{(1)}, \ldots w_{\mathrm{odd}}^{(2d+2)} \circ y_{\mathrm{odd}}^{(2d+2)} \in R'$ and $w_{\mathrm{even}}^{(1)} \circ y_{\mathrm{even}}^{(1)}, \ldots w_{\mathrm{even}}^{(2d+2)} \circ$

$y^{(2d+2)}_{\text{even}} \in R''$. Therefore,

$$\begin{aligned}
&\left|\{(y_j^{(1)}, \ldots, y_j^{(4d+3)}) : j \in \mathbb{N}\}\right| \\
&\quad \leq 2^{2d+1} \left|\{(y_j^{(1)}, \ldots, y_j^{(2d+2)}) : j \in \mathbb{N}\}\right| \\
&\quad \leq 2^{2d+1} \left(\left|\{(y_j^{(1)}, \ldots, y_j^{(2d+2)}) : j \in \mathbb{N}_{\text{odd}}\}\right| + \left|\{(y_j^{(1)}, \ldots, y_j^{(2d+1)}) : j \in \mathbb{N}_{\text{even}}\}\right|\right) \\
&\quad \leq 2^{2d+1} \cdot 2 \sum_{i=0}^{d} \binom{2d+2}{i} < 2^{2d+2} \cdot 2^{2d+1} = 2^{4d+3}.
\end{aligned}$$

The last line follows by the Sauer Lemma. Thus, R cannot shatter $4d+3$ points if R' and R'' cannot shatter d points.

□

Here's an example of how to apply our regex rules:

$$\begin{aligned}
\text{VC}_{\text{weak}}(1^*0(01)^\infty \cup 10^\infty) &\leq \text{VC}_{\text{weak}}(1^*0(01)^\infty) + \text{VC}_{\text{weak}}(10^\infty) \\
&\leq 1 + \text{VC}_{\text{weak}}(0(01)^\infty) + 1 + \text{VC}_{\text{weak}}(0^\infty) \\
&\leq 2 + 1 + \text{VC}_{\text{weak}}((01)^\infty) \\
&\leq 3 + 1 = 4.
\end{aligned}$$

Proof of Proposition 3.29 Recall that we consider the hypothesis class

$$\mathcal{H}_{f,t} := \{[[f^k]]_t : k \in \mathbb{N}\}$$

for symmetric unimodal f and $t \in (0, 1)$.

To build up the argument, we first bound the VC-dimension for two simple cases.

- First, we consider the case when f has no fixed point. Thus, for all $x \in (0, 1]$, $f(x) < x$, which means that the sequence $f(x), f^2(x), \ldots$ is decreasing.

  If the threshold t is 0 or is greater than $f(\frac{1}{2})$, then the sequence will be all 0's or 1's, which will imply that $\text{VC}(\mathcal{H}_{f,t}) = 0$. Thus, the only interesting thresholds are

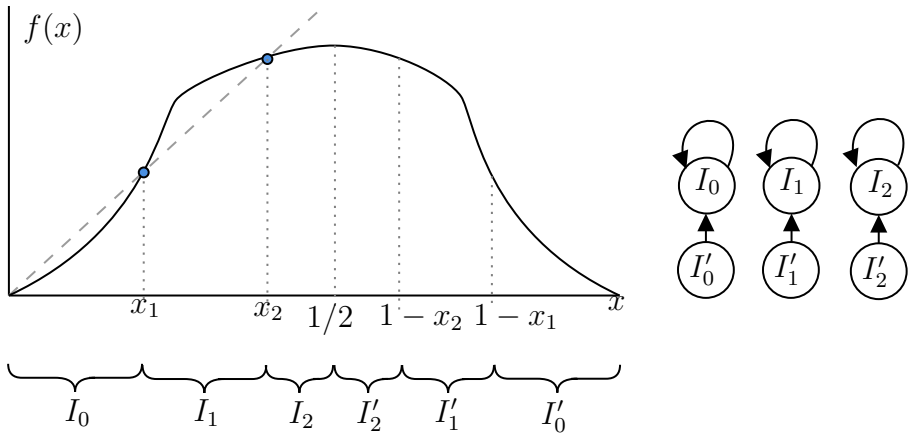


Figure 3.14: A plot of the domain of some f with two fixed points—both smaller than $\frac{1}{2}$—subdivided into intervals. The relationships of which intervals f maps onto one another are also visualized.

$t \in (0, f(\frac{1}{2})]$. Because the sequence is decreasing, $\mathcal{S}_{\mathcal{H}_{f,t}} = 1^*0^\infty$. From Lemma 3.30, $\text{VC}(\mathcal{H}_{f,t}) \leq \text{VC}_{\text{weak}}(\mathcal{H}_{f,t}) \leq 1$.

- Let $x_1 < \cdots < x_m$ be all the fixed points of f. Suppose $x_m \leq \frac{1}{2}$. By symmetry, for all $j \in [m]$, $f(1 - x_j) = x_j$.

  To analyze this function, we partition $[0, 1]$ into $2m + 2$ intervals: $I_0 = [0, x_1), I_0' = (1 - x_1, 1]$, $I_m = [x_m, \frac{1}{2}]$, $I_m' = (\frac{1}{2}, 1 - x_m]$, $I_j = [x_j, x_{j+1})$, and $I_j' = (1 - x_{j+1}, 1 - x_j]$ for all $j \in \{1, \ldots, m - 1\}$ (visualized in Figure 3.14).

  Because f is unimodal and because the edges of all intervals map to fixed points, for all $j \in \{0, \ldots, m\}$, $f(I_j') = f(I_j) = I_j$. In this case, it must be the case that $q = 0$ because f cannot have a 2-cycle. Such a cycle is impossible because it would have to be contained entirely in some I_j. In those intervals, it must be the case that either $\forall x \in I_j$, $f(x) \geq x$, or $\forall x \in I_j$, $f(x) \leq x$ (if this were not the case, then this would imply the existence of a fixed point other than x_j in I_j). Thus, cyclic behavior within an interval is impossible.

  Thus, we can construct a Regex to represent the itinerary of any $x \in [0, 1]$: $\bigcup_{j=0}^{m} I_j^\infty$.[14]

[14]This is a massive abuse of notation, but we use the same Regex notation to denote the intervals that are traversed as we use to denote the values of Boolean sequence.

Now, we consider all possible locations of threshold t:

- If $t \in I_j$, such that $f(x) \geq x$ for $x \in I_j$, then $\mathcal{S}_{\mathcal{H}_{f,t}} \subseteq 0^*1^\infty \cup 0^\infty \cup 1^\infty$. By Lemma 3.30, $\text{VC}_{\text{weak}}(\mathcal{H}_{f,t}) \leq 1$.
- If $t \in I_j$, such that $f(x) \leq x$ for $x \in I_j$, then $\mathcal{S}_{\mathcal{H}_{f,t}} \subseteq 1^*0^\infty \cup 0^\infty \cup 1^\infty$. By Lemma 3.30, $\text{VC}_{\text{weak}}(\mathcal{H}_{f,t}) \leq 1$.
- If $t \in \bigcup_{j=0}^{m} I'_j$, then $\mathcal{S}_{\mathcal{H}_{f,t}} = 0^\infty$, and $\text{VC}_{\text{weak}}(\mathcal{H}_{f,t}) = 0$.

Now, we give a lemma, which relates the VC-dimension of complex functions to that of simpler ones. Let $\mathcal{F}_q$ refer to the family of symmetric unimodal functions that have a 2^q-cycle but not a 2^{q+1}-cycle.

Lemma 3.31. *For any $f \in \mathcal{F}_q$ with fixed point $x^* > \frac{1}{2}$ and any $t \in [0,1]$,*

$$VC_{weak}(\mathcal{H}_{f,t}) \leq 4 \max_{g \in \mathcal{F}_{q-1}, t' \in [0,1]} VC_{weak}(\mathcal{H}_{g,t'}) + 10.$$

Proof. Consider some such f. Let $x_1 < \cdots < x_m$ be the fixed points of f where $x_m = x^* > \frac{1}{2}$. Because $\frac{1}{2}$ maximizes f, $f(\frac{1}{2}) \geq x_m > \frac{1}{2}$. This fixed point must the only fixed point no smaller than $\frac{1}{2}$; the existence of another such fixed point would contradict the fact that f is decreasing on $(\frac{1}{2}, 1]$. Thus, $x_1, \ldots, x_{m-1} < \frac{1}{2}$.

We build a recursive relationship by considering f^2 and relating some its output on some segments of $[0,1]$ to other maps with smaller q. For now, we instead attempt to upper-bound the VC-dimension of $\mathcal{H}_{f^2,t}$.

For all $j \in [m]$, unimodality implies that x_j and $1 - x_j$ are the only points that map to x_j and that the following ordering holds.

$$0 < x_1 < \ldots, < x_{m-1} < 1 - x_m < \frac{1}{2} < x_m < 1 - x_{m-1} < \ldots, 1 - x_1 < 1.$$

By the Intermediate Value Theorem, there exists some $x'_m \in (x_m, 1 - x_{m-1})$ such that $f(x'_m) = f(1 - x'_m) = 1 - x_m$ and $f^2(x'_m) = f^2(1 - x'_m) = x_m$.

We define intervals as follows:

- $I_0 = [0, x_1)$ and $I'_0 = (1 - x_1, 1]$.
- For all $j \in [m-2]$, $I_j = [x_j, x_{j+1})$ and $I'_j = (1 - x_{j+1}, 1 - x_j]$.
- $I_{m-1} = [x_{m-1}, 1 - x'_m)$ and $I'_{m-1} = (x'_m, 1 - x_{m-1}]$.
- $I_m = [1 - x'_m, 1 - x_m)$ and $I'_m = (x_m, x'_m]$.
- $I_{m+1} = [1 - x_m, \frac{1}{2})$, and $I'_{m+1} = [\frac{1}{2}, x_m]$.

For any $j \in \{0, \ldots, m+1\}$, f is increasing on all intervals I_j and decreasing on I'_j. By symmetry, $f(I_j) = f(I'_j)$. For all $j \in \{0, \ldots, m-2\}$, $f(I_j) = I_j$. $f(I_{m-1}) = I_{m-1} \cup I_m$, $f(I_m) = I_{m+1} \cup I'_{m+1}$, and $f(I_{m+1}) \subseteq I'_m$, because $f(\frac{1}{2}) \in [x_m, x'_m)$.[15]

From there, we obtain additional properties for f^2: $f^2(I_{m-1}) = I_{m-1} \cup I_m \cup I_{m+1} \cup I'_{m+1}$, $f^2(I_m) \subseteq I'_m$, and $f^2(I_{m+1}) \subset I_{m+1} \cup I'_{m+1}$. This suggests that there is recurrent structure that we can take advantage of to count all of the patterns.

Let $J_{m+1} := I_{m+1} \cup I'_{m+1}$. We create a Regex to track the behavior of iterates f^2, which we visualize in Figure 3.15:

$$\bigcup_{j=0}^{m-2} I_j^{\infty} \ \cup \ I_{m-1}^{*} I_m I_m'^{\infty} \ \cup \ I_m'^{\infty} \ \cup \ I_{m-1}^{*} J_{m+1}^{\infty}.$$

When an iterate of f^2 gets "stuck" in one of $I_0, I_1, \ldots, I_{m-1}$, it must either be at a fixed point, be strictly increasing, or be strictly decreasing. To suggest otherwise would imply the existence of another fixed point in those intervals, because f^2 is monotonically increasing or decreasing in all of those and either all x yield $f^2(x) \geq x$ or $f^2(x) \leq x$.

For the remaining intervals, one might notice in Figure 3.15 that zooming in on the intervals I_m, J_{m+1}, and I'_m for f^2 gives what looks like unimodal maps.[16] We take advantage

[15] This must be the case for the assumptions to be met. If $f(\frac{1}{2}) < x_m$, then x_m cannot be a fixed point because $\frac{1}{2}$ maximizes f. If $f(\frac{1}{2}) > x'_m$, then there exists a 3-cycle with points in I_{m+1}, I'_{m-1}, I_m, which contradicts the assumption that we only have power-of-two cycles.

[16] We use similar techniques here to those used in Section 3.3.2.1.

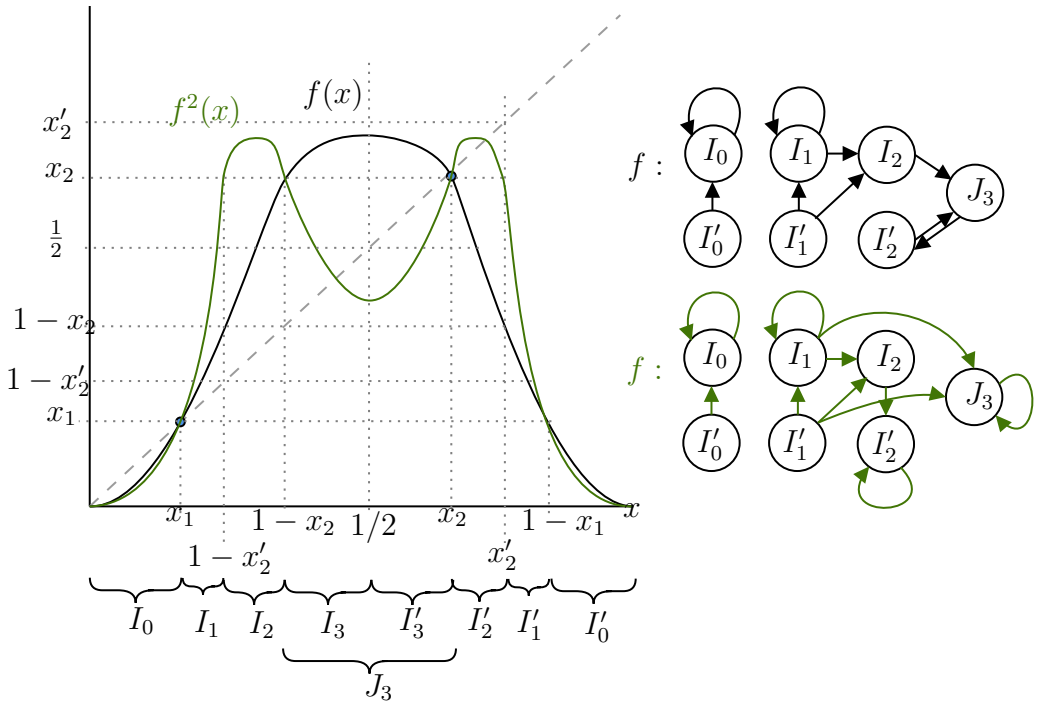


Figure 3.15: Like Figure 3.14, plot of f and f^2 with $m = 3$ fixed points with $x_m > \frac{1}{2}$ and visualizes the mappings between intervals.

of that structure to bound the complexity of the 0/1 Regexes for those intervals. We can formalize this by defining symmetric unimodal mappings h_m and h_{m+1} and bijective monotonic mappings $\phi_m : I'_m \to (0, 1]$ (increasing) and $\phi_{m+1} : J_{m+1} \to [0, 1]$ (decreasing) such that:

- For $x \in I_m$, $f^2(x) = \phi_m^{-1} \circ h_m \circ \phi_m(1 - x)$.
- For $x \in I'_m$, $f^2(x) = \phi_m^{-1} \circ h_m \circ \phi_m(x)$.
- For $x \in J_{m+1}$, $f^2(x) = \phi_{m+1}^{-1} \circ h_{m+1} \circ \phi_{m+1}(x)$.

Because f cannot have a cycle of length 2^{q+1}, h_m and h_{m+1} may not have cycles of length 2^q. Thus, we can reason inductively about how iterates behave when they're trapped in those intervals.

We do another case analysis of the 0/1 Regexes induced by different choices of t.

- If $t \in I_j$ for $j \in \{0, \ldots, m-1\}$, then $\mathcal{S}_{\mathcal{H}_{f^2,t}} \subseteq 0^\infty \cup 1^\infty \cup 0^*1^\infty \cup 1^*0^\infty$ because a sequence of iterates only crosses t if it enters the correct interval I_j, where the iterate

then will be stuck and must monotonically increase or decrease. By Lemma 3.30, $\mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{f^2,t}) \leq 2$.

- If $t \in I'_j$ for $j \in \{0, \ldots, m-1\}$, then $\mathcal{S}_{\mathcal{H}_{f^2,t}} = 0^\infty$, and $\mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{f^2,t}) = 0$.
- If $t \in I_m$, then $\mathcal{S}_{\mathcal{H}_{f^2,t}} = 0^\infty \cup 1^\infty \cup 0^*1^\infty$. Then, $\mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{f^2,t}) \leq 1$.
- If $t \in J_{m+1}$, then $\mathcal{S}_{\mathcal{H}_{f^2,t}} = 0^\infty \cup 0^*1^\infty \cup 0^*J^\infty_{m+1}$. Because h_{m+1} has at most a cycle of length 2^{q-1}, we have that

$$\mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{f^2,t}) \leq 2 + \max_{t'} \mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{h_{m+1},t'}).$$

- If $t \in I'_m$, then $\mathcal{S}_{\mathcal{H}_{f^2,t}} = 0^\infty \cup 0^*I'^\infty_m$. This gives us that

$$\mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{f^2,t}) \leq 1 + \max_{t'} \mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{h_m,t'}).$$

To get $\mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{f,t})$, notice that $\mathcal{S}_{\mathcal{H}_{f,t}} = \mathcal{S}_{\mathcal{H}_{f^2,t}} \oplus \mathcal{S}_{\mathcal{H}'_{f^2,t}}$, where $\mathcal{H}'_{f^2,t}$ refers to the outcome of all odd iterates of f. We show that $\mathcal{S}_{\mathcal{H}'_{f^2,t}} \subseteq \mathcal{S}_{\mathcal{H}_{f^2,t}}$ because the latter could induce all sequences produced by the former by starting with some x' such that $f^2(x') = f(x)$. Thus, by Lemma 3.30,

$$\begin{aligned}
\mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{f,t}) &\leq 4 \max_{t'} \mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{f^2,t'}) + 2 \\
&\leq 4 \max(2 + \max_{t'} \mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{h_{m+1},t'}), 1 + \max_{t'} \mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{h_m,t'})) + 2 \\
&\leq 4 \max_{g \in \mathcal{F}_{q-1},t'} \mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{g,t'}) + 10. \qquad \square
\end{aligned}$$

Now, we prove a bound on the VC-dimension for arbitrary q by induction with Lemma 3.31 to show that for $\mathrm{VC}(\mathcal{H}_{f,t}) \leq 18 \cdot 4^q$.

This holds when $q = 0$. There are two possible cases for the fixed point of such an f. If the the largest fixed point is smaller than $\frac{1}{2}$, then, by the simple cases explored at

the beginning, $\mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{f,t}) \leq 1$. Otherwise, we apply Lemma 3.31 along with the the other simple case—which tells us what happens when there are no fixed point—to get that $\mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{f,t}) \leq 4(1) + 10 = 14$. This trivially satisfies the proposition.

For the inductive step for arbitrary q, we iteratively apply Lemma 3.31 to obtain the final bound.

$$\begin{aligned}
\mathrm{VC}(\mathcal{H}_{f,t}) &\leq 4 \max_{g \in \mathcal{F}_{q-1}, t'} \mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{g,t'}) + 10 \\
&\leq 4^q \max_{g \in \mathcal{F}_0, t'} \mathrm{VC}_{\mathrm{weak}}(\mathcal{H}_{g,t'}) + 10 \sum_{i=0}^{q-1} 4^i \\
&\leq 14 \cdot 4^q + \frac{10}{3} 4^q \leq 18 \cdot 4^q.
\end{aligned}$$

3.3.2.4 Proof of Theorem 3.19, Claim 4

Proposition 3.32. *Suppose f is a symmetric unimodal function with a $2^q m$-cycle for odd m. Then for*

$$K = \exp\left(O\left(q + d \log(d + m)\right)\right),$$

$VC(\mathcal{H}_{f,K}) \geq d$ for $\mathcal{H}_{f,K} = \left\{[[f^k]]_{1/2} : k \in [K]\right\}$.

The claim holds by this proposition, since the VC-dimension of $\mathcal{H}_f$ is larger than every d and hence must be infinite.

Proof. The proof of this claim relies on the existence of a lemma that describes a characteristic of odd-period cycles of unimodal functions.

Lemma 3.33. *Let f be a symmetric unimodal function with some odd cycle $x_1, x_2, \ldots, x_m$ of length $m > 1$ such that $f(x_i) = x_{i+1}$ and $f(x_m) = x_1$. Then, there exists some i such that $x_i < \frac{1}{2}$ and $f(x_i) \geq \frac{1}{2}$.*

Proof. To prove the claim, it suffices to show that the following two cases are impossible: (1) $x_1, \ldots, x_m < \frac{1}{2}$ and (2) $x_1, \ldots, x_m \geq \frac{1}{2}$.

1. Suppose $x_1, \ldots, x_m < \frac{1}{2}$. By unimodality $x_j < x_{j'}$ implies that $f(x_j) < f(x_{j'})$. If x_1 is the smallest element of the cycle, then $f(x_1) > x_1$. For any other x_j, $f(x_j) > x_1$, which means that x_1 cannot be part of a cycle, which contradicts the odd cycle.

2. Suppose instead that $x_1, \ldots, x_m \geq \frac{1}{2}$.

   For this to be the case, $f(\frac{1}{2}) > \frac{1}{2}$ by unimodality. This fact paired with $f(1) < 1$ implies the existence of some $x^* \in (\frac{1}{2}, 1)$ with $f(x^*) = x^*$. Because f is decreasing on $[1/2, 1]$, $f([1/2, x^*)) \subseteq (x^*, 1]$ and $f((x^*, 1]) = [0, x^*)$.

   If $x_1 \in [\frac{1}{2}, x^*)$, then $x_2 \in (x^*, 1]$, and $x_3 \in [\frac{1}{2}, x^*)$. If apply this fact repeatedly, the oddness of m implies that $x_m \in [\frac{1}{2}, x^*)$ and $x_1 \in (x^*, 1]$, a contradiction. □

We show that $\mathrm{VC}(\mathcal{H}_{f^{2^q}, K/2^q}) > d$. If f has a cycle of length $2^q \cdot m$, then f^{2^q} has a cycle of length m. By Sharkovsky's Theorem, for all odd $m' > m$, f^{2^q} also has a cycle of length m'. Let $p_1 < \cdots < p_d$ be the smallest prime numbers greater than m. According to Lemma 3.34, $p_d \leq \left(\frac{K}{2^q}\right)^{1/d}$ for

$$K = 2^q \left(O(\max(d \log d, m)\right)^d = \exp\left(O\left(q + d \log(d + m)\right)\right).$$

For $j \in [m]$, let $x^{(j)}$ be the point guaranteed by Lemma 3.33 with $f^{2^q \cdot p_j}(x^{(j)}) = x^{(j)}$, $x^{(j)} < \frac{1}{2}$, and $f^{2^q}(x^{(j)}) \geq \frac{1}{2}$. Therefore, it follows that $f^{2^q \cdot \ell p_j}(x^{(j)}) < \frac{1}{2}$ and $f^{2^q(\ell p_j + 1)}(x^{(j)}) \geq \frac{1}{2}$ for all $\ell \in \mathbb{Z}_{\geq 0}$.

To show that $\mathcal{H}_{f^{2^q}}$ shatters $x^{(1)}, \ldots, x^{(d)}$, we show that for any labeling $\sigma \in \{0, 1\}^d$, there exists $h \in \mathcal{H}_{f^{2^q}, K/2^q}$ such that $h(x^{(j)}) = \sigma_j$.

- If $\sigma = (0, \ldots, 0)$, then consider $f^{2^q \cdot k}$, where $k = \prod_{j=1}^n p_j$. Then, for all j, $f^{2^q \cdot k}(x^{(j)}) < \frac{1}{2}$. Because $k \leq p_d^d \leq \frac{K}{2^q}$, there exists some $h \in \mathcal{H}_{f^{2^q}, K/2^q}$ that assigns zero to every $x^{(j)}$.

- Similarly, if $\sigma = (1, \ldots, 1)$, we instead consider $f^{2^q \cdot k}$ for $k = 1 + \prod_{j=1}^n p_j$. Now, for all j, $f^{2^q \cdot k}(x^{(j)}) \geq \frac{1}{2}$, and $k \leq p_d^d \leq \frac{K}{2^q}$, which means there exists satisfactory $h \in \mathcal{H}_{f^{2^q}, K/2^q}$.

- Otherwise, assume WLOG that $(\sigma_1, \ldots, \sigma_\ell) = (0, \ldots, 0)$ and $(\sigma_{\ell+1}, \ldots, \sigma_d) = (1, \ldots, 1)$ for $\ell \in (1, d)$. We satisfy the claim for $f^{2^q \cdot k}$ if we choose some k with $k = q_1 \prod_{i=1}^{\ell} p_i = 1 + q_2 \prod_{i=\ell+1}^{d} p_i$, for some $q_1, q_2 \in \mathbb{Z}_+$.

  We find $q_1 \in [\prod_{i=\ell+1}^{d} p_i]$ and $q_2 \in [\prod_{i=1}^{\ell} p_i]$ by choosing them such that:

$$q_1 \prod_{i=1}^{\ell} p_i \equiv 1 \pmod{\prod_{i=\ell+1}^{d} p_i}$$
$$q_2 \prod_{i=\ell+1}^{d} p_i \equiv -1 \pmod{\prod_{i=1}^{\ell} p_i}.$$

  This is possible because $p_1, \ldots, p_d$ are prime, and $\gcd\left(\prod_{i=1}^{\ell} p_i, \prod_{i=\ell+1}^{d} p_i\right) = 1$.

  Because $k \leq \prod_{i=1}^{d} p_i \leq p_d^d \leq \frac{K}{2^q}$, there must exist some satisfactory $h \in \mathcal{H}_{f^{2^q}, K/2^q}$. □

Lemma 3.34. *For $m \geq 3$ and any $d \geq 0$, there exist d primes such that $m \leq p_1 < \cdots < p_d$ for*

$$p_d = O(\max(d \log d, m)).$$

Proof. Let $\pi(x) = |\{y \in [x] : y \text{ is prime}\}|$ be the number of primes no larger than x. By the Prime Number Theorem,

$$\frac{x}{\log(x) + 2} \leq \pi(x) \leq \frac{x}{\log(x) - 4},$$

for all $x \geq 55$ (Rosser, 1941). Thus, for some $m' = O(\max(d \log d, m))$, the number of prime numbers smaller than m' is

$$\Omega\left(\frac{d \log d}{\log(d \log d)} + \frac{m}{\log m}\right) = \Omega\left(d + \frac{m}{\log m}\right),$$

and the number between m and m' is $\Omega(d)$. Thus, $p_d \leq m'$. □

3.4 Supplemental background on discrete dynamical systems and itineraries

In this section, we provide background information from the discrete dynamical systems literature that may aid the reader in understanding and contextualizing the results of previous sections. Section 3.4.1 provides more examples of unimodal mappings with characterizations of their cyclic itineraries. Section 3.4.2 discusses the ordering of cycle itineraries, as characterized by Metropolis, Stein, and Stein (1973). Finally, Section 3.4.3 provides a proof that the existence of an increasing p cycle can be numerically validated.

3.4.1 Examples of Itineraries

Let the tent map and logistic map be defined by $f_{\text{tent},r}(x) = 2r\max(x, 1-x)$ and $f_{\log,r}(x) = 4rx(1-x)$ respectively, for parameter $r \in (0,1)$.

Example 3.1. *For all $r \in (\frac{1}{2}, 1]$, there is a two-cycle C of itinerary 12 (which is the only itinerary for a 2-cycle) in $f_{tent,r}$ with*

$$C = \left(\frac{2r}{1+4r^2}, \frac{4r^2}{1+4r^2}\right).$$

Example 3.2. *When $r = \frac{1+\sqrt{5}}{4}$, there is a two-cycle C of $f_{log,r}$ with*

$$C = \left(\frac{1}{2}, \frac{1+\sqrt{5}}{4}\right).$$

Example 3.3. *When $r \in [\frac{1+\sqrt{5}}{4}, 1]$, $f_{tent,r}$ has a three-cycle C of itinerary 123 with*

$$C = \left(\frac{2r}{1+8r^3}, \frac{4r^2}{1+8r^3}, \frac{8r^3}{1+8r^3}\right).$$

Note that this and Example 3.1 are consistent with Sharkovsky's Theorem; whenever there exists a three-cycle, there also exists a two-cycle.

Example 3.4. *When $r \in [\frac{1}{2}, 1]$, there also exists a four-cycle C of itinerary 1324 for $f_{tent,r}$ with*

$$C = \left(\frac{8r^3 - 4r^2 + 2r}{16r^2 + 1}, \frac{16r^4 - 8r^3 + 4r^2}{16r^2 + 1}, \frac{16r^4 - 8r^3 + 2r}{16r^2 + 1}, \frac{16r^4 - 4r^2 + 2r}{16r^2 + 1} \right).$$

Again, this reaffirms Sharkovsky's Theorem, since this cycle always exists when the above three-cycle exists.

Example 3.5. *However, when $r \in (0.9196\ldots, 1]$, there also exists a four-cycle C of itinerary 1234 for $f_{tent,r}$ with*

$$C = \left(\frac{2r}{16r^2 + 1}, \frac{4r^2}{16r^2 + 1}, \frac{8r^3}{16r^2 + 1}, \frac{16r^4}{16r^2 + 1} \right).$$

This demonstrates a relationship beyond Sharkovsky's theorem: whenever a 1234 four-cycle exists, a 123 three-cycle also exists. This will be integral to the bounds we show.

Example 3.6. *The triangle map from Telgarsky, 2016, $f_{tent,1}$ has an increasing p-cycle C_p for every $p \in \mathbb{N}$ with*

$$C_p = \left(\frac{2}{1 + 2^p}, \frac{2^2}{1 + 2^p}, \ldots, \frac{2^p}{1 + 2^p} \right).$$

Thus Theorem 3.15 and Fact 3.4 retrieve the fact used by Telgarsky that $M(f_{tent,1}) = \Omega(2^k)$.

3.4.2 Orderings of Itineraries

As has been mentioned before, the existence of some cycles can be shown to imply the existence of other cycles. Sharkovsky's Theorem famously does this by showing that if $p \triangleright p'$, then the existence of a p-cycle implies the existence of a p'-cycle. Proposition 3.35 can be used to imply that the existence of a chaotic p-cycle implies the existence of a chaotic $(p-1)$-cycle. These pose a broader question: Is there a complete ordering on all cycle itineraries that can appear in unimodal mappings? And does this ordering coincide with the amount of "chaos" induced by a cycle?

Researchers of discrete dynamical systems have thoroughly investigated these questions; we refer interested readers to Metropolis, Stein, and Stein, 1973; Alsedà, Llibre, and Misiurewicz, 2000 for a more comprehensive survey. We introduce the basics of this theory as it relates to our results.

Metropolis, Stein, and Stein, 1973 present a partial ordering over cyclic itineraries present in unimodal mappings, which serves as a measurement of the complexity of the function. That is, two itineraries $\mathbf{a}$ and $\mathbf{a}'$ may be related analogously to Sharkovsky's Theorem with $\mathbf{a} \triangleright \mathbf{a}'$, if f having itinerary $\mathbf{a}$ implies that f has itinerary $\mathbf{a}'$. This ordering for all cycles of length at most 6 is illustrated in Table 3.3. For instance, if a unimodal map has a cycle with itinerary 12435, then it also has a cycle with itinerary 135246.

Table 3.3: For any unimodal function f, let $f_r(x) := rf(x)$ for $r > 0$. As r increases, any such family obtains new cycles in the same order, and those cycles are super-stable in the same order. This translates Table 1 of Metropolis, Stein, and Stein, 1973 to our notation and shows at what values of r, $f_{\log,r}$ has various super-stable cycles of length at most 6.

Cycle length p	Itinerary	Regime	Super-stable r	Cycle Type
2	12	Doubling	0.8090	Primary
4	1324	Doubling	0.8671	Primary
6	143526	Chaotic	0.9069	Primary
5	13425	Chaotic	0.9347	Stefan, Primary
3	123	Chaotic	0.9580	Stefan, Increasing, Primary
6	135246	Chaotic	0.9611	
5	12435	Chaotic	0.9764	
6	124536	Chaotic	0.9844	
4	1234	Chaotic	0.9901	Increasing
6	123546	Chaotic	0.9944	
5	12345	Chaotic	0.9976	Increasing
6	123456	Chaotic	0.9994	Increasing

We make several observations about the table and make connections to the itineraries discussed elsewhere in the paper.

- The table does not contradict Sharkovsky's Theorem. Note that $3 \triangleright 5 \triangleright 6 \triangleright 4 \triangleright 2$, and order in which the first itinerary occurs of a period is the same as the Sharkovsky

ordering:

$$12 \lhd 1324 \lhd 143526 \lhd 13425 \lhd 123.$$

- The last cycle to occur for a given period is its increasing cycle and it occurs as p increases (not with the Sharkovsky ordering of p):

$$12 \lhd 123 \lhd 1234 \lhd 12345 \lhd 123456.$$

- The first cycle to appear for every odd period is its *Stefan cycle* (123, 13425). This is proved by Alsedà, Llibre, and Misiurewicz, 2000 and justifies why Theorem 3.10 relies on the existence of a Stefan cycle whenever there is an odd period.

- There exist cycles of power-of-two length (e.g. 1234) that induce non-power-of-two cycles (e.g. 123).

Following the last bullet point, we distinguish between the 2^q-cycles that only induce cycles of length 2^i for $i < q$ and those that induce non-power-of-two cycles. To do so, we say that the itinerary of a p-cycle is *primary* if it induces no other p-cycle with a different itinerary.

We say that an itinerary $\mathbf{a}' = a'_1 \dots a'_{2p}$ of a $2p$-cycle is a *2-extension* of itinerary $\mathbf{a} = a_1 \dots a_p$ of a p-cycle if

$$a_i = \left\lceil \frac{a'_i}{2} \right\rceil = \left\lceil \frac{a'_{i+p}}{2} \right\rceil$$

for all i. For instance, 12 is a 2-extension of 1, 1324 is of 12, 15472638 is of 1324, and 135246 is of 123.

Theorem 2.11.1 of Alsedà, Llibre, and Misiurewicz (2000) characterizes which itineraries are primary. It critically shows that a power-of-two cycle is primary if and only if it is composed of iterated 2-extensions of the trivial fixed-point itinerary 1. As a result, 1324 is a primary itinerary and 1234 is not. This sheds further light on the warmup example

given in Section 3.1.4, where f^k_{1324} has a polynomial number of oscillations, while f^k_{1234} has an exponential number.

According to Theorem 2.12.4 of Alsedà, Llibre, and Misiurewicz (2000), the existence a non-primary itinerary of any period implies the existence of some cycle with period not a power of two. Hence, f can *only* be in the doubling regime (where all periods are powers of two) if all of those power-of-two periods are primary. The existence of any non-primary power-of-two period (such as 1234 or 13726548) implies that the f is in the chaotic regime.

This ordering can also be visualized using the bifurcation diagrams in Figure 3.16. The diagram plots the convergent behavior of $f^k_r(x)$ for large k, where r is some parameter and reflects the complexity of the unimodal function f_r. (When $r = 0$, $f_r = 0$; when $r = 1$, $x_{\max} = 1$, and $C_{0,1}(f^k) = 2^k$.) As r increases, the number of oscillations of f^k_r increases and with it, new cycles are introduced. Each new cycle has a *stable* region over parameters r where $f^k_r(x)$ converges to the cycle, and the bifurcation diagram visualizes when each of these stable regions occurs. While the three functions families f_r have different underlying unimodal functions, they produce qualitatively identical bifurcation diagrams that feature the same ordering of itineraries.

Our discussions of the *doubling* and *chaotic* regimes in Section 3.3 are inspired by these bifurcation diagrams. Parameter values r are naturally partitioned into two categories: those on the left side of the diagram where the plot is characterized by a branching of cycles (the doubling regime) and those on the right side where there are extended regions of chaos, interrupted by small stable regions (the chaotic regime).

3.4.3 Identifying Increasing Cycles in Unimodal Maps

It is straightforward to determine whether a symmetric and unimodal f has an increasing p-cycle. Algorithmically, one can do so by verifying that $f(\frac{1}{2}) > \frac{1}{2}$ and counting how many consecutive values of $k \geq 2$ satisfy $f^k(x_0) < \frac{1}{2}$.

Proposition 3.35. *Consider some $p \geq 2$ and a symmetric unimodal mapping f. f has an*

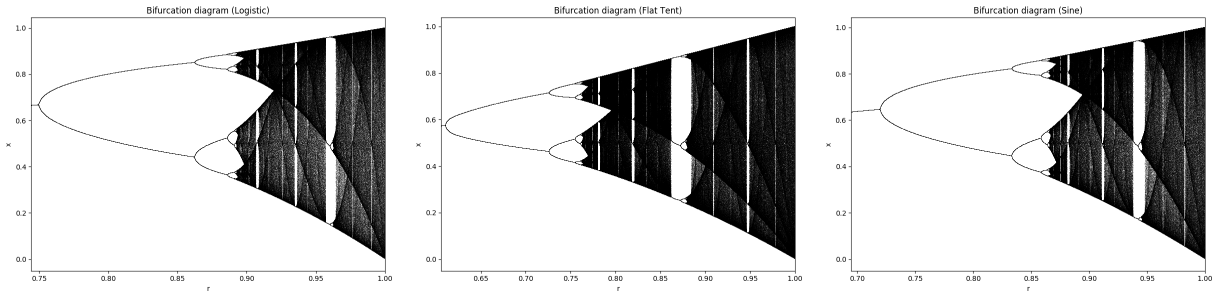


Figure 3.16: Bifurcation diagrams—which display the qualitative behavior of a family of functions f_r as the parameter $r \in [0,1]$ changes—showing the convergence behavior for iterates $f_r^k(x)$ for large k. For fixed r on the horizontal axis, the points plotted correspond to $f^k(x_0)$ for very large k. Regions of r where a vertical slice contains p discrete points indicates the existence of a *stable* p-cycle, since $f^k(x_0)$ converges exclusively to those points. Regions where the slice has a dispersed mass of points exhibit chaos. As r increases, cycles of different itineraries appear and experience stability in the same order indicated by Table 3.3. In the first plot, f_r is the logistic map $f_r(x) = f_{\log,r}(x) = 4rx(1-x)$. The second f_r is the"flat tent map," $f_r(x) = \min\{\frac{5rx}{2}, r, \frac{5rx}{2}(1-x)\}$, and the third is the sine map, $f_r(x) = r\sin(\pi x)$. The three are qualitatively identical and exhibit self-similarity.

increasing p-cycle if

$$f^2\left(\frac{1}{2}\right) < \cdots < f^p\left(\frac{1}{2}\right) \leq \frac{1}{2} < f\left(\frac{1}{2}\right),$$

then f has an increasing p-cycle.

Proof. Refer to Figure 3.17 for a visualization of the variables and inequalities defined.

Let $x' = f(\frac{1}{2})$. By the unimodality of f and the fact that $x' > \frac{1}{2}$, there exists some $x'' > \frac{1}{2}$ such that

$$f(x'') < f^2(x'') < \cdots < f^{p-1}(x'') = \frac{1}{2}.$$

Because f is monotonically increasing on $[0, \frac{1}{2}]$, the following string of inequalities hold.

$$f(x') \leq f(x'') < f^2(x') \leq f^2(x'') < \cdots < f^{p-1}(x') \leq f^{p-1}(x'') = \frac{1}{2} \tag{3.1}$$

It then must hold that $x' \geq x''$.

Let $g(x) = f^p(x) - x$ and note that g is continuous. Because $\frac{1}{2}$ maximizes f, it must be the case that $f^p(x') \leq x'$ and $g(x') \leq 0$. Because $f^p(x'') = x'$ and $x'' \leq x'$, $g(x'') \geq 0$. Hence,

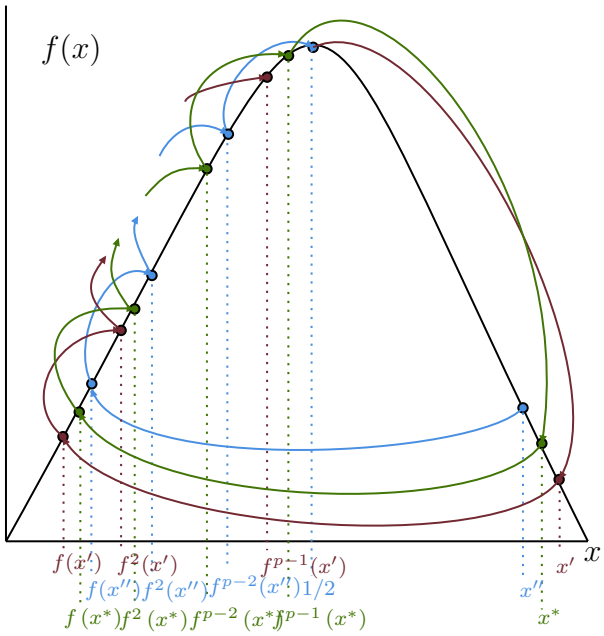


Figure 3.17: Visualizes the proof of Proposition 3.35.

there exists $x^* \in [x'', x']$ such that $g(x^*) = 0$ and $f^p(x^*) = x^*$.

Since $x^* \in [x'', x']$, it must also be the case that $f^j(x^*) \in [f^j(x'), f^j(x'')]$ for $j \in [p-1]$. By Equation (3.1), it follows that

$$f(x^*) < f^2(x^*) < \cdots < f^{p-1}(x^*) < f^p(x^*) = x^*.$$

Hence, there exists an increasing p-cycle. □

3.5 Conclusion

This chapter shares the connections drawn by Sanford and Chatziafratis (2022) between deep learning theory and dynamical systems. By applying a characterization of discrete dynamical systems by Metropolis, Stein, and Stein (1973), we obtain novel depth-width tradeoffs for the expressivity of neural networks. While prior works relied on Sharkovsky's

theorem and periodicity to provide families of functions that are hard to approximate with shallow neural networks, we go beyond periodicity. Studying the chaotic itineraries of unimodal mappings, we reveal subtle connections between expressivity and different types of periods, and we use them to shed new light on the benefits of depth in the form of enhanced width lower bounds and stronger approximation errors. More broadly, we believe that it is an exciting direction for future research to exploit similar tools and concepts from the literature of dynamical systems to improve our understanding of neural networks, e.g., their dynamics, optimization, and robustness properties.

These results complement those of Chapters 2 and 4 investigating the impact of depth on the expressivity of neural networks. Those chapters similarly consider intrinsically univariate targets with a measure of complexity associated with the frequency of oscillations, but these shallow models cannot utilize compositionality to fit the target function. This chapter provides a broad characterization of compositionality-based targets that can only be efficiently approximated by deep networks. These results reveal a fine line between targets that can be efficiently approximated by shallow models and those that require a linear scaling in network depth to achieve a given approximation error.

After employing dynamical systems to characterize a phase transition between different learning regimes, the authors would later identify another phase transition in the weight initialization of recurrent neural networks (Chatziafratis et al., 2022). Just as a small increase in the maximum value of a logistic mapping can introduce a chaotic itinerary that makes its iterates hard to approximate, a small perturbation to the variance of the initialization of an RNN can introduce exploding or vanishing gradients that make it hard to train. This later work captures this phase transition by applying similar tools from dynamical systems, such as Sharkovsky's theorem.

Finally, these results inspired the work in Chapter 6, where the importance of depth in modern sequential architectures is rigorously characterized by compositional target tasks. While these results use a different mathematical toolset and apply to a different class of

neural networks, they both share the core insight that certain iterated tasks can be solved much more efficiently with depth than with width.

Chapter 4: Intrinsic dimensionality of bounded-norm shallow neural network interpolants

This chapter studies the structural and statistical properties of $\mathcal{R}$-norm minimizing interpolants of datasets labeled by specific target functions. The $\mathcal{R}$-norm is the basis of an inductive bias for two-layer neural networks, recently introduced to capture the functional effect of controlling the size of network weights, independently of the network width. We find that these interpolants are intrinsically multivariate functions, even when there are ridge functions that fit the data, and also that the $\mathcal{R}$-norm inductive bias is not sufficient for achieving statistically optimal generalization for certain learning problems. Altogether, these results shed new light on an inductive bias connected to practical neural network training.

The research presented in this chapter reflects the work of Ardeshir, Hsu, and Sanford (2023).

4.1 Introduction

Research on neural network inductive biases is important for theoretical understanding and developing practical guidance in network training. Recent theories of generalization rely on inductive biases of training algorithms to explain how neural nets that (nearly) interpolate training data can be accurate out-of-sample (Neyshabur, Tomioka, and Srebro, 2015; Zhang et al., 2021). When inductive biases are made explicit and their effects are elucidated, they can be incorporated into training procedures when deemed appropriate for a problem.

In this chapter, we study the inductive bias for two-layer neural networks implied by a variational norm called the $\mathcal{R}$-*norm*, introduced by Savarese et al. (2019) and Ongie et al. (2019) to capture the functional effect of controlling the size of network weights. (A definition

is given in Section 4.2.2.1.) We focus on the *approximation* and *generalization* consequences of preferring networks with small $\mathcal{R}$-norm in the context of learning explicit target functions. It is well-known that the size of the weights can play a critical role in generalization properties of neural networks (Bartlett, 1996), and weight-decay regularization is a common practice in gradient-based training (Hinton, 1987; Hanson and Pratt, 1988). Thus, explicating the consequences of the $\mathcal{R}$-norm inductive bias may advance our understanding of generalization in practical settings.

We investigate the d-dimensional variational problem (VP), which seeks a neural net $g\colon \mathbb{B}^d \to \mathbb{R}$ of minimum $\mathcal{R}$-norm among those that perfectly fit a given labeled dataset $\{(x_i, y_i)\}_{i\in[n]} \subset \mathbb{B}^d \times \mathbb{R}$:

$$\inf_{g\colon \mathbb{B}^d \to \mathbb{R}} \|g\|_{\mathcal{R}} \quad \text{s.t.} \quad g(x_i) = y_i \qquad \forall i \in [n]; \tag{VP}$$

as well as a variant (ϵ-VP) that only requires g to uniformly approximate labels up to error $\epsilon \in (0, 1)$:

$$\inf_{g\colon \mathbb{B}^d \to \mathbb{R}} \|g\|_{\mathcal{R}} \quad \text{s.t.} \quad |g(x_i) - y_i| \leq \epsilon \quad \forall i \in [n]. \tag{ϵ-VP}$$

Here, $\mathbb{B}^d \subset \mathbb{R}^d$ is a d-dimensional domain of interest. We study the structural and statistical properties of solutions to these problems for datasets labeled by specific target functions in high dimensions.

The recent introduction of the $\mathcal{R}$-norm and its connections to weight-decay regularization have catalyzed research on the foundational properties of solutions to (VP). In particular, solutions in the one-dimensional ($d = 1$) setting have been precisely characterized and their generalization properties are now well-understood by their connections to splines (Debarre et al., 2022; Savarese et al., 2019; Parhi and Nowak, 2021a; Hanin, 2021). However, far less is known about the solutions of $\mathcal{R}$-norm-minimizing interpolants for the general d-dimensional case.

Key message. Inductive biases based on certain variational norms, such as the $\mathcal{R}$-norm, are believed to offer a way around the curse of dimensionality suffered by kernel methods because they are adaptive to low-dimensional structures. Researchers have pointed to this adaptivity property in non-parametric settings (Bach, 2017; Parhi and Nowak, 2021b) and specific learning tasks with low-dimensional structure (Wei et al., 2019) as mathematical evidence of the statistical advantage of neural networks over kernel methods. One may hypothesize that the $\mathcal{R}$-norm inductive bias achieves this advantage by favoring functions with low-dimensional structure. Indeed, many other forms of inductive bias used in statistics and machine learning are known to explicitly identify relevant low-dimensional structure (Candès, Romberg, and Tao, 2006; Donoho, 2006; Candès and Recht, 2009; Bhojanapalli, Neyshabur, and Srebro, 2016; Barak et al., 2022; Damian, Lee, and Soltanolkotabi, 2022; Frei, Chatterji, and Bartlett, 2022; Mousavi-Hosseini et al., 2022; Galanti et al., 2022). Our results provide theoretical evidence that this is not always the case with the $\mathcal{R}$-norm inductive bias and that this inability becomes more pronounced in higher dimensions.

We show that, even in cases where the dataset can be perfectly fit by an intrinsically one-dimensional function, the solutions g to (VP) or (ϵ-VP) are not necessarily the piecewise-linear ridge functions described in previous works (Savarese et al., 2019; Hanin, 2021). Rather, the $\mathcal{R}$-norm is far better minimized by a *multi-directional*[1] neural network g that averages several ridge functions pointing in different directions, each of which approximates a small fraction of the data.

4.1.1 Our contributions

Our results are summarized by the following informal theorems concerning the structural and generalization properties of $\mathcal{R}$-norm interpolation. Together, they show that the $\mathcal{R}$-norm inductive bias (1) leads to interpolants that are qualitatively different from those that minimize the width or intrinsic dimensionality of the learned network, and (2) is insufficient

[1]By a multi-directional function, we mean a function that does not *only* depend on a one-dimensional projection of its input—i.e., a function that is not a ridge function (defined in Section 4.2.1).

for obtaining optimal generalization for a well-studied learning problem.

Informal Theorem 4.1 ($\mathcal{R}$-norm minimizers of the parity dataset are not ridge functions)**.** *Suppose the dataset* $\{(x_i, y_i)\}_{i\in[n]} \subset \{-1,1\}^d \times \{-1,1\}$ *used in* (VP) *and* (ϵ-VP) *is the complete dataset of* 2^d *examples labeled by the* d*-variable parity function.*

- *The optimal value of* (VP) *is* $\Theta(d)$*.*
- *The optimal value of* (ϵ-VP) *for any* $\epsilon \in [0, 1/2)$*—with the additional constraint that* g *be a ridge function—is* $\Theta(d^{3/2})$*.*

This result is presented formally in Section 4.3. In Section 4.3.1, we show that every ridge function satisfying the constraints of (ϵ-VP) has $\mathcal{R}$-norm at least $\Omega(d^{3/2})$; this bound is tight for ridge functions, as there is a matching upper bound. Using an averaging strategy, we show in Section 4.3.2 the existence of multi-directional interpolants g of the parity dataset with $\|g\|_{\mathcal{R}} = O(d)$, and we also establish the optimality of this construction in Section 4.3.3. These results characterize the optimal value of (VP) in terms of the dimension d, and also establish the $\mathcal{R}$-norm-suboptimality of ridge function interpolants. (In Section 4.5, we extend the averaging strategy to other types of target functions, expanding the scope of our structural findings.)

Informal Theorem 4.2 (Min-$\mathcal{R}$-norm interpolation is sub-optimal for learning parities)**.** *Suppose the dataset* $\{(\mathbf{x}_i, \chi(\mathbf{x}_i))\}_{i\in[n]} \subset \{-1,1\}^d \times \{-1,1\}$ *used in* (VP) *is an i.i.d. sample, where* $\mathbf{x}_i \sim \mathrm{Unif}(\{-1,1\}^d)$ *is labeled by the* d*-variable parity function* χ *for all* $i \in [n]$*. If the sample size is* $n = o(d^2/\sqrt{\log d})$*, then with probability at least* $1/2$*, every solution to* (VP) *has mean squared error at least* $1 - o(1)$ *for predicting* χ *over* $\mathrm{Unif}(\{-1,1\}^d)$*.*

This result is presented formally in Section 4.4.1, and it is complemented by a sample complexity upper bound in Section 4.4.2. The results are stated for the parity function on all d variables, but the same holds for any parity function over $\Omega(d)$ variables. It is well-known that an i.i.d. sample of size $O(d)$ is sufficient for learning parity functions exactly

(Helmbold, Sloan, and Warmuth, 1992; Fischer and Simon, 1992), and hence we conclude that the $\mathcal{R}$-norm inductive bias is insufficient for achieving the statistically optimal sample complexity for this learning problem.

4.1.2 Related work

Variational norms and inductive biases of optimization methods. Many variational norms (such as $\mathcal{R}$-norm) from functional analysis can be regarded as representational costs that induce topologies in the space of infinitely wide neural networks with certain activation functions. Prior works have analyzed these norms for homogeneous activation functions like ReLU (e.g., Kurková and Sanguineti, 2001; Mhaskar, 2004; Bach, 2017; Savarese et al., 2019; Ongie et al., 2019); see Siegel and Xu (2021) and references therein for a comparison. In particular, the work of Ongie et al. (2019) provided analytical descriptions of $\mathcal{R}$-norm in terms of the Radon transform of the function itself. This work was extended to higher powers of ReLU by Parhi and Nowak (2021a).

The variational norms are also connected to the implicit biases of optimization methods for training neural networks. In the context of univariate functions, the dynamics of gradient descent were shown to be biased towards (adaptive) linear or cubic spline depending on the optimization regime (Williams et al., 2019; Shevchenko, Kungurtsev, and Mondelli, 2021; Maennel, Bousquet, and Gelly, 2018), and these results have been partially extended to the multivariate case (Jin and Montúfar, 2020). For classification problems, the implicit bias of gradient descent was connected to a variational problem related to $\mathcal{R}$-norm with margin constraints on the data (Bach and Chizat, 2021).

Solutions to the variational problem. Debarre et al. (2022) and Hanin (2021) fully characterized the form of all solutions of (VP) for one-dimensional datasets (as discussed above). However, pinning down even a single solution for general multidimensional datasets appears to be difficult; Ergen and Pilanci (2021) was able to do so for rank-one datasets,

where all the feature vectors lie on a line. The datasets we study do not satisfy the rank-one condition of Ergen and Pilanci (2021), and thus we require different techniques to analyze multi-directional functions.

Adaptivity. In the context of non-parametric regression, it is well-known that (deep) neural networks achieve minimax-optimal rates in the presence of low-dimensional structure in the target function (e.g., Schmidt-Hieber, 2020; Bauer and Kohler, 2019; Kohler and Krzyżak, 2005; Györfi et al., 2002). The convergence rates in these works depend only on the intrinsic dimension of the target function (and not the ambient dimension) and are achieved by optimally trading off accuracy and regularization in certain deep neural network architectures. Recent works (Klusowski and Barron, 2016; Parhi and Nowak, 2021b; Bach, 2017) consider two-layer neural networks with variational norm (similar to $\mathcal{R}$-norm) regularization, which also allows for adaptivity to low-dimensional structures. That is, a function $g\colon \mathbb{R}^d \to \mathbb{R}$ depending only on a k-dimensional projection of its input x, i.e., $g(x) = \phi(Ux)$ for some $U \in \mathbb{R}^{k\times d}$ (with orthonormal rows) and $\phi\colon \mathbb{R}^k \to \mathbb{R}$ has variational norm no greater than that of the corresponding low-dimensional function ϕ (Bach, 2017). In particular, Bach (2017) and Klusowski and Barron (2016) studied minimax rates under ridge target functions where $k = 1$. Our results on generalization are of a different flavor: rather than striking a careful balance between fitting and regularization to achieve minimax rates, we study the behavior of $\mathcal{R}$-norm-minimizing interpolation.

Regularization based on weight decay (equivalent to $\mathcal{R}$-norm for shallow networks) has also been used to obtain minimax rates for learning smooth target functions. Parhi and Nowak (2021b) do so by drawing analogies to spline theory, while Wang and Lin (2021) consider a connection to the Group Lasso. Zhang and Wang (2022) exploits depth to promote stronger sparsity regularizes. This is distinct from the low-dimensional structures studied in this work and mentioned above.

Learning ridge functions and parity functions with neural nets. Target functions that depend on low-dimensional projections of the input (of which ridge functions are the simplest case) have been long studied in statistics (see, e.g., Li, 2018), and learning such functions is one of the simplest problems where neural network training demonstrates adaptivity. Such demonstrations typically require going beyond the neural tangent kernel regime and have been used to explain the "feature learning" ability of neural networks (Frei, Chatterji, and Bartlett, 2022; Damian, Lee, and Soltanolkotabi, 2022; Mousavi-Hosseini et al., 2022; Bietti et al., 2022). Several recent works have considered the prospects of learning (sparse) parity functions by training neural nets with gradient-based algorithms (Abbe and Sandon, 2020; Daniely and Malach, 2020; Malach et al., 2021a; Barak et al., 2022; Telgarsky, 2022). The positive results express parities as low-weight linear combinations of (random) ReLUs, which motivates our focus on the variational norm of approximating neural nets. Our sample complexity lower bound shows that, even if computational and optimization considerations are set aside, the inductive bias imposed by the $\mathcal{R}$-norm may lead to suboptimal statistical performance.

Averaging and ensembling. Neural networks have been interpreted as forms of averaging or ensemble methods to explain their statistical properties (e.g., Bartlett, 1996; Baldi and Sadowski, 2013; Gal and Ghahramani, 2016; Olson, Wyner, and Berk, 2018). Our use of averaging is distinguished by its use as an approximation technique for achieving smaller $\mathcal{R}$-norm.

Weight lower bounds for other explicit functions. Representation costs for two-layer neural networks to approximate other explicit functions have been explored in several prior works (Martens et al., 2013; Daniely, 2017b; Safran and Shamir, 2017; Safran, Eldan, and Shamir, 2019). These works establish exponential lower bounds on the width of two-layer networks needed to approximate functions represented more compactly by three-layer networks. These results also imply lower bounds on the size of second-layer weights in a

two-layer network after fixing the width of the network. In contrast, our results hold for networks of unbounded width and for a target function that can be exactly represented by two-layer networks of poly(d) width.

4.2 Preliminaries

4.2.1 Notation

In this work, we consider real-valued functions over the radius-$\sqrt{d}$ Euclidean ball

$$\mathbb{B}^d = \left\{x \in \mathbb{R}^d : \|x\|_2 \le \sqrt{d}\right\}.$$

Let $\mathrm{Par}_S \colon \mathbb{B}^d \to \{-1, 1\}$ denote the multi-linear monomial $\mathrm{Par}_S(x) := \prod_{i \in S} x_i$ over variables indexed by $S \subseteq [d]$, and let $\mathrm{Par} := \mathrm{Par}_{[d]}$. On input $x \in \{-1, 1\}^d$, $\mathrm{Par}_S(x)$ computes the *parity* of $\{x_i : i \in S\}$.

We say $g \colon \mathbb{B}^d \to \mathbb{R}$ is a *ridge function* if $g(x) = \phi(v^\mathsf{T} x)$ for some unit vector $v \in \mathbb{S}^{d-1}$ and function $\phi \colon [-\sqrt{d}, \sqrt{d}] \to \mathbb{R}$. A function $\phi : \mathbb{R} \to \mathbb{R}$ is *ρ-periodic* if $\phi(z + \rho) = \phi(z)$ for all $z \in \mathbb{R}$. We say that $g : \mathbb{B}^d \to \mathbb{R}$ is *k-index* if there exists a matrix $U \in \mathbb{R}^{k \times d}$ and $\phi : \mathbb{R}^k \to \mathbb{R}$ such that $g(x) = \phi(Ux)$ for all $x \in \mathbb{B}^d$. (A ridge function is 1-index.)

For a matrix $M \in \mathbb{R}^{m \times n}$, we denote the i-th largest singular value of M by $\sigma_i(M)$ for $i = 1, \ldots, \min\{m, n\}$.

4.2.2 Neural networks and $\mathcal{R}$-norm

We consider two-layer neural networks (of infinite and finite width) with ReLU activations $\mathrm{ReLU}(z) := \max\{0, z\}$. Let $\mathcal{M}$ denote the space of signed measures over $\mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}]$. For $\mu \in \mathcal{M}$, let $g_\mu : \mathbb{B}^d \to \mathbb{R}$ denote the infinite-width neural network given by

$$g_\mu(x) := \int_{\mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}]} \mathrm{ReLU}(w^\mathsf{T} x + b)\, \mu(\mathrm{d}w, \mathrm{d}b).$$

The total variation norm of μ is

$$|\mu| := \int_{\mathbb{S}^{d-1}\times[-\sqrt{d},\sqrt{d}]} |\mu|\,(\mathrm{d}w, \mathrm{d}b),$$

where $|\mu|\,(\mathrm{d}w, \mathrm{d}b)$ is the corresponding total variation measure (somewhat abusing notation). The *bias-corrected network* $\bar{g}_\mu$ is given by $\bar{g}_\mu(x) := g_\mu(x) - g_\mu(0)$; equivalently,

$$\bar{g}_\mu(x) = \int (\mathrm{ReLU}(w^\mathsf{T} x + b) - \mathrm{ReLU}(b))\,\mu(\mathrm{d}w, \mathrm{d}b).$$

The width-m neural network g_θ with parameters $\theta = (a^{(j)}, w^{(j)}, b^{(j)})_{j\in[m]} \in (\mathbb{R}\times\mathbb{S}^{d-1}\times[-\sqrt{d},\sqrt{d}])^m$ is given by

$$g_\theta(x) := \sum_{j=1}^m a^{(j)}\mathrm{ReLU}(w^{(j)\mathsf{T}} x + b^{(j)}).$$

We regard g_θ as an infinite-width neural network with the "sparse" atomic measure

$$\mu_\theta = \sum_{j=1}^m a^{(j)}\delta_{(w^{(j)},b^{(j)})}.$$

Observe that $g_\theta = g_{\mu_\theta}$ and $|\mu_\theta| = \sum_{j=1}^m |a^{(j)}| = \|a\|_1$.

Our constructions frequently use *sawtooth functions*, a family of ridge functions that are composed of $t+1$ repetitions of a triangular wave that draw inspiration from a construction of Yehudai and Shamir (2019, Proposition 4.2). For $t \in \{0,\dots d\}$ with $t \equiv d \pmod 2$ and $w \in \{-1,1\}^d$, let

$$s_{w,t}(x) := (-1)^{d-t}\mathrm{Par}(w)\phi_t(w^\mathsf{T} x),$$

where $\phi_t : \mathbb{R}\to\mathbb{R}$ is a function that forms a piecewise affine interpolation between the points

$$\{(-t-1, 0)\} \cup \{(t-2\tau, (-1)^\tau) : \tau \in \{0,\dots,t\}\} \cup \{(t+1, 0)\},$$

and $\phi_t(z) = 0$ for all $z \le -t-1$ and $z \ge t+1$. We refer to t as the *width* of the sawtooth function $s_{w,t}$. Note that $s_{w,t}$ is $\sqrt{d}$-Lipschitz and $s_{w,t}(x) = \mathrm{Par}(x)\mathbb{1}\{\left|w^\mathsf{T}x\right| \le t\}$ for all $x \in \{-1,1\}^d$. Also, $s_{w,t}$ can be expressed as a neural network g_θ with width $m \le O(t+1)$ and $|a^{(i)}| \le O(\sqrt{d})$ for each $i \in [m]$.

Let $\nu := \mathrm{Unif}(\{-1,1\}^d)$ denote the uniform distribution on $\{-1,1\}^d$, and let $\boldsymbol{\nu}_n$ denote the empirical distribution on $\mathbf{x}_1, \dots, \mathbf{x}_n \sim_{\mathrm{iid}} \nu$. We use the following inner products and norms over the vector space of real-valued functions on $\{-1,1\}^d$ with respect to a distribution ν_0 (such as ν or $\boldsymbol{\nu}_n$):

$$\langle g, h\rangle_{L^2(\nu_0)} := \mathop{\mathbb{E}}_{\mathbf{x}\sim\nu_0}[g(\mathbf{x})h(\mathbf{x})], \quad \|g\|_{L^2(\nu_0)} := \langle g, g\rangle^{1/2}_{L^2(\nu_0)}, \quad \|g\|_{L^\infty(\nu_0)} := \max_{x\in\mathrm{supp}(\nu_0)} |g(x)|.$$

4.2.2.1 $\mathcal{R}$-norm and the variational problem

We now recall the definition of the $\mathcal{R}$-norm of a function $g\colon \mathbb{B}^d \to \mathbb{R}$, presented here in a variational form as the minimum cost of representing g as an infinite-width neural network with a "skip-connection":

$$\|g\|_\mathcal{R} := \inf_{\mu\in\mathcal{M}, v\in\mathbb{R}^d, c\in\mathbb{R}} |\mu| \quad \text{s.t.} \quad g(x) = g_\mu(x) + v^\mathsf{T}x + c \ \ \forall x \in \mathbb{B}^d. \tag{$\mathcal{R}$-norm}$$

Indeed, $\|\cdot\|_\mathcal{R}$ is a semi-norm on the space of functions with finite $\mathcal{R}$-norm. It was initially introduced by Ongie et al. (2019) along with explicit characterizations in terms of the Radon transform. See the works of Ongie et al. (2019), Parhi and Nowak (2021a), and Siegel and Xu (2021) for more discussion about the $\mathcal{R}$-norm and its connections to other function spaces.

The appearance of the affine component $v^\mathsf{T}x + c$ in the definition of $\mathcal{R}$-norm has implications for how the bias terms are treated. Notice that a neuron $x \mapsto \mathrm{ReLU}(w^\mathsf{T}x + b)$ with bias $|b| \ge \sqrt{d}$ behaves as an affine function over the domain of interest $\mathbb{B}^d$, so it can be absorbed into the "free" affine component (in the definition of $\mathcal{R}$-norm) so as to not be counted towards the $\mathcal{R}$-norm. Other works (e.g., Siegel and Xu, 2021) consider a different variational

norm, $\|\cdot\|_{\mathcal{V}_2}$, which does not have "free" affine components, but instead permits biases b to be in the larger range $[-2\sqrt{d}, 2\sqrt{d}]$. These differences in the way affine components are accommodated do not lead to different function spaces (see Parhi and Nowak, 2021b, Theorem 6), and the results of this paper for $\mathcal{R}$-norm also hold for these other variational norms, as we demonstrate in Section 4.6.

Although the $\mathcal{R}$-norm is defined in terms of an infimum, it has been shown by Parhi and Nowak (2021b, Lemma 2) that the infimum is always achieved by a particular signed measure $\mu \in \mathcal{M}$.

Proposition 4.3 (Lemma 2 in Parhi and Nowak, 2021b)**.** *For any $g : \mathbb{B}^d \to \mathbb{R}$ with $\|g\|_{\mathcal{R}} < \infty$, there exists an even Radon measure[2] μ over $\mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}]$, and $v \in \mathbb{R}^d, c \in \mathbb{R}$, such that g admits an integral of the form*

$$g(x) = \int_{\mathbb{S}^{d-1}\times[-\sqrt{d}\times\sqrt{d}]} \mathrm{ReLU}(w^\mathsf{T} x + b)\, \mu(\mathrm{d}w, \mathrm{d}b) + v^\mathsf{T} x + c \quad \forall x \in \mathbb{B}^d.$$

Moreover, μ attains the ($\mathcal{R}$-norm)*, i.e., $\|g\|_{\mathcal{R}} = |\mu|$.*

Since the total variation norm is sparsity-inducing, the objective in (VP) favors finite-width networks. It can be shown, using an extension of Caratheodory's theorem (Rosset et al., 2007), that (VP) in fact always has a finite-width solution. That is, (VP) is solved by the sum of an affine function $x \mapsto v^\mathsf{T} x + c$ and a width-m neural network, for some $m \leq \max\{0, n-(d+1)\}$. The following theorem of Parhi and Nowak (2021b) formalizes the fact that the $\mathcal{R}$-norm-minimizing interpolant of a d-dimensional dataset can be represented as a finite-width neural network. Thus, considering finite-width neural networks is sufficient to determine the value of (VP).[3]

Theorem 4.4. *For any dataset $\{(x_i, y_i)\}_{i\in[n]}$ from $\mathbb{B}^d \times \mathbb{R}$, the infimum in* (VP) *is achieved*

[2]Evenness of μ should interpreted in the distributional sense, but it roughly means $\mu(w, b) = \mu(-w, -b)$ when μ has a density.

[3]We note that the finite-width solution to (VP) is not necessarily unique; Hanin (2021) discusses this issue in the one-dimensional case ($d = 1$) under general data models.

by the sum of an affine function $x \mapsto v^\mathsf{T} x + c$ and a finite-width neural network g of the form

$$g(x) = \sum_{j=1}^{m} a^{(j)} \mathrm{ReLU}(w^{(j)} x + b^{(j)}) + v^\mathsf{T} x + c,$$

with $m \leq \max\{0, n-(d+1)\}$ and $(w_j, b_j) \in \mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}]$ for all $i \in [m]$.

Proof. By Theorem 5 of Parhi and Nowak (2021b) (see also the proof of Theorem 1 of Parhi and Nowak (2021a) which covers the interpolation form of the optimization problem), there exists a neural network $x \mapsto \sum_{j=1}^{m'} a^{(j)} \mathrm{ReLU}(w^{(j)\mathsf{T}} x + b^{(j)})$ of width $m' \leq n-(d+1)$, and an affine function $x \mapsto v^{(0)\mathsf{T}} x + c^{(0)}$, such that their sum achieves the infimum in (VP). We can divide neurons of the neural network into two sets based on whether their corresponding bias term is smaller or larger than $\sqrt{d}$ in absolute value. Since every $x \in \mathbb{B}^d$ satisfies $\|x\|_2 \leq \sqrt{d}$, without loss of generality (by possibly flipping the sign of some $a^{(j)}$ and $w^{(j)}$), assume the first m neurons satisfy $\left|b^{(j)}\right| \leq \sqrt{d}$ and the rest satisfy $b^{(j)} > \sqrt{d}$. Then we have

$$\begin{aligned}
g(x) &= \sum_{j=1}^{m} a^{(j)} \mathrm{ReLU}(w^{(j)\mathsf{T}} x + b^{(j)}) + \sum_{j=m+1}^{m'} a^{(j)} \mathrm{ReLU}(w^{(j)\mathsf{T}} x + b^{(j)}) + v^{(0)\mathsf{T}} x + c^{(0)} \\
&= \sum_{j=1}^{m} a^{(j)} \mathrm{ReLU}(w^{(j)\mathsf{T}} x + b^{(j)}) + \sum_{j=m+1}^{m'} a^{(j)} (w^{(j)\mathsf{T}} x + b^{(j)}) + v^{(0)\mathsf{T}} x + c^{(0)} \\
&= \sum_{j=1}^{m} a^{(j)} \mathrm{ReLU}(w^{(j)\mathsf{T}} x + b^{(j)}) + \underbrace{\left(v^{(0)} + \sum_{j=m+1}^{m'} a^{(j)} w^{(j)}\right)^\mathsf{T}}_{=:v} x + \underbrace{\sum_{j=m+1}^{m'} b^{(j)} + c^{(0)}}_{=:c}.
\end{aligned}$$

Therefore, g has the desired form with $m \leq m' \leq n-(d+1)$. □

The following lemma, which is a minor elaboration on Lemma 25 of Parhi and Nowak (2021a), relates the $\mathcal{R}$-norm of a finite-width network to the ℓ_1-norm of its top-layer weights.

Lemma 4.5. *Let $v \in \mathbb{R}^d$, $c \in \mathbb{R}$, and $\theta = (a^{(j)}, w^{(j)}, b^{(j)})_{j \in [m]} \in (\mathbb{R} \times \mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}])^m$ be the set of parameters of a finite neural network where $(w^{(i)}, b^{(i)}) \neq (w^{(j)}, b^{(j)})$ for all $i \neq j$.*

(i) The $\mathcal{R}$-norm of the sum of g_θ and an affine function $v^\mathsf{T}x + c$ satisfies

$$\left\|g_\theta(x) + v^\mathsf{T}x + c\right\|_\mathcal{R} \leq \|a\|_1 = |a^{(1)}| + \cdots + |a^{(m)}|. \tag{4.1}$$

(ii) Moreover, if the measure μ_θ is even in a distributional sense, that is

$$\mu_\theta(w, b) = \mu_\theta(-w, -b),$$

then the inequality in (4.1) *holds with equality.*

Note that our assumption that μ_θ is even in Lemma 4.5(ii) precludes the case where $a^{(i)} = -a^{(j)}$ and $(w^{(i)}, b^{(i)}) = (-w^{(j)}, -b^{(j)})$ for some $i \neq j$. This is needed because if such a case were allowed, we would have $a^{(i)}\mathrm{ReLU}(w^{(i)\mathsf{T}}x + b^{(i)}) + a^{(j)}\mathrm{ReLU}(w^{(j)\mathsf{T}}x + b_j) = a^{(i)}(w^{(i)\mathsf{T}}x + b^{(i)})$ for all $x \in \mathbb{B}^d$—an affine function. After ruling out these cases, we can apply the argument of Parhi and Nowak (2021a) to prove Lemma 4.5(ii).

4.2.2.2 $\mathcal{R}$-norm of ridge functions

Prior works illuminate precise formulations of the $\mathcal{R}$-norm, and characterize solutions to (VP), albeit only for the one-dimensional setting (Hanin, 2021; Savarese et al., 2019; Ergen and Pilanci, 2021). These results are nevertheless useful for analyzing ridge functions in d-dimensional space.

Theorem 4.6. *For any ridge function $g\colon \mathbb{B}^d \to \mathbb{R}$ of the form $g(x) = \phi(w^\mathsf{T}x)$ where $w \in \mathbb{S}^{d-1}$ and $\phi\colon [-\sqrt{d}, \sqrt{d}] \to \mathbb{R}$ is Lipschitz, we have*

$$\|g\|_\mathcal{R} = \|\phi'\|_{\mathrm{TV}} \coloneqq \operatorname*{ess\,sup}_{-\sqrt{d}\leq t_0<t_1<\cdots<t_r\leq\sqrt{d};\, r\in\mathbb{N}} \sum_{i=1}^{r} \left|\phi'(t_i) - \phi'(t_{i-1})\right|,$$

where ϕ' is a right continuous derivative of ϕ.[4]

[4]Take $\phi'(u) = \lim_{t\downarrow 0} \frac{\phi(u+t)-\phi(u)}{t}$; the limit exists almost everywhere by Rademacher's theorem.

Remark 4.1. *If ϕ is twice differentiable, then $\|g\|_{\mathcal{R}} = \int_{-\sqrt{d}}^{\sqrt{d}} |\phi''(u)| \, \mathrm{d}u = \|\phi''\|_1$. Intuitively, this ℓ_1-norm penalty induces sparsity in the second derivative, leading to representations that use few neurons. In contrast, minimizing the ℓ_2-norm penalty $\|\phi''\|_2$ on the second derivative yields a cubic spline (Kimeldorf and Wahba, 1971).*

Proof. Without loss of generality, g only depends on the first coordinate x_1 due to the invariance of the $\mathcal{R}$-norm to rotation (cf. Proposition 11 of Ongie et al., 2019). The result then follows from Remark 4 of Parhi and Nowak (2021b). □

This bound on the $\mathcal{R}$-norm for ridge functions (and univariate functions) is critical for analyses of the solutions to (VP) for $d = 1$ (Hanin, 2021; Savarese et al., 2019). It suggests a potential approach for our high-dimensional setting: project the dataset to every one-dimensional subspace, interpolate the data with a ridge function that points in that direction, directly compute the $\mathcal{R}$-norm of each using Theorem 4.6, and return the ridge function with the lowest $\mathcal{R}$-norm. In the sequel, we examine the optimality of this approach, and find that ridge functions *cannot* be optimal solutions to (VP), even when the dataset can be perfectly fit by a ridge function.

4.2.3 Concentration inequalities

Our proofs make extensive use of textbook probability concentration inequalities. We provide those results below.

A random variable $\mathbf{u}$ is *c-subgaussian* if $\|\mathbf{u}\|_{\psi_2} := \inf\{t \geq 0 : \mathbb{E}[\exp(\mathbf{u}^2/t^2)] \leq 2\} \leq c$, and a random vector $\mathbf{v}$ is *σ^2-subgaussian* if every one-dimensional projection of $\mathbf{v}$ is c-subgaussian.

Lemma 4.7 (Hoeffding's inequality; Theorem 2.8 in Boucheron, Lugosi, and Massart, 2013)**.** *Let $\mathbf{u}_1, \ldots, \mathbf{u}_n$ be independent, mean-zero random variables such that $\mathbf{u}_i$ takes value in $[a_i, b_i]$ almost surely for all $i \in [n]$. Then, for any $t > 0$,*

$$\Pr\left[\sum_{i=1}^{n} \mathbf{u}_i \geq t\right] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

Lemma 4.8 (Multiplicative Chernoff bound; Theorem 4.4 in Mitzenmacher and Upfal, 2017)**.** *Let* $\mathbf{u}_1, \ldots, \mathbf{u}_n$ *be independent Bernoulli random variables with* $\Pr[\mathbf{u}_i = 1] = p \in [0,1]$ *for all* $i \in [n]$*. Then, for any* $\eta \in (0,1]$*,*

$$\Pr\left[\sum_{i=1}^{n} \mathbf{u}_i \geq (1+\eta)p\right] \leq \exp\left(-\frac{p\eta^2}{3}\right).$$

Lemma 4.9 (Bernstein's inequality; Corollary 2.11 in Boucheron, Lugosi, and Massart, 2013)**.** *Let* $\mathbf{u}_1, \ldots, \mathbf{u}_n$ *be independent, mean-zero random variables with* $\mathbf{u}_i \leq K$ *almost surely for all* $i \in [n]$*, and let* $v \coloneqq \sum_{i=1}^{n} \mathbb{E}[\mathbf{u}_i^2]$*. Then, for any* $t > 0$*,*

$$\Pr\left[\sum_{i=1}^{n} \mathbf{u}_i \geq t\right] \leq \exp\left(-\frac{t^2}{2(v + Kt/3)}\right).$$

Lemma 4.10 (McDiarmid's inequality; Theorem 6.2 in Boucheron, Lugosi, and Massart, 2013)**.** *Let* $\mathbf{u}_1, \ldots, \mathbf{u}_n$ *be independent random variables, and let* f *be a measurable function. Suppose, for each* $i \in [n]$*, the value of* $f(u_1, \ldots, u_n)$ *can change by at most* $c_i \geq 0$ *by changing the value of* u_i*. Then, for any* $t > 0$*,*

$$\Pr\left[f(\mathbf{u}_1, \ldots, \mathbf{u}_n) - \mathbb{E}\left[f(\mathbf{u}_1, \ldots, \mathbf{u}_n)\right] \geq t\right] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{n} c_i^2}\right).$$

Lemma 4.11 (Properties of subgaussian random variables)**.** *Let* $\mathbf{u}_1, \ldots, \mathbf{u}_n$ *be independent random variables with* $\|\mathbf{u}_i\|_{\psi_2} < \infty$ *for all* $i \in [n]$*. There is a universal constant* $C > 0$ *such that the following hold.*

(i) (Concentration; Section 2.5.2 in Vershynin, 2018) For any $t > 0$*,* $\Pr[|\mathbf{u}_1| \geq t] \leq 2\exp(-t^2/(C\|\mathbf{u}_1\|_{\psi_2}))$*.*

(ii) (Maximum; Exercise 2.5.10 in Vershynin, 2018)

$$\mathbb{E}\left[\max_{i\in[n]} |\mathbf{u}_i|\right] \leq C\sqrt{\ln n} \max_{i\in[n]} \|\mathbf{u}_i\|_{\psi_2}.$$

(iii) (Averaging: Proposition 2.6.1 in Vershynin, 2018) $\|\sum_{i=1}^n \mathbf{u}_i\|_{\psi_2}^2 \leq C \sum_{i=1}^n \|\mathbf{u}_i\|_{\psi_2}^2$.

(iv) (Centering; Lemma 2.6.8 in Vershynin, 2018) $\|\mathbf{u}_1 - \mathbb{E}[\mathbf{u}_1]\|_{\psi_2} \leq C \|\mathbf{u}_1\|_{\psi_2}$.

(v) (Lipschitzness) For any 1*-Lipschitz function* $\phi\colon \mathbb{R} \to \mathbb{R}$ *with* $\phi(0) = 0$, $\|\phi(\mathbf{u}_1)\|_{\psi_2} \leq \|\mathbf{u}_1\|_{\psi_2}$.

Proof. The only property not already proved in (Vershynin, 2018) is (v). Since ϕ is 1-Lipschitz and $\phi(0) = 0$,

$$|\phi(\mathbf{u}_1)| = |\phi(\mathbf{u}_1) - \phi(0)| \leq |\mathbf{u}_1 - 0| = |\mathbf{u}_1|.$$

Hence

$$\mathbb{E}\left[\exp(\phi(\mathbf{u}_1)^2/\|\mathbf{u}_1\|_{\psi_2}^2)\right] \leq \mathbb{E}\left[\exp(\mathbf{u}_1^2/\|\mathbf{u}_1\|_{\psi_2}^2)\right] \leq 2,$$

which implies $\|\phi(\mathbf{u}_1)\|_{\psi_2} \leq \|\mathbf{u}_1\|_{\psi_2}$. □

Lemma 4.12 (Singular values of random matrices; Theorem 4.6.1 in Vershynin, 2018)**.** *There is a positive constant* $C > 0$ *such that the following holds. Let* $\mathbf{A}$ *be a* $m \times n$ *random matrix whose rows are independent, mean-zero,* v*-subgaussian random vectors in* $\mathbb{R}^n$*. For any* $t \geq 0$*, with probability at least* $1 - 2e^{-t}$*, the singular values* $\sigma_1(\mathbf{A}), \sigma_2(\mathbf{A}), \ldots, \sigma_n(\mathbf{A})$ *of* $\mathbf{A}$ *satisfy*

$$\sqrt{m} - Cv(\sqrt{n} + \sqrt{t}) \leq \sigma_i(\mathbf{A}) \leq \sqrt{m} + Cv(\sqrt{n} + \sqrt{t}) \quad \textit{for all } i \in [n].$$

Lemma 4.13 (MGF of Unif($\{-1, 1\}$); Lemma 2.2 in Boucheron, Lugosi, and Massart, 2013)**.** *If* $\mathbf{u} \sim \text{Unif}(\{-1, 1\})$*, then* $\mathbb{E}[\exp(t\mathbf{u})] \leq \exp(t^2/2)$ *for all* $t \in \mathbb{R}$.

4.2.4 Covering numbers

Let $\mathcal{N}(\epsilon, A, \gamma)$ denote the *covering number* over a metric space A with metric $\gamma : A \times A \to \mathbb{R}_+$. That is, $\mathcal{N}(\epsilon, A, d) = \min |\mathcal{N}_\epsilon|$ for $\mathcal{N}_\epsilon \subset A$ such that for all $x \in A$, there exists $x' \in \mathcal{N}_\epsilon$

with $\gamma(x, x') \leq \epsilon$. Note that $\mathcal{N}(\epsilon, [-1, 1], |\cdot|) \leq \frac{2}{\epsilon}$.

Lemma 4.14 (Covering numbers of $\mathbb{S}^{d-1}$; Corollary 4.2.13 in Vershynin, 2018)**.** *For any* $\epsilon > 0$, $\mathcal{N}(\epsilon, \mathbb{S}^{d-1}, \|\cdot\|_2) \leq (\frac{2}{\epsilon} + 1)^d$. *If* $\epsilon \in [0, 1]$, *then* $\mathcal{N}(\epsilon, \mathbb{S}^{d-1}, \|\cdot\|_2) \leq (\frac{3}{\epsilon})^d$.

4.3 Intrinsic dimensionality of solutions to the variational problem for parity

In this section, we study the $\mathcal{R}$-norm of neural networks that solve VP or ϵ-VP for the (full) *parity dataset* $\{(x, \mathrm{Par}(x)) : x \in \{-1, 1\}^d\}$, which has size $n = 2^d$. For simplicity, the labels are provided by the parity function Par over all d variables, although the same quantitative results (up to constant factor differences) hold for any Par_S with $|S| = \Theta(d)$.

The high level message is that, despite the fact that this dataset can be exactly fit using ridge functions, the solutions to (VP) and (ϵ-VP) are *not* ridge functions and instead must be multi-directional.

4.3.1 Every ridge parity interpolant has $\mathcal{R}$-norm $\Omega(d^{3/2})$

We first show that any ϵ-approximate interpolant of the parity dataset *that is also a ridge function* must have $\mathcal{R}$-norm $\Omega(d^{3/2})$. This lower-bound is established even for $\epsilon = 1/2$.

Theorem 4.15. *For* $d \geq 2$, *let* Ridge_d *be the set of functions* $g \colon \mathbb{B}^d \to \mathbb{R}$ *such that* $g(x) = \phi(w^\mathsf{T} x)$ *for some* $w \in \mathbb{S}^{d-1}$ *and Lipschitz continuous* $\phi \colon [-\sqrt{d}, \sqrt{d}] \to \mathbb{R}$. *Then*

$$\inf\{\|g\|_\mathcal{R} : g \in \mathrm{Ridge}_d,\ \|g - \chi\|_{L^\infty(\nu)} \leq 1/2\} \geq d^{3/2}/(2\sqrt{2}).$$

This lower bound in is tight up to constant factors, because the sawtooth function $s_{\vec{1},d}$ satisfies the constraints of (VP) and has $\left\|s_{\vec{1},d}\right\|_\mathcal{R} = O(d^{3/2})$.

Its proof constructs a labeled dataset of $d + 1$ points, and shows that any ridge function $g(x) = \phi(w^\mathsf{T} x)$ that approximates that dataset must have many high-magnitude oscillations. These oscillations imply a lower bound on $\|\phi'\|_{\mathrm{TV}}$, which proves the claim by way of Theorem 4.6.

Proof. Take any $g \in \mathrm{Ridge}_d$ of the form $g(x) = \phi(w^\mathsf{T} x)$ for some function ϕ and vector w satisfying the approximation constraint $\|g - \mathrm{Par}\|_{L^\infty(\nu)} \le 1/2$. Suppose for sake of contradiction that $w_i = 0$ for some $i \in [d]$. Then, there exists a pair of points $x, x' \in \{-1, 1\}^d$ that are identical except in the i-th positions, x_i and x'_i. Thus, $\mathrm{Par}(x) = -\mathrm{Par}(x')$, but $w^\mathsf{T} x = w^\mathsf{T} x'$ and hence $g(x) = g(x')$; this contradicts the approximation constraint. So, we may henceforth assume that $w_i \neq 0$ for all $i \in [d]$.

For each $i \in \{0, 1, \ldots, d\}$, define

$$x^{(i)} := (\mathrm{sign}(w_1), \ldots, \mathrm{sign}(w_i), -\mathrm{sign}(w_{i+1}), \ldots, -\mathrm{sign}(w_d)).$$

Because the parity of $x^{(i)}$ alternates with i, i.e., $\mathrm{Par}(x^{(i)}) \neq \mathrm{Par}(x^{(i+1)})$, $\mathrm{sign}(g(x^{(i)}))$ also alternates because g satisfies the approximation constraint. Furthermore, again due to the approximation constraint, we have $|g(x^{(i)}) - g(x^{(i+1)})| \ge 1$. We claim that, because ϕ interpolates $d+1$ well-separated data points $(w^\mathsf{T} x^{(i)}, \phi(w^\mathsf{T} x^{(i)}))$ that satisfy $w^\mathsf{T} x^{(i)} < w^\mathsf{T} x^{(i+1)}$ for all $i \in \{0, 1, \ldots, d-1\}$, there must be a large cost for representing ϕ using a neural network. By Theorem 4.6, it suffices to obtain a lower bound on $\|\phi'\|_{\mathrm{TV}}$, since this will imply a lower bound on $\|g\|_{\mathcal{R}}$.

By Lemma 4.16 (a modification of the mean value theorem, which is presented following the proof) for every $i \in \{0, 1, \ldots, d-1\}$, there exists $A_i \subseteq [w^\mathsf{T} x^{(i)}, w^\mathsf{T} x^{(i+1)}]$ with Lebesgue measure $\mathrm{Leb}(A_i) > 0$ such that, for every $z^{(i)} \in A_i$, we have

$$|\phi'(z^{(i)})| \ge \frac{1}{2} \cdot \frac{|\phi(w^\mathsf{T} x^{(i+1)}) - \phi(w^\mathsf{T} x^{(i)})|}{w^\mathsf{T} x^{(i+1)} - w^\mathsf{T} x^{(i)}},$$

and $\mathrm{sign}\phi'(z^{(i)}) = \mathrm{sign}\phi(w^\mathsf{T} x^{(i+1)}) - \phi(w^\mathsf{T} x^{(i)})$. The fact that the signs of $\phi(w^\mathsf{T} x^{(i)})$ alternate with i implies that the signs of $\phi'(z^{(i)})$ also alternate with i. We now lower-bound the total variation of ϕ' using the fact that $\prod_{i=1}^d \mathrm{Leb}(A_i) > 0$ and taking advantage of the alternating

signs:

$$
\begin{aligned}
2\left\|\phi'\right\|_{\mathrm{TV}} &= 2 \operatorname*{ess\,sup}_{-\sqrt{d}\le t_0<t_1<\cdots<t_r\le\sqrt{d};\, r\in\mathbb{N}} \sum_{i=1}^{r} |\phi'(t_i)-\phi'(t_{i-1})| \\
&\ge 2\sum_{i=1}^{d-1} |\phi'(z^{(i)})-\phi'(z^{(i-1)})| = 2\sum_{i=1}^{d-1}(|\phi'(z^{(i)})|+|\phi'(z^{(i-1)})|) \ge 2\sum_{i=0}^{d-1}|\phi'(z^{(i)})| \\
&\ge \sum_{i=0}^{d-1} \frac{|\phi(w^{\mathsf{T}}x^{(i+1)})-\phi(w^{\mathsf{T}}x^{(i)})|}{w^{\mathsf{T}}x^{(i+1)}-w^{\mathsf{T}}x^{(i)}} \ge \sum_{i=0}^{d-1}\frac{1}{w^{\mathsf{T}}x^{(i+1)}-w^{\mathsf{T}}x^{(i)}} \\
&\ge \frac{d^2}{\sum_{i=0}^{d-1} w^{\mathsf{T}}x^{(i+1)}-w^{\mathsf{T}}x^{(i)}} = \frac{d^2}{w^{\mathsf{T}}x^{(d)}-w^{\mathsf{T}}x^{(0)}} \ge \frac{d^2}{\|w\|_2\|x^{(d)}-x^{(0)}\|_2} = \frac{d^{3/2}}{\sqrt{2}}.
\end{aligned}
$$

The second-to-last inequality is a consequence of Cauchy-Schwarz: for any $a_1,\dots,a_d>0$, $d^2=(\sum_i\sqrt{a_i}/\sqrt{a_i})^2\le(\sum_i a_i)(\sum_i 1/a_i)$. Therefore, $\|g\|_{\mathcal{R}}=\|\phi'\|_{\mathrm{TV}}\ge d^{3/2}/(2\sqrt{2})$. □

The proof of Theorem 4.15 employs the following lemma, which is essentially a robust variant of the mean value theorem.

Lemma 4.16. *Suppose $\phi:\mathbb{R}\to\mathbb{R}$ is Lipschitz continuous on the interval $[t_1,t_2]$ with $\phi(t_1)\ne\phi(t_2)$. Define*

$$
A := \Bigg\{ t\in[t_1,t_2] : \phi'(t) \text{ exists}, \\
\qquad |\phi'(t)| \ge \frac{1}{2}\cdot\frac{|\phi(t_2)-\phi(t_1)|}{t_2-t_1},\ \operatorname{sign}(\phi'(t))=\operatorname{sign}(\phi(t_2)-\phi(t_1)) \Bigg\}.
$$

(The factor $1/2$ in the definition of A can be replaced by any constant in $(0,1)$.) Then, $\mathrm{Leb}(A)>0$, *where* Leb *is the Lebesgue measure.*

Proof. Recall that ϕ' denotes the right continuous derivative of g (or the right-hand Dini derivative) which is guaranteed to exist except on a null set by Rademacher's theorem. Let $s:=\operatorname{sign}\phi(t_2)-\phi(t_1)$. By the assumption $\phi(t_1)\ne\phi(t_2)$ and the Fundamental Theorem of Calculus, we have

$$
0<|\phi(t_2)-\phi(t_1)| = s\left(\phi(t_2)-\phi(t_1)\right) = \int_{t_2}^{t_1} s\phi'(z)\,\mathrm{d}z \le (t_2-t_1)\operatorname*{ess\,sup}_{z\in[t_1,t_2]} s\phi'(z).
$$

Recall that, by definition,

$$\operatorname*{ess\,sup}_{z\in[t_1,t_2]} s\phi'(z) = \inf\left\{a : \mathrm{Leb}(\{z \in [t_1,t_2] : \phi'(z) \text{ exists},\ s\phi'(z) > a\}) = 0\right\},$$

and thus

$$B := \{z \in [t_1,t_2] : \phi'(z) \text{ exists},\ s(\phi(t_2)-\phi(t_1)) \le 2\cdot(t_2-t_1)s\phi'(z)\}$$

satisfies $\mathrm{Leb}(B) > 0$. Observe that $B = A$, so $\mathrm{Leb}(A) > 0$, concluding the proof. □

4.3.2 Existence of a multi-directional parity interpolant with $\mathcal{R}$-norm $O(d)$

We now show that the $\Omega(d^{3/2})$ $\mathcal{R}$-norm lower-bound from Theorem 4.15 for ridge functions can be avoided by neural networks that are not ridge functions. The main idea is to employ an *averaging strategy* that combines a collection of distinct ridge functions, each of which perfectly fits a small fraction of the parity dataset—those on the "equator" relative to the ridge direction—while ignoring the "outliers" in that direction. Because all points on the cube are "outliers" for some directions and on the "equator" for others, this strategy ultimately ensures that every example is perfectly fit.

Theorem 4.17. *For any even*[5] *d, there exists a neural network $g : \mathbb{B}^d \to \mathbb{R}$ having $g(x) = \mathrm{Par}(x)$ for all $x \in \{-1,1\}^d$ such that $\|g\|_{\mathcal{R}} \le O(d)$.*

Proof. Recall that the sawtooth function $s_{w,0} : \mathbb{B}^d \to \mathbb{R}$ satisfies

$$s_{w,0}(x) = \mathrm{Par}(x)\mathbb{1}\left\{w^{\mathsf{T}}x = 0\right\}$$

for all $x \in \{-1,1\}^d$. By construction, $s_{w,0}$ is a ridge function that is a single "bump" around zero in the direction of w, and $\|s_{w,0}\|_{\mathcal{R}} \le O(\sqrt{d})$. Consider $\mathbf{w} \sim \mathrm{Unif}(\{-1,1\}^d)$. By sym-

[5] Our results also hold for odd d, but the proofs are more tedious.

metry, $\Pr\left[\mathbf{w}^\mathsf{T} x = 0\right] = \Pr\left[\mathbf{w}^\mathsf{T} x' = 0\right]$ for all $x, x' \in \{-1,1\}^d$, so

$$\begin{aligned}\mathbb{E}\left[s_{\mathbf{w},0}(x)\right] &= \mathrm{Par}(x) \cdot \Pr\left[\mathbf{w}^\mathsf{T} x = 0\right] \\ &= \mathrm{Par}(x) \cdot 2^{-d} \cdot \left|\{v \in \{-1,1\}^d : v^\mathsf{T} x = 0\}\right| \\ &= \mathrm{Par}(x) \cdot q,\end{aligned}$$

where $q := \binom{d}{d/2}/2^d = \Theta(1/\sqrt{d})$. Define $g(x) := \frac{1}{q2^d} \sum_{w \in \{-1,1\}^d} s_{w,0}(x)$. Then

$$g(x) = \frac{1}{q}\,\mathbb{E}\left[s_{\mathbf{w},0}(x)\right] = \mathrm{Par}(x)$$

for each $x \in \{-1,1\}^d$, i.e., g interpolates the parity dataset. Finally, we bound the $\mathcal{R}$-norm:

$$\|g\|_{\mathcal{R}} \le \frac{1}{q2^d} \sum_{w \in \{-1,1\}^d} \|s_{w,0}\|_{\mathcal{R}} \le \frac{1}{q} \cdot O(\sqrt{d}) \le O(d). \qquad \square$$

While Theorem 4.17 successfully exhibits a neural network g that perfectly fits the parity dataset with $\|g\|_{\mathcal{R}} = O(d)$, the width of g is $\Omega(2^d)$. We next show that by allowing non-zero $L^\infty(\nu)$ error in the approximation, we can achieve a construction with both $O(d)$ $\mathcal{R}$-norm and poly(d) width.

Theorem 4.18. *There is a universal constant $c > 0$ such that the following holds. For any even d, any $\epsilon \in (0,1)$, and any even $t \in \{0, 2, \ldots, d\}$, there exists a function $g\colon \mathbb{B}^d \to \mathbb{R}$ that can be represented by a width-m neural network such that $\|g - \mathrm{Par}\|_{L^\infty(\nu)} \le \epsilon$, where*

$$\begin{aligned}&m \le O(d^{3/2}\sqrt{\log(1/\epsilon)}/\epsilon^2) \quad &&\text{and} \quad \|g\|_{\mathcal{R}} \le O(d\log(1/\epsilon)) \quad &&\text{if } t \le c\sqrt{d\log(1/\epsilon)}; \\ &m \le O(d^2/(\epsilon t)) &&\text{and} \quad \|g\|_{\mathcal{R}} \le O(t\sqrt{d}) &&\text{otherwise.}\end{aligned}$$

Moreover, g can be expressed as a linear combination of width-t sawtooth functions.

Remark 4.2. *Suppose ϵ is a constant. Using $t = \Theta(d)$, we obtain a neural network of width $m = O(d)$ and $\|g\|_{\mathcal{R}} = O(d^{3/2})$, matching the properties of the sawtooth (ridge function)*

interpolant $s_{w,d}$. Using $t = \Theta(1)$, we obtain a neural network of width $m = O(d^{3/2})$ and $\|g\|_{\mathcal{R}} = O(d)$, matching the properties of the interpolant from Theorem 4.17 but with almost exponentially smaller width.

4.3.2.1 Proof of Theorem 4.18

A more detailed version of Theorem 4.18—which also specifies the *intrinsic dimensionality* of g—is stated and proved in below. The proof uses a similar technique as that of Theorem 4.17, but instead averages randomly sampled sawtooth functions $s_{\mathbf{w}^{(1)},t}, \ldots, s_{\mathbf{w}^{(k)},t}$ for $\mathbf{w}^{(j)} \sim \text{Unif}(\{-1,1\}^d)$ of width t. We show that for sufficiently large k, every $x \in \{-1,1\}^d$ lies the in the "active" region of about the same number of sawtooth functions; this yields a good approximation of $\text{Par}(x)$ for all x.

Theorem 4.19 (Detailed version of Theorem 4.18)**.** *There exists a universal constant C such that for any even d, even sawtooth width $t \in \{0, 2, \ldots d\}$, and accuracy $\epsilon \in (0, \frac{1}{2})$, there exists a k-index width-m neural network g with $\|g - \text{Par}\|_{L^\infty(\nu)} \le \epsilon$ such that:*

1. *If $t \le C\sqrt{d \ln \frac{1}{\epsilon}}$, then $k = O(\frac{d^{3/2}}{t+1} \cdot \frac{\log^{1/2} \epsilon}{\epsilon^2})$, $m = O(d^{3/2} \frac{\log^{1/2} \epsilon}{\epsilon^2})$, and $\|g\|_{\mathcal{R}} = O(d \log \frac{1}{\epsilon})$.*
2. *Otherwise, $k = O(\frac{d^2}{\epsilon t^2})$, $m = O(\frac{d^2}{\epsilon t})$, and $\|g\|_{\mathcal{R}} = O(t\sqrt{d})$.*

Proof. For $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(k)} \sim_{\text{iid}} \text{Unif}(\{-1,1\}^d)$ and

$$q := \Pr\left[\left|\mathbf{w}^{(1)\mathsf{T}} x\right| \le t\right], \text{ for any } x \in \{-1,1\}^d,$$

let

$$\mathbf{g}(x) := \frac{1}{kq} \sum_{j=1}^{k} s_{\mathbf{w}^{(j)},t}(x).$$

Because $\mathbb{E}\left[s_{\mathbf{w},t}(x)\right] = q \cdot \text{Par}(x)$, we have $\mathbb{E}\left[\mathbf{g}(x)\right] = \text{Par}(x)$. Following the arguments in the proof of Theorem 4.17, we have $\|\mathbf{g}\|_{\mathcal{R}} = O(\frac{(t+1)\sqrt{d}}{q})$, and $\mathbf{g}$ has width $O(k(t+1))$.

It remains to place a lower bound on q and to show that with non-zero probability, $\mathbf{g}$ uniformly approximates Par. By applying a union bound, it suffices to show that

$\Pr\left[|\mathbf{g}(x) - \mathrm{Par}(x)| \geq \epsilon\right] \leq \frac{1}{2^{d+1}}$ for any fixed $x \in \{-1, 1\}^d$.

For fixed x, let

$$\mathbf{r}(x) := |\{j \in [k] : |x^{\mathsf{T}}\mathbf{w}^{(j)}| > t\}|$$

denote the number of sawtooth functions $s_{\mathbf{w}^{(j)},t}$ that are inactive at x. We upper-bound the accuracy of approximation of $\mathrm{Par}(x)$ by $\mathbf{g}(x)$ in terms of $\mathbf{r}(x)$:

$$\begin{aligned}
|\mathbf{g}(x) - \mathrm{Par}(x)| &= \left| \frac{1}{qk} \sum_{j=1}^{k} \mathbb{1}\{|x^{\mathsf{T}}\mathbf{w}^{(j)}| \leq t\} - q \right| \\
&= \left| \frac{(k - \mathbf{r}(x))(1-q)}{qk} - \frac{\mathbf{r}(x)q}{qk} \right| \\
&= \left| \frac{1-q}{q} - \frac{\mathbf{r}(x)}{qk} \right|.
\end{aligned}$$

Fix threshold

$$T := 2\left\lceil \sqrt{(d/2)\ln(8/\epsilon)} \right\rceil,$$

and note that $\Pr\left[|\mathbf{w}^{\mathsf{T}}x| \geq T\right] \leq \frac{\epsilon}{4}$. The proof naturally divides into two cases, depending on the value of t.

Case 1: $t \leq T$**.** We first lower-bound q. Because $\mathbf{w}^{\mathsf{T}}x$ is a shifted symmetric binomial distribution around $\mathbf{w}^{\mathsf{T}}x = 0$, if $|t'| \geq |t|$ and $t' \equiv t \pmod 2$, then $\Pr\left[\mathbf{w}^{\mathsf{T}}x = t'\right] \leq$

$\Pr\left[\mathbf{w}^\mathsf{T} x = t\right]$. Then, for any $t \le T$:

$$\begin{aligned}
q &= \sum_{\tau=-t/2}^{t/2} \Pr\left[\mathbf{w}^\mathsf{T} x = 2\tau\right] \\
&= (t+1) \cdot \frac{1}{t+1} \sum_{\tau=-t/2}^{t/2} \Pr\left[\mathbf{w}^\mathsf{T} x = 2\tau\right] \\
&\ge (t+1) \cdot \frac{1}{T+1} \sum_{\tau=-T/2}^{T/2} \Pr\left[\mathbf{w}^\mathsf{T} x = 2\tau\right] \\
&= \frac{t+1}{T+1} \Pr\left[|\mathbf{w}^\mathsf{T} x| \le T\right] \\
&\ge \frac{(t+1)(1-\frac{\epsilon}{2})}{2\sqrt{d \ln \frac{4}{\epsilon}}} \\
&\ge \frac{t+1}{4\sqrt{d \ln \frac{4}{\epsilon}}}.
\end{aligned}$$

Now, we bound $\mathbf{r}(x)$ by Bernstein's inequality (Lemma 4.9) by taking $k \ge \frac{Cd^{3/2}\sqrt{\ln \frac{1}{\epsilon}}}{\epsilon^2(t+1)}$:

$$\begin{aligned}
\Pr\left[|\mathbf{g}(x) - \mathrm{Par}(x)| > \epsilon\right] &= \Pr\left[\left|\mathbf{r}(x) - \mathbb{E}[\mathbf{r}(x)]\right| > \epsilon q k\right] \\
&\le 2\exp\left(-\frac{\epsilon^2 q^2 k^2}{2(kq(1-q) + \epsilon q k/3)}\right) \\
&\le 2\exp\left(-\frac{\epsilon^2 k(t+1)}{8(1+\epsilon/3)\sqrt{d \ln \frac{4}{\epsilon}}}\right) \le \frac{1}{2^{d+1}}.
\end{aligned}$$

Case 2: $t \ge T$**.** By Hoeffding's inequality (Lemma 4.7) and the assumption on t, we have

$$q \ge 1 - 2\exp(-\frac{2t^2}{d}) \ge 1 - \frac{\epsilon}{4} \ge \frac{3}{4}.$$

Observe that $\mathbb{E}\left[\mathbf{r}(x)\right] = (1-q)k \le \frac{\epsilon k}{4}$.

We show that

$$|\mathbf{g}(x) - \mathrm{Par}(x)| = |\frac{1-q}{q} - \frac{\mathbf{r}(x)}{qk}| \le \epsilon$$

by showing that $\mathbf{r}(x) \le (1-q)k + \epsilon q k$ and $\mathbf{r}(x) \ge (1-q)k - \epsilon q k$. Because $1 - q \le \frac{\epsilon}{4}$ and $q \ge \frac{3}{4}$, $(1-q)k - \epsilon q k \le -\frac{\epsilon}{2}k$, so the second inequality is always satisfied because $\mathbf{r}(x) \ge 0$. For the

former inequality, it suffices to show that $\mathbf{r}(x) \leq \frac{3\epsilon k}{4}$ with probability at least $1 - 2^{-(d+1)}$. We take $k \geq \frac{Cd^2}{\epsilon t^2}$, which implies that

$$k \geq C\left(\frac{d}{2\epsilon} + \frac{de^{-2t^2/d}}{2\epsilon^2}\right) \geq C\left(\frac{d}{2\epsilon} + \frac{d(1-q)}{4\epsilon^2}\right)$$

by the bounds on t and q. Then, by Bernstein's inequality (Lemma 4.9), we have

$$\begin{aligned}
\Pr\left[|\mathbf{g}(x) - \mathrm{Par}(x)| > \epsilon\right] &= \Pr\left[\left|\mathbf{r}(x) - \mathbb{E}[\mathbf{r}(x)]\right| > \frac{3\epsilon k}{4}\right] \\
&\leq 2\exp\left(-\frac{9\epsilon^2 k^2/16}{2(kq(1-q) + \epsilon k/4)}\right) \\
&\leq \frac{1}{2^{d+1}},
\end{aligned}$$

so the claim follows. □

4.3.3 Every parity interpolant has $\mathcal{R}$-norm $\Omega(d)$

Finally, we show that $\mathcal{R}$-norm upper-bounds from Theorems 4.17 and 4.18 are tight. That is, we show that every solution to (ϵ-VP) for the parity dataset has $\mathcal{R}$-norm $\Omega(d)$, even for constant ϵ. This is implied by the following stronger result, which requires only $L^2(\nu)$ approximation, as opposed to $L^\infty(\nu)$.

Theorem 4.20. *For any $d \geq 8$ and $\alpha \in (0,1)$, $\inf\{\|g\|_{\mathcal{R}} : \|g - \mathrm{Par}\|_{L^2(\nu)} \leq 1 - \alpha\} \geq \alpha d/8$.*

A result analogous to Theorem 4.20 also holds for most *sampled parity datasets* (defined in Section 4.4). This result is stated and proved in Section 4.3.3.1.

The core of proof strategy is to bound the correlation of any fixed ReLU neuron with the parity function Par.

Proof. Consider any measure μ over $\mathbb{S}^{d-1} \times [-2\sqrt{d}, 2\sqrt{d}]$, $v \in \mathbb{R}^d$, and $c \in \mathbb{R}$ such that $g(x) = g_\mu(x) + v^\mathsf{T}x + c = \int_{\mathbb{S}^{d-1}\times\mathbb{R}} \mathrm{ReLU}(w^\mathsf{T}x + b)\mu(\mathrm{d}w, \mathrm{d}b)$ for all $\|x\|_2 \leq d$. We prove the claim by showing that $|\mu| \geq \frac{\alpha d}{8}$ for any such μ.

By Fact 4.21, $\|g - \mathrm{Par}\|_{L^2(\nu)} \le 1 - \alpha$ implies that $\langle g, \mathrm{Par}\rangle \ge \alpha$. We show that this inner product bound is only possible if $|\mu|$ is sufficiently large. By Lemma 4.22, any fixed neuron $r_{w,b}(x) := \mathrm{ReLU}(w^\mathsf{T} x + b)$ has $|\langle r_{w,b}, \mathrm{Par}\rangle| \le \frac{8}{d}$. Because the inner-product over $\{-1,1\}^d$ is a discrete sum and Par is orthogonal to any affine function (such as $x \mapsto v^\mathsf{T} x + c$), we can upper-bound the ability of g to correlate with Par as follows:

$$\begin{aligned}
\langle g, \mathrm{Par}\rangle_{L^2(\nu)} &= \int_{\mathbb{S}^{d-1}\times\mathbb{R}} \langle r_{w,b}, \mathrm{Par}\rangle_{L^2(\nu)}\, \mu(\mathrm{d}w, \mathrm{d}b) + \mathbb{E}\left[(v^\mathsf{T}\mathbf{x} + c)\mathrm{Par}(\mathbf{x})\right] \\
&\le \int_{\mathbb{S}^{d-1}\times\mathbb{R}} \left|\langle r_{w,b}, \mathrm{Par}\rangle_{L^2(\nu)}\right| |\mu(\mathrm{d}w, \mathrm{d}b)| \\
&\le \frac{8}{d} \int_{\mathbb{S}^{d-1}\times\mathbb{R}} |\mu(\mathrm{d}w, \mathrm{d}b)| = \frac{8\,|\mu|}{d}.
\end{aligned}$$

Thus, $\langle g_\mu, \mathrm{Par}\rangle \ge \alpha$ only if $|\mu| \ge \frac{\alpha d}{8}$. □

Fact 4.21. *For any measure ν_0 over $\{-1,1\}^d$, $g \in L^2(\nu_0)$, $h : \{-1,1\}^d \to \{-1,1\}$, and $\alpha \in (0,1)$, if $\|g - h\|_{L^2(\nu_0)} \le 1 - \alpha$, then $\langle g, h\rangle_{L^2(\nu_0)} \ge \alpha$.*

Proof. The claim is a consequence of the fact $\langle h, h\rangle_{L^2(\nu_0)} = 1$ and Cauchy-Schwarz:

$$\begin{aligned}
\langle g, h\rangle_{L^2(\nu_0)} &= \langle h, h\rangle_{L^2(\nu_0)} + \langle g - h, h\rangle_{L^2(\nu_0)} \\
&= 1 + \langle g - h, h\rangle_{L^2(\nu_0)} \\
&\ge 1 - \|g - h\|_{L^2(\nu_0)} \\
&\ge 1 - (1 - \alpha) = \alpha.
\end{aligned}$$

□

Lemma 4.22. *For $d \ge 8$, $w \in \mathbb{R}^d$ with $\|w\|_2 \le 1$, and $b \in [-2\sqrt{d}, 2\sqrt{d}]$, the neuron $r_{w,b}(x) := \mathrm{ReLU}(w^\mathsf{T} x + b)$ satisfies $|\langle r_{w,b}, \mathrm{Par}\rangle| \le \frac{8}{d}$.*

Remark 4.3. *Lemma 4.22 is asymptotically tight. For even d, consider the "single-blade" sawtooth function*

$$s_{\vec{1},0}(x) = \sqrt{d}(r_{\vec{1}/\sqrt{d},1/\sqrt{d}}(x) - 2r_{\vec{1}/\sqrt{d},0}(x) + r_{\vec{1}/\sqrt{d},-1/\sqrt{d}}(x))$$

that satisfies $s_{\vec{1},0}(x) = \mathrm{Par}(x)\mathbb{1}\{\vec{1}^\mathsf{T}x = 0\}$. *Then,*

$$\langle s_{\vec{1},0}, \mathrm{Par}\rangle_{L^2(\nu)} = \frac{1}{2^d}\binom{d}{d/2} \geq \frac{1}{\sqrt{2d}},$$

and thus there exists b *with* $|\langle r_{\vec{1}/\sqrt{d},b}, \mathrm{Par}\rangle_{L^2(\nu)}| \geq \frac{1}{4\sqrt{2d}}$.

Proof. We directly bound the inner product by showing that we can bound a discrete second derivative. For any $x \in \{-1,1\}^d$, let $x^j \in \{-1,1\}^d$ denote x with a flipped jth bit. That is, $x_i^j = (-1)^{\mathbb{1}\{i=j\}}x_i$. Observe that $\mathrm{Par}(x) = -\mathrm{Par}(x^j)$.

$$\begin{aligned}
&|\langle r_{w,b}, \mathrm{Par}\rangle| \\
&\quad= \frac{1}{2^d}\left|\sum_x r_{w,b}(x)\mathrm{Par}(x)\right| \\
&\quad= \frac{1}{4\cdot 2^d}\left|\sum_x (r_{w,b}(x)\mathrm{Par}(x) + r_{w,b}(x^j)\mathrm{Par}(x^j) + r_{w,b}(x^{j'})\mathrm{Par}(x^{j'}) + r_{w,b}(x^{j,j'})\mathrm{Par}(x^{j,j'}))\right| \\
&\quad= \frac{1}{4\cdot 2^d}\left|\sum_x \mathrm{Par}(x)(r_{w,b}(x) - r_{w,b}(x^j) - r_{w,b}(x^{j'}) + r_{w,b}(x^{j,j'}))\right| \\
&\quad\leq \frac{1}{4\cdot 2^d}\sum_x |r_{w,b}(x) - r_{w,b}(x^j) - r_{w,b}(x^{j'}) + r_{w,b}(x^{j,j'})|.
\end{aligned}$$

We say that (x, x^j) is *cut* and denote $(x, x^j) \in C_j$ if x and x^j lie on the opposite side of the "hinge" of the neuron $r_{w,b}$, that is $\mathrm{sign}(w^\mathsf{T}x - b) \neq \mathrm{sign}(w^\mathsf{T}x^j - b)$. Let $S_{x,j,j'} = \{x, x^j, x^{j'}, x^{j,j'}\}$ represent a "square" in $\{-1,1\}^d$, and let $S_{x,j,j'} \in C_{j,j'}$ if any of its edges $(x, x^j), (x, x^{j'}), (x^j, x^{j,j'}), (x^{j'}, x^{j,j'})$ are cut. We bound the term inside the sum by considering two cases.

1. If $S_{x,j,j'} \notin C_{j,j'}$, then $|r_{w,b}(x) - r_{w,b}(x^j) - r_{w,b}(x^{j'}) + r_{w,b}(x^{j,j'})| = 0$.

2. Otherwise, the quantity is bounded by Lipschitzness:

$$\begin{aligned}
&|r_{w,b}(x) - r_{w,b}(x^j) - r_{w,b}(x^{j'}) + r_{w,b}(x^{j,j'})| \\
&\quad \le |r_{w,b}(x) - r_{w,b}(x^j)| + |r_{w,b}(x^{j'}) - r_{w,b}(x^{j,j'})| \\
&\quad \le |w^\mathsf{T} x - w^\mathsf{T} x^j| + |w^\mathsf{T} x^{j'} - w^\mathsf{T} x^{j,j'}| = 4|w_j|.
\end{aligned}$$

Therefore, $|\langle r_{w,b}, \mathrm{Par}\rangle| \le \min_{j \neq j'} \frac{1}{2^d} |C_{j,j'}|\,|w_j|$. It remains to bound $|C_{j,j'}|$ and $|w_j|$ for some j and j'. By employing a bound on the total number of cut edges (O'Neil, 1971):

$$\frac{1}{d}\sum_{j=1}^{d} |C_j| \le \frac{1}{2d} \cdot \left\lceil \frac{d}{2} \right\rceil \binom{d}{\lfloor d/2 \rfloor} \le \frac{2^d}{2\sqrt{d}}.$$

As a result, at most $\frac{d}{2}$ choices of j satisfy $|C_j| \ge 2^d/\sqrt{d}$. Because $\|w\|_2 \le 1$, at most $\frac{d}{4}$ coordinates j have $|w_j| \ge 2/\sqrt{d}$. Thus, there exist at *least* $\frac{d}{4}$ coordinates j satisfying both $|C_j| \le 2^d/\sqrt{d}$ and $|w_j| \le 2/\sqrt{d}$. Assuming $d \ge 8$, let j, j' be two of those coordinates. Since $|C_{j,j'}| \le 2|C_j| + 2|C_{j'}|$, we conclude that $|C_{j,j'}| \le 4 \cdot 2^d/\sqrt{d}$, which gives the desired bound on the inner product. □

4.3.3.1 $\mathcal{R}$-norm lower bound for sampled parity datasets

Theorem 4.23. *Fix any $\delta \in (0,1)$ and $\alpha = \omega(1/d)$, and assume $n \ge O(d^3(\log d + \log(1/\delta)))$. With probability at least $1-\delta$, $\inf\{\|g\|_\mathcal{R} : \|g - \mathrm{Par}\|_{L^2(\boldsymbol{\nu}_n)} \le 1-\alpha\} \ge \Omega(\alpha d)$.*

Proof. Let g be a function with finite $\mathcal{R}$-norm which satisfies the $L^2(\boldsymbol{\nu}_n)$ approximability condition, which admits an integral representation due to Proposition 4.3. That is,

$$g(x) = \int_{\mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}]} (\mathrm{ReLU}(w^\mathsf{T} x + b) - \mathrm{ReLU}(b))\mu(\mathrm{d}w, \mathrm{d}b) + c + v^\mathsf{T} x \quad \forall x \in \mathbb{B}^d$$

for some measure μ and $v \in \mathbb{R}^d, c = g(0)$. Moreover, g can be represented compactly as $g(x) = \bar{g}_\mu(x) + v^\mathsf{T} x + c$ where $\bar{g}_\mu(x) = g_\mu(x) - g_\mu(0)$.

By Fact 4.21, $\|g - \mathrm{Par}\|_{L^2(\boldsymbol{\nu}_n)} \leq 1-\alpha$ only if $\langle g, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)} \geq \alpha$. We use this correlation to prove lower bounds on $|\mu|$ (the total variation of measure μ). At a high level, we upper-bound

$$\langle g, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)} = \langle \bar{g}_\mu, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)} + \left\langle v^\mathsf{T} x + c, \mathrm{Par}(x)\right\rangle_{L^2(\boldsymbol{\nu}_n)}$$

in terms of $|\mu|$ by relating quantities in $L^2(\boldsymbol{\nu}_n)$ with their $L^2(\nu)$ counterparts. We show that each component of the sum is small for sufficiently large n and d.

We first bound the correlation of the linear combination of neurons with parity, proving upper bounds on $\langle g, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)}$. We denote $\bar{r}_{w,b}(x) = \mathrm{ReLU}(w^\mathsf{T} x + b) - \mathrm{ReLU}(b)$ to be the adjusted ReLU. By the triangle inequality,

$$\begin{aligned}\langle \bar{g}_\mu, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)} &\leq \int_{\mathbb{S}^{d-1}\times[-\sqrt{d},\sqrt{d}]} \left|\langle \bar{r}_{w,b}, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)}\right| |\mu|\,(dw, db) \\ &\leq |\mu| \sup_{w\in\mathbb{S}^{d-1}, b\in[-\sqrt{d},\sqrt{d}]} \left|\langle \bar{r}_{w,b}, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)}\right|.\end{aligned}$$

Lemmas 4.22 and 4.26 together bound the correlation of any neuron $\bar{r}_{w,b}$ with Par. That is, for any $w \in \mathbb{S}^{d-1}$ and $b \in [-\sqrt{d}, \sqrt{d}]$, with probability at least $1 - \delta/3$:

$$\left|\langle \bar{r}_{w,b}, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)}\right| \leq \left|\langle \bar{r}_{w,b}, \mathrm{Par}\rangle_{L^2(\nu)}\right| + C_1\sqrt{\frac{d(\ln n + \ln(3/\delta))}{n}} \leq \frac{8}{d} + 2C_1\sqrt{\frac{d\ln n}{n}} \leq \frac{C_2}{d},$$

where C_1 is the constant from Lemma 4.26 and $n > C(d^3(\log d + \log(1/\delta)))$ by assumption.

We now show that the linear components cannot be substantially correlated with the parity function and bound $\left\langle v^\mathsf{T} x + c, \mathrm{Par}\right\rangle_{L^2(\boldsymbol{\nu}_n)}$. Because no linear term correlates with the full parity dataset, Lemma 4.25 provides an upper bound on the inner product between the linear perturbation and sampled parity dataset and implies the following bound with

probability at least $1 - \delta/3$:

$$\begin{aligned}\left|\left\langle v^\mathsf{T} x + c, \mathrm{Par}\right\rangle_{L^2(\boldsymbol{\nu}_n)}\right| &\le 8\max\{|\mu|, 1\} \sup_{\substack{|c|\le 1\\ \|v\|_2 \le 1}} \left|\left\langle v^\mathsf{T} x + c, \mathrm{Par}\right\rangle_{L^2(\boldsymbol{\nu}_n)}\right| \\ &= 8\max\{|\mu|, 1\}\left(\left|\frac{1}{n}\sum_{i=1}^n \mathbf{y}_i\right| + \left\|\frac{1}{n}\sum_{i=1}^n \mathbf{y}_i \mathbf{x}_i\right\|_2\right).\end{aligned}$$

By Lemma 4.24 and our assumptions on n, we bound the two data-dependent terms with probability at least $1 - \frac{\delta}{3}$ for some absolute constant C_2:

$$\begin{aligned}\left|\left\langle v^\mathsf{T} x + c, \mathrm{Par}\right\rangle_{L^2(\boldsymbol{\nu}_n)}\right| &\le 8\max\{|\mu|, 1\}\left(\sqrt{\frac{2\ln(12/\delta)}{n}} + 2\sqrt{\frac{d}{n}}\right) \\ &\le \frac{C_2}{d}\max\{|\mu|, 1\}.\end{aligned}$$

Combining both bounds, we have with probability at least $1 - \delta$,

$$\alpha \le \left\langle \bar{g}_\mu(x) + v^\mathsf{T} x + c, \mathrm{Par}\right\rangle_{L^2(\boldsymbol{\nu}_n)} \le \frac{C_2}{d}\left(|\mu| + \max\{|\mu|, 1\}\right) \le \frac{2C_2}{d}\max\{|\mu|, 1\}.$$

Therefore, we conclude

$$|\mu| \ge \frac{\alpha d}{2C_2} - 1. \qquad \square$$

Lemma 4.24. *Fix any $\delta \in (0,1)$. Assume $n \ge O(\log(1/\delta))$ and $n = \omega(d)$, let $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i\in[n]}$ be the sampled parity dataset (where $\mathbf{y}_i = \mathrm{Par}(\mathbf{x}_i)$ for all $i \in [n]$), and let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be the data matrix containing all samples. All of the following hold with probability $1 - \delta$:*

(i) $\left|\frac{1}{n}\sum_{i=1}^n \mathbf{y}_i\right| \le \sqrt{\frac{2\ln(4/\delta)}{n}}$;

(ii) $\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\right\|_2 \le 2\sqrt{\frac{d}{n}}$;

(iii) $\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{y}_i\mathbf{x}_i\right\|_2 \le 2\sqrt{\frac{d}{n}}$; *and*

(iv) $\frac{3}{4}\sqrt{n} \le \sigma_d(\mathbf{X}) \le \sigma_1(\mathbf{X}) \le 2\sqrt{n}$.

Proof. Claim *(i)* holds with probability at least $1 - \frac{\delta}{2}$ as a result of a standard application of Hoeffding's inequality (Lemma 4.7) to a sum of Rademacher random variables.

Claim *(iv)* also holds with probability at least $1 - \frac{\delta}{2}$, since Lemma 4.12 and the assumptions on n imply that

$$\sigma_1(\mathbf{X}) \le \sqrt{n} + C\left(\sqrt{d} + \sqrt{\ln\frac{2}{\delta}}\right) \le 2\sqrt{n}$$

and

$$\sigma_d(\mathbf{X}) \ge \sqrt{n} - C\left(\sqrt{d} - \sqrt{\ln\frac{2}{\delta}}\right) \ge \frac{3}{4}\sqrt{n}.$$

Claims *(ii)* and *(iii)* follow from the singular value bounds on $\mathbf{X}$.

$$\left\|\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\right\|_2 \le \frac{1}{n}\sqrt{\operatorname{tr}(\mathbf{X}^\mathsf{T}\mathbf{X})} \le \frac{1}{n}\cdot 2\sqrt{nd} = 2\sqrt{\frac{d}{n}};$$
$$\left\|\frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i\mathbf{x}_i\right\|_2 \le \frac{1}{n}\sqrt{\mathbf{y}^\mathsf{T}\mathbf{X}\mathbf{X}^\mathsf{T}\mathbf{y}} \le \frac{1}{n}\cdot\sigma_1(\mathbf{X})\sqrt{d} \le 2\sqrt{\frac{d}{n}}.$$ □

Lemma 4.25. *Fix any $\delta \in (0,1)$. Assume $n \ge O(\log(1/\delta))$ and $n = \omega(d)$. With probability at least $1-\delta$ over the random measure $\boldsymbol{\nu}_n$, if $\mu \in \mathcal{M}$ satisfies*

$$\left\|g_\mu(\mathbf{x}) + c + v^\mathsf{T}\mathbf{x} - \mathrm{Par}(\mathbf{x})\right\|_{L^2(\boldsymbol{\nu}_n)} \le 1,$$

then $\max\{|c + g_\mu(0)|, \|v\|_2\} \le 8\max\{|\mu|, 1\}$.

Proof. We draw inspiration from the fact that the full parity dataset is orthogonal to any linear term and can never be well-approximated with large linear components. In other words, the square loss on approximating the full parity dataset with a linear function is minimized by the constant-zero function and strictly worsens as the linear terms increase. That is, orthogonality ensures that $\|c + v^\mathsf{T}x - \mathrm{Par}(x)\|^2_{L^2(\nu)} = 1 + |c|^2 + \|v\|_2^2$. Thus, having an upper bound on the squared error imposes similar upper bounds on the norms of the linear terms. We make a similar argument for the *sampled* parity dataset, where we replace

ν with $\boldsymbol{\nu}_n$.

Without loss of generality, we incorporate $g_\mu(0)$ into c and define $\bar{g}_\mu(x) = g_\mu(x) - g_\mu(0)$ which can be also represented as $\bar{g}_\mu(x) = \int \bar{r}_{w,b}(x)\mu(dw, db)$ where $\bar{r}_{w,b} = \mathrm{ReLU}(w^\mathsf{T}x + b) - \mathrm{ReLU}(b)$. Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be the collection of samples $\mathbf{x}_i$ and let $\mathbf{y}_i = \mathrm{Par}(\mathbf{x}_i)$. We bound the squared loss of the linear component $v^\mathsf{T}x + c$, ignoring the neural network $\bar{g}_\mu$:

$$\begin{aligned}\left\|c + v^\mathsf{T}x - \mathrm{Par}(x)\right\|^2_{L^2(\boldsymbol{\nu}_n)} &= 1 + c^2 + v^\mathsf{T}\left(\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\mathsf{T}\right)v - \frac{2}{n}v^\mathsf{T}\left(\sum_{i=1}^n (\mathbf{y}_i - c)\mathbf{x}_i\right) - \frac{2c}{n}\sum_{i=1}^n \mathbf{y}_i \\ &\geq 1 + c^2 + \frac{1}{n}\left\|v\right\|_2^2 \sigma_d(\mathbf{X})^2 \\ &\quad - 2\left\|v\right\|_2\left(\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{y}_i\mathbf{x}_i\right\|_2 + |c|\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\right\|_2\right) - 2|c|\left|\frac{1}{n}\sum_{i=1}^n \mathbf{y}_i\right|.\end{aligned}$$

With probability $1-\delta$, all events of Lemma 4.24 hold, and we use them to lower-bound the squared loss.

$$\begin{aligned}\left\|c + v^\mathsf{T}x - \mathrm{Par}(x)\right\|^2_{L^2(\boldsymbol{\nu}_n)} &\geq 1 + c^2 + \frac{9}{16}\left\|v\right\|_2^2 - 4\sqrt{\frac{d}{n}}(1 + |c|)\left\|v\right\|_2 - \frac{2\sqrt{2\ln(8/\delta)}}{\sqrt{n}}|c| \\ &\geq \frac{1}{4}\max\{|c|, \left\|v\right\|_2\}^2.\end{aligned}$$

where we have used the assumptions on n and the AM/GM inequality. We now provide upper bounds on the square loss based on measure μ using the triangle inequality:

$$\left\|c + v^\mathsf{T}x - \mathrm{Par}(x)\right\|_{L^2(\boldsymbol{\nu}_n)} \leq \left\|\bar{g}_\mu\right\|_{L^2(\boldsymbol{\nu}_n)} + \left\|\bar{g}_\mu(x) + c + v^\mathsf{T}x - \mathrm{Par}(x)\right\|_{L^2(\boldsymbol{\nu}_n)} \leq \left\|\bar{g}_\mu\right\|_{L^2(\boldsymbol{\nu}_n)} + 1.$$

We may now connect $L^2(\boldsymbol{\nu}_n)$ norm of $\bar{g}_\mu$ to its variational norm. We bound the output of $\bar{g}_\mu$ on a single input $\mathbf{x}_i$ by employing Cauchy-Schwarz:

$$\begin{aligned}\bar{g}_\mu(\mathbf{x}_i)^2 &\leq \left(\int |\bar{r}_{w,b}(\mathbf{x}_i)|\,|\mu|\,(\mathrm{d}w, \mathrm{d}b)\right)^2 \\ &\leq |\mu|\int \bar{r}_{w,b}(\mathbf{x}_i)^2\,|\mu|\,(\mathrm{d}w, \mathrm{d}b).\end{aligned}$$

We sum over all i to bound the norm of $\bar{g}_\mu$:

$$\begin{aligned}\|\bar{g}_\mu(x)\|^2_{L^2(\boldsymbol{\nu}_n)} &\le |\mu| \int \|\bar{r}_{w,b}(x)\|^2_{L^2(\boldsymbol{\nu}_n)} |\mu| \,(\mathrm{d}w, \mathrm{d}b) \le |\mu|^2 \sup_{w\in\mathbb{S}^{d-1}, |b|\le\sqrt{d}} \|\bar{r}_{w,b}\|^2_{L^2(\boldsymbol{\nu}_n)} \\ &\le |\mu|^2 \sup_{w\in\mathbb{S}^{d-1}} \frac{1}{n}\sum_{i=1}^n |w^\mathsf{T}\mathbf{x}_i|^2 = |\mu|^2 \frac{\sigma_1(\mathbf{X})^2}{n} \le 4\,|\mu|^2 .\end{aligned}$$

The second inequality relies on the Lipschitzness of ReLU. Combining all the above,

$$\frac{1}{2}\max\{|c|, \|v\|_2\} \le \left\|c + v^\mathsf{T}x - \mathrm{Par}(x)\right\|_{L^2(\boldsymbol{\nu}_n)} \le 1 + \|g_\mu\|_{L^2(\boldsymbol{\nu}_n)} < 2 + 2\,|\mu| . \qquad \square$$

Lemma 4.26. *For $\bar{r}_{w,b}(x) = \mathrm{ReLU}(w^\mathsf{T}x + b) - \mathrm{ReLU}(b)$ and $n \ge d$, there exists an absolute constant C such that for any $\delta \in (0,1)$ with probability at least $1-\delta$,*

$$\left|\langle \bar{r}_{w,b}, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)}\right| \le \left|\langle \bar{r}_{w,b}, \mathrm{Par}\rangle_{L^2(\nu)}\right| + C\sqrt{\frac{d(\ln n + \ln(1/\delta))}{n}},$$

for all $w \in \mathbb{S}^{d-1}, b \in [-\sqrt{d}, \sqrt{d}]$.

Proof. Observe that the inner product over the sampled parity dataset is an unbiased estimate of the inner product over the full parity dataset,

$$\mathbb{E}\left[\langle \bar{r}_{w,b}, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)}\right] = \langle \bar{r}_{w,b}, \mathrm{Par}\rangle_{L^2(\nu)} .$$

Let $\mathbf{Z}_{w,b}$ denote the deviation from the mean, i.e.

$$\mathbf{Z}_{w,b} = \langle \bar{r}_{w,b}, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)} - \langle \bar{r}_{w,b}, \mathrm{Par}\rangle_{L^2(\nu)} .$$

We use standard concentration of measure techniques for the following steps:

1. $\mathbf{Z}_{w,b}$ is Lipschitz in terms of its parameterization (w,b) in the sense that

$$|\mathbf{Z}_{w_1,b_1} - \mathbf{Z}_{w_2,b_2}| \le 4\sqrt{d}\gamma((w_1,b_1),(w_2,b_2)),$$

where γ is a distance defined later on.

2. $\mathbf{Z}_{w,b}$ is $O(\frac{1}{\sqrt{n}})$-subgaussian for fixed w, b.

3. $\mathbb{E}\left[\sup_{w\in\mathbb{S}^{d-1}, b\in[-\sqrt{d},\sqrt{d}]} |\mathbf{Z}_{w,b}|\right] = O(\sqrt{\frac{d}{n}})$ using a covering argument.

4. The maximum of $|\mathbf{Z}_{w,b}|$ is close to its expectation due to the bounded difference inequality.

(Step 1) Using the fact that ReLU is 1-Lipschitz and triangle inequality,

$$
\begin{aligned}
&|\mathbf{Z}_{w_1,b_1} - \mathbf{Z}_{w_2,b_2}| \\
&\quad \le \left|\langle \bar{r}_{w_1,b_1}, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)} - \langle \bar{r}_{w_2,b_2}, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)}\right| + \left|\langle \bar{r}_{w_1,b_1}, \mathrm{Par}\rangle_{L^2(\nu)} - \langle \bar{r}_{w_2,b_2}, \mathrm{Par}\rangle_{L^2(\nu)}\right| \\
&\quad \le 2\left\|\bar{r}_{w_1,b_1} - \bar{r}_{w_2,b_2}\right\|_{L^\infty(\nu)} \\
&\quad \le 2(\left\|\bar{r}_{w_1,b_1} - \bar{r}_{w_2,b_1}\right\|_{L^\infty(\nu)} + \left\|\bar{r}_{w_2,b_1} - \bar{r}_{w_2,b_2}\right\|_{L^\infty(\nu)}) \\
&\le 2\left(\max_{x\in\{-1,1\}^d} (w_1 - w_2)^\mathsf{T} x + 2\left|b_1 - b_2\right|\right) \\
&\quad \le 4\sqrt{d}\left(\left\|w_1 - w_2\right\|_2 + \frac{|b_1 - b_2|}{\sqrt{d}}\right) =: 4\sqrt{d}\gamma((w_1,b_1),(w_2,b_2)).
\end{aligned}
$$

Thus $\mathbf{Z}_{w,b}$ is $4\sqrt{d}$-Lipschitz with respect to γ.

(Step 2) We bound the subgaussianity of $\mathbf{Z}_{w,b}$.

$$
\begin{aligned}
\left\|\mathbf{Z}_{w,b}\right\|_{\psi_2} &\le C_1 \left\|\langle \bar{r}_{w,b}, \mathrm{Par}\rangle_{L^2(\boldsymbol{\nu}_n)}\right\|_{\psi_2} = C_1 \left\|\sum_{i=1}^n \mathbf{y}_i(\mathrm{ReLU}(w^\mathsf{T}\mathbf{x}_i + b) - \mathrm{ReLU}(b))\right\|_{\psi_2} \\
&\le \frac{C_2}{\sqrt{n}} \left\|\mathbf{y}_1(\mathrm{ReLU}(w^\mathsf{T}\mathbf{x}_1 + b) - \mathrm{ReLU}(b))\right\|_{\psi_2} \\
&\le \frac{C_2}{\sqrt{n}} \left\|\mathrm{ReLU}(w^\mathsf{T}\mathbf{x}_1 + b) - \mathrm{ReLU}(b)\right\|_{\psi_2} \\
&\le \frac{C_2}{\sqrt{n}} \left\|w^\mathsf{T}\mathbf{x}_1\right\|_{\psi_2} \le \frac{2C_2}{\sqrt{n}}
\end{aligned}
$$

The first, second, and fourth inequalities rely on the centering, averaging, and Lipschitzness properties of subgaussian random variables in Lemma 4.11. The third inequality follows

from $|\mathbf{y}_1| = 1$, and the final is due to the 2-subgaussianity of a vector with i.i.d. Rademacher components.

(Step 3) Let $\mathcal{N}_\epsilon$ be an ϵ-covering of $\mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}]$ with respect to γ. We bound its size using the standard ϵ-net result in Lemma 4.14 for $\epsilon \leq 2$.

$$\begin{aligned}\mathcal{N}\left(\epsilon, \mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}], \gamma\right) &\leq \mathcal{N}\left(\frac{\epsilon}{2}, \mathbb{S}^{d-1}, \|\cdot\|_2\right) \times \mathcal{N}\left(\frac{\epsilon}{2}, [-1,1], |\cdot|\right) \\ &\leq \left(\frac{6}{\epsilon}\right)^d \cdot \frac{4}{\epsilon} \leq \left(\frac{6}{\epsilon}\right)^{d+1}.\end{aligned}$$

We bound the expected maximum deviation over all w and b by employing a bound on the expected maximum of subgaussian random variables (Lemma 4.11), applying the covering numbers argument, letting $\pi(w,b) = \arg\min_{(w',b')\in\mathcal{N}_\epsilon} \gamma((w,b),(w',b'))$, and setting $\epsilon \coloneqq 1/\sqrt{n}$.

$$\begin{aligned}\mathbb{E}\left[\sup_{w\in\mathbb{S}^{d-1}, b\in[-\sqrt{d},\sqrt{d}]} |\mathbf{Z}_{w,b}|\right] &\leq \mathbb{E}\left[\sup_{w,b} \left|\mathbf{Z}_{w,b} - \mathbf{Z}_{\pi(w,b)}\right|\right] + \mathbb{E}\left[\sup_{(w,b)\in\mathcal{N}_\epsilon} |\mathbf{Z}_{w,b}|\right] \\ &\leq 4\sqrt{d}\epsilon + \frac{2C_2}{\sqrt{n}}\sqrt{\ln \mathcal{N}\left(\epsilon, \mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}], \gamma\right)} \\ &\leq 4\sqrt{d}\epsilon + 2C_2\sqrt{\frac{d+1}{n}\ln\frac{6}{\epsilon}} \leq C_3\sqrt{\frac{d\ln n}{n}}.\end{aligned}$$

(Step 4) We conclude by showing that $\sup_{w,b} |\mathbf{Z}_{w,b}|$ is close to its expectation with high probability due to the McDiarmid's inequality (Lemma 4.10). Consider a perturbation where $\mathbf{x}_i$ is replaced by some $\mathbf{x}_i' \in \{-1,1\}^d$ with $\mathbf{y}_i' = \mathrm{Par}(\mathbf{x}_i')$, and let $\mathbf{Z}_{w,b}^i$ denote the resulting deviation term.

$$\begin{aligned}\left|\sup_{w,b} |\mathbf{Z}_{w,b}| - \sup_{w,b} \left|\mathbf{Z}_{w,b}^i\right|\right| &\leq \sup_{w,b} \left|\mathbf{Z}_{w,b} - \mathbf{Z}_{w,b}^i\right| = \frac{1}{n}\sup_{w,b} |\mathbf{y}_i \bar{r}_{w,b}(\mathbf{x}_i) - \mathbf{y}_i' \bar{r}_{w,b}(\mathbf{x}_i')| \\ &\leq \frac{1}{n}\sup_{w,b} \left[|\bar{r}_{w,b}(\mathbf{x}_i) - \bar{r}_{w,b}(\mathbf{x}_i')| + |(\mathbf{y}_i - \mathbf{y}_i')\bar{r}_{w,b}(\mathbf{x}_i)|\right] \\ &\leq \frac{1}{n}\left[\|\mathbf{x}_i - \mathbf{x}_i'\|_2 + 2\|\mathbf{x}_i\|_2\right] \leq \frac{4\sqrt{d}}{n}\end{aligned}$$

Hence, with probability at least $1 - \delta$:

$$\sup_{w,b} |\mathbf{Z}_{w,b}| \leq \sqrt{\frac{8d \ln 1/\delta}{n}} + \mathbb{E}\left[\sup_{w,b} |\mathbf{Z}_{w,b}|\right] \leq C_4 \sqrt{\frac{d(\ln n + \ln 1/\delta)}{n}}.$$

The bound in the lemma statement immediately follows. □

4.4 Generalization properties of solutions to the variational problem

In this section, we consider the generalization properties of a learning algorithm that returns a solution to (VP) for a *sampled parity dataset*

$$\{(\mathbf{x}_i, \mathrm{Par}(\mathbf{x}_i)) : i \in [n]\}, \text{ for } \mathbf{x}_1, \ldots, \mathbf{x}_n \sim_{\mathrm{iid}} \nu.$$

(Again, for simplicity, we label data using Par, but the same results hold for any Par_S with $|S| = \Theta(d)$.)

We show that $n = o(d^2/\sqrt{\log d})$ results in a predictor with nearly trivial accuracy. Note that information-theoretically, $n \geq O(d)$ is sufficient for learning any parity function (Helmbold, Sloan, and Warmuth, 1992; Fischer and Simon, 1992). This means that the inductive bias based on $\mathcal{R}$-norm is not sufficient to achieve statistically optimal sample complexity for learning parity functions.

4.4.1 Poor generalization with $n \ll d^2/\sqrt{\log d}$ samples

We first give a lower bound on the sample size needed for non-trivial generalization for learning parity functions by solving (VP) with the sampled parity dataset.

Theorem 4.27. *If $n = o(d^2/\sqrt{\log d})$, then with probability at least $1/2$, every solution $\mathbf{g}\colon \mathbb{B}^d \to \mathbb{R}$ to* (VP) *for the sampled parity dataset has $\|\mathbf{g} - \mathrm{Par}\|_{L^2(\nu)} \geq 1 - o(1)$.*

Its proof relies on the following approximation lemma, which shows the existence of a low-$\mathcal{R}$-norm network $\mathbf{g}$ that perfectly fits all n samples. The lemma defines $\mathbf{g}$ with the same

"cap construction" used in Theorem 1 of Bubeck, Li, and Nagaraj (2021).

Lemma 4.28. *There is an absolute constant $c > 0$ such that the following holds. If $n \le cd^2$, and $\mathbf{x}_1, \dots \mathbf{x}_n \sim_{\text{iid}} \nu$, then with probability at least $1/2$, there exists $\mathbf{g}\colon \mathbb{B}^d \to \mathbb{R}$ with $\mathbf{g}(\mathbf{x}_i) = \mathrm{Par}(\mathbf{x}_i)$ for all $i \in [n]$ and $\|\mathbf{g}\|_{\mathcal{R}} \le 4n\sqrt{\ln d}/d$.*

We conclude that generalization fails in this low-sample regime because Theorem 4.20 shows that no network with sufficiently small $\mathcal{R}$-norm can correlate with parity.

Proof of Theorem 4.27. Let $\alpha := 64n\sqrt{\ln d}/d^2$, so $\alpha = o(1)$ by assumption on n. By Theorem 4.20, every $g\colon \mathbb{B}^d \to \mathbb{R}$ with $\|g - \mathrm{Par}\|_{L^2(\nu)} \le 1 - \alpha$ has $\|g\|_{\mathcal{R}} \ge \alpha d/8 \ge 8n\sqrt{\ln d}/d$. However, by Lemma 4.28, with probability at least $1/2$, every solution $\mathbf{g}$ to (VP) for the dataset $\{(\mathbf{x}_i, \mathrm{Par}(\mathbf{x}_i))\}_{i\in[n]}$ has $\|\mathbf{g}\|_{\mathcal{R}} \le 4n\sqrt{\ln d}/d$. In this event, the solutions $\mathbf{g}$ have $\|\mathbf{g} - \mathrm{Par}\|_{L^2(\nu)} \ge 1 - \alpha = 1 - o(1)$. □

The proof of Lemma 4.28 utilizes a proof that a random sample from the boolean hypercube is subgaussian when conditioned on its parity.

Lemma 4.29. *Fix $S \subseteq [d]$ with $|S| \ge 3$, and let $\mathbf{x} \sim \mathrm{Unif}(\{-1, 1\}^d)$. Conditional on the value of $\mathrm{Par}_S(\mathbf{x})$, the random vector $\mathbf{x}$ is mean-zero, isotropic, and satisfies*

$$\mathbb{E}\left[\exp(u^{\mathsf{T}}\mathbf{x}) \mid \mathrm{Par}_S(\mathbf{x})\right] \le \exp(\|u\|_2^2)$$

for all $u = (u_1, \dots, u_d) \in \mathbb{R}^d$.

Proof. The assumption $|S| \ge 3$ implies that, conditioned on $\mathrm{Par}_S(\mathbf{x})$, the $\{\mathbf{x}_i\}_{i\in[d]}$ are mean-zero and pairwise uncorrelated. So it remains to show that, for any vector $u = (u_1, \dots, u_d) \in \mathbb{R}^d$,

$$\mathbb{E}\left[\exp(u^{\mathsf{T}}\mathbf{x}) \mid \mathrm{Par}_S(\mathbf{x})\right] \le \exp(\|u\|_2^2).$$

So fix u, and fix any $i \in S$. Let u_{-i} (respectively, $\mathbf{x}_{-i}$) be the vector obtained from u (respectively, $\mathbf{x}$) by removing the i-th entry. Observe that $\mathbf{x}_{-i} \mid \mathrm{Par}_S(\mathbf{x}) \sim \mathrm{Unif}(\{-1, 1\}^{d-1})$,

and also that $\mathbf{x}_i \mid \mathrm{Par}_S(\mathbf{x}) \sim \mathrm{Unif}(\{-1,1\})$. (But, of course, $\mathbf{x}_{-i}$ and $\mathbf{x}_i$ are not conditionally independent given $\mathrm{Par}_S(\mathbf{x})$.) Therefore, using Cauchy-Schwarz,

$$\begin{aligned}
\mathbb{E}\left[\exp(u^{\mathsf{T}}\mathbf{x}) \mid \mathrm{Par}_S(\mathbf{x})\right] &= \mathbb{E}\left[\exp(u_{-i}^{\mathsf{T}}\mathbf{x}_{-i})\exp(u_i\mathbf{x}_i) \mid \mathrm{Par}_S(\mathbf{x})\right] \\
&\le \sqrt{\mathbb{E}\left[\exp(2u_{-i}^{\mathsf{T}}\mathbf{x}_{-i}) \mid \mathrm{Par}_S(\mathbf{x})\right]}\sqrt{\mathbb{E}\left[\exp(2u_i\mathbf{x}_i) \mid \mathrm{Par}_S(\mathbf{x})\right]} \\
&\le \sqrt{\exp(\|2u_{-i}\|_2^2/2)}\sqrt{\exp((2u_i)^2/2)} \\
&= \exp(\|u\|_2^2).
\end{aligned}$$

Above, the second inequality uses the moment generating function bound from Lemma 4.13, as well as the conditional independence of $\{\mathbf{x}_j : j \neq i\}$ given $\mathrm{Par}(\mathbf{x})$. □

Now, we are ready to prove Lemma 4.28.

Proof of Lemma 4.28. Throughout, we take $C > 0$ to be a suitably large constant, and we assume $n \le d^2/C$. The construction of $\mathbf{g}\colon \mathbb{B}^d \to \mathbb{R}$ is based on typical statistical behavior of the random examples $(\mathbf{x}_1,\mathbf{y}_1),\dots,(\mathbf{x}_n,\mathbf{y}_n)$, where $\mathbf{y}_i := \mathrm{Par}_S(\mathbf{x}_i)$ for each $i \in [n]$. We may assume that $n \ge d$, since otherwise the examples can be perfectly fit with a linear function $\mathbf{g}$, and this function has $\|\mathbf{g}\|_{\mathcal{R}} = 0$. So, combining the assumption $n \ge d$ with the assumption $n \le d^2/C$ implies that $d \ge C$. Observe that $\mathbf{y}_1,\dots,\mathbf{y}_n$ are i.i.d. $\mathrm{Unif}(\{-1,1\})$ random variables. Since $n \ge d \ge C$, it follows by standard binomial tail bounds that with probability at least $5/6$ over the realizations of $\mathbf{y}_1,\dots,\mathbf{y}_n$, the number of $\mathbf{y}_i$ that are equal to 1 is at least $n/3$, and also that the number of $\mathbf{y}_i$ that are equal to -1 is also at least $n/3$. We henceforth condition on this "good event" (which depends only on $\mathbf{y}_1,\dots,\mathbf{y}_n$).

To help define our construction of $\mathbf{g}\colon \mathbb{B}^d \to \mathbb{R}$ and set up the rest of the analysis, we partition $[n]$ into disjoint groups $G_1, G_2, \dots, G_m$ so that for each $j \in [m]$, (i) the size $n_j := |G_j|$ of the j-th group is between $c_1 d/\ln d$ and $2c_1 d/\ln d$, and (ii) all $\mathbf{y}_i$ for $i \in G_j$ are the same (i.e., all $+1$ or all -1). Here, with foresight, we set $c_1 := 1/256$; by using $d \ge C$, we ensure that each group is non-empty, and also that $n_j < d$. The feasibility of this partitioning

is ensured because, in the "good event" (and using $d \geq C$), the number of $i \in [n]$ with $\mathbf{y}_i = 1$ is at least $n/3 \geq d/3 \geq c_1 d/\ln d$, and same for the number of $i \in [n]$ with $\mathbf{y}_i = -1$. Let $z^{(j)}$ denote the common $\mathbf{y}_i$ value for all $i \in G_j$. Finally, note that the number of groups m satisfies $m \leq n\ln(d)/(c_1 d)$.

We now define our construction of $\mathbf{g}\colon \mathbb{B}^d \to \mathbb{R}$. Let $\mathbf{A}_j$ denote the random $n_j \times d$ matrix whose rows are the $\mathbf{x}_i^\mathsf{T}$ for $i \in G_j$, and define the random vector $\mathbf{w}^{(j)} := \mathbf{A}_j^\dagger(z^{(j)}\vec{1})$. Observe that $\mathbf{w}^{(j)}$ is a least squares solution to the system of linear equations $\{\mathbf{x}_i^\mathsf{T} w = \mathbf{y}_i : i \in G_j\}$, since $\mathbf{y}_i = z^{(j)}$ for all $i \in G_j$. We define $\mathbf{g}$ as follows:

$$\mathbf{g}(x) = \sum_{j=1}^{m} z^{(j)}\mathrm{ReLU}(2z^{(j)}\mathbf{w}^{(j)\mathsf{T}}x - 1).$$

To analyze our construction, we consider the realizations of $\mathbf{x}_1, \ldots, \mathbf{x}_n$, and establish some basic properties that hold with sufficiently high probability (conditional on the "good event"). Note that within a group G_j, the $\{\mathbf{x}_i\}_{i\in G_j}$ are (conditionally) iid, and the realizations across groups are also (conditionally) independent.

We claim that with probability at least 5/6 (conditional on the "good event"),

- (P1) $\mathbf{w}^{(j)\mathsf{T}}\mathbf{x}_i = \mathbf{y}_i$ for all $j \in [m]$ and $i \in G_j$;
- (P2) $\|\mathbf{w}^{(j)}\|_2 \leq 2\sqrt{n_j/d}$ for all $j \in [m]$.

To establish this claim, we lower-bound the n_j-th largest singular value $\sigma_{n_j}(\mathbf{A}_j)$. Note that $\sigma_{n_j}(\mathbf{A}_j)$ is at least the corresponding singular value of the $n_j \times (d-1)$ submatrix $\mathbf{B}_j$ obtained from $\mathbf{A}_j$ by removing the t-th column of $\mathbf{A}_j$ for some $t \in S$. (If S is empty, we may remove any column.) Since the rows of $\mathbf{A}_j$ are independent, and since the entries of $\mathbf{x}_i$ after removing the t-th one are iid $\mathrm{Unif}(\{-1,1\})$ random variables, it follows that the $n_j \times (d-1)$ entries of $\mathbf{B}_j$ are iid $\mathrm{Unif}(\{-1,1\})$ random variables. Hence, the rows of $\mathbf{B}_j^\mathsf{T}$ are independent, mean-zero, isotropic, and $O(1)$-subgaussian. By Lemma 4.12 and a union bound, with probability

at least $1 - 2m\exp(-\min_{j\in[m]}\{n_j\})$,

$$\sigma_{n_j}(\mathbf{A}_j) \geq \sigma_{n_j}(\mathbf{B}_j^{\mathsf{T}}) \geq \sqrt{d-1} - C_2\sqrt{n_j} \geq \left(\sqrt{1-\frac{1}{d}} - C_2\sqrt{\frac{c_1}{\ln d}}\right)\sqrt{d} \quad \text{for all } j \in [m],$$

where $C_2 > 0$ is twice the absolute constant from Lemma 4.12, and the final inequality uses the upper-bound on n_j. The fact $d \geq C$ and the upper-bounds on m and n altogether imply that the probability of the above event is at least $5/6$, and also that $\sqrt{1-1/d} - C_2\sqrt{c_1/\ln d} \geq 1/2$. So in this event, for each $j \in [m]$, the column space of $\mathbf{A}_j$ has rank n_j, so the system of linear equations defining $\mathbf{w}^{(j)}$ is feasible, and

$$\|\mathbf{w}^{(j)}\|_2 = \|\mathbf{A}_j^{\dagger}(z^{(j)}\vec{1})\|_2 \leq \sigma_1(\mathbf{A}_j^{\dagger})\|\vec{1}\|_2 = \frac{\sqrt{n_j}}{\sigma_{n_j}(\mathbf{A}_j)} \leq 2\sqrt{\frac{n_j}{d}}.$$

This establishes P1 and P2 in the event as claimed.

We further claim that with probability at least $5/6$ (conditional on the "good event"),

- (P3) $|\mathbf{w}^{(j)\mathsf{T}}\mathbf{x}_i| \leq 4\|\mathbf{w}^{(j)}\|_2\sqrt{\ln d}$ for all $j \in [m]$ and $i \in [n] \setminus G_j$.

To establish this claim, first observe that $\mathbf{x}_i$ and $\mathbf{w}^{(j)}$ are independent for $i \notin G_j$. Moreover, by Lemma 4.29, conditional on $\mathbf{w}^{(j)}$ (with $G_j \not\ni i$), $\mathbf{x}_i^{\mathsf{T}}\mathbf{w}^{(j)}$ is a mean-zero random variable satisfying

$$\mathbb{E}\left[\exp(\mathbf{w}^{(j)\mathsf{T}}\mathbf{x}) \mid \mathrm{Par}_S(\mathbf{x}), \mathbf{w}^{(j)}\right] \leq \exp(\|\mathbf{w}^{(j)}\|_2^2).$$

So, by Markov's inequality and a union bound, we have with probability at least $5/6$,

$$|\mathbf{w}^{(j)\mathsf{T}}\mathbf{x}_i| \leq (\sqrt{2}\|\mathbf{w}^{(j)}\|_2)\sqrt{2\ln(12mn)} \quad \text{for all } j \in [m] \text{ and } i \in [n] \setminus G_j.$$

Using $d \geq C$ and the upper-bounds on m and n, we obtain $\sqrt{\ln(12mn)} \leq 2\sqrt{\ln d}$, and hence we deduce P3 from the above inequality.

So, by a union bound, with probability at least $2/3$ (conditional on the "good event"), the properties P1, P2, and P3 all hold simultaneously. We can now establish the desired

properties of $\mathbf{g}$. Using $d \geq C$, P2, and the upper-bounds on m and n, we obtain

$$\|\mathbf{g}\|_{\mathcal{R}} \leq 2\sum_{j=1}^{m} \|\mathbf{w}^{(j)}\|_2 \leq 4\sum_{j=1}^{m} \sqrt{\frac{n_j}{d}} \leq 4\sqrt{m\sum_{j=1}^{m} \frac{n_j}{d}} = 4\sqrt{\frac{mn}{d}} \leq \frac{4n\sqrt{\ln d}}{d}.$$

Furthermore, by P1, we have for any $j \in [m]$ and $i \in G_j$,

$$2z^{(j)}\mathbf{w}^{(j)\mathsf{T}}\mathbf{x}_i - 1 = 2z^{(j)}\mathbf{y}_i - 1 = 1.$$

And by P2, P3, and the upper-bound on n_j, we have for any $j \in [m]$ and $i \in [n] \setminus G_j$,

$$2z^{(j)}\mathbf{w}^{(j)\mathsf{T}}\mathbf{x}_i - 1 \leq 2|\mathbf{w}^{(j)\mathsf{T}}\mathbf{x}_i| - 1 \leq 16\sqrt{\frac{n_j \ln d}{d}} - 1 \leq 16\sqrt{c_1} - 1 = 0,$$

and hence $\mathrm{ReLU}(2z^{(j)}\mathbf{w}^{(j)\mathsf{T}}\mathbf{x}_i - 1) = 0$. Therefore, for any $i \in [n]$, if $i \in G_j$,

$$\mathbf{g}(\mathbf{x}_i) = z^{(j)}\mathrm{ReLU}(2z^{(j)}\mathbf{w}^{(j)\mathsf{T}}\mathbf{x}_i - 1) = z^{(j)} = \mathbf{y}_i. \qquad \square$$

4.4.2 Good generalization with $n \gtrsim d^3$ samples

We complement the lower bound in Theorem 4.27 with the following sample complexity upper bound.

Theorem 4.30. *There is an absolute constant $C > 0$ such that the following holds. For any $\epsilon \in (0,1)$ and $\delta \in (0,1)$, if $n \geq C(\log(1/\delta) + d^3/\epsilon^2)$, then with probability at least $1-\delta$, every solution $\mathbf{g}\colon \mathbb{B}^d \to \mathbb{R}$ to* (VP) *for the sampled parity dataset satisfies $\|\mathrm{Par} - \mathrm{clip} \circ \mathbf{g}\|_{L^2(\nu)}^2 \leq \epsilon$, where $\mathrm{clip}(t) := \min\{\max\{t, -1\}, 1\}$.*

We note that there is a gap between our lower bound (Theorem 4.27) and upper bound (Theorem 4.30): roughly a factor of $d\sqrt{\log d}$. We believe that this gap could be narrowed if one resolves the open question raised by Bubeck, Li, and Nagaraj (2021) about the minimum Lipschitz constant achievable by two-layer ReLU networks of width m networks that interpolate a sample of size n; Lemma 4.28 is derived from a theorem that produces networks with

smoothness conjectured to be sub-optimal. Nevertheless, our lower bound in Theorem 4.27 is already high enough to establish the statistical suboptimality of solutions to (VP).

For technical reasons, we only bound the $L^2(\nu)$ error of the natural truncation of a solution to (VP). The proof is based on standard Rademacher complexity arguments.

Proof. Let $\mathbf{G}$ denote all solutions to (VP) on the sampled parity dataset, so $\|\mathbf{g} - \mathrm{Par}\|_{L^2(\boldsymbol{\nu}_n)} = 0$ for all $\mathbf{g} \in \mathbf{G}$. By Proposition 4.3, we can write each $\mathbf{g}$ as $\mathbf{g}(x) = g_{\boldsymbol{\mu}}(x) + \mathbf{v}^\mathsf{T} x + \mathbf{c}$, where $\boldsymbol{\mu} \in \mathcal{M}$, $\mathbf{v} \in \mathbb{R}^d$, and $\mathbf{c} \in \mathbb{R}$. Furthermore, we can assume that $g_{\boldsymbol{\mu}}(0) = 0$ by absorbing the value of $g_{\boldsymbol{\mu}}(0)$ into $\mathbf{c}$ (at the cost of losing the evenness of $\boldsymbol{\mu}$, but evenness is not needed in the sequel). Lemma 4.5, Theorem 4.4, and Theorem 4.17 together imply that every $\mathbf{g} \in \mathbf{G}$ satisfies $\|\mathbf{g}\|_{\mathcal{R}} \le Cd$ for some absolute constant $C > 0$. Let $\mathcal{E}$ be the event that $\max\{|\mathbf{c}|, \|\mathbf{v}\|_2\} \le 8Cd$ (for all $\mathbf{g} \in \mathbf{G}$), and let $\mathcal{E}^{\mathrm{c}}$ be its complement; event $\mathcal{E}$ occurs with probability at least $1 - \delta/2$ by Lemma 4.25, for another absolute constant $C' > 0$.

Since, for each $\mathbf{g} \in \mathbf{G}$, we have $\mathbf{g}(\mathbf{x}_i) = \mathrm{Par}(\mathbf{x}_i)$ for every example $(\mathbf{x}_i, \mathrm{Par}(\mathbf{x}_i))$ in the sampled parity dataset, it follows that $\|\mathrm{clip} \circ \mathbf{g} - \mathrm{Par}\|_{L^2(\boldsymbol{\nu}_n)} = \|\mathbf{g} - \mathrm{Par}\|_{L^2(\boldsymbol{\nu}_n)} = 0$ for all such $\mathbf{g} \in \mathbf{G}$. For any $t > 0$,

$$
\begin{aligned}
&\Pr\left[\sup_{\mathbf{g}\in\mathbf{G}} \|\mathrm{clip} \circ \mathbf{g} - \mathrm{Par}\|^2_{L^2(\nu)} \ge t\right] \\
&\le \Pr\left[\sup_{\mathbf{g}\in\mathbf{G}} \|\mathrm{clip} \circ \mathbf{g} - \mathrm{Par}\|^2_{L^2(\nu)} \ge t \mid \mathcal{E}\right] + \Pr\left[\mathcal{E}^{\mathrm{c}}\right] \\
&= \Pr\left[\sup_{\mathbf{g}\in\mathbf{G}} \|\mathrm{clip} \circ \mathbf{g} - \mathrm{Par}\|^2_{L^2(\nu)} - \|\mathrm{clip} \circ \mathbf{g} - \mathrm{Par}\|^2_{L^2(\boldsymbol{\nu}_n)} \ge t \mid \mathcal{E}\right] + \Pr\left[\mathcal{E}^{\mathrm{c}}\right] \\
&\le \Pr\left[\sup_{g\in\mathcal{G}_0} \|\mathrm{clip} \circ g - \mathrm{Par}\|^2_{L^2(\nu)} - \|\mathrm{clip} \circ g - \mathrm{Par}\|^2_{L^2(\boldsymbol{\nu}_n)} \ge t\right] + \delta/2,
\end{aligned}
$$

where

$$
\mathcal{G}_0 := \left\{x \mapsto g(x) + v^\mathsf{T} x + c : \|g\|_{\mathcal{R}} \le Cd,\ \max\{\|v\|_2, |c|\} \le 8Cd\right\}.
$$

Define

$$t_0 := 4\underbrace{\mathbb{E}\left[\sup_{g\in\mathcal{G}_0}\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\epsilon}_i g(\mathbf{x}_i)\right]}_{\mathrm{Rad}_n(\mathcal{G}_0)} + 4\sqrt{\frac{\log(2/\delta)}{n}}.$$

Above, $\mathrm{Rad}_n(\mathcal{G}_0)$ denotes the Rademacher complexity of $\mathcal{G}_0$, where

$$\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n \sim_{\mathrm{iid}} \mathrm{Unif}(\{-1, 1\}),$$

independent of $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Since, for any $y \in \{-1, 1\}$, the mapping

$$z \mapsto (y - \mathrm{clip}(z))^2 = (1 - y\,\mathrm{clip}(z))^2$$

is 4-Lipschitz and has range $[-4, 4]$, it follows by standard Rademacher complexity arguments (see, e.g., Meir and Zhang, 2003, Theorem 8) that

$$\Pr\left[\sup_{g\in\mathcal{G}_0}\|\mathrm{clip}\circ g - \mathrm{Par}\|^2_{L^2(\nu)} - \|\mathrm{clip}\circ g - \mathrm{Par}\|^2_{L^2(\boldsymbol{\nu}_n)} \geq t_0\right] \leq \delta/2.$$

So it remains to show that $t_0 \leq \epsilon$ under the assumption $n \geq C_0((d^3 + \log(1/\delta))/\epsilon^2)$ for suitably large absolute constant $C_0 > 0$. The second term in the definition of t_0 is at most $\epsilon/2$ provided that C_0 is chosen large enough. To bound the first term ($\mathrm{Rad}_n(\mathcal{G}_0)$), we use the fact that

$$\mathrm{Rad}_n(\mathcal{G}_0) = \mathrm{Rad}_n(\mathcal{G}_1) + \mathrm{Rad}_n(\mathcal{G}_2)$$

where $\mathcal{G}_1 := \{g : \|g\|_{\mathcal{R}} \leq Cd\}$ and $\mathcal{G}_2 := \{x \mapsto v^{\mathsf{T}}x + c : \max\{\|v\|_2, |c|\} \leq 8Cd\}$. Theorem 10 of Parhi and Nowak (2021a) implies

$$\mathrm{Rad}_n(\mathcal{G}_1) \leq \frac{2\cdot(Cd)\cdot\sqrt{d}}{\sqrt{n}} = O\left(\sqrt{\frac{d^3}{n}}\right),$$

while Theorem 3 of Kakade, Sridharan, and Tewari (2008) implies

$$\mathrm{Rad}_n(\mathcal{G}_2) \le \sqrt{d+1} \cdot \sqrt{(8Cd)^2} \cdot \sqrt{\frac{2}{n}} = O\left(\sqrt{\frac{d^3}{n}}\right).$$

By choosing C_0 large enough, it follows that $\mathrm{Rad}_n(\mathcal{G}_0) \le \epsilon/8$. Hence, we have shown that $t_0 \le \epsilon$ as required. □

4.5 Generality of the averaging technique for minimizing $\mathcal{R}$-norm

In this section, we show how the benefit of averaging goes beyond the parity dataset. We consider an *f-dataset* $\{(x, f(x)\}_{x\in\{-1,1\}^d}$, a generalization of the parity dataset where $f(x) = \phi(v^\mathsf{T} x)$ is a ridge function with L-Lipschitz and ρ-periodic ϕ. For another dataset generated by oscillatory ridge functions, we prove the same contrast between minimum-$\mathcal{R}$-norm interpolation with and without ridge constraints, so long as the periodicity ρ is not too small (specifically, $\rho \ge 1/\sqrt{d}$). More concretely, suppose the dataset $\{(x_i, f(x_i))\}_{i\in[n]} \subset \{-1,1\}^d \times \{-1,1\}$ used in (VP) and (ϵ-VP) is the *f-dataset*, where $v \in \{\pm\frac{1}{\sqrt{d}}\}^d$ and ϕ is ρ-periodic and $\frac{1}{\rho}$-Lipschitz. Then we have the following:

- The optimal value of (ϵ-VP) for constant $\epsilon \in (0, 1/2)$ is $\tilde{O}(\sqrt{d}/\rho)$. (Theorem 4.31)
- The optimal value of (ϵ-VP) for constant $\epsilon \in (0, 1/2)$—with the additional constraint that g be a ridge function—is $\Omega(\sqrt{d}/\rho^2)$. (Theorem 4.32)

Because the parity dataset is an f-dataset with a $1/\sqrt{d}$-periodic and $\sqrt{d}$-Lipschitz choice of ϕ, the above results closely match those of Informal Theorem 4.1. We give both results, starting with an upper bound on the minimum-$\mathcal{R}$-norm approximate interpolant, which parallels Theorem 4.18.

Theorem 4.31. *Suppose $f\colon \mathbb{B}^d \to [-1,1]$ is given by $f(x) = \phi(v^\mathsf{T} x)$ for some unit vector $v \in \mathbb{S}^{d-1}$ and some $\phi\colon [-\sqrt{d}, \sqrt{d}] \to [-1,1]$ that is L-Lipschitz and ρ-periodic for $\rho \in [\|v\|_\infty, 1]$. Fix any $\epsilon \in (0,1)$. There exists a function $g\colon \mathbb{B}^d \to \mathbb{R}$ represented by a width-m neural*

network such that:

$$\|f-g\|_{L^\infty(\nu)} \le \epsilon; \quad m \le dL\,\mathrm{polylog}(1/\epsilon)\sqrt{\rho\|v\|_1}/\epsilon^2; \quad \|g\|_{\mathcal{R}} \le L^2\,\mathrm{polylog}(d/\epsilon)\rho\|v\|_1/\epsilon.$$

Remark 4.4. *Suppose* $f(x) = \cos(\frac{2\pi}{\rho}v^\mathsf{T}x)$ *for* $v \in \{\pm\frac{1}{\sqrt{d}}\}^d$ *and* $\rho \in [\frac{1}{\sqrt{d}}, 1]$. *Theorem 4.31 implies that there exists an* ϵ*-approximate interpolating neural network* g *of width* $\tilde{O}(\frac{d^{5/4}}{\sqrt{\rho}\epsilon^2})$ *and* $\|g\|_{\mathcal{R}} = \tilde{O}(\frac{\sqrt{d}}{\rho\epsilon})$. *If* d *is even and* $\rho = 4/\sqrt{d}$, *then* $f(x) = \mathrm{Par}(x)$ *for* $x \in \{-1,1\}^d$, *and the width and* $\mathcal{R}$*-norm bounds of Theorem 4.18 for small* t *are approximately recovered.*

A detailed version of Theorem 4.31 appears in Section 4.5.1. The construction is more delicate than that in Theorem 4.18 due to the potential lack of symmetries that had existed in the parity dataset.

We give the lower bound on the $\mathcal{R}$-norm of all approximately interpolanting ridge functions, whose proof in Section 4.5.2 relies a reduction to the argument of Theorem 4.15.

Theorem 4.32. *Assume* d *is even. Let* Ridge_d *be the set of functions* $g\colon \mathbb{B}^d \to \mathbb{R}$ *such that* $g(x) = \phi(w^\mathsf{T}x)$ *for some* $w \in \mathbb{S}^{d-1}$ *and Lipschitz continuous* $\phi\colon [-\sqrt{d}, \sqrt{d}] \to \mathbb{R}$. *Let* $\rho := 4q/\sqrt{d}$ *for* $q \in \{1, 2, \ldots, \lfloor\sqrt{d}/4\rfloor\}$ *and* $f(x) := \cos((2\pi/(\rho\sqrt{d}))\vec{1}^\mathsf{T}x)$. *Then*

$$\inf\{\|g\|_{\mathcal{R}} : g \in \mathrm{Ridge}_d,\ \|g-f\|_{L^\infty(\nu)} \le 1/2\} = \Omega(\sqrt{d}/\rho^2).$$

Remark 4.5. *By contrasting the above result to the* $\tilde{O}(\frac{\sqrt{d}}{\rho\epsilon})$ $\mathcal{R}$*-norm of the averaging-based construction from Remark 4.4, ridge functions are suboptimal solutions to* ϵ*-VP for constant* ϵ.

Remark 4.6. *Lemma 4.34 (in Section 4.5.1) implies the existence of a neural network* $g_{\mathrm{Ridge}} \in \mathrm{Ridge}_d$ *that point-wise approximates* f *(i.e.,* $\|g_{\mathrm{Ridge}} - f\|_{L^\infty(\nu)} \le \epsilon$*) and has*

$$\|g_{\mathrm{Ridge}}\|_{\mathcal{R}} = O(\frac{\sqrt{d}}{\rho^2\epsilon}).$$

Hence, the lower bound in Theorem 4.32 is tight when ϵ *is constant.*

4.5.1 Proof of Theorem 4.31

Theorem 4.33 (Detailed version of Theorem 4.31)**.** *Suppose $f\colon \mathbb{B}^d \to [-1,1]$ is given by $f(x) = \phi(v^\mathsf{T} x)$ for some unit vector $v \in \mathbb{S}^{d-1}$ and some $\phi\colon [-\sqrt{d}, \sqrt{d}] \to [-1,1]$ that is L-Lipschitz and ρ-periodic for $\rho \in [\|v\|_\infty, 1]$. Let $\sigma_{\rho,v} := \sqrt{2\rho\|v\|_1 - 1}$, and fix any $\epsilon \in (0,1)$. There exists a function $g\colon \mathbb{B}^d \to \mathbb{R}$ such that the following properties hold:*

1. *$|f(x) - g(x)| \le \epsilon$ for all $x \in \{-1,1\}^d$;*
2. *g is represented by a neural network of width at most*

$$O\left(\frac{dL(\sigma_{\rho,v}\sqrt{\log(1/\epsilon)} + \rho\log(1/\epsilon))}{\epsilon^2}\right);$$

3. *g satisfies*

$$\|g\|_{\mathcal{R}} = O\left(\frac{L^2(\sigma_{\rho,v}\sqrt{\log(1/\epsilon)} + \rho\log(1/\epsilon))(\sigma_{\rho,v} + \sqrt{\log(d/\epsilon)})}{\epsilon}\right).$$

Proof. We first describe the (randomized) construction of our approximating neural network $\mathbf{g}\colon \mathbb{R}^d \to \mathbb{R}$. For $w \in \mathbb{Z}^d$, define $h_w\colon \mathbb{R}^d \to \mathbb{R}$ by $h_w(x) := \phi(v^\mathsf{T} x + \rho w^\mathsf{T} x)$. Let $\mathbf{w} \in \mathbb{Z}^d \setminus \{-(1/\rho)v\}$ be a random vector with distribution to be specified later in the proof. Let $\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(k)}$ be i.i.d. copies of $\mathbf{w}$ for a positive integer $k > (9(d+1)\ln(2))/\epsilon$, and let $\mathbf{h}^{(j)} := h_{\mathbf{w}^{(j)}}$ for each j. Observe that each $\mathbf{h}^{(j)}$ can be written as $\mathbf{h}^{(j)}(x) = \phi^{(j)}(x^\mathsf{T}\mathbf{u}^{(j)})$ for

$$\mathbf{u}^{(j)} := \frac{1}{\|v + \rho\mathbf{w}^{(j)}\|_2}(v + \rho\mathbf{w}^{(j)}) \in \mathbb{S}^{d-1} \quad \text{and} \quad \phi^{(j)}(z) := \phi(\|v + \rho\mathbf{w}^{(j)}\|_2 z),$$

where $\phi^{(j)}\colon \mathbb{R} \to [-1,1]$ is L_j-Lipschitz for $L_j := L\|v + \rho\mathbf{w}^{(j)}\|_2$ (using the L-Lipschitzness of ϕ). Let $\tau > 0$ be a value (depending on ρ, v, and ϵ) also to be specified later. By Lemma 4.34 (with $t := \tau/\|v + \rho\mathbf{w}^{(j)}\|_2$ and $\delta := \epsilon/3$), there exist $\tilde{\mathbf{h}}^{(1)}, \dots, \tilde{\mathbf{h}}^{(k)}$ such that:

- (H1) $\tilde{\mathbf{h}}^{(j)}\colon \mathbb{R}^d \to \mathbb{R}$ is represented by a neural network of width at most $O(\tau L/\epsilon)$;

- (H2) $\|\tilde{\mathbf{h}}^{(j)}\|_{\mathcal{R}} = O(\tau L^2 \|v + \rho \mathbf{w}^{(j)}\|_2/\epsilon)$;
- (H3) $|\tilde{\mathbf{h}}^{(j)}(x) - \mathbf{h}^{(j)}(x)| \leq \epsilon/3$ for all $x \in \{-1,1\}^d$ such that $|x^\mathsf{T}\mathbf{u}^{(j)}| \leq \tau/\|v + \rho\mathbf{w}^{(j)}\|_2$;
- (H4) $|\tilde{\mathbf{h}}^{(j)}(x) - \mathbf{h}^{(j)}(x)| \leq 1$ for all $x \in \mathbb{R}^d$.

Our approximating neural network $\mathbf{g}\colon \mathbb{R}^d \to \mathbb{R}$ is defined by

$$\mathbf{g}(x) := \frac{1}{k}\sum_{j=1}^{k} \tilde{\mathbf{h}}^{(j)}(x).$$

By construction and using properties H1 and H2 (above), the following properties of $\mathbf{g}$ are immediate:

- (G1) $\mathbf{g}$ is represented by a neural network of width at most $O(k\tau L/\epsilon)$;
- (G2) $\max\{\|\mathbf{g}\|_{\mathcal{R}}, \|\mathbf{g}\|_{\mathscr{S}_2}\} = O(\tau L^2 \max_{j\in[k]} \|v + \rho\mathbf{w}^{(j)}\|_2/\epsilon)$.

Note that these properties are given in terms of τ, which has yet to be specified, as well as $\max_{j\in[k]} \|v + \rho\mathbf{w}^{(j)}\|_2$, which is a random variable. So, in the remainder of the proof, we choose a particular distribution for $\mathbf{w}$ (and hence also for $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(k)}$) and a value of τ that, together, will ultimately allow us to establish the existence of an approximating neural network with the desired properties via the probabilistic method.

We first specify the probability distribution of $\mathbf{w}$ and establish some of its properties. We let $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_d)$ be a vector of independent random variables $\mathbf{w}_1, \ldots, \mathbf{w}_d$ with

$$p_i := \Pr[\mathbf{w}_i = -2\mathrm{sign}v_i] = |v_i|/(2\rho)$$

and $\Pr[\mathbf{w}_i = 0] = 1 - p_i$. Note that $p_i \in [0, 1/2]$ for all i since we have assumed $\rho \geq \|v\|_\infty$, so the distribution of $\mathbf{w}$ is well-defined. Furthermore, observe that $\mathbf{w} \neq -(1/\rho)v$ almost surely (since $v \neq 0$ and $\rho \geq \|v\|_\infty$ by assumption), $\mathbb{E}[v + \rho\mathbf{w}] = 0$, and

$$\mathbb{E}\left[\|v + \rho\mathbf{w}\|_2^2\right] = \sum_{i=1}^{d} \mathrm{Var}(\rho\mathbf{w}_i) = \sum_{i=1}^{d} 4\rho^2 \cdot \frac{|v_i|}{2\rho} \cdot \left(1 - \frac{|v_i|}{2\rho}\right) = 2\rho\|v\|_1 - \|v\|_2^2 = \sigma_{\rho,v}^2.$$

Moreover, $\|v+\rho\mathbf{w}\|_2$ is a function of independent random variables $\mathbf{w}_1, \ldots, \mathbf{w}_d$ that satisfies the $(2|v_1|, \ldots, 2|v_d|)$-bounded differences property. By McDiarmid's inequality (Lemma 4.10) and Jensen's inequality,

$$\Pr\left[\|v+\rho\mathbf{w}\|_2 \geq \sigma_{\rho,v} + \sqrt{2\ln(1/\delta)}\right] \leq \delta \quad \text{for all } \delta \in (0,1). \tag{4.2}$$

Finally, for any fixed $x \in \{-1,1\}^d$, $x^\mathsf{T}(v+\rho\mathbf{w}) = \sum_{i=1}^d x_i(v_i+\rho\mathbf{w}_i)$ is a sum of d independent, mean-zero random variables, with variance $\mathrm{Var}\left(x^\mathsf{T}(v+\rho\mathbf{w})\right) = \sigma_{\rho,v}^2$ and $|x_i(v_i+\rho\mathbf{w}_i)| \leq 2\rho$ almost surely for each i. By Bernstein's inequality (Lemma 4.9),

$$\Pr\left[\left|x^\mathsf{T}(v+\rho\mathbf{w})\right| \geq \sigma_{\rho,v}\sqrt{2\ln(2/\delta)} + 2\rho\ln(2/\delta)/3\right] \leq \delta \quad \text{for all } x \in \{-1,1\}^d,\ \delta \in (0,1). \tag{4.3}$$

We now show that $\mathbf{g}$ has the desired properties with positive probability. Since $w^\mathsf{T}x$ is an integer for any $w \in \mathbb{Z}^d$ and $x \in \{-1,1\}^d$, and since $v+\rho\mathbf{w}^{(j)} \neq 0$ almost surely, the ρ-periodicity of ϕ implies that $g_{\mathbf{w}^{(j)}}(x) = f(x)$ for all $x \in \{-1,1\}^d$ and all $j \in [k]$. Therefore, the intermediate (random) function $\mathbf{g}_1 \colon \mathbb{R}^d \to [-1,1]$ defined by $\mathbf{g}_1(x) := \frac{1}{k}\sum_{j=1}^k \mathbf{h}^{(j)}(x)$ satisfies $\mathbf{g}_1(x) = f(x)$ for all $x \in \{-1,1\}^d$. For each $x \in \{-1,1\}^d$, let

$$\mathbf{r}(x) := |\{j \in [k] : |x^\mathsf{T}(v+\rho\mathbf{w}^{(j)})| \geq \tau\}| = |\{j \in [k] : |x^\mathsf{T}\mathbf{u}^{(j)}| \geq \tau/\|v+\rho\mathbf{w}^{(j)}\|_2\}|.$$

Using the approximation properties of $\tilde{\mathbf{h}}^{(j)}$ (i.e., H3 and H4 from above), we have for each $x \in \{-1,1\}^d$,

$$|\mathbf{g}(x) - \mathbf{g}_1(x)| = \frac{1}{k}\left|\sum_{j=1}^k (\tilde{\mathbf{h}}^{(j)}(x) - \mathbf{h}^{(j)}(x))\right| \leq \left(1 - \frac{\mathbf{r}(x)}{k}\right)\cdot\frac{\epsilon}{3} + \frac{\mathbf{r}(x)}{k}\cdot 1.$$

This final expression is at most ϵ if $\mathbf{r}(x) \leq 2k\epsilon/3$. We choose τ such that for any $x \in \{-1,1\}^d$,

we have $\Pr[|x^\mathsf{T}(v+\rho\mathbf{w})| > \tau] \le \epsilon/3$. By (4.3), it suffices to choose

$$\tau := \sigma_{\rho,v}\sqrt{2\ln(6/\epsilon)} + 2\rho\ln(6/\epsilon)/3.$$

By a multiplicative Chernoff bound (Lemma 4.8) and a union bound over all $x \in \{-1,1\}^d$,

$$\Pr\left[\exists x \in \{-1,1\}^d \text{ s.t. } \mathbf{r}(x) > 2k\epsilon/3\right] \le 2^d \cdot e^{-k\epsilon/9} < \frac{1}{2},$$

where the final inequality uses the choice of $k > (9(d+1)\ln 2)/\epsilon$. Therefore, with probability more than $1/2$, we have $\mathbf{r}(x) \le 2k\epsilon/3$ for all $x \in \{-1,1\}^d$, and hence

$$|\mathbf{g}(x) - f(x)| = |\mathbf{g}(x) - \mathbf{g}_1(x)| \le \epsilon \quad \text{for all } x \in \{-1,1\}^d. \tag{4.4}$$

Finally, by (4.2) an a union bound over all $j \in [k]$, we have that with probability more than $1/2$,

$$\max_{j\in[k]} \|v + \rho\mathbf{w}^{(j)}\|_2 \le \sigma_{\rho,v} + \sqrt{2\ln(2k)}. \tag{4.5}$$

So, there is a positive probability that both (4.4) and (4.5) hold simultaneously, and in this event, it can be checked (via G1 and G2 above) that the function $\mathbf{g}$ satisfies the desired properties in the theorem. □

Lemma 4.34. *Suppose $f(x) = \phi(v^\mathsf{T}x)$ is an L-Lipschitz function for $v \in \mathbb{S}^{d-1}$, $\phi : \mathbb{R} \to [-1,1]$, and $L \ge 1$. For any $t \in [1, \sqrt{d}-1]$ and $\delta \in (0,1)$, there exists a neural network g of width $O(\frac{tL}{\delta})$ such that:*

1. $\|g\|_\mathcal{R} = O(\frac{tL^2}{\delta})$;
2. $|f(x) - g(x)| \le \delta$ *for all* x *with* $\left|v^\mathsf{T}x\right| \le t$;
3. $|f(x) - g(x)| \le 1$ *for all* $x \in \mathbb{R}^d$;
4. $g(x) = 0$ *for all* x *with* $\left|v^\mathsf{T}x\right| \ge t + \frac{1}{L}$; *and*

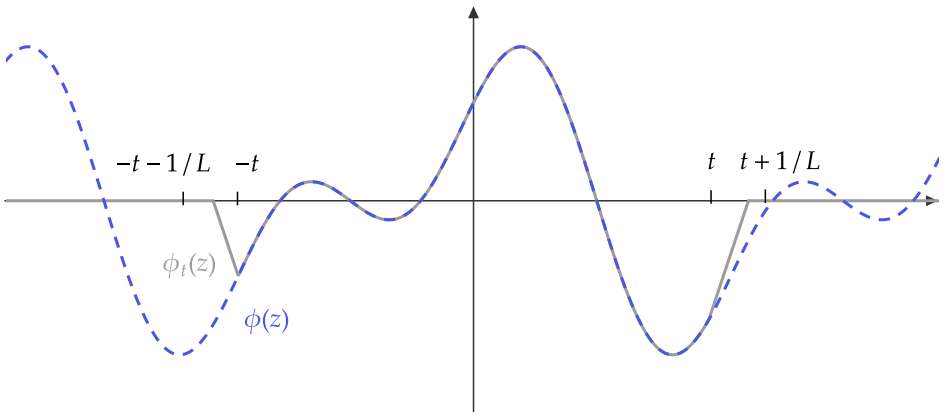


Figure 4.1: A visualization of how the truncated ϕ_t (gray) is generated from ϕ (blue), t, and L.

5. *g is a ridge function that in direction v.*

Proof. We first introduce an L-Lipschitz function ϕ_t (visualized in Figure 4.1) that perfectly fits ϕ on the interval $[-t, t]$ and is zero in $(\infty, -t - \frac{1}{L}] \cup [t + \frac{1}{L}, \infty]$:

$$\phi_t(z) := \begin{cases} \phi(z) & \text{if } z \in [-t, t]; \\ \text{sign}\phi(t) \max\{|\phi(t)| - L(z - t)), 0\} & \text{if } z \geq t; \\ \text{sign}\phi(-t) \max\{|\phi(-t)| - L(-z + t)), 0\} & \text{if } z \leq -t. \end{cases}$$

Then, there exists a piecewise-linear function ψ_t that

- point-wise approximates ϕ_t to accuracy δ;
- has $\psi_t(z) = \phi_t(z)$ for all $z \notin [-t, t]$;
- has $\frac{2tL}{\delta}$ evenly-spaced knots on the interval $[-t, t]$ where ψ_t exactly fits ϕ_t; and
- is L-Lipschitz.

As a result ψ_t can be written as a neural network with $\psi_t(z) = \sum_{j=1}^{m} a^{(j)} \text{ReLU} z - b^{(j)}$ where $m = \frac{2Lt}{\delta}$, $b^{(j)} \in [-t - \frac{1}{L}, t + \frac{1}{L}]$, and $\left|a^{(j)}\right| \leq 2L$.

By taking $g(x) := \psi_t(v^\mathsf{T} x)$, we have a neural network that satisfies conditions 2, 3, 4, and 5. The bound on $\|g\|_\mathcal{R}$ is immediate from the fact that g can be expressed as a neural network with $O(\frac{tL}{\delta})$ neurons with unit weights, biases in $[-\sqrt{d}, \sqrt{d}]$, and bounded coefficients $a^{(j)}$. □

4.5.2 Proof of Theorem 4.32

Theorem 4.32. *Assume d is even. Let* Ridge_d *be the set of functions $g\colon \mathbb{B}^d \to \mathbb{R}$ such that $g(x) = \phi(w^\mathsf{T} x)$ for some $w \in \mathbb{S}^{d-1}$ and Lipschitz continuous $\phi\colon [-\sqrt{d}, \sqrt{d}] \to \mathbb{R}$. Let $\rho := 4q/\sqrt{d}$ for $q \in \{1, 2, \ldots, \lfloor \sqrt{d}/4 \rfloor\}$ and $f(x) := \cos((2\pi/(\rho\sqrt{d}))\vec{1}^\mathsf{T} x)$. Then*

$$\inf\{\|g\|_\mathcal{R} : g \in \text{Ridge}_d,\ \|g - f\|_{L^\infty(\nu)} \le 1/2\} = \Omega(\sqrt{d}/\rho^2).$$

Proof. We prove the claim by a reduction to Theorem 4.15. That is, we show that an interpolant with better $\mathcal{R}$-norm than the bound stipulates can be used to construct a neural network that contradicts Theorem 4.15.

To do so, we consider a lower dimension $d' = 4\lfloor d/4q \rfloor - 4$ and create a mapping from points $z \in \{-1, 1\}^{d'}$ to $x_z \in \{-1, 1\}^d$. We define $a \in [0, 4q-1]$ such that $2a \equiv d \pmod{4q}$. For any z, we define x_z as follows:

$$x_z = (\underbrace{z_1, \ldots, z_1}_{q}, \ldots, \underbrace{z_{d'}, \ldots, z_{d'}}_{q}, \underbrace{1, \ldots, 1}_{a}, \underbrace{-1, \ldots, -1}_{d - d'q - a}).$$

Observe that

$$\vec{1}^\mathsf{T} x_z = q\vec{1}^\mathsf{T} z + 2a - d + d'q \equiv q\vec{1}^\mathsf{T} z \pmod{4q}.$$

Due to the periodicity of cosine and the fact that d' is a multiple of 4,

$$\cos(\frac{2\pi}{\rho} v^\mathsf{T} x_z) = \cos(\frac{\pi}{2}\vec{1}^\mathsf{T} z) = \text{Par}(z).$$

Consider some $g(x) = \phi(w^\mathsf{T} x)$ with $\|g - \cos(\frac{2\pi}{\rho} v^\mathsf{T} \cdot)\|_\infty \le \frac{1}{2}$. Define $w' \in \mathbb{R}^{d'}$ such that

$w'_i := \sum_{j=1}^{q} w_{(i-1)q+j}$. Observe that $\|w'\|_2 \leq \sqrt{q}$ and that $w^\mathsf{T} x_z = w'^\mathsf{T} z + c_w$, where c_w depends only on the remaining elements of w. Define $\tilde{g}(z) = \phi(w'^\mathsf{T} z + c_w)$. Then,

$$|\tilde{g}(z) - \chi(z)| = |\phi(w^\mathsf{T} x_z) - \cos(\frac{2\pi}{\rho} v^\mathsf{T} x_z)| \leq \frac{1}{2}$$

for all $z \in \{-1, 1\}^{d'}$. Since translation can only decrease the $\mathcal{R}$-norm (by exhausting some neurons to effectively behave linearly in the domain) namely, $\|\tilde{g}\|_\mathcal{R} \leq \|w'\|_2 \|\phi'\|_{\mathrm{TV}} = \|w'\|_2 \|g\|_\mathcal{R}$, Theorem 4.15 implies that $\|g\|_\mathcal{R} = \Omega(d'^{3/2}/\sqrt{q})$. The theorem statement follows by plugging in q and d'. □

4.6 An alternative variational norm

This chapter considers the approximation and generalization implications of bounding the complexity of shallow neural networks with the $\mathcal{R}$-norm. However, $\mathcal{R}$-norm is not the only weight-based complexity measurement, and other works employ slightly different norms for similar purposes. This section demonstrates that our results are not peculiarities of our formulation of $\mathcal{R}$-norm and extend to other variational norms. One alternative—which we refer to as the $\mathcal{V}_2$-*norm*—omits the linear component of the neural network whose measure determines the $\mathcal{R}$-norm and instead permits ReLU neurons whose thresholds lie outside the domain $\mathbb{B}^d$.

We first introduce notation for an infinite-width neural network that permits such thresholds. Let $\mathcal{M}'$ denote the space of probability measures over $\mathbb{S}^{d-1} \times [-2\sqrt{d}, 2\sqrt{d}]$. For some measure $\tilde{\mu} \in \mathcal{M}'$, let $\tilde{g}_\mu : \mathbb{B}^d \to \mathbb{R}$ be an infinite-width neural network with

$$\tilde{g}_{\tilde{\mu}}(x) = \int_{\mathbb{S}^{d-1} \times [-2\sqrt{d}, 2\sqrt{d}]} \mathrm{ReLU}(w^\mathsf{T} x + b) \, \tilde{\mu}(\mathrm{d}w, \mathrm{d}b),$$

which has total variation norm $|\tilde{\mu}| = \int_{\mathbb{S}^{d-1} \times [-2\sqrt{d}, 2\sqrt{d}]} |\tilde{\mu}| \, (\mathrm{d}w, \mathrm{d}b)$. Now, we introduce the

$\mathcal{V}_2$-norm for some $g : \mathbb{B}^d \to \mathbb{R}$:

$$\|g\|_{\mathcal{V}_2} = \inf_{\tilde{\mu} \in \mathcal{M}'} |\tilde{\mu}| \quad \text{s.t.} \quad g(x) = \tilde{g}_{\tilde{\mu}}(x), \quad \forall x \in \mathbb{B}^d. \tag{$\mathcal{V}_2$-norm}$$

In the same spirit as Lemma 4.5, for a discrete network $g(x) = g_\theta(x)$ with

$$\theta = (a^{(j)}, w^{(j)}, b^{(j)})_{j \in [m]} \in (\mathbb{R} \times \mathbb{S}^{d-1} \times [-2\sqrt{d}, 2\sqrt{d}])^m,$$

we have $\|g\|_{\mathcal{V}_2} \leq \|a\|_1$.

Our definition of the $\mathcal{V}_2$-norm was introduced by Siegel and Xu (2021) as the norm corresponding to their variation space $\mathbb{P}_1$ with constants $c_1 = -2\sqrt{d}$ and $c_2 = 2\sqrt{d}$. They relate the $\mathcal{V}_2$-norm to the *Barron norm* of E, Ma, and Wu (2019) and the Radon norm of Ongie et al. (2019). We show that the $\mathcal{V}_2$-norm and the $\mathcal{R}$-norm are closely related and that all of our bounds apply equivalently to the $\mathcal{V}_2$-norm. We first place upper and lower bounds on the $\|g\|_{\mathcal{V}_2}$ in terms of $\|g\|_{\mathcal{R}}$ and then explain why each of our results transfers to this new variational norm.

Theorem 4.35. *Suppose $g : \mathbb{B}^d \to \mathbb{R}$ has $\|g\|_{\mathcal{R}} < \infty$. Then, $\|g\|_{\mathcal{R}} \leq \|g\|_{\mathcal{V}_2}$. If g is bounded near the origin (i.e., $|g(x)| \leq K$ for all x with $\|x\|_2 \leq 1$), then $\|g\|_{\mathcal{V}_2} \leq 12 \|g\|_{\mathcal{R}} + 18K$.*

As a result, all of our results that apply to the $\mathcal{R}$-norm translate modulo constants to the $\mathcal{V}_2$-norm. Because $\|g\|_{\mathcal{V}_2} \geq \|g\|_{\mathcal{R}}$ always holds, every theorem that places lower bounds on an $\mathcal{R}$-norm exactly translates to $\|g\|_{\mathcal{V}_2}$, including Theorems 4.15, 4.20, 4.23, and 4.30. The upper-bounds hold up to constants by observing that every target function we consider is bounded by some K on $\mathbb{B}^d$.

- Because every sawtooth $s_{w,t}$ is bounded by 1, the averages of sawtooths g in Theorems 4.17 and 4.18 are bounded by $K = \frac{1}{q} = O(\sqrt{d})$. Hence, $\|g\|_{\mathcal{V}_2} = O(d)$, just like $\|g\|_{\mathcal{R}}$.

- For the "cap" construction $\mathbf{g}$ of Theorem 4.27, there are $k = O(n \log(d)/d)$ neurons, none of which are active at the origin. Their biases are negative and—under the "good event"—their weight norms are $O(1/\log d)$. Thus, no neuron can output a value greater than $O(1/\log d)$, so even if all k neurons activate, every x with $\|x\|_2 \le 1$ has $|\mathbf{g}(x)| = O(n/d)$, which is dominated by the $\mathcal{R}$-norm of $O(n\sqrt{\log d}/d)$.
- The construction g of Theorem 4.31 computes an average of functions bounded on $[-1, 1]$. Therefore, $\mathbf{g}$ is bounded by 1, and its $\mathcal{V}_2$-norm is no more than its $\mathcal{R}$-norm.

Proof of Theorem 4.35. We show separately that $\|g\|_{\mathscr{V}_2} \ge \|g\|_{\mathcal{R}}$, and then that $\|g\|_{\mathscr{V}_2} \le 12\|g\|_{\mathcal{R}} + 18K$ under the additional hypothesis that $|g(x)| \le K$ for all $x \in \mathbb{B}^d$ such that $\|x\|_2 \le 1$.

Lower bound on $\mathcal{V}_2$-norm: Fix any $\xi > 0$. By the definition of $\mathcal{V}_2$-norm, there exists $\tilde{\mu} \in \mathcal{M}'$ such that $g(x) = \tilde{g}_{\tilde{\mu}}(x)$ for all $x \in \mathbb{B}^d$ and $|\tilde{\mu}| \le \|g\|_{\mathscr{V}_2} + \xi$.[6] We show that there exists g_μ (where μ is $\tilde{\mu}$ with the support of b restricted to $[-\sqrt{d}, \sqrt{d}]$), v, and c such that $\tilde{g}_{\tilde{\mu}}(x) = g_\mu(x) + v^\mathsf{T} x + c$ for all $x \in \mathbb{B}^d$. Observe that for any $x \in \mathbb{B}^d$, $w^\mathsf{T} x + b > 0$ if $b > \sqrt{d}$ and $w^\mathsf{T} x + b < 0$ if $b < -\sqrt{d}$.

$$
\begin{aligned}
\tilde{g}_{\tilde{\mu}}(x) &= \int_{\mathbb{S}^{d-1}\times[-2\sqrt{d},-\sqrt{d}]} \mathrm{ReLU}(w^\mathsf{T} x + b)\tilde{\mu}(\mathrm{d}w, \mathrm{d}b) + \int_{\mathbb{S}^{d-1}\times[-\sqrt{d},\sqrt{d}]} \mathrm{ReLU}(w^\mathsf{T} x + b)\tilde{\mu}(\mathrm{d}w, \mathrm{d}b) \\
&\quad + \int_{\mathbb{S}^{d-1}\times[\sqrt{d},2\sqrt{d}]} \mathrm{ReLU}(w^\mathsf{T} x + b)\tilde{\mu}(\mathrm{d}w, \mathrm{d}b) \\
&= 0 + \int_{\mathbb{S}^{d-1}\times[-\sqrt{d},\sqrt{d}]} \mathrm{ReLU}(w^\mathsf{T} x + b)\mu(\mathrm{d}w, \mathrm{d}b) + \int_{\mathbb{S}^{d-1}\times[\sqrt{d},2\sqrt{d}]} (w^\mathsf{T} x + b)\tilde{\mu}(\mathrm{d}w, \mathrm{d}b) \\
&= g_\mu(x) + \sum_{i=1}^{d} x_i \underbrace{\int_{\mathbb{S}^{d-1}\times[\sqrt{d},2\sqrt{d}]} w_i \tilde{\mu}(\mathrm{d}w, \mathrm{d}b)}_{:= v_i} + \underbrace{\int_{\mathbb{S}^{d-1}\times[\sqrt{d},2\sqrt{d}]} b\tilde{\mu}(\mathrm{d}w, \mathrm{d}b)}_{:= c} \\
&= g_\mu(x) + v^\mathsf{T} x + c.
\end{aligned}
$$

As a result, $\|g\|_{\mathcal{R}} \le |\mu| \le |\tilde{\mu}| \le \|g\|_{\mathscr{V}_2} + \xi$. Because the argument holds simultaneously for all $\xi > 0$, we conclude that $\|g\|_{\mathcal{R}} \le \|g\|_{\mathscr{V}_2}$.

[6] This relies on the assumption that $\|g\|_{\mathscr{V}_2} < \infty$, but if it is not, then the claim trivially follows because $\|g\|_{\mathcal{R}} < \infty$.

Upper bound on $\mathcal{V}_2$-norm: By Proposition 4.3, there exist $\mu \in \mathcal{M}'$, $v \in \mathbb{R}^d$, and $c \in \mathbb{R}$ such that $g(x) = g_\mu(x) + v^\mathsf{T} x + c$ for all $x \in \mathbb{B}^d$ and $|\mu| = \|g\|_\mathcal{R}$. We construct $\tilde{\mu} \in \mathcal{M}'$ such that $g_\mu(x) + v^\mathsf{T} x + c = \tilde{g}_{\tilde{\mu}}(x)$ for all $x \in \mathbb{B}^d$:[7]

$$\tilde{\mu}(w,b) = \begin{cases} \mu(w,b) & \text{if } b \in [-\sqrt{d}, \sqrt{d}]; \\ \left(-3\|v\|_2 + \frac{2c}{\sqrt{d}}\right) \delta\left((w,b) - \left(v, 2\sqrt{d}\right)\right) & \\ \quad + \left(4\|v\|_2 - \frac{2c}{\sqrt{d}}\right) \delta\left((w,b) - \left(v, \frac{3}{2}\sqrt{d}\right)\right) & \text{otherwise.} \end{cases}$$

Fix any $x \in \mathbb{B}^d$. Then:

$$\begin{aligned} \tilde{g}_{\tilde{\mu}}(x) - g_\mu(x) &= \left(-3\|v\|_2 + \frac{2c}{\sqrt{d}}\right) \mathrm{ReLU}\left(\frac{v^\mathsf{T}}{\|v\|_2} x + 2\sqrt{d}\right) \\ &\quad + \left(4\|v\|_2 - \frac{2c}{\sqrt{d}}\right) \mathrm{ReLU}\left(\frac{v^\mathsf{T}}{\|v\|_2} x + \frac{3}{2}\sqrt{d}\right) \\ &= \left(-3\|v\|_2 + \frac{2c}{\sqrt{d}}\right)\left(\frac{v^\mathsf{T}}{\|v\|_2} x + 2\sqrt{d}\right) + \left(4\|v\|_2 - \frac{2c}{\sqrt{d}}\right)\left(\frac{v^\mathsf{T}}{\|v\|_2} x + \frac{3}{2}\sqrt{d}\right) \\ &= v^\mathsf{T} x + c. \end{aligned}$$

Therefore, $|\tilde{\mu}| \le |\mu| + |-3\|v\|_2 + \frac{2c}{\sqrt{d}}| + |4\|v\|_2 - \frac{2c}{\sqrt{d}}| \le |\mu| + 7\|v\|_2 + \frac{4|c|}{\sqrt{d}}$. It suffices to bound $\|v\|_2$ and $|c|$.

- Let $x_0 \coloneqq \frac{v}{\|v\|_2}$. By boundedness, the triangle inequality, and several applications of Holder's inequality:

$$\begin{aligned} |g(x_0) - g(0)| &\ge \left|v^\mathsf{T} x_0\right| - |g_\mu(x_0) - g_\mu(0)| \\ &= \|v\|_2 - \left|\int_{\mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}]} (\mathrm{ReLU}(w^\mathsf{T} x_0 + b) - \mathrm{ReLU}(b)) \mu(\mathrm{d}w, \mathrm{d}b)\right| \\ &\ge \|v\|_2 - \int_{\mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}]} \left|\mathrm{ReLU}(w^\mathsf{T} x_0 + b) - \mathrm{ReLU}(b)\right| |\mu| (\mathrm{d}w, \mathrm{d}b) \\ &\ge \|v\|_2 - \|x_0\|_2 |\mu| = \|v\|_2 - |\mu|. \end{aligned}$$

[7] In the event that $v = \vec{0}$, we use $\frac{v}{\|v\|_2} \coloneqq \frac{1}{\sqrt{d}}\vec{1} \in \mathbb{S}^{d-1}$.

Hence, $\|v\|_2 \leq |g(x_0) - g(0)| + |\mu| \leq 2K + |\mu|$.

- We similarly employ our bound on $g(0)$:

$$K \geq |g(0)| \geq |c| - \left| \int_{\mathbb{S}^{d-1} \times [-\sqrt{d}, \sqrt{d}]} \mathrm{ReLU}(b) \mu(\mathrm{d}w, \mathrm{d}b) \right| \geq |c| - |\mu| \sqrt{d}.$$

  As a result, $|c| \leq K + |\mu| \sqrt{d}$.

Therefore, $\|g\|_{\mathcal{V}_2} \leq |\tilde{\mu}| \leq 12 |\mu| + 18K \leq 12 \|g\|_{\mathcal{R}} + 18K$. □

4.7 Conclusion

In this work, we shed light on the $\mathcal{R}$-norm inductive bias for learning neural networks, but numerous questions remain. We are particularly interested in understanding the solutions to (VP) for other datasets, as well as the generality of the averaging techniques used in our constructions. Extensions of the $\mathcal{R}$-norm to deeper networks and the analysis of solutions to (VP) for other high dimensional datasets could also be useful for proving depth-separation results that focus on the variational norm. Progress in this direction would complement existing research on bounded-width approximation (Telgarsky, 2016; Eldan and Shamir, 2016; Martens et al., 2013; Daniely, 2017b; Safran and Shamir, 2017; Safran, Eldan, and Shamir, 2019). Finally, our work suggests that minimizing $\mathcal{R}$-norm yields neural networks that are intrinsically high-dimensional, and we are interested in whether this phenomenon is borne out in architectures beyond two-layer fully-connected networks.

This chapter concludes the first part of this thesis, which studies the representational properties of various feed-forward neural network topologies and training regimes. Moreso than their predecessors, the contributions of this chapter establish a link between representational trade-offs of neural architectures and their generalization properties. While Chapters 2 and 3 characterize the hardness of efficiently representing certain single-index functions with a variety of neural networks, this chapter provides an example of a surprising representational effect that occurs among training datasets with low intrinsic dimensionality. This

work leaves a blueprint for research on the representational implications of other inductive biases and their interplay with generalization.

The remaining chapters of the dissertation focus primarily on the representational properties of transformers, with less emphasis on generalization and inductive bias. However, the flavor of this chapter's results suggests a future direction for a wave of theoretical work on transformers, which could identify architecture-specific inductive biases and their implications for generalization.

Chapter 5: Associative capabilities of multi-headed attention layers

Attention layers, as commonly used in transformers, form the backbone of modern deep learning, yet there is no mathematical description of their benefits and deficiencies over other architectures. This chapter establishes both positive and negative results on the representation power of attention layers, with a focus on intrinsic complexity parameters such as width, depth, and embedding dimension. On the positive side, we present a *sparse averaging task*, where recurrent networks and feedforward networks all have complexity scaling polynomially in the input size, whereas transformers scale merely *logarithmically* in the input size; furthermore, we use the same construction to show the necessity and role of a large embedding dimension in a transformer. On the negative side, we present a *triple detection* task, where attention layers in turn have complexity scaling linearly in the input size; as this scenario seems rare in practice, we also present natural variants that can be efficiently solved by attention layers. The proof techniques introduce communication complexity to the analysis of transformers and related models. They further establish the role of sparse averaging as a prototypical attention task, which even finds use in the analysis of triple detection.

The research presented in this chapter reflects the work of Sanford, Hsu, and Telgarsky (2023).

5.1 Introduction

In recent years, transformer networks (Vaswani et al., 2017) have been established as a fundamental neural architecture powering state-of-the-art results in many applications, including language modeling (OpenAI, 2023), computer vision (Dosovitskiy et al., 2021),

and protein folding (Jumper et al., 2021). The key building block of transformer models is the *self-attention unit*, a primitive that represents interactions among input elements as inner products between low-dimensional embeddings of these elements.

The success of transformer models is linked to their ability to scale their training and generalization performance to larger datasets and sequence lengths. Their representational capacity, however, underlies this scaling power, and is tied to the inductive biases of their learning algorithms. Empirically, transformer models trained with gradient-based learning algorithms exhibit biases towards certain algorithmic primitives (Edelman et al., 2022; Liu et al., 2022) and learn representations that may encode domain-specific information in the self-attention units (Clark et al., 2019; Hewitt and Manning, 2019; Rogers, Kovaleva, and Rumshisky, 2020; Chen et al., 2022). These examples indicate that transformer architectures not only provide computational benefits but also have representational capabilities that are particularly well-matched to practical tasks.

In this paper, we investigate these inductive biases by identifying "natural" computational tasks for which transformers are well-suited, especially compared to other neural network architectures, as well as tasks that highlight the limitations of transformers. The tasks—sparse averaging, pair-matching, and triples-matching—represent primitive operations that aggregate structural information encoded in embeddings. We use these tasks to elucidate the relationship between the embedding dimension m of a self-attention unit and its expressivity, and showcase the fundamental representational limitations of self-attention layers.

In our model, the primary computational bottleneck faced by a transformer in computing a "sequence-to-sequence"[1] function $f\colon \mathcal{X}^N \to \mathcal{Y}^N$ is the constrained processing of pairs of input elements $\{x_i, x_j\} \in \binom{\mathcal{X}}{2}$; we allow transformers unbounded computational power when processing the individual elements $x_i \in \mathcal{X}$. This is motivated by modern scaling regimes where the context length N has rapidly increased, the self-attention embedding dimension m

[1]Note, however, that attention units are permutation equivariant, so the order of elements in the input "sequence" $X \in \mathcal{X}^N$ is irrelevant. In practice, *positional encodings* are used when the sequence order is relevant.

remains much smaller than N, and the parameterization of multi-layer perceptrons (MLPs) that operate on individual elements is much larger than m. Indeed, the largest GPT-3 model (Brown et al., 2020) features a context length $N = 2048$, an embedding dimension $m = 128$, and MLPs with a 12288-dimensional parameterization; the context length of GPT-4 is as large as $N = 32000$. As such, we are interested in the capabilities of transformers with $N^{o(1)}$ total "size", as opposed to $N^{\Omega(1)}$. The nature of the bottleneck in our model makes the tools of communication complexity indispensable for formalizing computational limits.

5.1.1 Our contributions

Sparse averaging separations among atomic self-attention units. The *q-sparse averaging task* qSA aims to capture the essential approximation-theoretic properties of self-attention units. In qSA, the ith input x_i is a pair (y_i, z_i), where $z_i \in \mathbb{R}^{d'}$ is the *data* part of x_i, simply a vector in $\mathbb{R}^{d'}$, and $y_i \in \binom{[N]}{q}$ is the *indexing* part, which specifies q locations in the input sequence; the ith output element in qSA is obtained by averaging the q *data* parts z_j given by $j \in y_i$, meaning

$$q\mathrm{SA}\left((y_1, z_1), \ldots, (y_N, z_N)\right) = \left(\frac{1}{q}\sum_{j \in y_1} z_j, \ldots, \frac{1}{q}\sum_{j \in y_N} z_j\right).$$

(See also Definition 5.4.) As summarized in the following informal theorem, our analysis of qSA in Sections 5.3 and 5.4 illustrates the ability of the self-attention primitive to associate arbitrary subsets of input elements (as opposed to just "local" subsets, as specified by some sequential/topological structure), measures the expressive power accrued by increasing the embedding dimension m of a self-attention unit, and indicates the representational limitations of "traditional" neural architectures on basic computational tasks.

Informal Theorem 5.1. *The task* qSA *for* $q \in \mathbb{Z}_+$ *satisfies the following properties (see Definition 5.4 for a formal definition and approximation metric).*

1. *There exists a unit of self-attention* f *with an* m*-dimensional embedding that approxi-*

mates qSA *if and only if* $m \gtrsim q$ *(Theorems 5.4 and 5.6).*

2. *Any fully-connected neural network whose output approximates* qSA *requires its first hidden layer to have width at least* $\Omega(Nd)$ *(Theorem 5.14).*

3. *Any recurrent neural network whose iterates approximate* qSA *requires a hidden state of at least* $\Omega(N)$ *bits (Theorem 5.15).*

We consider the qSA implementation in Item 1 *efficient* since the dimension of the model parameters grows with $\text{poly}(q, d, \log N)$, whereas the latter two are *inefficient* since their parameter (or state) dimension grows as $\text{poly}(N)$. The proofs of the positive results employ embeddings for each index j and each subset y_i that have large inner products if and only if $j \in y_i$. The negative results involve communication complexity reductions and geometric arguments. These arguments naturally introduce a dependence on bits of precision, which we suppress above within the notation "$\gtrsim$"; we note that these bounded-precision results are arguably more relevant to modern networks, which use as few as 4 or even 2 bits of numerical precision.

Contrast between pairwise and triple-wise matching with self-attention layers. We frame standard transformer architectures as being able to efficiently represent functions that are decomposable into sparse pairwise interactions between inputs. To do so, we introduce two sequential tasks and prove a collection of constructions and hardness results that characterize the abilities of transformers to solve these tasks.

Given an input sequence $X = (x_1, \ldots, x_N) \in [M]^N$ (for some $M = \text{poly}(N)$), we formalize the problems of *similar pair detection* (Match2) and *similar triple detection* (Match3) as

$$\text{Match2}(X)_{i\in[N]} = \mathbb{1}\left\{\exists j \text{ s.t. } x_i + x_j = 0 \pmod M\right\}, \tag{5.1}$$

$$\text{Match3}(X)_{i\in[N]} = \mathbb{1}\left\{\exists j_1, j_2 \text{ s.t. } x_i + x_{j_1} + x_{j_2} = 0 \pmod M\right\}. \tag{5.2}$$

For both tasks, note that the output is an N-dimensional vector whose ith element is 1 if

and only if the sequence X includes a pair or triple *containing* x_i. In this sense, the problems differ from 2SUM and 3SUM, which are not sequence-to-sequence tasks.

We believe these two tasks are intrinsically "pairwise" and "triple-wise", respectively; moreover, since we also believe self-attention performs a fundamentally "pairwise" operation, we will use Match2 and Match3 to show a sharp gap in the representation power of self-attention.

Informal Theorem 5.2.

1. *A single unit of standard self-attention with input and output MLPs and an $O(d)$-dimensional embedding can compute* Match2 *(Theorem 5.16).*

2. *A single layer of standard multi-headed self-attention cannot compute* Match3 *unless its number of heads H or embedding dimension m grows polynomially in N (Theorem 5.17).*

3. *A standard transformer model* can *efficiently compute a modified version of* Match3 *that makes assumptions about embedding structure or locality (Theorems 5.18 and 5.19).*

4. *Under a generalized notion of "third-order tensor self-attention" introduced in Section 5.5.4,* Match3 *is efficiently computable with a single unit of third-order attention (Theorem 5.20).*

While the above result demonstrates the limitations of multi-headed self-attention and illustrates the importance of learning embeddings with contextual clues, we believe a stronger result exists. Specifically, we conjecture that even multi-layer transformers cannot efficiently compute Match3 without hints or augmentation.

Informal Conjecture 5.3. Every multi-layer transformer that computes Match3 must have width, depth, embedding dimension, or bit complexity at least $N^{\Omega(1)}$.

In Appendices 5.5.6 and 5.5.7, we give a heuristic information-theoretic argument to support this conjecture, prove a matching upper-bound, and finally, prove analogous results for graph-augmented transformers applied to the undirected and directed cycle detection problems.

5.1.2 Related work

Several computational and learning-theoretic aspects of transformers, distinct from but related to the specific aims of the present paper, have been mathematically studied in previous works.

Universality and Turing-completeness. To demonstrate the power of transformers, universal approximation results for transformers (Yun et al., 2020; Wei, Chen, and Ma, 2022)—analogous to results for feedforward networks (Hornik, Stinchcombe, and White, 1989)—establish the capability for sufficiently large networks to accurately approximate general classes of functions. Note, however, that the precise minimal dependence of the required size (e.g., number of attention units, depth of the network) as a function of the input size N does not directly follow from such results, and it is complicated by the interleaving of other neural network elements between attention layers. (Approximate) Turing-completeness of transformers demonstrates their power differently, and such results have been established, first assuming infinite precision weights (Pérez, Marinković, and Barceló, 2019) and later with finite-precision (Wei, Chen, and Ma, 2022). Such results are more closely aligned with our aims because Turing machines represent a uniform model of computation on inputs of arbitrary size. Wei, Chen, and Ma (2022) showed that Turing machines that run for T steps can be approximated by "encoder-decoder" transformers of depth $\log(T)$ and size polynomial in $\log(T)$ and the number of states of the Turing machine (but the decoder runs for T steps).

Formal language recognition. The ubiquity of transformers in natural language understanding has motivated the theoretical study of their ability to recognize formal languages.

On the positive side, Bhattamishra, Ahuja, and Goyal (2020) constructed transformers that recognize counter languages, and Yao et al. (2021) showed that transformers of bounded size and depth can recognize Dyck languages of bounded stack depth. Liu et al. (2022) showed that the computations of finite-state automata on sequences of length N can be performed by transformers of depth $\log(N)$ and size polynomial in the number of states. On the negative side, Hahn (2020) showed limitations of modeling distributions over formal languages (including Dyck) with fixed-size transformers (though this result does not imply quantitative lower bounds on the size of the transformer). Hahn (2020), as well as Hao, Angluin, and Frank (2022), also establish the inability of "hard attention" Transformers to recognize various formal languages and circuit classes by leveraging depth reduction techniques from circuit complexity (Furst, Saxe, and Sipser, 1984).

Learnability. The sample complexity of learning with low-weight transformers can be obtained using techniques from statistical learning theory and, in turn, establish learnability of certain boolean concept classes (e.g., sparse parity) (Edelman et al., 2022; Bhattamishra et al., 2022) using transformer-based hypothesis classes. Our qSA function is inspired by these classes, and we establish concrete size lower bounds for approximation (and hence also learnability) by transformers. We note that our constructions use bounded-size weights, so in principle, the aforementioned sample complexity bounds combined with our expressivity results provide an upper bound on the sample complexity of empirical risk minimization for transformers. Prior work of Likhosherstov, Choromanski, and Weller (2021) also shows how sparse attention patterns can be achieved by self-attention units (via random projection arguments); however, when specialized to qSA, their construction is suboptimal in terms of the sparsity level q.

Related models. Graph neural networks (GNNs), like transformers, process very large inputs (graphs) using neural networks that act only on small collections of the input parts (vertex neighborhoods). Many classes of GNNs are universal approximators for classes of

invariant and equivariant functions (Maron et al., 2019; Keriven and Peyré, 2019). At the same time, they are restricted by the distinguishing power of certain graph isomorphism tests (Xu et al., 2018; Morris et al., 2019; Chen et al., 2019), and lower bounds have been established on the network size to approximate such tests (Aamand et al., 2022). Loukas (2019) established a connection between GNNs and the Local (Angluin, 1980) and Congest (Peleg, 2000) models for distributed computation, and hence directly translates lower bounds for Congest—notably cycle detection problems—into size lower bounds for GNNs. Our lower bounds for cycle detection using transformers also leverage a connection to the Congest model. However, transformers do not have the same limitations as GNNs, since the computational substrate of a transformer does not depend on the input graph in the way it does with GNNs. Thus, we cannot directly import lower bounds for Congest to obtain lower bounds for transformers.

Transformers are also related to other families of invariant and equivariant networks. Our focus on Match2 and Match3 (and related problems) was inspired by the separation results of Zweig and Bruna (2022) between models for processing sets: Deep Sets (Qi et al., 2017; Zaheer et al., 2017), which are "singleton symmetric", and the more expressive Relational Pooling networks (Santoro et al., 2017), which are only "pairwise symmetric".

5.1.3 Conclusion and future work

Our primary contributions are to present a multi-faceted story about transformer approximation: First, qSA separates transformer models approximation-theoretically from RNNs and MLPs. Because the minimum embedding dimension of an attention unit that computes qSA scales linearly with q, qSA furthermore witnesses a fine-grained characterization of transformer representation power as a function of the embedding dimension. Second, while single units of self-attention can solve the Match2 task, even wide layers of self-attention with high-dimensional embeddings cannot solve Match3, and we believe that deeper models cannot as well. This question of deeper models is stated as a formal conjecture and ad-

dressed heuristically in Section 5.5.7, using both information- and communication-theoretic proof techniques, both of which we feel are significant steps towards a complete proof.

While our investigation is purely approximation-theoretic, we also include in Section 5.3.4 a preliminary empirical study, showing that attention can learn qSA with vastly fewer samples than recurrent networks and MLPs; we feel this further emphasizes the fundamental value of qSA, and constitutes an exciting direction for future work.

Beyond the explicit open question in Informal Conjecture 5.3, we anticipate that future research could connect the separation results proved in this work to formal linguistic theory and empirical work on attention matrix interpretation. This work examines Match2 and Match3 because we believe that the former could represent a key primitive for language processing tasks such as co-referencing, while the latter represents a natural extension of the former that likely is *not* necessary for language modeling. Rather, it may be possible that language modeling performs triple-wise modeling for tasks such as the identification of subject, verb, and object components by relying on pairwise matching constructions and "clues" learned within an embedding, such as those encoded in the toy problems Match3Bigram and Match3Local. That is, transformers serve as a useful foundational model for language modeling because of their abilities to integrate contextual clues and pairwise communication, and while they are not extensible to "purely triple-wise problems," most practical sequential problems have some efficient decomposition to pairwise structures that can be easily exploited by these architectures. Future work by linguists, theoretical computer scientists, and empirical NLP practitioners could assess how foundational our primitives are and study whether there are any practical triple-wise problems that transformer models fail to solve.

5.2 Preliminaries

Let $\mathbb{B}^d = \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$ denote the unit ball in $\mathbb{R}^d$, and let $[n] = \{1, 2, \ldots, n\}$ denote the first n positive integers. The expression $\mathbb{1}\{P\}$ equals 1 if predicate P is true and

0 otherwise. The row-wise softmax operator applied to matrix $A \in \mathbb{R}^{N \times M}$ returns

$$\text{softmax}(A)_{i,j} = \frac{\exp(A_{i,j})}{\sum_{j'=1}^{M} \exp(A_{i,j'})}.$$

5.2.1 Attention units and transformer architectures

We first introduce the concept of self-attention, which is used as the building block of all transformer architectures included in this paper.

Definition 5.1. For input dimension d, output dimension d', embedding dimension m, precision p, and matrices $Q, K \in \mathbb{R}^{d \times m}$ and $V \in \mathbb{R}^{d \times d'}$ (encoded using p-bit fixed-point numbers), a *self-attention unit* is a function $f_{Q,K,V} : \mathbb{R}^{N \times d} \to \mathbb{R}^{N \times d'}$ with

$$f_{Q,K,V}(X) = \text{softmax}(XQK^{\mathsf{T}}X^{\mathsf{T}})XV.$$

Let $\mathsf{Attn}_{d,m,d',p} = \{f_{Q,K,V} : Q, K, V\}$ denote all such self-attention units.

Self-attention units can be computed in parallel to create multi-headed attention.

Definition 5.2. For head-count H and self-attention units $f_1, \ldots, f_H \in \mathsf{Attn}_{d,m,d',p}$, a *multi-headed attention layer* is a function $L_{f_1,\ldots,f_H} : \mathbb{R}^{N \times d} \to \mathbb{R}^{N \times m}$ with

$$L_{f_1,\ldots,f_H}(X) = \sum_{h=1}^{H} f_h(X).$$

Let $\mathsf{Attn}^{H}_{d,m,d',p}$ contain all such $L_{f_1,\ldots,f_H}$.

Transformer models are composed of two components: multi-headed attention layers (as above) and element-wise multi-layer perceptrons. Due to universal approximation results, we model multi-layer perceptrons as arbitrary functions mapping fixed-precision vectors to themselves.

Definition 5.3. A *multi-layer perceptron (MLP) layer* is represented by some $\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$, whose real-valued inputs and outputs can be represented using p-bit fixed-precision numbers. We apply ϕ to each element (i.e., row) of an input $X \in \mathbb{R}^{N \times d}$, abusing notation to let $\phi(X) = (\phi(x_1), \ldots, \phi(x_N)) \in \mathbb{R}^{N \times d'}$. Let $\Phi_{d,d',p}$ denote all such MLPs.

We concatenate the notation of each class of functions to denote function composition. For example, for output dimension d', we use $\mathsf{Attn}'_{d,m,d',p} := \mathsf{Attn}_{m,m,d',p}\Phi_{d,m,p}$ and $\mathsf{Attn}^{H\prime}_{d,m,d',p} := \mathsf{Attn}^{H}_{m,m,d',p}\Phi_{d,m,p}$ to represent single-headed and multi-headed attention units with an input MLP respectively. (The capabilities and limitations of these models are studied in Section 5.3.) For depth D, we let

$$\mathsf{Transformer}^{D,H}_{d,m,d',p} = \Phi_{m,d',p}(\mathsf{Attn}^{H\prime}_{m,m,m,p})^{D-1}\mathsf{Attn}^{H\prime}_{d,m,m,p}$$

represent a full transformer model comprising D layers of H-headed self-attention with interspersed MLPs.

While two key features of transformer architectures—the residual connection and the positional embedding—are conspicuously missing from this formalism, the two can be implemented easily under the framework. We can include a positional embedding by encoding the index as a coordinate of the input, i.e. $x_{i,1} = i$. Then, the subsequent MLP transformation $\phi(X)$ can incorporate i suitably into the embedding. A residual connection can be included additively as input to a multi-layer perceptron layer (as is standard) by implementing an "approximate identity" attention head f with Q, K and $V = I_m$ set to ensure that $f(X) \approx X$.[2]

We periodically consider transformers implemented with real-valued arithmetic with infinite bit complexity; in those cases, we omit the bit complexity p from the notation.

Finally, we assume for the proof of Theorem 5.5 that the model is permitted to append a single `<END>` token at the end of a sequence. That is, we say that a model $f \in \mathsf{Transformer}^{D,H}_{d,m,d',p}$

[2]A simple construction involves letting $XQ = XK$ with iid Gaussian columns fixed for every index i. Then, the diagonals of $XQK^\mathsf{T}X^\mathsf{T}$ are far larger than all other entries and its softmax is approximately I_N.

represents a target $h : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d'}$ if $f(X')_{1:N} = g(X)$ when $X' = (x_1, \ldots, x_N, x')$ for constant-valued $x' \in \mathbb{R}^d$.

5.3 Sparse averaging and self-attention embedding dimension

We present the sparse averaging task to highlight the ability of transformer architectures to simulate a wide range of meaningful interactions between input elements. This task demonstrates how the embedding dimension of a self-attention unit modulates the expressive capabilities of the architecture, while showcasing the inabilities of fully-connected and recurrent neural networks to capture similar interactions (see Section 5.4).

Definition 5.4. For sparsity q, problem dimension d', and input dimension $d = d' + q + 1$, consider an input $X = (x_1, \ldots, x_N) \in \mathbb{R}^{N\times d}$ with $x_i = (z_i; y_i; i)$ for $z_i \in \mathbb{B}^{d'}$ and $y_i \in \binom{[N]}{q}$.[3] Let the *q-sparse average* be

$$q\mathrm{SA}(X) = \left(\frac{1}{q}\sum_{j=1}^{q} z_{y_{i,j}}\right)_{i\in[N]}.$$

For accuracy $\epsilon > 0$, a function $f : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d'}$ *ϵ-approximates* qSA if for all X,

$$\max_{i\in[N]} \|f(X)_i - q\mathrm{SA}(X)_i\|_2 \le \epsilon.$$

Figure 5.1a visualizes the sparse averaging task as a bipartite graph between subsets y_i and elements z_i with corresponding averages. Theorems 5.4 and 5.6 jointly show that the minimum embedding dimension m of single self-attention units $\mathsf{Attn}'_{d,m,d',p}$ that $O(\frac{1}{q})$-approximate qSA scales linearly with q. We believe that the sparse averaging problem is thus a canonical problem establishing the representational capabilities and inductive biases of self-attention units.

Section 5.3.1 presents positive results that show that the sparse averaging task can be

[3] We may encode a q element subset of $[N]$ as a vector in $[N]^q$ constrained to have distinct components.

solved using self-attention units with embedding dimension m growing linearly with q. Section 5.3.2 states and proves nearly matching lower bounds on the embedding dimension m necessary to solve the sparse averaging task in the finite-precision setting. Section 5.3.3 shares a negative result pertaining to the restricted attention units in the infinite-precision setting. Section 5.3.4 presents empirical results that support the theoretical findings of this section by demonstrating that trained transformer models can indeed solve the sparse averaging task in an interpretable manner. Finally, Section 5.3.5 contains the technically involved proofs of the theorems in Section 5.3.1.

5.3.1 Self-attention can approximate qSA when $m \gtrsim q$

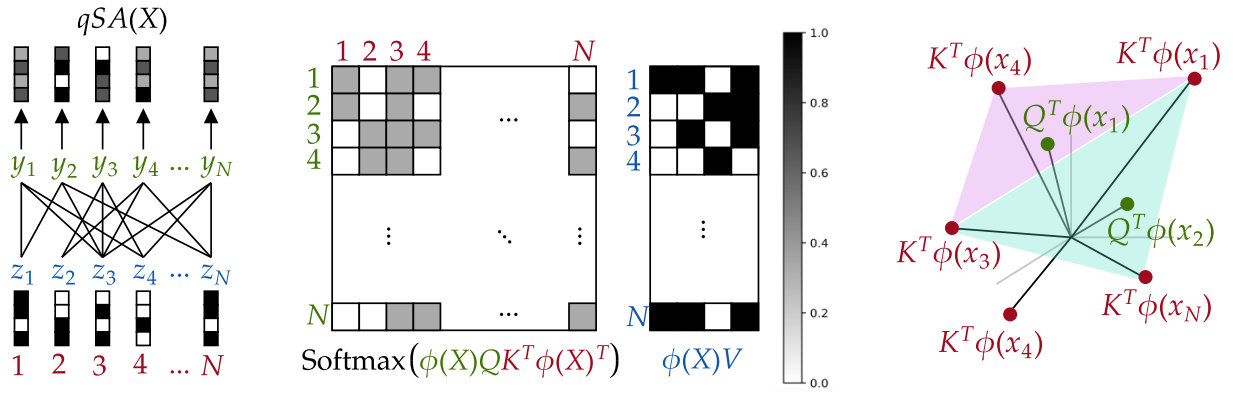
Our principle positive result shows that the sparse averaging task qSA can be approximately solved using fixed-precision arithmetic self-attention units with embedding dimension m growing with $q \log N$.

Theorem 5.4 (Fixed-precision)**.** *For any N, any $m \geq \Omega(d' + q \log N)$, any $\epsilon \in (0, 1)$, and $p = \Omega(\log(\frac{q}{\epsilon} \log N))$, there exists some $f \in \mathsf{Attn}'_{d,m,d',p}$ that ϵ-approximates qSA.*

While the full proof appears in Section 5.3.5.1, we briefly sketch the argument here. Because the output of a self-attention unit is a convex combination of rows of the value matrix $\phi(X)V \in \mathbb{R}^{N \times d'}$, a natural way to approximate qSA with a unit of self-attention is to let each value be the corresponding vector in the average (i.e. $V^\mathsf{T}\phi(x_i) = z_i$) and choose the key and query functions in order to ensure that the attention matrix satisfies

$$\operatorname{softmax}(\phi(X)QK^\mathsf{T}\phi(X)^\mathsf{T})_{i,j} \approx \begin{cases} \frac{1}{q} & \text{if } j \in y_i, \\ 0 & \text{otherwise.} \end{cases}$$

To do so, let each key $K^\mathsf{T}\phi(x_i)$ represent a fixed vertex on a convex polytope, which depends only on index i and is constructed from random binary vectors. We select each query $Q^\mathsf{T}\phi(x_i)$ to ensure that $\phi(x_i)^\mathsf{T}QK^\mathsf{T}\phi(x_j)$ is a fixed large value if $j \in y_i$ and a slightly smaller value

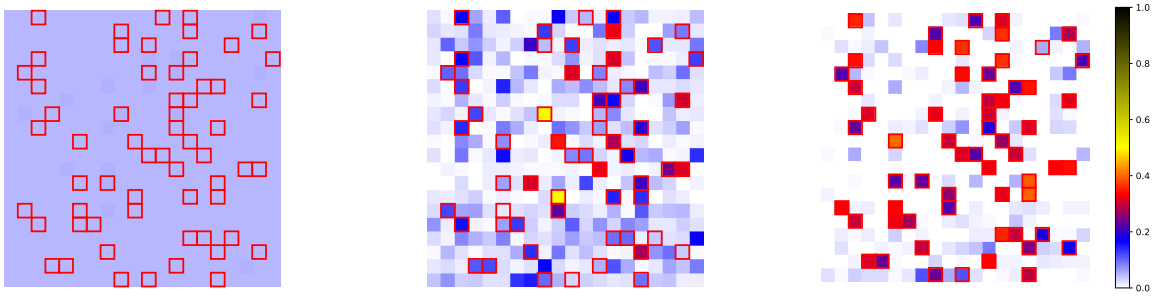


(a) Bipartite graph relating y_i and z_i in $q\mathrm{SA}(X)$.

(b) Attention and value matrices used for the self-attention construction of $q\mathrm{SA}(X)$ in Theorem 5.4.

(c) Key and query embeddings that produce the self-attention matrix in (b).

Figure 5.1: A visualization of the qSA function outputs given a sequence of inputs $(z_i; y_i; i)_{i\in[N]}$ as a bipartite graph between subsets y_i and vectors z_i (a), and of the attention matrix (b) and underlying embeddings (c) that produce the self-attention construction in Theorem 5.4.



(a) $T = 0$.

(b) $T = 1000$.

(c) $T = 40000$.

Figure 5.2: Attention matrix $\mathrm{softmax}(\phi(X)QK^{\mathsf{T}}\phi(X)^{\mathsf{T}}) \in \mathbb{R}^{20\times 20}$ for a fixed example after T epochs of training a self-attention unit to solve qSA for $q = 3$. Each row i corresponds to subset y_i, and each cell $j \in y_i$ is outlined in red. See Section 5.3.4 for experimental details.

otherwise. We obtain the precise query, key, and value embeddings by employing tools from dual certificate analysis from the theory of compressed sensing.

We visualize this construction in Figure 5.1b and 5.1c for $q = 3$ and $d' = 4$, which presents the associated attention and value matrices necessary for the construction, and plots a polytope of keys (red dots) with each face corresponding to each subset y_i (green dots). The construction is empirically relevant; Figure 5.2 shows that a unit of self-attention trained on data generated by the qSA task recovers a similar attention matrix to the one

stipulated in our construction and visualized in Figure 5.1b.

The logarithmic dependence of the embedding dimension m on the sequence length N can be eliminated by considering self-attention units with real-valued arithmetic with infinite bit complexity.

Theorem 5.5 (Infinite-precision)**.** *For fixed N, $m \geq \Omega(d' + q)$ and $\epsilon > 0$, there exists some $f \in \mathsf{Attn}'_{d,m,d'}$ that ϵ-approximates qSA.*

The proof of Theorem 5.5 employs a similar polytope-based construction in Section 5.3.5.2, relying on a cyclic polytope rather than one drawn from discrete boolean vectors. Theorem 5.8 proves the near-optimality of *that* bound by employing a geometric argument to show that a variant of qSA can only be approximated by a restricted family of self-attention units with a sufficiently high-dimensional embedding.

5.3.2 Self-attention cannot approximate qSA when $m \lesssim q$

We show that the construction used to prove Theorem 5.4 is nearly optimal.

Theorem 5.6. *For any sufficiently large q, any $N \geq 2q + 1$, and any $d' \geq 1$, there exists a universal constant c such that if $mp \leq cq$, then no $f \in \mathsf{Transformer}^{1,1}_{d,m,d',p}$ exists that $\frac{1}{2q}$-approximates qSA.*

(By choosing $p = O(\log(q \log N))$, Theorem 5.4 is shown to be optimal up to logarithmic factors of q and doubly-logarithmic factors of N.)

The proof of Theorem 5.6 employs a standard communication complexity argument based on a reduction from the following *set disjointness* problem in the two-party communication model, in which each party possesses a subset of an n element domain (encoded as n-bit strings), and they wish to jointly determine whether their subsets are disjoint. We note that communication complexity is commonly-used technique for proving lower bounds on the representational power of circuits and feedforward neural networks (see, e.g., Karchmer

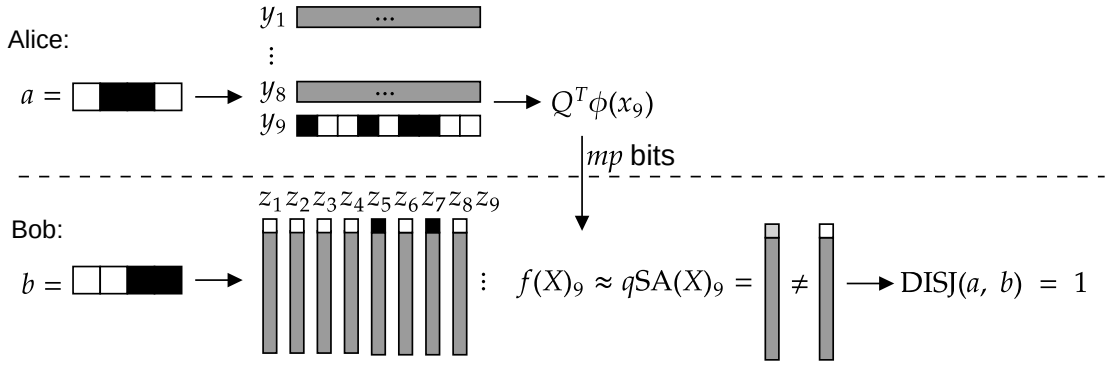


Figure 5.3: The mp-bit communication protocol used to reduce the hardness of computing qSA with a single unit of self-attention to the hardness of solving the DISJ communication problem for the proof of Theorem 5.6 for $q = 4$.

and Wigderson, 1988; Ben-David, Eiron, and Simon, 2002; Martens et al., 2013; Vardi et al., 2021).

Fact 5.7 (Set disjointness communication lower bound (Yao, 1979))**.** *Suppose Alice and Bob are given inputs $a, b \in \{0,1\}^n$, respectively, with the goal of jointly computing* $\text{DISJ}(a,b) = \max_i a_i b_i$ *by alternately sending a single bit message to the other party over a sequence of communication rounds. Any deterministic protocol for computing* $\text{DISJ}(a,b)$ *requires at least n rounds of communication.*

Our proof designs a communication protocol that Alice and Bob use to jointly compute $\text{DISJ}(a,b)$ when $n = q$ in $O(mp)$ rounds of communication, under the assumption that such an f exists that closely approximates qSA.

- Alice encodes her input a in a single subset by letting $y_{2q+1} = \{2i + a_i - 1 : i \in [q]\}$.
- Bob uses his input b to assign z_{2i-1} to $2b_i - 1$ and $z_{2i} = -1$ for all $i \in [q]$.
- All other input components are set to constant values known by both parties.

Alice sends her mp-bit query embedding $Q^\mathsf{T}\phi(x_{2q+1})$ bit-by-bit to Bob, who approximately computes qSA by determining the outcome of f. The crux of the reduction shows that

$q\mathrm{SA}(X)_{2q+1} = -1$ if and only if $a_i b_i = 0$ for all $i \in [q]$, which allows Bob to determine $\mathrm{DISJ}(a, b)$.

We visualize the protocol in Figure 5.3. The proofs of Theorems 5.17, 5.15, 5.23, and 5.25 employ similar communication complexity reductions to DISJ.

Proof of Theorem 5.6. We first embed every instance of DISJ with $n = q$ into an instance of qSA and prove that they correspond. We assume the existence of the a transformer $f \in \mathsf{Transformer}^{1,1}_{d,m,d',p}$ that $\frac{1}{2q}$-approximates qSA and implies the existence of an $O(mp)$-bit communication protocol that computes DISJ. An application of Fact 5.7 concludes the proof.

Consider an instance of DISJ with $a \in \{0,1\}^q$ and $b \in \{0,1\}^q$ known by Alice and Bob respectively. We design an instance $X = (z_i; y_i; i)_{i \in [N]}$ of qSA. For each $j \in [2q]$, let $y_{2q+1} = \{2i + a_i - 1 : \ i \in [q]\}$. Additionally, let

$$z_j = \begin{cases} e_1 & \text{if } j \text{ is odd and } b_{(j-1)/2} = 1, \\ -e_1 & \text{otherwise.} \end{cases}$$

All other inputs are set arbitrarily. Then,

$$\begin{aligned} q\mathrm{SA}(X)_{2q+1} &= \frac{1}{q}\left|\left\{j \in [2q] : \ j \in y_{2q+1},\ j \text{ is odd, and } a_{(j-1)/2} = 1\right\}\right| e_1 \\ &\quad - \frac{1}{q}\left|\left\{j \in [2q] : \ j \in y_{2q+1} \text{ and } (j \text{ is even or } a_{(j-1)/2} = 0)\right\}\right| e_1 \\ &= \frac{|\{i \in [q] : a_i b_i = 1\}| - |\{i \in [q] : a_i b_i = 0\}|}{q} e_1. \end{aligned}$$

Hence, $q\mathrm{SA}(X)_{2q+1} = -e_1$ if and only if $\mathrm{DISJ}(a, b) = 0$.

It remains to show that this implies the existence of an efficient communication protocol that computes $\mathrm{DISJ}(a, b)$. By the existence of f, there exist $Q, K, V : \mathbb{R}^d \to \mathbb{R}^m$ and

$\psi : \mathbb{R}^m \to \mathbb{R}^{d'}$ such that

$$f(X)_{2q+1} = \psi\left(\frac{\sum_{i=1}^{N} \exp\left(Q(x_{2q+1})^\mathsf{T} K(x_i)\right) V(x_i)}{\sum_{i=1}^{N} \exp\left(Q(x_{2q+1})^\mathsf{T} K(x_i)\right)}\right).$$

The protocol is as follows:

1. From a, Alice determines y_{2q+1} and then computes $Q(x_{2q+1}) \in \mathbb{R}^m$, which she sends to Bob. This transmission uses $O(mp)$ bits.

2. Bob determines $z_1, \ldots, z_{2q}$ from b. Using those and the information from Alice, he computes $f(X)_{2q+1}$. He returns 1 if and only if $f(X)_{2q+1}^\mathsf{T} e_1 \geq -1 + \frac{1}{q}$.

The protocol computes $\mathrm{DISJ}(a, b)$ because f is a $\frac{1}{2q}$-approximation of qSA. Because any such protocol requires sharing $\Omega(q)$ bits of information, we conclude that $mp \leq cq$ for some c. $\square$

5.3.3 Optimality of Theorem 5.5 under restricted architectures

While the near-optimality of the bounded-precision self-attention construction in Theorem 5.4 is assured by the communication complexity argument of Theorem 5.6, it is not immediately apparent whether Theorem 5.5 is similarly optimal among infinite-precision self-attention models. Theorem 5.8 proves that this is indeed the case for a restricted family of architectures that resembles *cross-attention* rather than self-attention.

Theorem 5.8. *For input $x_1, \ldots, x_N$ satisfying $x_i = (z_i; y_i; i)$, suppose $\phi(x_i)^\mathsf{T} Q = w(y_i, i)$, $\phi(x_i)^\mathsf{T} K = u(i)$, and $\phi(x_i)^\mathsf{T} V = z_i$. Then, for any $q < N$ and $m \leq q(1 - C \log_N q)$ for some universal C, there do not exist $w : \mathbb{R}^d \times [N] \to \mathbb{R}^m$ and $u : [N] \to \mathbb{R}^m$ such that the resulting self-attention unit $\frac{1}{2q}$-approximates qSA.*

The architectural assumptions of this statement are strong. For each element $x_i = (z_i; y_i; i)$, its value embedding must reproduce its target z_i; its key embedding depends exclusively on the index i; and its query embedding only on the indices y_i and i. Indeed this attention unit more closely resembles *cross-attention* rather than self-attention, in which the

problem is formulated as two sequences $((z_1, 1), \ldots, (z_N, N))$ and $(y_1; 1), \ldots, (y_N; N)$ that are passed to the key and value inputs and the query inputs respectively. We leave open the problem of generalizing this result to include all infinite-precision cross-attention or self-attention architectures, but we note that the constructions in Theorems 5.4 and 5.5 can be implemented under such architectural assumptions.

The proof relies on a geometric argument about how the convex hull of fixed key embeddings $U = (u(1), \ldots, u(N))$ lacks neighborliness and hence cannot separate every size-q subsets of values embeddings $z_1, \ldots, z_N$ from the other values.

Proof. It suffices to show that for any fixed key embedding U, there exists some y_i and setting of $z_1, \ldots, z_N$ such that

$$\left\| (\text{softmax}(w(X)U^\mathsf{T})Z)_i - \frac{1}{q} \sum_{i' \in y_i} z_{i'} \right\|_2 \geq \frac{1}{2q},$$

where $w(X) = (w(y_1, 1), \ldots, w(y_N, N)) \in \mathbb{R}^{N \times m}$ and $U = (u(1), \ldots, u(N)) \in \mathbb{R}^{N \times m}$.

By Fact 5.9, for some $y_1 \in \binom{[N]}{q}$, there are no w and $\tau \in \mathbb{R}$ satisfying $w(y_1, 1)^\mathsf{T} u_{i'} \geq \tau$ if and only if $i' \in y_1$. Hence, for any fixed w, there exists $i_1 \in y_1$ and $i_2 \in [N] \setminus y_1$ such that $w(y_1, 1)^\mathsf{T} u_{i_2} > w(y_1, 1)^\mathsf{T} u_{i_1}$. Given the value embeddings $z_{i_1} = e_1, z_{i_2} = e_2$ and $z_i = e_3$ for all $i \notin \{i_1, i_2\}$, we have

$$\begin{aligned}
&\left\| (\text{softmax}(w(X)U^\mathsf{T})Z)_1 - \frac{1}{q} \sum_{i' \in y_1} z_{i'} \right\|_2^2 \\
&\quad \geq \left(\text{softmax}(w(X)U^\mathsf{T})Z)_{1,i_1} - \frac{1}{q} \right)^2 + \left(\text{softmax}(w(X)U^\mathsf{T})Z)_{1,i_2} \right)^2 \\
&\quad \geq \max \left(\left(\text{softmax}(w(X)U^\mathsf{T})Z)_{1,i_1} - \frac{1}{q} \right)^2, \text{softmax}(w(X)U^\mathsf{T})Z)_{1,i_1}^2 \right) \\
&\quad \geq \frac{1}{4q^2}.
\end{aligned}$$

□

Fact 5.9. *If $m' < q(1 - \log_N Cq)$, then the columns of any $U = (u_1, \ldots, u_N) \in \mathbb{R}^{N \times m'}$ can be partitioned into sets U_1 and U_2 with $|U_1| = q$ that are not linearly separable. Hence,*

$\mathrm{Conv}(u_1, \ldots, u_N)$ *is not q-neighborly.*

Proof. By the Sauer-Shelah Lemma (Sauer, 1972; Shelah, 1972; Vapnik and Chervonenkis, 1968) and the fact that the VC dimension of m'-dimensional linear thresholds is $m' + 1$, the maximum number of partitions of the columns of U that can be linearly separated is at most

$$\sum_{k=0}^{m'+1} \binom{N}{i} \leq C' N^{m'+1} < C' \cdot \frac{N^q}{(Cq)^q} \leq \binom{N}{q},$$

for a sufficiently large choice of C given universal constant C'. If the fact were to be false, then at least $\binom{N}{q} \geq (\frac{N}{q})^q$ such partitions must exist, which contradicts the above bound. □

5.3.4 Experimental details

This section describes the experimental setup behind Figure 5.2, and provides further experiments suggesting an *implicit bias* of transformers for qSA, in particular when compared with MLPs and RNNs.

Experimental setup. Experiments used synthetic data, generated for qSA with $n = 1000$ training and testing examples, a sequence length $N = 20$, $q = 3$, with the individual inputs described in more detail as follows.

- The positional encoding of element i is a random vector sampled uniformly from the sphere in $\mathbb{R}^{d_0}$ with $d_0 := \lceil 1 + 2\ln(N) \rceil$, a quantity which agrees with the theory but was not tuned.

- A sequence element then consists of the data portion $z \in \mathbb{R}^{d_1}$ where $d_1 = 4$, also sampled from the unit sphere, then the positional encoding of this sequence element, and then q further positional encodings identifying elements to average to produce the output; this differs from (and is more tractable than) the presentation in Section 5.3, where the positional encoding is provided as an integer and the MLP layer input to our attention layers is expected to choose a sufficient positional encoding.

As such, the total dimension of a sequence element is $d_1 + (q+1)d_0 = 32$. The architectures are detailed as follows.

- The attention is identical to the description in the paper body, with the additional detail of the width and embedding dimension m being fixed to 100.
- Figure 5.4 also contains an MLP, which first flattens the input, then has a single hidden ReLU layer of width 256, before a final linear layer and an output reshaping to match the desired output sequence shapes.
- Figure 5.4 also contains an LSTM, which is a standard `pytorch` LSTM with 2 layers and a hidden state size 800, which is 200 times larger than the target output dimension 4.

Experiments fit the regression loss using Adam and a minibatch size of 32, with default precision, and take a few minutes to run on an NVIDIA TITAN XP, and would be much faster on standard modern hardware.

Further discussion of Figure 5.2 and Figure 5.5. In Figure 5.2 and Figure 5.5, we plot (post-softmax) alignment matrices after $T \in \{0, 1000, 40000\}$ iterations of Adam. The alignment matrices in Figure 5.2 are taken from the training example whose loss is the median loss across all examples. Figure 5.5 is similar, but additionally shows the examples of minimal and maximal loss.

Further discussion of Figure 5.4. Figure 5.4 plots training and testing error curves for the same attention architecture as in Figure 5.2, but with further MLP and LSTM architectures as described above. but also an MLP trained on flattened (vectorized) error bars reflect 5 separate training runs from random initialization. A few variations of these architectures were attempted, however curves did not qualitatively change, and in particular, only the attention layer achieves good generalization across all attempts.

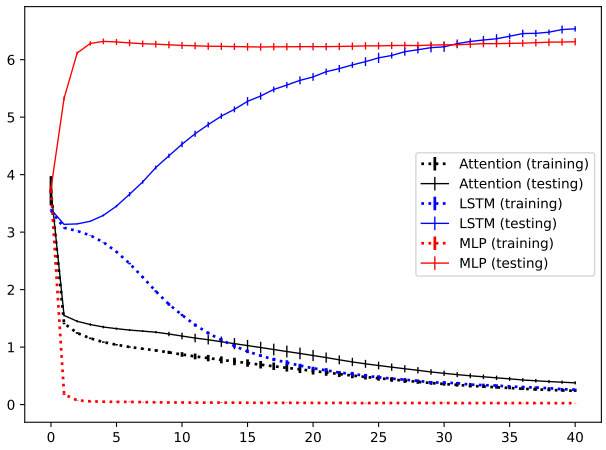


Figure 5.4: Test and train error curves of fitting various architectures to qSA, where the horizontal axis denotes thousands of training iterations, and the vertical axis denotes the regression objective; see Section 5.3.4 for further details.

5.3.5 Proofs for Section 5.3

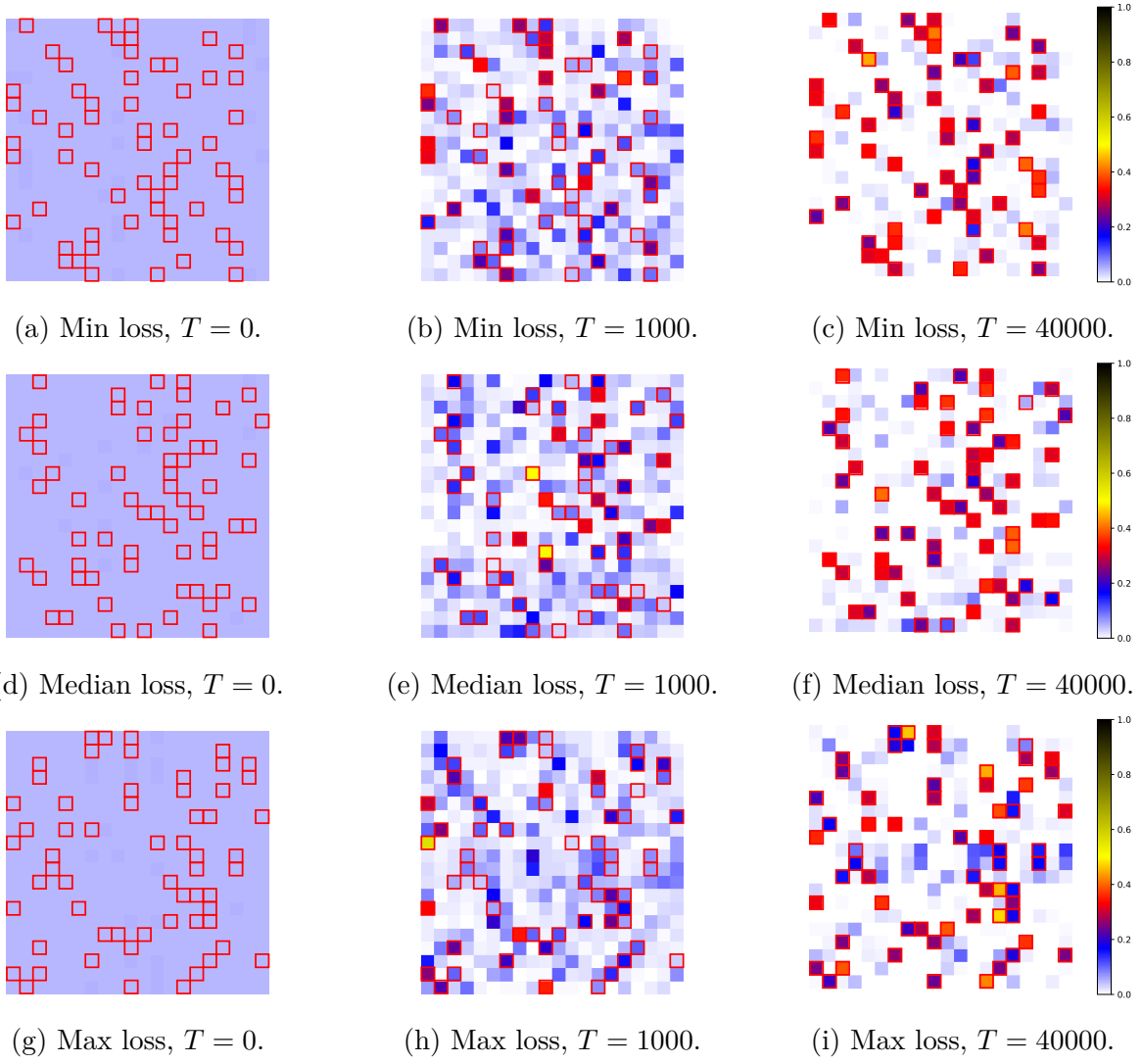
5.3.5.1 Proof of Theorem 5.4

Theorem 5.4 (Fixed-precision)**.** *For any N, any $m \geq \Omega(d' + q \log N)$, any $\epsilon \in (0, 1)$, and $p = \Omega(\log(\frac{q}{\epsilon} \log N))$, there exists some $f \in \mathsf{Attn}'_{d,m,d',p}$ that ϵ-approximates qSA.*

Proof. Before explaining how they are produced by the input MLP, we introduce the corresponding key, value, and query inputs. The values will simply be $\phi(X)V = (z_1, \ldots, z_N)$. For some $m' = \frac{m-d}{2}$, let $\phi(X)K = (u_1, \ldots, u_N) \in \mathbb{R}^{N \times m'}$ be embedded key vectors, where $u_1, \ldots, u_N \in \{\pm 1/\sqrt{m'}\}^{m'}$ are the columns of a $m' \times N$ matrix satisfying the $(q, 1/4)$-restricted isometry and orthogonality property (Definition 5.5), as guaranteed to exist by Lemma 5.10 and the assumption on m'. Let $\alpha := \lceil 2 \log(4N/\epsilon) \rceil$. By Lemma 5.11, for each $y \in \binom{[N]}{q}$, there exists $w_y \in \mathbb{R}^{m'}$ with $\|w_y\|_2 \leq 2\sqrt{q}$ satisfying

$$\begin{aligned} \langle u_{i'}, w_y \rangle &= 1 \quad \text{for all } i' \in y, \\ |\langle u_{i'}, w_y \rangle| &\leq \frac{1}{2} \quad \text{for all } i' \notin y. \end{aligned}$$

Given the bounded precision of the model, we are not free to represent the vectors w_y



(a) Min loss, $T = 0$. (b) Min loss, $T = 1000$. (c) Min loss, $T = 40000$.

(d) Median loss, $T = 0$. (e) Median loss, $T = 1000$. (f) Median loss, $T = 40000$.

(g) Max loss, $T = 0$. (h) Max loss, $T = 1000$. (i) Max loss, $T = 40000$.

Figure 5.5: Alignment plots as in Figure 5.2, but using examples with minimum, median, and maximum loss, whereas Figure 5.2 only uses the example with median loss.

exactly. Under p-bit precision for p sufficiently large, we there exists a vector of p-bit floating point numbers $\widetilde{w_y} \in \mathbb{R}^{m'}$ for every w_y with $\|w_y\|_2 \leq 2\sqrt{q}$ satisfying $\|\widetilde{w_y} - w_y\|_2 \leq \frac{\epsilon}{4\alpha}$. As an immediate consequence, $|\langle u_{i'}, \widetilde{w_y}\rangle - \langle u_{i'}, w_y\rangle| \leq \frac{\epsilon}{4\alpha}$ for all i' and y (by Cauchy-Schwarz). The remainder of the proof demonstrates that the necessary properties of the argument hold even with this approximation.

We now describe how to structure the neural network. We define an MLP $\phi : \mathbb{R}^d \to \mathbb{R}^m$ as $\phi(x_i) = \phi(z_i; y_i; i) = (z_i; \alpha\widetilde{w_{y_i}}; u_i)$, which works simply by using a look-up table on the values of u_i and $\widetilde{w_{y_i}}$ from keys i and y_i respectively. Then, we define Q, K, V as sparse

boolean-valued matrices that simply copy their respective elements from $\phi(X)$.

We analyze the output of the softmax. If $i' \in y_i$, then

$$\begin{aligned}
\operatorname{softmax}(\phi(X)QK^{\mathsf{T}}\phi(X)^{\mathsf{T}})_{i,i'} &= \frac{\exp(\alpha \langle u_i, \widetilde{w_{i'}} \rangle)}{\sum_{i'' \in y_i} \exp(\alpha \langle u_i, \widetilde{w_{i''}} \rangle) + \sum_{i'' \notin y_i} \exp(\alpha \langle u_i, \widetilde{w_{i''}} \rangle)} \\
&\geq \frac{\exp(\alpha - \frac{\epsilon}{4})}{q \exp(\alpha + \frac{\epsilon}{4}) + N \exp(\frac{\alpha}{2} + \frac{\epsilon}{4})} = \frac{e^{\alpha}}{qe^{\alpha} + Ne^{\alpha/2}} \cdot \exp\left(-\frac{\epsilon}{2}\right) \\
&\geq \left(\frac{1}{q} - \frac{Ne^{\alpha/2}}{qe^{\alpha}}\right)\left(1 - \frac{\epsilon}{2}\right) \geq \frac{\left(1 - \frac{\epsilon}{4}\right)\left(1 - \frac{\epsilon}{4}\right)}{q} \geq \frac{1}{q}\left(1 - \frac{\epsilon}{2}\right).
\end{aligned}$$

An analogous argument shows that

$$\operatorname{softmax}(\phi(X)QK^{\mathsf{T}}\phi(X)^{\mathsf{T}})_{i,i'} \leq \frac{1}{q}\left(1 + \frac{\epsilon}{2}\right).$$

Likewise, if $i' \notin y_i$, then

$$\operatorname{softmax}(\phi(X)QK^{\mathsf{T}}\phi(X)^{\mathsf{T}})_{i,i'} \leq \frac{\exp(\frac{\alpha}{2} + \frac{\epsilon}{4})}{q \exp(\alpha - \frac{\epsilon}{4})} \leq \exp\left(-\frac{\alpha}{2} + \frac{\epsilon}{2}\right) \leq \frac{\epsilon}{2N}.$$

We thus conclude that that we meet the desired degree of approximation for such m:

$$\begin{aligned}
\|f(X)_i - q\mathrm{SA}(X)_i\|_2 &= \left\| \sum_{i' \in y_i} \left(\frac{1}{q} - \operatorname{softmax}(\phi(X)QK^{\mathsf{T}}\phi(X)^{\mathsf{T}})_{i,i'}\right) z_{i'} \right\|_2 \\
&\quad + \left\| \sum_{i' \notin y_i} \left(\operatorname{softmax}(\phi(X)QK^{\mathsf{T}}\phi(X)^{\mathsf{T}})_{i,i'}\right) z_{i'} \right\|_2 \\
&\leq q \cdot \frac{\epsilon}{2q} + (N - q) \cdot \frac{\epsilon}{2N} \leq \epsilon. \qquad \square
\end{aligned}$$

Restricted isometry and orthogonality property The proof relies on the restricted isometry and orthogonality property from the compressed sensing literature. For $v \in \mathbb{R}^N$, let $\operatorname{supp}(v) = \{i \in [N] : v_i \neq 0\}$.

Definition 5.5. We say a matrix $U \in \mathbb{R}^{m \times N}$ satisfies the (q, δ)-*restricted isometry and*

orthogonality property if

$$\|Uv\|_2^2 \in [(1-\delta)\|v\|_2^2, (1+\delta)\|v\|_2^2] \quad \text{and} \quad |\langle Uv, Uv'\rangle| \leq \delta\|v\|_2\|v'\|_2$$

for all vectors $v, v' \in \mathbb{R}^N$ with $|\operatorname{supp}(v)| \leq q$, $|\operatorname{supp}(v')| \leq 2q$, and $\operatorname{supp}(v) \cap \operatorname{supp}(v') = \emptyset$.

The first result shows the existence of a sign-valued matrix U that satisfies the desired distance-preserving property.

Lemma 5.10 (Consequence of Theorem 2.3 of Mendelson, Pajor, and Tomczak-Jaegermann, 2007 and Lemma 1.2 of Candes and Tao, 2005)**.** *There is an absolute constant $C > 0$ such that the following holds. Fix $\delta \in (0, 1/2)$ and $q \in \mathbb{N}$. Let U denote a random $m \times N$ matrix of independent Rademacher random variables scaled by $1/\sqrt{m}$. If $m \geq C(q \log N)/\delta^2$, then with positive probability, U satisfies the (q, δ)-restricted isometry and orthogonality property.*

Sparse subsets of the columns of such a U can then be linearly separated from all other columns.

Lemma 5.11 (Consequence of Lemma 2.2 in Candes and Tao, 2005)**.** *Fix $\delta \in (0, 1/2)$ and $q \in \mathbb{N}$. Let matrix $U = [u_1, \ldots, u_N] \in \mathbb{R}^{m \times N}$ satisfy the (q, δ)-restricted isometry and orthogonality property. For every vector $v \in \{0, 1\}^N$ with $\operatorname{supp}(v) \leq q$, there exists $w \in \mathbb{R}^m$ satisfying*

$$\begin{aligned} \|w\|_2 &\leq \sqrt{q}/(1-2\delta), \\ \langle u_i, w\rangle &= 1 && \text{if } v_i = 1, \\ |\langle u_i, w\rangle| &\leq \delta/(1-2\delta) && \text{if } v_i = 0. \end{aligned}$$

5.3.5.2 Proof of Theorem 5.5

Theorem 5.5 (Infinite-precision)**.** *For fixed N, $m \geq \Omega(d' + q)$ and $\epsilon > 0$, there exists some $f \in \mathsf{Attn}'_{d,m,d'}$ that ϵ-approximates qSA.*

The proof relies on the properties of *neighborly polytopes*, which we define.

Definition 5.6 (Ziegler (2006))**.** A polytope P is *q-neighborly* if every subset of $q' \leq q$ vertices forms a $(q'-1)$-face.

We give a q-neighborly polytope below that we use for the construction. For vectors $v_1, \ldots, v_N \in \mathbb{R}^{m'}$, let $\mathrm{Conv}(v_1, \ldots, v_N) = \{\sum_{i=1}^N \alpha_i v_i : \alpha \in [0,1]^N, \sum_i \alpha_i = 1\}$ denote their convex hull.

Fact 5.12 (Theorem 1 of Gale (1963))**.** *For $t \in \mathbb{R}$, let $\theta(t) = (t, \ldots, t^{m'}) \in \mathbb{R}^{m'}$. Then, for all distinct $t_1, \ldots, t_N \in \mathbb{R}$, the* cyclic polytope $\mathrm{Conv}(\theta(t_1), \ldots, \theta(t_N))$ *is $\frac{m'}{2}$-neighborly.*

The proof of Theorem 5.5 is immediate from the aforementioned fact and the following lemma.

Lemma 5.13. *Suppose there exists $u_1, \ldots, u_N \in \mathbb{R}^{m'}$ such that $\mathrm{Conv}(u_1, \ldots, u_N)$ is q-neighborly. Then, for any $\epsilon > 0$, there exists some $f \in \mathsf{Attn}'_{d,m,d'}$ with fixed key vectors $\phi(X)K = (u_1, \ldots, u_N)$ that ϵ-approximates q*SA*.*

Proof. The construction employs a similar look-up table MLP ϕ to the one used in the proof of Theorem 5.4. We let the key and value embeddings be

$$\phi(X)K = ((u_1, 1), \ldots, (u_N, 1)) \in \mathbb{R}^{N \times (m'+1)}, \text{ and } \phi(X)V = (z_1, \ldots, z_N) \in \mathbb{R}^{N \times d}.$$

To set the query vectors, observe that for any face F of a polytope P, there exists a hyperplane H_F such that $F \subset H_F$ and $P \setminus F$ lies entirely on one side of H_F. Thus, for every $y \in \binom{[N]}{q}$, there exists $w'_y \in \mathbb{R}^{m'}$ and $b_y \in \mathbb{R}$ such that

$$w_y'^{\mathsf{T}} u_i + b_y \begin{cases} = 1 & \text{if } i \in y, \\ < 1 & \text{otherwise.} \end{cases}$$

For $\alpha > 0$, let $\phi(x_i)^{\mathsf{T}} Q = \alpha w_y = \alpha(w'_y, b_y)$.

We construct the MLP to satisfy $\phi(x_i) = (z_k; w_{y_i}; u_i; 1) \in \mathbb{R}^m$ for $m = 2m' + 2$ and set parameter weights accordingly. Following the softmax analysis of Theorem 5.5, a sufficiently large choice of α ensures that $\max_{i\in[N]} \|f(X)_i - q\mathrm{SA}(X)_i\|_2 \leq \epsilon$. □

5.4 Sparse averaging and limitations of alternative architectures

In this section, we show that fully-connected neural networks (Section 5.4.1) and recurrent neural networks (Section 5.4.2) cannot efficiently approximate qSA. These results employ similar communication complexity reductions to those used elsewhere in the paper. While the results are perhaps unsurprising given the nature of the task, they provide a clean distillation of the kinds of tasks for which attention layers are particularly well-suited.

5.4.1 Only wide fully-connected neural networks can approximate qSA

In this section, we show that any fully-connected neural network that approximates $q\mathrm{SA} : \mathbb{R}^{Nd} \to \mathbb{R}^{Nd'}$ must have width $m = \Omega(N)$.[4] We consider networks of the form $f(x) = g(Wx)$ for some weight matrix $W \in \mathbb{R}^{m\times Nd}$ (the first layer weights) and arbitrary function $g : \mathbb{R}^m \to \mathbb{R}^{Nd'}$ (computed by subsequent layers of a neural network).

Theorem 5.14. *Suppose $q \leq \frac{N}{2}$. Any fully-connected neural network f defined as above that $\frac{1}{2q}$-approximates qSA satisfies $m \geq \mathrm{Rank}(W) \geq \frac{Nd'}{2}$.*

Proof. For simplicity, we arrange the input as

$$x = (1; \ldots; N; y_1; \ldots; y_N; z_1; \ldots; z_N)$$

and $W = [\tilde{W}; V_1; \ldots; V_N]$ with $z_1, \ldots, z_N \in \mathbb{B}^{d'}$, $\tilde{W} \in \mathbb{R}^{m\times N(d-d')}$, and $V_1, \ldots, V_N \in \mathbb{R}^{m\times d'}$. If $\mathrm{Rank}(W) \leq \frac{Nd'}{2} - 1$, then so too is $\mathrm{Rank}([V_q; \ldots; V_N]) \leq \frac{Nd'}{2} - 1$, and $[V_q; \ldots; V_N]$ has a

[4] We regard inputs as Nd-dimensional vectors rather than $N \times d$ matrices.

nontrivial null space containing a nonzero vector $u = (u_q; \ldots; u_N) \in \mathbb{R}^{(N-q)d'}$. Let

$$\xi = \frac{1}{\max_{j \in \{q,\ldots,N\}} \|u_j\|_2} (u_q; \ldots; u_N),$$

$z = (\vec{0}; \ldots; \vec{0}; \xi_q; \ldots; \xi_N)$, and $z' = (\vec{0}; \ldots; \vec{0}; -\xi_q; \ldots; -\xi_N)$. Then,

1. $z_j, z'_j \in \mathbb{B}^{d'}$ for all $j \in [N]$;
2. $V_j z_j = V_j z'_j = 0$ for all $j \in [N]$; and
3. $\|z_{j^*} - z'_{j^*}\|_2 = 2$ for some $j^* \in \{q, \ldots, N\}$.

Therefore, for any $y_1, \ldots, y_N \in \binom{[N]}{q}$, respective $x = (1; \ldots; N; y_1; \ldots; y_N; z_1; \ldots; z_N)$ and $x' = (1; \ldots; N; y_1; \ldots; y_N; z'_1; \ldots; z'_N)$ satisfy $f(x) = f(x')$. Consider y with $y_j = (1, \ldots, q-1, j)$ for each $j \in \{q, \ldots, N\}$. Then,

$$q\mathrm{SA}(x)_j = \frac{1}{q}\xi_j \text{ and } q\mathrm{SA}(x')_j = -\frac{1}{q}\xi_j.$$

Hence, $\|q\mathrm{SA}(x)_{j^*} - q\mathrm{SA}(x')_{j^*}\|_2 \geq \frac{2}{q}$. Because $f(x) = f(x')$,

$$\max\left(\|f(x) - q\mathrm{SA}(x)_{j^*}\|_2, \|f(x') - q\mathrm{SA}(x')_{j^*}\|_2\right) \geq \frac{1}{q},$$

so f can approximate qSA to accuracy no better than $\frac{1}{q}$. □

5.4.2 Only high-memory recurrent neural networks can approximate qSA

In this section, we show that any memory-bounded algorithm that approximates qSA : $\mathbb{R}^{N \times d} \to \mathbb{R}^{N \times d'}$ must use a large "hidden state" (memory) as it processes the input elements. This lower bound applies to various recurrent neural network (RNN) architectures.

A memory-bounded algorithm with an m-bit memory processes input $X \in \mathbb{R}^{N \times d}$ sequentially as follows. There is an initial memory state $h_0 \in \{0,1\}^m$. For $i = 1, 2, \ldots, N$, the

algorithm computes the i-th output $f(X)_i \in \mathbb{R}^{d'}$ and the updated memory state h_i as a function of the input $x_i \in \mathbb{R}^d$ and previous memory state h_{i-1}:

$$(f(X)_i, h_i) = g_i(x_i, h_{i-1}),$$

where $g_i \colon \mathbb{R}^d \times \{0,1\}^m \to \mathbb{R}^{d'} \times \{0,1\}^m$ is permitted to be an arbitrary function, and $f \colon \mathbb{R}^{N \times d} \to \mathbb{R}^{N \times d'}$ is the function computed by the algorithm.

Our lower bound applies to algorithms that only need to solve the subclass of "causal" instances of qSA in which the input $X = ((z_i, y_i, i))_{i \in [N]} \in \mathbb{R}^{N \times d}$ is promised to satisfy $y_i = \emptyset$ for all $i \leq N/2 + 1$, and $y_i \subseteq \{1, \ldots, N/2 + 1\}$ for all $i > N/2 + 1$.

Theorem 5.15. *For any $\varepsilon \in (0,1)$, any memory-bounded algorithm that ε-approximates qSA (for $q = 1$ and $d' = 1$) on the subclass of "causal" instances must have memory $m \geq (N-1)/2$.*

Proof. Consider an m-bit memory-bounded algorithm computing a function $f \colon \mathbb{R}^{N \times d} \to \mathbb{R}^N$ that ε-approximates qSA (for $q = 1$ and $d' = 1$). We construct, from this algorithm, a communication protocol for DISJ (with $N = 2n + 1$) that uses m bits of communication.

Let $a, b \in \{0,1\}^n$ be the input for DISJ provided to Alice and Bob, respectively. The protocol is as follows.

1. Alice constructs inputs $x_i = (z_i, \emptyset, i)$ for $i = 1, \ldots, n+1$, where for each $i = 1, \ldots, n$,

$$z_i = \begin{cases} +1 & \text{if } a_i = 0, \\ -1 & \text{if } a_i = 1, \end{cases}$$

and

$$z_{n+1} = +1.$$

Bob constructs inputs $x_{n+1+i} = (0, y_{n+1+i}, n+1+i)$ for $i = 1, \ldots, n$, where

$$y_{n+1+i} = \begin{cases} \{n+1\} & \text{if } b_i = 0, \\ \{i\} & \text{if } b_i = 1. \end{cases}$$

Observe that, for this input $X = (x_1, \ldots, x_{2n+1})$, we have

$$q\mathrm{SA}(X)_{n+1+i} = \begin{cases} +1 & \text{if } a_i b_i = 0, \\ -1 & \text{if } a_i b_i = 1. \end{cases}$$

2. Alice simulates the memory-bounded algorithm on the first $n+1$ inputs $x_1, \ldots, x_{n+1}$, and sends Bob the m-bit memory state h_{n+1}. This requires m bits of communication.

3. Starting with h_{n+1}, Bob continues the simulation of the memory-bounded algorithm on these n additional inputs $x_{n+2}, \ldots, x_{2n+1}$.

4. If any output $f(X)_{n+1+i}$ for $i = 1, \ldots, n$ satisfies

$$f(X)_{n+1+i} < 0,$$

then Bob outputs 1 (not disjoint); otherwise Bob outputs 0 (disjoint).

The approximation guarantee of f implies that $\mathrm{sign}(f(X)_{n+1+i}) = q\mathrm{SA}(X)_{n+1+i}$ for all $i = 1, \ldots, n$, so Bob outputs 1 if and only if a and b are not disjoint. Because this protocol for DISJ uses m bits of communication, by Fact 5.7, it must be that $m \geq n = (N-1)/2$. $\square$

We note that the proof of Theorem 5.15 can be simplified by reducing from the INDEX problem, which has a 1-way communication lower bound of n bits. This suffices for "single pass" algorothms, such as standard RNNs. However, the advantage of the above argument (and reducing from DISJ) is that it easily extends to algorithms that make multiple passes over the input. Such algorithms are able to capture bidirectional recurrent neural net and

related models. A straightforward modification of the protocol in the proof of Theorem 5.15 shows that $\Omega(N)$ memory is required for any algorithm that makes $O(1)$ passes over the input (and computes the outputs in a final pass).

5.5 Pairwise and triple-wise tasks

In this section, we argue that the standard transformer architecture is unable to efficiently represent functions that do not decompose into a small number of pairwise-symmetric functions. We do this by contrasting the (in)approximability of intrinsically pairwise and triple-wise functions, respectively Match2 and Match3 (defined in Equations (5.1) and (5.2)), and their variants.

Section 5.5.1 shows that Match2 can be efficiently represented using a single self-attention unit with constant embedding dimension. In contrast, Section 5.5.2 demonstrates the limited triple-wise capabilities of self-attention layers by proving that Match3 cannot be efficiently represented using a single-layer transformer. Section 5.5.3 shows that this limitation is due to the adversarial nature of the task and that simpler variants of Match3 *can* be efficiently represented by shallow transformers. Motivated by overcoming these triple-wise limitations, Section 5.5.4 introduces a new type of self-attention, *third-order tensor self-attention,* and Section 5.5.5 shows that Match3 can be efficiently computed using this architecture. Sections 5.5.6 and 5.5.7 discuss the broader conjectured hardness result—that even multi-layer standard transformers cannot efficiently represent Match3—and argues for its veracity using heuristic arguments and results about a further modified transformer architecture.

5.5.1 Efficient computation of Match2 with standard self-attention

We first show that Match2 can be efficiently approximated by a single standard (pairwise) self-attention unit.

Theorem 5.16. *For any input size N, input range $M = N^{O(1)}$, and fixed-precision bit complexity $p = O(\log M)$, there exists a transformer architecture $f \in \mathrm{tr}^{1,1}_{1,m,1,p}$ with a sin-*

gle self-attention unit with embedding dimension $m = 3$ such that for all $X \in [M]^N$, $f(X) =$ Match2(X).

The proofuses both a "blank token" and a trigonometric positional embedding, which ensures that

$$\phi(x_i)^\mathsf{T} Q K^\mathsf{T} \phi(x_j) = c \sum_{k=1}^{d} \cos\left(\frac{2\pi(x_{i,k} + x_{j,k})}{M}\right)$$

for some sufficiently large constant c. This embedding ensures that a cell of the attention matrix $\text{softmax}(\phi(X) Q K^\mathsf{T} \phi(X)^\mathsf{T})_{i,j}$ is extremely close to zero, unless $x_i = -x_j \pmod M$.

Proof. As discussed in Section 5.2.1, we allow a single blank token to be appended to the end of the sequence X and assume the existence of a positional encoding. That is, we consider input $X' = (x_1, \ldots, x_N, x')$ with $x_{i,0} = i$ and $x' = \vec{0}$ to be the input to the target attention model. We define input MLP $\phi : \mathbb{R} \to \mathbb{R}^3$ and parameterizations $Q, K, V \in \mathbb{R}^{3\times 3}$ such that

$$Q^\mathsf{T} \phi(x_i) = c\left(\cos\left(\frac{2\pi x_i}{M}\right), \sin\left(\frac{2\pi x_i}{M}\right), 1\right),$$

$$K^\mathsf{T} \phi(x_i) = \left(\cos\left(\frac{2\pi x_i}{M}\right), -\sin\left(\frac{2\pi x_i}{M}\right), 0\right),$$

$V^\mathsf{T}\phi(x_i) = \vec{1}$, $Q^\mathsf{T}\phi(x') = \vec{0}$, $K^\mathsf{T}\phi(x') = e_3$, and $V^\mathsf{T}\phi(x') = \vec{0}$. By elementary trigonometric identities, the following is true about the corresponding inner products:

$$(Q^\mathsf{T}\phi(x_i))^\mathsf{T} K^\mathsf{T} \phi(x_j) = c \cos\left(\frac{2\pi(x_i + x_j)}{M}\right)$$
$$(Q^\mathsf{T}\phi(x_i))^\mathsf{T} K^\mathsf{T} \phi(x') = cd.$$

As a result, $(Q^\mathsf{T}\phi(x_i))^\mathsf{T} K^\mathsf{T}\phi(x_j) = cd$ if and only if $x_i + x_j = 0 \pmod M$. Otherwise, $(Q^\mathsf{T}\phi(x_i))^\mathsf{T} K^\mathsf{T}\phi(x_j) \le c(1 - \frac{1}{M^2})$. (Here, the $O(\log M)$-bit fixed-precision arithmetic is sufficient to numerically distinguish the two cases.) For each $i \in [N]$ let

$$\beta_i = |\{j \in [N] : x_i + x_j = 0 \pmod M\}|$$

represent the total number of matches the input belongs to. If we take $c = M^2 \log(6N)$, then

$$(\text{softmax}(\phi(X)QK^\mathsf{T}\phi(X)^\mathsf{T}))_{i,j} \in \begin{cases} [0, \frac{1}{6N}] & \text{if } x_i + x_j \neq 0 \pmod M \text{ and } \ i,j \in [N]; \\ [\frac{1}{\beta_i+1} \pm \frac{1}{6N}] & \text{if } x_i + x_j = 0 \pmod M \text{ and } \ i,j \in [N]; \\ [\frac{1}{\beta_i+1} \pm \frac{1}{6N}] & \text{if } i \in [N], j = N+1. \end{cases}$$

We conclude that for any $i \in [N]$,

$$(\text{softmax}(\phi(X)QK^\mathsf{T}\phi(X)^\mathsf{T})V\phi(X))_i \begin{cases} \leq \frac{1}{6} \cdot \vec{1} & \text{if } \ \nexists j \text{ s.t. } x_i + x_j = 0 \pmod M \\ \geq \left(\frac{\beta_i}{\beta_i+1} - \frac{1}{6}\right) \cdot \vec{1} & \text{if } \exists j \text{ s.t. } x_i + x_j = 0 \pmod M, \end{cases}$$

where $\leq$ is a partial ordering with $v \leq v'$ if $v_i \leq v'_i$ for all i. Since the latter case holds only when $\beta_i \geq 1$, the final step of the proof is design an output MLP ψ such that $\psi(z) = 1$ if $z \geq \frac{1}{3}$ and $\psi(z) = 0$ if $z \leq \frac{1}{6}$, which can be crafted using two ReLU gates. □

5.5.2 Hardness of computing Match3 with a multi-headed self-attention layer

Although Match2 can be efficiently represented using a single unit of standard self-attention, representing Match3 using an entire layer of multi-headed attention units is impossible unless either the number of heads H, the embedding dimension m, or the precision p grows as $N^{\Omega(1)}$.

Theorem 5.17. *There is universal constant $c > 0$ such that for sufficiently large N, and any $M \geq N + 1$, if $mpH \leq cN/\log\log N$, then there is no $f \in \text{tr}^{1,H}_{1,m,1,p}$ satisfying $f(X) = \text{Match3}(X)$ for all $X \in [M]^N$.*

Like that of Theorem 5.6, the proof relies on a reduction from set disjointness in two-party communication. The proof of the lower bound applies a domain-restricted variant of Match3, which actually makes the problem substantially simpler to solve. In Remark 5.1, we show how this variant of Match3 introduces a *depth separation* between the representational

powers of single-layer and two-layer transformer models.

As mentioned in the introduction, we also conjecture that multiple layers of multi-headed attention are subject to the same impossibility (Conjecture 5.21). The impossibility is specific to standard (pairwise) attention; in Section 5.5.5, we show that Match3 *can* be efficiently computed with a single unit of *third-order* self-attention.

Proof. The proof employs a reduction to Fact 5.7 that embeds inputs to the set-disjointness problem of cardinality $n = \frac{N-1}{2}$ into a subset of instances passed to Match3. For the sake of simplicity, we assume in the construction that N is odd; if it were not, we could replace it with $N-1$ and set the final element such that it never belongs to a triple.

We consider the following family of inputs to Match3:

$$x_i \in \begin{cases} \{0\} & \text{if } i = 1, \\ \{1, i\} & \text{if } i \in \{2, \ldots, \frac{N+1}{2}\}, \\ \{1, (M - i + \frac{N-1}{2})\} & \text{if } i \in \{\frac{N+3}{2}, \ldots, N\}. \end{cases} \tag{5.3}$$

Note that $\text{Match3}(X)_1 = 1$ if and only if there exists $i \in \{2, \ldots, \frac{N+1}{2}\}$ such that $x_i = i$ and $x_{i+\frac{N-1}{2}} = (M - i)$. Given input $(a, b) \in \{0,1\}^n \times \{0,1\}^n$ to DISJ, let $x_{i+1} = 1$ if and only if $a_i = 0$, and let $x_{i+\frac{N+1}{2}} = 1$ if and only if $b_i = 0$. Then, $\text{Match3}(X)_1 = 1$ iff $\text{DISJ}(a, b) = 1$.

Suppose $f(X) = \text{Match3}(X)$ for all $X \in [M]^N$ for some $f \in \text{tr}^{1,H}_{1,m,1,p}$. We show that f simulates an $O(mpH)$-bit communication protocol for testing DISJ. By definition of the standard self-attention unit with multi-layer perceptrons, note that $f(X)_1 = \psi(\sum_{h=1}^{H} f_h(\phi(X)))$ for $\phi : \mathbb{R} \to \mathbb{R}^m$, $\psi : \mathbb{R}^m \to \{0,1\}$, and

$$f_h(X) = \frac{\sum_{i=1}^{N} \exp(Q_h(x_1)^\intercal K_h(x_i)) V_h(x_i)}{\sum_{i=1}^{N} \exp(Q_h(x_1)^\intercal K_h(x_i))},$$

for $Q_h, K_h, V_h : \mathbb{R}^{m \times m}$.

If we assume that this construction exists and is known explicitly by both Alice and Bob,

we design a communication protocol for Alice and Bob to solve DISJ by sharing $O(mpH)$ bits with one another. Let Alice possess $a \in \{0,1\}^n$ and Bob $b \in \{0,1\}^n$, with $n = \frac{N-1}{2}$.

1. Alice and Bob compute $(x_2, \ldots, x_{\frac{N+1}{2}})$ and $(x_{\frac{N+3}{2}}, \ldots, x_N)$ from a and b respectively.

2. Alice computes an $O(p \log\log N)$-bit approximation of the logarithm of the first half of the softmax normalization term for each attention head and sends the result to Bob. That is, she sends Bob

$$L_{h,a} = \log\left(\sum_{i=1}^{\frac{N+1}{2}} \exp(Q_h(\phi(x_1))^\mathsf{T} K_h(\phi(x_i)))\right)$$

   for each $h \in [H]$. This requires transmitting $O(pH \log\log N)$ bits.

3. Bob finishes the computation of normalization terms

$$L_h = \log\left(\exp(L_{h,a}) + \sum_{i=\frac{N+3}{2}}^{N} \exp(Q_h(\phi(x_1))^\mathsf{T} K_h(\phi(x_i)))\right)$$

   for each h and sends the result back to Alice (up to $O(p \log\log N)$-bits of precision). This again requires transmitting $O(pH \log\log N)$ bits.

4. Alice computes the partial convex combination of the first $\frac{N+1}{2}$ value vectors stipulated by the attention matrix

$$S_{h,a} = \frac{\sum_{i=1}^{\frac{N+1}{2}} \exp(Q_h(\phi(x_1))^\top K_h(\phi(x_i)))V_h(\phi(x_i))}{\exp(L_h)} \in \mathbb{R}^m$$

   for each h and sends the partial combinations to Bob. This requires transmitting $O(mpH \log\log N)$ bits (using the same precision as above).

5. Bob finishes the computation of the convex combinations

$$f_h(X) = S_{h,a} + \frac{\sum_{i=\frac{N+3}{2}}^{N} \exp(Q_h(\phi(x_1))^\top K_h(\phi(x_i)))V_h(\phi(x_i))}{\exp(L_h)} \in \mathbb{R}^m.$$

Bob concludes the protocol by computing and outputting $f(X)_1$, using his knowledge of each $f_h(X)$ and of ψ.

By the equivalences previously established, Bob returns 1 if and only if $\text{DISJ}(a, b) = 1$. Because the protocol requires $O(mpH \log \log N)$ bits of communication, we can only avoid contradicting Fact 5.7 if $mpH \geq \Omega(n/\log \log N) = \Omega(N/\log \log N)$. □

Remark 5.1. *The domain restrictions to* Match3 *stipulated in Equation* (5.3) *make the* Match3 *problem substantially easier to solve than the full-domain case. Indeed, under the domain restrictions,*

$$\text{Match3}(X)_1 = \max_{i \in \{2, \ldots, \frac{N+1}{2}\}} \text{Match2}(X)_i,$$

which is computable by a two-layer single-headed transformer network with constant embedding dimension. The first layer computes each $\text{Match2}(X)_i$ *with the construction in the proof of Theorem 5.16, and the second computes the maximum of the previous outputs by using those outputs as key vectors.*

While Informal Conjecture 5.3 suggests that two layers are insufficient to compute the full-domain version of Match3*, this restricted variant introduces a concise* depth separation *(see Eldan and Shamir, 2016; Telgarsky, 2016; Daniely, 2017a) between one- and two-layer transformer models.*

5.5.3 More efficient constructions for simplified Match3 computations

While the previous sections suggests that no efficient construction exists to compute Match3 with standard transformer models, practical examples of triple detection abound. For example, a transformer-based language model will likely succeed in linking a subject/verb/object triple because all three tokens likely inhabit the same local region and because the model could agglomerate the triple by first identifying a pair and then adding the third. Here, we introduce two variants on the Match3 problem that have additional structure to serve as hints. The first variant specifies triple sums comprising the input element

and a neighboring pair elsewhere in the sequence: for each $i \in [N]$,

$$\text{Match3Bigram}(X)_i = \mathbb{1}\left\{\exists j \text{ s.t. } x_i + x_j + x_{j+1} = 0 \pmod{M}\right\}.$$

The second focuses on localized sums, where are all components of a triple must be within a fixed range of constant width $K \ll N$: for each $i \in [N]$,

$$\text{Match3Local}(X)_i = \mathbb{1}\left\{\exists j_1, j_2 \text{ s.t. } x_i + x_{j_1} + x_{j_2} = 0 \pmod{M}, |i - j_1|, |i - j_2| \leq K\right\}.$$

We show that the two can be efficiently represented using compact standard transformer models.

Theorem 5.18. *For any N, $M = N^{O(1)}$, and $p = O(\log M)$, there exists a transformer architecture $f \in \text{tr}^{D,1}_{1,m,1,p}$ with embedding dimension $m = 3$ and depth $D = 2$ such that for all $X \in [M]^{N \times d}$, $f(X) = \text{Match3Bigram}(X)$.*

Informally, the first layer of the construction uses a sinusoidal positional encoding to compute each bigram sum $x_j + x_{j+1}$ in the jth element of the sequence. The second layer applies the Match2 construction provided by Theorem 5.16 to determine whether there exists a j for each i such that $x_i + x_j + x_{j+1} = 0 \pmod{M}$.

Theorem 5.19. *For any d, N, $M = N^{O(1)}$, $p = O(\log M)$, and $K \leq N$, there exists a transformer architecture $f \in \text{tr}^{1,1}_{1,m,1,p}$ with embedding dimension $m = O(K \log N)$ and bit-complexity $p = O(\log(K \log N))$ such that for all $X \in [M]^{N \times d}$, $f(X) = \text{Match3Local}(X)$.*

Proof. We implement the localized construction by using Theorem 5.4 to construct a specific sparse simultaneous average of the inputs with $q := 2K + 1$ and $d' := 2K + 1$. To do so, we use the input MLP to convert x_i to the embedding $(z_i; y_i; i)$, for zero-padded input

$$z_i = x_i e_{\bar{i}} \in \mathbb{R}^{2K+1}$$

for $\bar{i} = i \pmod{2K+1}$ and subset

$$y_i = \{i-K, i-K+1, \ldots, i+K\} \in \binom{[N]}{2K+1}.$$

This construction ensures that the ith element of self-attention output computes (a rotation of) $(x_{i-K}, x_{i-K+1}, \ldots, x_{i+K})$. An output MLP can then verify whether any matching triples involving x_i exist among those vectors. □

5.5.4 Higher-order tensor attention

We introduce a novel category of higher-order tensor-based transformer models in order to show that problems like Match3 that are hard to compute with standard transformer models can be made solvable. An s-order transformer is designed to efficiently compute dense s-wise interactions among input elements in an analogous manner to how standard transformers compute pairwise interactions. (We think of a standard transformer as second-order.) Before defining the new type of attention, we introduce notation to express the needed tensor products.

For vectors $v^1 \in \mathbb{R}^{N_1}$ and $v^2 \in \mathbb{R}^{N_2}$, let $v^1 \otimes v^2 \in \mathbb{R}^{N_1 N_2}$ denote their *Kronecker product* by $(v^1 \otimes v^2)_{(i_1-1)N_2+i_2} = v^1_{i_1} v^2_{i_2}$. The *column-wise Kronecker product* of matrices $A^1 \in \mathbb{R}^{N_1 \times m}$ and $A^2 \in \mathbb{R}^{N_2 \times m}$ is

$$A^1 \star A^2 = [A^1_1 \mid \cdots \mid A^1_m] \star [A^2_1 \mid \cdots \mid A^2_m] = [A^1_1 \otimes A^2_1 \mid \cdots \mid A^1_m \otimes A^2_m] \in \mathbb{R}^{N_1 N_2 \times m}.$$

The following generalizes the definition of self-attention.

Definition 5.7. For order $s \geq 2$, input dimension d, output dimension d', embedding dimension m, bit complexity p, and matrices $Q, K^1, \ldots, K^{s-1} \in \mathbb{R}^{d \times m}$ and $V^1, \ldots, V^{s-1} \in \mathbb{R}^{d \times d'}$ (encoded with p-bit fixed-point numbers), an *s-order self-attention unit* is a function

$f_{Q,K,V} : \mathbb{R}^{N \times d} \to \mathbb{R}^{N \times d'}$ with

$$f_{Q,K,V}(X) = \text{softmax}(\underbrace{XQ}_{\in \mathbb{R}^{N \times m}} \underbrace{((XK^1) \star \cdots \star (XK^{s-1}))^\mathsf{T}}_{\in \mathbb{R}^{m \times N^{s-1}}}) \underbrace{((XV^1) \star \cdots \star (XV^{s-1}))}_{\in \mathbb{R}^{N^{s-1} \times d'}}.$$

The input to the row-wise softmax is an $N \times N^{s-1}$ matrix. Let $\mathsf{Attn}^{\otimes s}_{d,m,d',p}$ denote the set containing all such attention units.

Note that $\mathsf{Attn}^{\otimes 2}_{d,m,d',p} = \mathsf{Attn}_{d,m,d',p}$. Because s-order self-attention units have the same domain and codomain as standard self-attention, multiple units can be analogous combined to construct multi-headed attention units and full transformer models. We define $\mathsf{Attn}^{M,\otimes s}_{d,m,d',p}$ and $\mathsf{Transformer}^{D,H,\otimes s}_{d,m,d',p}$ accordingly.

The purpose of the s-order transformer model as a theoretical construct is to posit how strictly generalizing the architecture in order to permit higher order outer products transfers the expressive powers of standard transformer architectures to more sophisticated interactions among elements of the input sequence X. The model is not defined to be immediately practical, due to its steep computational cost of evaluation.

However, the trade-offs involved in using such architectures resemble those already made by using transformer models instead of fully-connected networks. Transformers are already computationally wasteful relative to the number of the parameters, and these models likely succeed only because extremely efficient factorized parameterization exist. Likewise, third-order transformers could indeed be practical if even more factorization proves useful, since the computational costs may prove mild if the embedding dimension m, number of heads H, and depth D necessary to succeed on a task exceed the sequence length N for standard second-order transformers.

5.5.5 Efficient representation of Match3 with third-order self-attention

Theorem 5.20 (Match3 construction with third-order self-attention)**.** *For any sequence length N, input range $M = N^{O(1)}$, and fixed-precision bit complexity $p = O(\log M)$, there*

exists a third-order transformer architecture $f \in \mathsf{Transformer}^{1,1,\otimes 3}_{1,m,1,p}$ *with a single self-attention unit with embedding dimension* $m = 5$ *such that for all* $X \in [M]^N$, $f(X) = \text{Match3}(X)$.

Proof of Theorem 5.20. The proof is almost identical to that of Theorem 5.16, except that we instead use a different key and query transforms to express a different trigonometric function:

$$\begin{aligned}
Q\phi(x_i) &= c\left(\cos\left(\frac{2\pi x_i}{M}\right), -\cos\left(\frac{2\pi x_i}{M}\right), \sin\left(\frac{2\pi x_i}{M}\right), \sin\left(\frac{2\pi x_i}{M}\right), 1\right), \\
K^1\phi(x_i) &= \left(\cos\left(\frac{2\pi x_i}{M}\right), \sin\left(\frac{2\pi x_i}{M}\right), -\cos\left(\frac{2\pi x_i}{M}\right), \sin\left(\frac{2\pi x_i}{M}\right), 0\right), \\
K^2\phi(x_i) &= \left(\cos\left(\frac{2\pi x_i}{M}\right), \sin\left(\frac{2\pi x_i}{M}\right), \sin\left(\frac{2\pi x_i}{M}\right), -\cos\left(\frac{2\pi x_i}{M}\right), 0\right).
\end{aligned}$$

Together, these ensure that the resulting tensor products reduce to a trigonometric expression that is maximized when $x_i + x_{j_1} + x_{j_2} = 0 \pmod M$. That is,

$$(\phi(X)Q((\phi(X)K^1) \star (\phi(X)K^2))^\mathsf{T})_{i,(j_1-1)+j_2} = c\cos\left(\frac{2\pi(x_i + x_{j_1} + x_{j_2})}{M}\right).$$

We similarly let $V^1\phi(x_i) = V^2\phi(x_i) = \vec{1}$ and $V^1\phi(x') = V^2\phi(x') = \vec{0}$. The remaining choice of c and the output MLP, and the analysis of the softmax proceeds identically to the previous proof. □

5.5.6 Heuristic argument for Informal Conjecture 5.3

Conjecture 5.21 (Formal version of Informal Conjecture 5.3)**.** *For sufficiently large* N *and any* $d \geq 1$, *for all* $M \geq N + 1$ *and* $mpHD \leq N^{\Omega(1)}$, *there is no* $f \in \mathsf{Transformer}^{D,H}_{1,m,1,p}$ *satisfying* $f(X) = \text{Match3}(X)$ *for all* $X \in [M]^N$.

We believe that the conjecture holds due to a heuristic information-theoretic argument. Define the distribution $\mathcal{D}$ over inputs $X \in \mathbb{R}^N$ that will be used to show that the model cannot compute Match3 for $M = N^4$ with high probability. We draw $\mathbf{X}$ from $\mathcal{D}$ as follows:

(E_1) With probability $\frac{1}{2}$, draw each $\mathbf{x}_i$ iid from $\text{Unif}([M])$.

(E_2) With probability $\frac{1}{2}$, draw j_1, j_2, j_3 iid from Unif($\binom{[N]}{3}$). For all $i \neq j_3$, draw each $\mathbf{x}_i$ iid from Unif($[M]$). Let $\mathbf{x}_{j_3} = -\mathbf{x}_{j_1} - \mathbf{x}_{j_2} \pmod{M}$.

Note that under event E_1, a three matching elements exist with probability at most $\frac{1}{N}$, and

$$\Pr\left[\text{Match3}(\mathbf{X}) = \vec{0} \mid E_1\right] \geq 1 - \frac{1}{N}.$$

Under event E_2, a triple of matching elements is always planted, so $\text{Match3}(\mathbf{X}) \neq \vec{0}$. It would suffice to prove that—unless a transformer is sufficiently large—it is impossible to determine whether $\text{Match3}(\mathbf{X}) = \vec{0}$ with probability at least 0.9.

Under $\mathcal{D}$, any subset of $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ consists of iid integers drawn uniformly from $[M]$, unless all of $\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \mathbf{x}_{j_3}$ appear in the subset. Consider a transformer architecture with p-bit precision, m-dimensional embeddings, H heads per layer, and D layers. We argue informally that a single-element output of a self-attention unit can take into account information about mp more inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$ than that it had in the previous layer. By induction, after D layers of H-headed self-attention with interleaved MLPs, each element is a function of at most $mpHD$ inputs. Until an element exists that is a function of at least two of the three of $\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \mathbf{x}_{j_3}$, we assume that the elements "known" by each output are chosen independently of the indices j_1, j_2, j_3. (Given two elements of the triple, the third element can be identified with a single self-attention unit.) Hence, we argue that it suffices to show that the probability any two elements of the triple j_1, j_2, j_3 occurring within any of the N sets of $mpHD$ inputs is vanishingly small for sufficiently large transformer parameters. The probability of single collection having any of two of the three inputs is at most

$$\frac{3\binom{mpHD}{2}}{\binom{N}{2}} \leq 3\left(\frac{empHD}{N}\right)^2.$$

Thus, the probability that any collection has all three inputs is no more than $3(empHD)^2/N$. If $mpHD = O(\sqrt{N})$, then the randomly chosen triple will not jointly appear as the outcome

of a single element of a self-attention unit with probability at least 0.9, and the transformer will be unexpected to successfully distinguish between the two cases.

Should the conjecture hold, it would represented a tight lower bound on the size of the smallest standard transformer architecture necessary to compute Match3.

Theorem 5.22 (Tightness of Conjecture 5.21)**.** *For any sequence length N, if the input range satisfies $M = N^{O(1)}$ and the transformer size parameters satisfy $p \geq \log(M)$, $H = 1$, $m \geq 4$, and $mD \geq CN^2$ for some universal constant C, then there exists a transformer architecture $f \in \mathsf{Transformer}^{D,H}_{1,m,1,p}$ such that $f(X) = \text{Match3}(X)$.*

Proof. We construct an architecture that collects a group of candidate pairs in each layer of single-headed self-attention and verifies whether there exists a triple incorporating each pair that satisfies the summation property. Then, all candidate triples are disposed of, and the subsequent layer collects a new family of candidates.

To do so, we first let $\ell := \left\lfloor \frac{m}{2} \right\rfloor - 1 \geq 1$ represent the total number of pairs shared in each layer of attention. We let $P = \binom{[N]}{2}$ represent a collection of all pairs of indices and partition it into D subsets $P_1, \ldots, P_D$, each containing ℓ distinct pairs. (Since $|P| = \frac{N(N+1)}{2}$, any D satisfying the theorem's preconditions is sufficiently large for this to be a proper partition.) Our construction ensures that there exist $x_i + x_{j_1} + x_{j_2} = 0 \pmod{M}$ for $(j_1, j_2) \in P_k$, then the kth layer of self attention will verify its existence and mark x_i as belonging to the match. Throughout the network, we maintain that the first two dimensions of any embedding of the ith element correspond to $x_i \in [M]$ and a bit indicating whether a match has been found yet containing x_i.

Consider the first layer of self-attention, and let $P_1 = \{(i_1, j_1), \ldots, (i_\ell, j_\ell)\}$. We set the input MLP $\phi_1 : \mathbb{R}^d \to \mathbb{R}^m$ and respective matrices $Q^1, K^1 \in \mathbb{R}^{m \times m}$ such that

$$Q^1\phi_1(x_i) = ce_1 \text{ and } K^1\phi_1(x_i) = \begin{cases} e_1 & \text{if } i \in P_1 \\ \vec{0} & \text{otherwise,} \end{cases}$$

for sufficiently large c. We additionally let

$$V^1\phi_1(x_i) = \begin{cases} (2\ell+1)\cdot(x_i;0;\vec{0}) & i \notin P_1, \\ (2\ell+1)\cdot(x_i;0;x_i e_{2\iota-1}) & i = i_\iota, \\ (2\ell+1)\cdot(x_i;0;x_i e_{2\iota}) & i = j_\iota. \end{cases}$$

By making use of a residual connection, we ensure that the ith outcome of the self-attention is $(x_i, 0, x_{i_1}, x_{j_1}, \ldots, x_{i_\ell}, x_{j_\ell})$. We encode an MLP to compute

$$(x_i, 0, x_{i_1}, x_{j_1}, \ldots, x_{i_\ell}, x_{j_\ell}) \mapsto \left(x_i, \mathbb{1}\left\{\exists \iota \in [\ell] \text{ s.t. } x_i + x_{i_\iota} + x_{j_\iota} = \vec{0} \pmod M\right\}; \vec{0}\right).$$

We repeat this construction D times, with the only modifications being the replacement of P_1 and the fact that the second dimension of the embedding remains 1 after being set to that value. After D layers, the final MLP outputs the value of the second dimension, which will be 1 if and only if the respective x_i belongs to a three-way match. □

5.5.7 Sharper separations for embedded subgraph detection problems

In pursuit of proving separations analogous to the one between Theorem 5.20 and Conjecture 5.21, we draw techniques for proving lower bounds for graph problems in the CONGEST model of distributed computation with restricted bandwidth (Peleg, 2000).[5]

The problems we consider take, as input, the adjacency matrix $X \in \{0,1\}^{N\times N}$ of an N-vertex graph $G = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = [N]$, so $x_{i,j} = \mathbb{1}\{(i,j) \in \mathcal{E}\}$. We may regard each row of X as a high-dimensional ($d = N$) embedding of the i-th vertex containing information about which (outgoing) edges are incident to the i-th vertex. We consider the following

[5] At a high level, the CONGEST model features N players that communicate in synchronous rounds over a network (an undirected graph with $[N]$ as its vertices) to solve a computational problem Peleg, 2000. In each round, each player can send a message to each of its neighbors. The computation that each player does with the messages received from its neighbors is unrestricted; the primary resources considered in CONGEST is the number of rounds of communication and the message sizes. Although CONGEST is often studied for solving computational problems on input graphs with vertices $[N]$, the input graph need not be the same as the communication network.

problems:

$$\text{DirectedCycle3}(X) = \left(\mathbb{1}\left\{\exists j_1, j_2 \in [N] \text{ s.t. } x_{i,j_1} x_{j_1,j_2} x_{j_2,i} = 1\right\}\right)_{i\in[N]};$$

$$\text{Cycle5}(X) = \left(\mathbb{1}\left\{\exists j_1, j_2, j_3, j_4 \in [N] \text{ s.t. } x_{i,j_1} x_{j_1,j_2} x_{j_2,j_3} x_{j_3,j_4} x_{j_4,i} = 1\right\}\right)_{i\in[N]},$$

$$\text{with dom}(\text{Cycle5}) = \{X : \ X = X^{\mathsf{T}}\}.$$

The former treats X as a directed graph (where X need not be symmetric) and asks whether each input belongs to a directed 3-cycle. The latter insists that X be an undirected graph by enforcing symmetry and determines membership in (undirected) 5-cycles.

However, solving these problems with any transformer model of constant order trivially requires having the product of the precision p, embedding dimension m, heads per layer H, and depth D grow polynomially with N, since each attention unit is limited to considering at most pm bits of information from each input. Such a lower bound is not interesting for dense graphs, where every vertex may have $\Omega(N)$ incident edges; the bottleneck is not due to any feature of standard attention units (and would persist with higher-order attention).

To circumvent this issue, we consider an augmented self-attention unit, which permits each element of the self-attention tensor to depend on both its respective inner product and on the presence of edges among corresponding inputs.

Definition 5.8. For order $s \geq 2$, input dimension d, output dimension d', embedding dimension m, bit complexity p, matrices $Q, K^1, \ldots, K^{s-1} \in \mathbb{R}^{d\times m}$ and $V^1, \ldots, V^{s-1} \in \mathbb{R}^{d\times d'}$ (encoded with p-bit fixed-point numbers), and cell-wise attention tensor function $\kappa : \{0,1\}^{s(s-1)} \times \mathbb{R} \to \mathbb{R}$, an *$s$-order graph self-attention unit* is a function $f_{Q,K,V} : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d'}$ with

$$f_{Q,K,V}(X) = \text{softmax}(\kappa(X, XQ((XK^1) \star \cdots \star (XK^{s-1}))^{\mathsf{T}}))((XV^1) \star \cdots \star (XV^{s-1})).$$

For attention tensor $A \in \mathbb{R}^{N^{\otimes s}}$, we abuse notation by writing $\kappa(X, A)$ as short-hand for

the particular cell-wise application of a fixed function, incorporating information about all relevant edges:

$$\kappa(X, A)_{i_1,\ldots,i_s} = \kappa(x_{i_1,i_2}, x_{i_1,i_3}, \ldots, x_{i_s,i_{s-1}}, x_{i_s,i_{s-2}}, A_{i_1,\ldots,i_s}).$$

Let $\mathsf{GraphAttn}^{\otimes s}_{d,m,d',p}$ and $\mathsf{GraphTransformer}^{D,H,\otimes s}_{d,m,d',p}$ denote all such attention units and all such transformers respectively.

Now, we provide four results that exhibit separations between orders of graph self-attention.

Theorem 5.23 (Hardness of representing Cycle5 with standard graph transformer)**.** *For sufficiently large N, any $f \in \mathsf{GraphTransformer}^{D,H}_{N,m,1,p}$ satisfying $f(X) = \text{Cycle5}(X)$ for all $X \in \{0,1\}^{N\times N}$ with $X = X^\mathsf{T}$ requires $mpHD = \Omega(N/\log^2 N)$.*

Theorem 5.24 (Efficient construction of Cycle5 with fifth-order graph transformer)**.** *For sequence length N and bit-complexity $p = O(\log N)$, there exists a fourth-order graph transformer architecture $f \in \mathsf{GraphTransformer}^{1,1,\otimes 5}_{N,1,1,p}$ with a single graph self-attention unit such that for all $X \in \{0,1\}^{N\times N}$ with $X = X^\mathsf{T}$, $f(X) = \text{Cycle5}(X)$.*

Theorem 5.25 (Hardness of representing DirectedCycle3 with standard graph transformer)**.** *For sufficiently large N, any $f \in \mathsf{GraphTransformer}^{D,H}_{N,m,1,p}$ satisfying $f(X) = \text{DirectedCycle3}(X)$ for all $X \in \{0,1\}^{N\times N}$ requires $mpHD = \Omega(N/\log^2 N)$.*

Theorem 5.26 (Efficient construction of DirectedCycle3 with fourth-order graph transformer)**.** *For sequence length N and bit-complexity $p = O(\log N)$, there exists a third-order graph transformer architecture $f \in \mathsf{GraphTransformer}^{1,1,\otimes 3}_{N,1,1,p}$ with a single graph self-attention unit such that for all $X \in \{0,1\}^{N\times N}$, $f(X) = \text{DirectedCycle3}(X)$.*

The proofs of Theorems 5.24 and 5.26 are immediate from the construction. Because each cell of the self-attention tensor has explicit access the the existence of all relevant edges, κ can be configured to ensure that cell's value is large if and only if the requisite edges for the

desired structure all exist. Taking a softmax with a blank element (like in Theorem 5.16) ensures that the outcome of the self-attention unit for a given element distinguishes between whether or not it belongs to a 5-cycle or a directed 3-cycle. The output MLP ensure that the proper output is returned.

We prove Theorems 5.23 and 5.25 by introducing a particular CONGEST communication graph that can be used to simulate any model in $\mathsf{GraphTransformer}^{D,H}_{d,m,d',p}$ (and hence, also any model in $\operatorname{tr} d, m, d', pD, H$) in $O(mHD \log N)$ rounds of communication. Then, we show for each problem that we can encode each instance of the set disjointness communication problem as an instance of Cycle5 (or DirectedCycle3) and derive a contradiction from the communication graph.

5.5.7.1 A CONGEST *communication graph that generalizes standard graph transformer computation*

The key principle of our analysis is that the predominant limitation of a transformer model is in its communication bandwidth and *not* its computational abilities. We model transformers as having element-wise multi-layer perceptron units with unbounded computational ability (but bounded precision inputs and outputs) and self-attention units, which compute linear combinations of inputs in a carefully regimented way that limits the ability of individual elements to share information with one another. Here, we introduce a specific CONGEST graph for each sequence length N and show that every transformer has a communication protocol that simulates its computation in this graph.

For fixed N, we design an undirected CONGEST graph $G^N = (V^N, E^N)$ with $O(N^2)$ nodes, each having degree at most 3. (Note that this graph is *not* the same as the graph provided as input X to a transformer; this graph is consistent across all transformers taking input of sequence size N.) Let $u_1, \ldots, u_N$ be nodes in V^N corresponding to each input. For every pair $i, j \in [N]$, let $v_{i,j}$ be a node as well. For each $i \in [N]$, let $B_i = (V_i, E_i)$ be a balanced binary trees having root u_i and leaves $v_{i,1}, \ldots, v_{i,N}, v_{1,i}, \ldots, v_{N,i}$. Hence, each

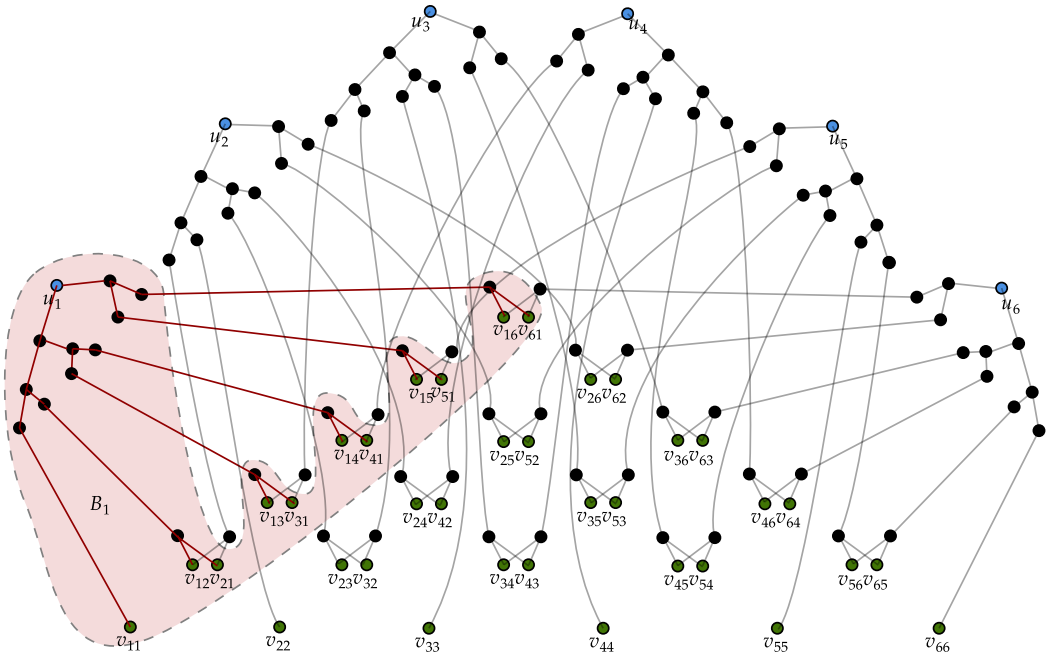


Figure 5.6: The CONGEST graph G^N visualized for $N = 6$ with root nodes $\{u_i\}_{i\in[N]}$ in blue, leaf nodes $\{v_{i,j}\}_{i,j\in[N]}$ in green, and the nodes V_1 of the binary tree B_1 shaded red and edges E_1 colored red.

B_i has $O(N)$ vertices of degree 3 and is of depth $O(\log N)$. Let $V^N = V_1 \cup \cdots \cup V_N$ and $E^N = E_1 \cup \cdots \cup E_N$. Noting that $E_1, \ldots, E_N$ are disjoint and that $V_1, \ldots, V_N$ are disjoint, except for leaves $v_{i,j}$, we ascertain that G^N contains $O(N^2)$ vertices of degree at most 3 and has diameter $O(\log N)$. We visualize the graph G^N with a highlighted tree B_1 in Figure 5.6.

Lemma 5.27. *For any transformer* $f \in \mathsf{GraphTransformer}^{D,H}_{d,m,d',p}$ *and any* $X \in \mathbb{R}^{N\times d}$ *with* p*-bit fixed-precision numbers, there exists a* CONGEST communication protocol on the graph G^N that shares p bits of information between adjacent vertices per round satisfying the following characteristics:

- Before any communication begins, each node u_i is provided with x_i and each node $v_{i,j}$ is provided with $x_{i,j}$ and $x_{j,i}$.
- After $T = O(HD(m + \log N))$ rounds of communication, each node u_i outputs $f(X)_i$.

Proof. It suffices to give a protocol that computes the outcome of a single-headed unit of

graph self-attention with parameters $Q, K, V \in \mathbb{R}^{m \times m}$ and $\kappa : \{-1, 1\}^2 \times \mathbb{R} \to \mathbb{R}$ and transmits its ith output back to u_i in $O(m \log N)$ rounds of p-bit communication. The remainder of the argument involves computing the outcomes of all element-wise MLPs within respective vertices $u_1, \ldots, u_N$ (since we assume each node to have unbounded computational power in the CONGEST model) and to repeat variants of the protocol HD times for every individual self-attention unit. Because the protocol is designed for a particular transformer architecture f, we can assume that every node in the CONGEST graph has knows every parameter of f.

We give the protocol in stages. We assume inductively that every input to f, $y_1, \ldots, y_N \in \mathbb{R}^m$, is known by its respective vertex $u_1, \ldots, u_N$.

1. Every vertex u_i computes $Q^\mathsf{T} y_i \in \mathbb{R}^m$ and propagates it to every vertex $v_{i,1}, \ldots, v_{i,N}$. This can be done in $O(m + \log N)$ rounds by transferring one p-bit fixed-precision number per round from an element of the binary tree B_i to each of its children per round. Because the respective edges $E_1, \ldots, E_N$ are disjoint, this operation can be carried out in parallel.

2. Each u_i computes $K^\mathsf{T} y_i, V^\mathsf{T} y_i \in \mathbb{R}^m$ and propagates them to $v_{1,i}, \ldots, v_{N,i}$ in $O(m + \log N)$ rounds.

3. Each $v_{i,j}$, using their knowledge of $x_{i,j}$ and $x_{j,i}$, computes

$$\alpha_{i,j} := \exp(\kappa(x_{i,j}, x_{j,i}, y_i^\mathsf{T} Q K^\mathsf{T} y_j)).$$

   This takes zero rounds.

4. Each u_i computes $\sum_{j=1}^N \alpha_{i,j}$ by propagating each $\alpha_{i,j}$ in $v_{i,j}$ up B_i to u_i, iteratively summing terms passed up. This takes $O(\log N)$ rounds.

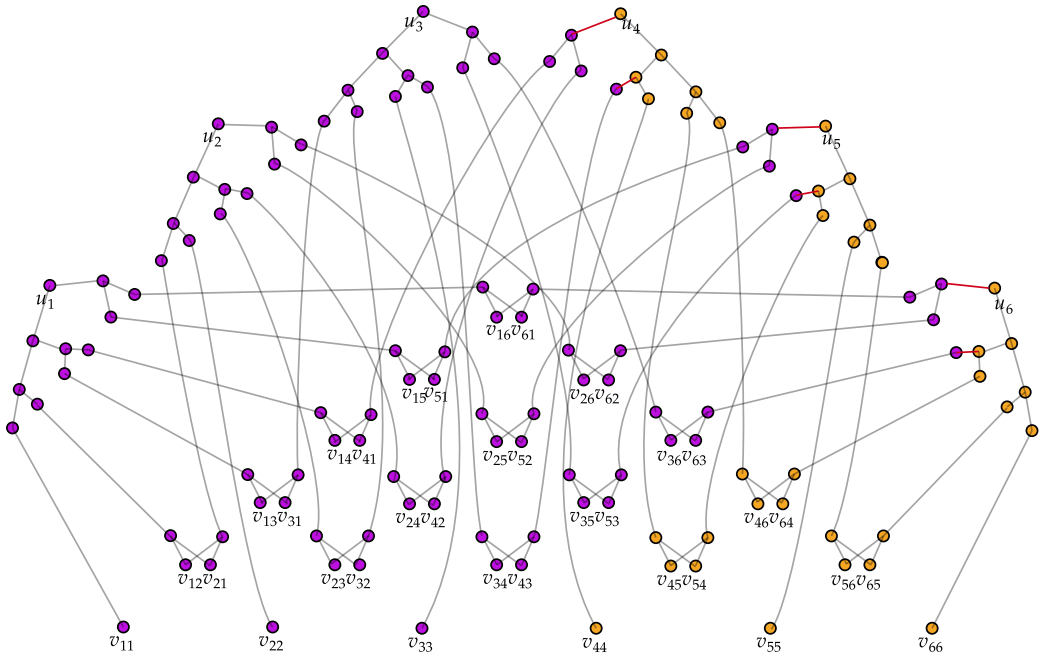


Figure 5.7: The CONGEST graph G^N with vertices partitioned into sets V_a^N (violet) and V_b^N (orange) for $N = 6$. The six edges cut by the partition are colored red.

5. Similarly, u_i computes $\sum_{j=1}^{N} \alpha_{i,j} V^\mathsf{T} y_j$ in $O(m \log N)$ rounds. Then, it computes

$$\frac{\sum_{j=1}^{N} \alpha_{i,j} V^\mathsf{T} y_j}{\sum_{j=1}^{N} \alpha_{i,j}},$$

which is the target output of the self-attention unit.

Because all steps are achievable in parallel with $O(m + \log N)$ rounds, the claim follows. □

5.5.7.2 Reduction from set disjointness

Before proving Theorems 5.23 and 5.25 by embedding an instance of a transformer model into an instance of each subgraph identification problem, we first introduce a partition of the vertices V^N of the CONGEST graph into those possessed by Alice and Bob for use in a two-party communication protocol. We call those two sets V_a^N and V_b^N.

Note that the previous section made no assumptions about the organization of edges in the binary tree. We thus add an additional condition: that each binary tree B_i can be

oriented to respect the left-to-right ordering $v_{i,1}, v_{1,i}, \ldots, v_{i,N}, v_{N,i}$. Let $u_i \in V_a^N$ if and only if $i \leq \frac{N}{2}$, and $v_{i,j} \in V_a^N$ if and only if $\min(i,j) \leq \frac{N}{2}$. We label are remaining nodes in B_i by labeling a parent node w_p as a function of its child nodes w_ℓ and w_r using the following rules:

(a) If $w_\ell, w_r \in V_a^N$, then let $w_p \in V_a^N$.

(b) If $w_\ell, w_r \in V_b^N$, then let $w_p \in V_b^N$.

(c) Otherwise, let $w_p \in V_a^N$ if and only if root $u_i \in V_a^N$.

This partition, which we visualize in Figure 5.7, bounds the number of bits Alice and Bob can exchange by simulating a protocol on CONGEST graph G^N.

Lemma 5.28. *Suppose Alice and Bob simulate an R-round p-bit protocol on* CONGEST communication graph G^N where Alice has access to all vertices V_a^N and Bob V_b^N. No other communication is permitted besides sharing bits as permitted by the CONGEST protocol between neighboring vertices. Then, Alice and Bob exchange at most $O(pRN \log N)$ bits.

Proof. It suffices to show that the partition V_a^N, V_b^N induces a cut of size at most $O(N \log N)$; this ensures that each can send no more than $O(pN \log N)$ bits per round.

Per the rules defined above, an edge in (w_p, w_ℓ) and (w_p, w_r) is cut if and only if they are described by case (c). Within each tree B_i under the orientation described above, an inductive argument shows that in every layer, all elements in V_a^N are to the left of all elements in V_b^N. Thus, there exists at most one parent of that layer that belongs to case (c), and thus, no more than one cut edge per layer. Because each tree has $O(\log N)$ layers and because there are N trees, the partition cuts at most $O(N \log N)$ edges. □

It remains to embed an instance of DISJ in V_a^N, V_b^N for each problem such that its output corresponds identically with that of DISJ.

Proof of Theorem 5.23. Assume for the sake of simplicity that N is divisible by 5. Let $a, b \in \{0,1\}^n$ for $n = \frac{N^2}{25}$ be an input to DISJ, and let Alice and Bob possess a and b respectively.

We index those vectors as $a = (a_{1,1}, a_{1,2}, \ldots, a_{N/5,N/5-1}, a_{N/5,N/5})$ and $b = (b_{1,1}, \ldots, b_{N/5,N/5})$ for ease of analysis. We design input matrix $X \in \{0,1\}^{N \times N}$ as follows:

- If $i \in (0, \frac{N}{5}]$ and $j \in (\frac{N}{5}, \frac{2N}{5}]$, then $x_{i,j} = x_{j,i} = a_{i,j-N/5}$.
- If $i \in (\frac{N}{5}, \frac{3N}{5}]$ and $j \in (\frac{2N}{5}, \frac{4N}{5}]$, then $x_{i,j} = x_{j,i} = \delta_{i,j-N/5}$.
- If $i \in (\frac{3N}{5}, \frac{4N}{5}]$ and $j \in (\frac{4N}{5}, N]$, then $x_{i,j} = x_{j,i} = b_{j-4N/5,i-3N/5}$.
- If $i \in (\frac{4N}{5}, N]$ and $j \in (0, \frac{N}{5}]$, then $x_{i,j} = x_{j,i} = \delta_{i,j+4N/5}$.
- Otherwise, $x_{i,j} = 0$.

This ensures that X has a 5-cycle if and only there exist $i, j \in (0, \frac{N}{5}]$ such that $a_{i,j}b_{i,j} = 1$[6]. In addition, note that under the protocol in Lemma 5.27, Alice's and Bob's inputs a and b are known exclusively by nodes belonging to V_a^N and V_b^N respectively.

Consider any transformer architecture $f \in \mathsf{GraphTransformer}_{N,m,1,p}^{D,H}$ that computes Cycle5. By Lemma 5.27, there exists a protocol on the CONGEST graph G^N that computes Cycle5 after $O(HD(m + \log N))$ rounds of communication of p-bits each. If Alice and Bob simulate this protocol, and output 1 if and only if at least one of their outputs indicates the existence of a Cycle5, then they successfully decide DISJ. By Lemma 5.28, this communication algorithm solves DISJ after exchanging $O(mpHDN \log^2 N)$ bits of communication. However, Fact 5.7 implies that no communication algorithm can do so without exchanging $\Omega(n) = \Omega(N^2)$ bits, which concludes the proof. □

Proof of Theorem 5.25. The proof is identical to its predecessor, but uses a different embedding of an instance $a, b \in \{0,1\}^n$ to DISJ. Let $n = \frac{N^2}{16}$. Then:

- If $i \in (0, \frac{N}{4}]$ and $j \in (\frac{N}{2}, \frac{3N}{4}]$, then $x_{i,j} = a_{i,j-N/2}$.
- If $i \in (\frac{N}{2}, \frac{3N}{4}]$ and $j \in (\frac{3N}{4}, N]$, then $x_{i,j} = b_{j-3N/4,i-N/2}$.

[6]We consider 5-cycles rather than 4-cycles because a spurious 4-cycle could exist among edges $\{x_{i,j} : i \in (0, \frac{N}{5}],\ j \in (\frac{N}{5}, \frac{2N}{5}]\}$.

- If $i \in (\frac{3N}{4}, N]$ and $j \in (0, \frac{N}{4}]$, then $x_{i,j} = \delta_{i,j+3N/4}$.
- Otherwise, $x_{i,j} = 0$.

This construction ensures that a directed 3-cycle exists if and only if a corresponding pair of elements in a and b are both 1. □

5.6 Conclusion

This chapter introduces a novel communication complexity lens on the representational powers of attention units. Using this lens, we design a collection of tasks that crystallize the strengths and limitations of transformer models and sharply characterize the representational powers afforded by changing the embedding dimension. In identifying the intrinsic pairwise aspect of transformers, this work introduced the concept of higher-order tensor attention, which would be further studied by Alman and Song (2023).

While at first glance, the results of this chapter fail to provide the exponential width separations of Chapters 2 and 3, the results here convey a similarly sharp separation in the regime where the context length N is exponentially large and the embedding dimension is thought of as the width. These results differ more broadly from the previous chapters in their modeling assumptions—in particular, the attention to bit-precision, the scaling of N, and the arbitrarily expressive MLP units. However, by making these assumptions, these results are more directly applicable to the study of transformers in practice and provide a more fine-grained understanding of the representational power of attention layers.

This chapter directly leads into the next, where we consider the role of depth in the representational power of transformers. While the research project culminating in the subsequent work originally aspired to resolve Conjecture 5.21, it would ultimately consider a broader computational model of deep transformers that would characterize the powers of depth in terms of compositional algorithms. This subsequent work subsumes the communication complexity lens into a novel *distributed computing lens* into the limitations of transformers.

Taken together, these two works pose the following question:

> *Is the communication lens on transformers merely a proof technique for negative results, or does it provide a fundamental understanding of the representational power of transformers?*

The next chapter indicates that the answer to this question may be the latter, due to the ability of self-attention units to simulate complex communication protocols between model inputs.

Chapter 6: Parallelizability of deep transformer networks

We show that a self-attention layer can efficiently simulate—and be simulated by—a fixed number of communication rounds of *Massively Parallel Computation*. Consequently, we show that logarithmic depth is sufficient for transformers to solve basic computational tasks that cannot be efficiently solved by several other neural sequence models and sub-quadratic transformer approximations. We thus establish parallelism as a key distinguishing property of transformers.

The research presented in this chapter reflects the work of Sanford, Hsu, and Telgarsky (2024).

6.1 Introduction

The transformer (Vaswani et al., 2017) has emerged as the dominant neural architecture for many sequential modeling tasks such as machine translation (Radford et al., 2019) and protein folding (Jumper et al., 2021). Reasons for the success of transformers include suitability to modern hardware and training stability: unlike in recurrent models, inference and training can be efficiently parallelized, and training is less vulnerable to vanishing and exploding gradients. However, the advantages of transformers over other neural architectures can be understood more fundamentally via the lens of *representation*, which regards neural nets as parameterized functions and asks what they can efficiently compute.

Many previous theoretical studies of transformers establish (approximation-theoretic and computational) universality properties, but only at large model sizes (Yun et al., 2020; Pérez, Barceló, and Marinkovic, 2021). These results are not unique to transformers and reveal little about which tasks can be solved in a *size-efficient* manner. Several other works (e.g., Hahn,

2020; Merrill and Sabharwal, 2022; Sanford, Hsu, and Telgarsky, 2023) give fine-grained representational results in the scaling regime where context length grows but model depth is constant. In this regime, basic algorithmic tasks like matching parentheses and evaluating Boolean formulas are impossible.

In this work, we identify parallelism as a key to distinguishing transformers from other architectures. While recurrent architectures process their inputs serially, transformers allow independent interactions between the input tokens, mediated by the inner products between query and key embeddings in self-attention units. We leverage this property of self-attention to establish a formal connection between transformers and *Massively Parallel Computation (MPC)* (Karloff, Suri, and Vassilvitskii, 2010). Concretely, we design transformers that simulate MPC protocols (and vice versa), and in doing so, we exhibit a wide range of computational tasks solved by logarithmic-depth transformers, including those that cannot be efficiently solved with other architectures such as graph neural nets (GNNs) and recurrent models.

6.1.1 Our results

We advance the understanding of transformers' representational capabilities with the following results.

1. The algorithmic capabilities and limitations of logarithmic-depth transformers are captured by the MPC model (Section 6.3).
2. There is a simple sequential task that (i) is solved by (and, empirically, learned from data using) logarithmic-depth transformers, but (ii) *cannot* be efficiently solved by several alternative architectures (Sections 6.4 and 6.6).

In more detail, our first collection of results, Theorems 6.3 and 6.8, show that any R-round MPC protocol can be implemented by a transformer of depth $O(R)$, and that any depth-L transformer can be simulated by an $O(L)$-round MPC protocol. The former implies that several graph problems are solved by logarithmic-depth transformers (Corollary 6.5); the

latter suggests the near-optimality of these transformers (Corollary 6.9), under the assumption of a well-known conjecture about the limitations of MPC algorithms (Conjecture 6.1). A key technical step (Lemma 6.4) shows how transformers can implement the simultaneous message-passing used in MPC protocols to communicate between machines. While Chapter 5 uses communication complexity to understand the representational limitations of self-attention layers, our results show the benefits of the communication lens for understanding the strengths of transformers.

Our second set of results concerns the *k-hop induction heads* task, a synthetic sequential task that draws inspiration from the induction heads primitive of Elhage et al. (2021). The theoretical results of Section 6.4 prove that depth $L = \Theta(\log k)$ is necessary and sufficient for efficient transformer representation. An accompanying empirical investigation reveals that transformers trained on the task obey the same threshold and recover a similar model to the theoretical construction. In contrast, Section 6.6 illustrates that bounded-size non-parallelizable recurrent architectures—including state-space models like Mamba (Gu and Dao, 2023)—cannot solve the task. Moreover, well-known transformer models with computationally efficient alternatives to self-attention, like Performer (Choromanski et al., 2022) and Longformer (Beltagy, Peters, and Cohan, 2020), and shallow transformers with chain-of-thought prompting sacrifice their abilities to implement parallel algorithms, as evidenced by their proven inability to solve this task.

6.1.2 Related work

Some of the types of lower bounds we sought in this work were inspired by the literature on depth-separation for feed-forward neural networks (e.g., Eldan and Shamir, 2016; Daniely, 2017a; Telgarsky, 2016), which exhibit functions that are efficiently approximated by deep networks, but not by shallower networks.

Many theoretical approaches have been used to understand the representational capabilities of transformers and self-attention units in various scaling regimes. Some works model

(variants of) transformers as machines for recognizing formal languages, such as the Dyck languages (Hahn, 2020; Bhattamishra, Ahuja, and Goyal, 2020; Yao et al., 2021; Hao, Angluin, and Frank, 2022) and star-free regular languages (Angluin, Chiang, and Yang, 2023). These approaches reveal the inability of fixed-size transformers to handle arbitrarily long inputs. Other works show how transformers can simulate finite-state automata (Liu et al., 2022) with logarithmic depth, and Turing machines with (unrolled) depth (or chain-of-thought length) scaling polynomially with total runtime (Wei, Chen, and Ma, 2022; Malach, 2023; Merrill and Sabharwal, 2023a). However, it is unclear if these results are near-optimal or even transformer-specific.

Theoretical results about the limitations of constant-depth transformers have been articulated by way of analogy to circuit complexity (Merrill and Sabharwal, 2023b; Merrill, Sabharwal, and Smith, 2022; Merrill and Sabharwal, 2022; Strobl, 2023; Strobl et al., 2023), implying the inability of constant-depth transformers to solve tasks like graph connectivity and Boolean formula evaluation. Other works characterize the representational capabilities of one-layer transformers (Likhosherstov, Choromanski, and Weller, 2021; Sanford, Hsu, and Telgarsky, 2023), but these approaches do not apply to deeper models. Sanford, Hsu, and Telgarsky study multi-headed attention using communication complexity, a framing that informs this work's connection to distributed computing.

The MPC model (Karloff, Suri, and Vassilvitskii, 2010; Beame, Koutris, and Suciu, 2017; Goodrich, Sitchinava, and Zhang, 2011; Andoni et al., 2014b; Im et al., 2023) was introduced to study distributed computing frameworks such as MapReduce (Dean and Ghemawat, 2004). A major goal is to design protocols that require few rounds of communication for setups in which each machine's local memory is sublinear in the input size. Many advances have been made in MPC algorithms for important problems (see, e.g., Im et al., 2023, for a recent survey). However, a basic problem that has resisted progress is connectivity in sparse graphs, where all MPC protocols in this memory regime appear to require $\Omega(\log n)$ rounds for input graphs on n vertices. Lower bounds in MPC and related models were stud-

ied by Beame, Koutris, and Suciu (2017), Roughgarden, Vassilvitskii, and Wang (2018), and Charikar, Ma, and Tan (2020). The conjectured impossibility of $o(\log n)$-round protocols for connectivity is now used as the basis for conditional lower bounds (Ghaffari, Kuhn, and Uitto, 2019).

Simulation of transformers by recurrent models (Oren et al., 2024) and simulation of graph neural nets (GNNs) by transformers (Kim et al., 2022) offer some coarse-grain insight into the relationship between these architectures, but separations are not implied by these previous works. Our connection between transformers and MPC most closely resembles the association established by Loukas (2019) between GNNs and the CONGEST model of distributed computation. Both works establish positive and negative results by identifying neural architectures with communication protocols. In Section 6.6.1, we show that the MPC connection allows transformers to solve graph connectivity more efficiently than GNNs.

Our k-hop induction heads task is designed as a k-fold composition of its standard analog (Elhage et al., 2021). It is similar to a special case of the LEGO reasoning task (Zhang et al., 2023), which reveals the super-linear benefit of depth relative to k; in our case, we theoretically and empirically exhibit an exponential benefit. We also draw a connection to the well-studied problem of pointer-chasing (Papadimitriou and Sipser, 1982; Duris, Galil, and Schnitger, 1984; Nisan and Wigderson, 1993), which enables the proof of our separation between parallel and serial architectures. Our fine-grained empirical interpretability analysis for synthetic tasks draws inspiration from similar analyses of sequential algorithms like sorting and reversal (Li and McClelland, 2022).

6.2 Preliminaries

6.2.1 Massively Parallel Computation model

We use the definition of MPC from Andoni et al. (2018).

Definition 6.1 (MPC protocol)**.** For any global and local memory constants $\gamma, \delta > 0$,

- $\texttt{Input} = (\texttt{Input}_1, \ldots, \texttt{Input}_{n_{\text{in}}}) \in \mathbb{Z}_{2^p}^{n_{\text{in}}}$ is distributed across local memories of machines $1 \le i \le \lceil \frac{n_{\text{in}}}{s} \rceil$:

$$\texttt{MachineIn}_i^{(1)} = \{(\texttt{Input}_\iota, \iota) : \iota \in \{(s-1)i+1, \ldots, \min\{n_{\text{in}}, si\}\}\}.$$

- For round $r = 1, \ldots, R$:
  - Each machine i computes messages $(\texttt{MsgOut}_{i,j}^{(r)})_{j=1,2,\ldots}$ to send to machines $(\texttt{Dest}_{i,j}^{(r)})_{j=1,2,\ldots}$ as function of $\texttt{MachineIn}_i^{(r)}$:

$$\texttt{MachineOut}_i^{(r)} = \textsc{Local}_{r,i}(\texttt{MachineIn}_i^{(r)}) = \{(\texttt{MsgOut}_{i,j}^{(r)}, \texttt{Dest}_{i,j}^{(r)}) \in \mathbb{Z}_{2^p}^{d_j} \times [q] : j = 1, 2, \ldots\};$$
$$\textstyle\sum_j d_j \le s \text{ is ensured.}$$

  - All messages are simultaneously transmitted; the messages in local memory of machine i for round $r+1$ are:

$$\texttt{MachineIn}_i^{(r+1)} = \{(\texttt{Msg}, \texttt{Src}) : (\texttt{Msg}, i) \in \texttt{MachineOut}_{\texttt{Src}}^{(r)}\};$$
$$\textstyle\sum_{(\texttt{Msg},\texttt{Src}) \in \texttt{MachineIn}_i^{(r+1)}} |\texttt{Msg}| \le s \text{ is ensured.}$$

- $\texttt{Output} = f(\texttt{Input})$ comes from

$$\texttt{MachineIn}_i^{(R+1)} = \{(\texttt{Output}_\iota, \texttt{Src}) : \iota \in \{(s-1)i+1, \ldots, \min\{n_{\text{out}}, si\}\}\}$$

  for $1 \le i \le \lceil \frac{n_{\text{out}}}{s} \rceil$.

Figure 6.1: Formal execution of an MPC protocol for computing $f\colon \mathbb{Z}_{2^p}^{n_{\text{in}}} \to \mathbb{Z}_{2^p}^{n_{\text{out}}}$. ($|\texttt{Msg}|$ is the number of words in $\texttt{Msg}$.)

a (γ, δ)-*MPC protocol* for a function $f : \mathbb{Z}_{2^p}^{n_{\text{in}}} \to \mathbb{Z}_{2^p}^{n_{\text{out}}}$ specifies a distributed computing protocol for $q = \Theta(n_{\text{in}}^{1+\gamma-\delta})$ machines, each with $s = O(n_{\text{in}}^{\delta})$ words[1] of local memory to jointly compute $f(\texttt{Input})$ for any given $\texttt{Input} \in \mathbb{Z}_{2^p}^{n_{\text{in}}}$ as follows. The $\texttt{Input} \in \mathbb{Z}_{2^p}^{n_{\text{in}}}$ is distributed across the local memories of the first $\lceil n_{\text{in}}/s \rceil$ machines. Computation proceeds in rounds. In each round, each machine computes an arbitrary function of its local memory to prepare at most s words to send to other machines; messages are simultaneously transmitted, and the protocol ensures that each machine receives at most s words at the end of the round. After the final round, the $\texttt{Output} = f(\texttt{Input}) \in \mathbb{Z}_{2^p}^{n_{\text{out}}}$ is in the local memories of the first $\lceil n_{\text{out}}/s \rceil$ machines. See Figure 6.1 for details.

[1] We assume the word size is $p = \Theta(\log n_{\text{in}})$ bits. For convenience, we regard words as elements of $\mathbb{Z}_{2^p}$ (integers mod 2^p).

Our negative results in Section 6.3.2 are conditional on the well-known "one-versus-two cycle" conjecture (Beame, Koutris, and Suciu, 2017; Roughgarden, Vassilvitskii, and Wang, 2018; Ghaffari, Kuhn, and Uitto, 2019).

Conjecture 6.1 (see, e.g., Ghaffari, Kuhn, and Uitto, 2019)**.** *For any $\gamma > 0$, $\delta < 1$, and N, if π is an (γ, δ)-MPC protocol that distinguishes a single cycle on N nodes and a union of two cycles each on $N/2$ nodes, then π uses $\Omega(\log N)$ rounds.*

6.2.2 Transformers

6.2.2.1 Transformer definition

We first define a self-attention head, the core primitive of a transformer. The *softmax* operator is $\mathrm{softmax}(v) = (\exp(v_1), \ldots, \exp(v_N))/\sum_{j=1}^{N} \exp(v_j)$ for $v \in \mathbb{R}^N$. We apply softmax to matrices $A \in \mathbb{R}^{N \times N}$ row-wise, i.e. $\mathrm{softmax}(A)_i = \mathrm{softmax}((A_{i,1}, \ldots, A_{i,N}))$.

Definition 6.2 (Self-attention head)**.** A *self-attention head* is a mapping $f_{Q,K,V} : \mathbb{R}^{N \times m} \to \mathbb{R}^{N \times m}$ defined by

$$f_{Q,K,V}(X) = \mathrm{softmax}(Q(X)K(X)^{\mathsf{T}})V(X)$$

and parameterized by row-wise *query*, *key*, and *value embeddings* $Q, K, V \colon \mathbb{R}^{N \times m} \to \mathbb{R}^{N \times m}$ (e.g., $Q(X) = (Q_1(X_1), \ldots, Q_N(X_N))$. Let Attn_m^N denote the set of all self-attention heads with embedding dimension m and context length N.

A transformer composes L layers of H self-attention heads per layer, plus an output multi-layer perceptron (MLP).

Definition 6.3 (Transformer)**.** A *transformer* is a mapping $T : \mathbb{R}^{N \times d_{\mathrm{in}}} \to \mathbb{R}^{N \times d_{\mathrm{out}}}$ specified by self-attention heads $(f_{\ell,h} \in \mathsf{Attn}_m^L)_{\ell \in [L], h \in [H]}$ and an element-wise output MLP $\psi = (\psi_1, \ldots, \psi_N) : \mathbb{R}^{N \times m} \to \mathbb{R}^{N \times d_{\mathrm{out}}}$. Upon input $X \in \mathbb{R}^{N \times d_{\mathrm{in}}}$, the transformer computes intermediate embeddings $X^0, \ldots, X^L \in \mathbb{R}^{N \times m}$ with $X^0 = X$ and

$$X^\ell = X^{\ell-1} + \sum_{h=1}^{H} f_{\ell,h}(X^{\ell-1}),$$

and returns $T(X) = \psi(X^L)$ as output. Let $\mathsf{Transformer}^N_{m,L,H,d_{\text{in}},d_{\text{out}}}$ denote the set of all such transformers, and $\mathsf{Transformer}^N_{m,L,H} := \mathsf{Transformer}^N_{m,L,H,1,1}$.

Modeling assumptions. We treat the transformer as a computational model that permits arbitrary element-wise computation, but restricts the manner in which multiple elements are processed together. This manifests in our decisions to model query/key/value embeddings and MLPs as arbitrary functions on the embedding space; Loukas (2019) employs a similar modeling assumption for GNNs. Note that the element-wise embeddings and MLPs may be index-specific, obviating the need for positional embeddings.

Our theoretical results cover the scaling regime where the context length N is the main asymptotic parameter; while the embedding dimension m, the number of heads H, and the depth L grow sub-linearly in N. This reflects real-world trends in large-language models, where context length has sharply increased in recent years.

Throughout, we assume all intermediate computations in transformers are represented by p-bit precision numbers for $p = \Theta(\log N)$. Limiting the precision is consistent with recent practice of using low-precision arithmetic with transformers (e.g., Wang et al., 2022; Dettmers et al., 2022). We discuss this precision assumption in greater detail in Section 6.2.2.2, along with other minor technical assumptions (such as the inclusion of a "start token" for mathematical convenience).

Masked transformers. We also consider *masked self-attention*, where only certain inner products influence the softmax output. Let $\Lambda \in \{-\infty, 0\}^{N \times N}$ be a *masking matrix* with at least one zero entry in every row. Then, a Λ-*masked self-attention* unit is defined by

$$f^{\Lambda}_{Q,K,V}(X) = \text{softmax}(Q(X)K(X)^{\mathsf{T}} + \Lambda)V(X).$$

Let $\Lambda\text{-}\mathsf{Attn}^N_m$ and $\Lambda\text{-}\mathsf{Transformer}^N_{m,L,H}$, respectively, denote the sets of all Λ-masked self-attention heads and all transformers comprised of those heads. We define *causally-masked*

transformers by $\mathsf{MaskAttn}^N_m := \Gamma\text{-}\mathsf{Attn}^N_m$ and $\mathsf{MaskTransformer}^N_{m,L,H} := \Gamma\text{-}\mathsf{Transformer}^N_{m,L,H}$, where Γ is the lower-triangular mask with $\Gamma_{i,j} = 0$ iff $i \geq j$.

6.2.2.2 Techical details

We discuss a few minor technicalities and modifications of the self-attention unit (Definition 6.2) and transformer model (Definition 6.3) defined in Section 6.2.2 that are necessary for readers looking for a comprehensive understanding of the proofs of our theoretical results.

Fixed-bit precision arithmetic. As discussed in Section 6.2.2, we assume that all numbers that appear in the intermediate products and outputs of self-attentions are representable with p-bit precision arithmetic, where $p = \Theta(\log N)$. While the details of fixed-precision arithmetic will be uninteresting to most readers, it is necessary to explain precisely what we mean in order to ensure that proofs of results like Theorem 6.8 are sound. Throughout the paper, we allow p to depend on of constants, such as γ, δ, and ϵ.

Concretely, we assume that all query, key, and value embeddings $Q(X), K(X), V(X)$ evaluated on all inputs contain scalar values $z \in \mathbb{R}$ that are polynomially bounded (i.e. $|z| \leq \exp(O(p)) = N^\zeta$ for sufficiently large constant exponent $\zeta > 0$) and are inverse-polynomially discretized (i.e. $z \cdot N^\zeta \in \mathbb{Z}$). Depending on the desired exponent ζ, some $p = \Theta(\log N)$ can be chosen to guarantee this property. While we do not formally analyze the precision needed to approximate the particular embeddings employed by our proofs, we note that our recurring sinusoidal embeddings (e.g. Lemma 6.20) can be discretized without losing their central properties and that discretizations of the restricted isometry embeddings of Proposition 6.10 are analyzed by Chapter 5.

Rather than stipulating a particular bounded-precision implementation that computes the output of a self-attention unit must be implemented, we specify a rounding constraint that any computational implementation of a self-attention unit must satisfy. Precisely, we require that any output round to the same inverse-polynomial discretization as the true

mathematical attention.

Definition 6.4. For a self-attention unit $f \in \mathsf{Attn}_m^N$, let $\hat{f}$ be an finite-precision implementation of that unit. We say that $\hat{f}$ is a *valid implementation* if

$$\sup_{X \in \mathbb{R}^{N \times m}} \left\| f(X) - \hat{f}(X) \right\|_\infty = O\left(\frac{1}{2^p}\right).$$

This definition is only to establishing the fact that self-attention units with sufficient margins can precisely compute hardmax outputs in Lemma 6.2 and to showing that MPC models can indeed compute the outputs precisely in Theorem 6.8.

Hardmax attention. While we exclusively consider attention units with the softmax, our constructions periodically rely on the exact computation of averages of embeddings. We define the *hardmax* operator to allow the consideration of discrete averaging operations. For some $v \in \mathbb{R}^N$, let

$$\text{hardmax}(X)_i = \begin{cases} \frac{1}{|I_{\max}(v)|}, & \text{if } i \in I_{\max}(v) \\ 0 & \text{otherwise,} \end{cases}$$

where $I_{\max}(v) = \{i \in [N] : v_i = \max_{i'} v_{i'}\}$.

We show that bounded-precision softmax self-attention units that satisfy a margin property can be modified slightly to have identical outputs to an analogous hardmax unit.

Lemma 6.2. *Let $f \in \mathsf{Attn}_m^N$ be a self-attention unit with precision $p = \Theta(\log N)$ and embedding functions Q, K, V such that for some fixed $1 \geq \xi = N^{-O(1)}$ and every $X \in \mathbb{R}^{N \times m}$ and $i \in [N]$:*

$$A(X)_{i,i'} \leq \max_{i''} A(X)_{i,i''} - \xi, \ \forall i' \notin I_{\max}(A(X)_i),$$

where $A(X) = Q(X)K(X)^\mathsf{T}$. Then there exists a self-attention unit $f' \in \mathsf{Attn}_m^N$ with a valid p'-bit implementation with $p' = O(p)$ satisfying

$$f'(X) = \text{hardmax}(A(X))V(X).$$

The proof of Lemma 6.2 is provided in Section 6.7.

Start tokens. Our technical proofs are occasionally simplified by including a "dummy token" whose value is passed in self-attention layers as a default or null value. For example, in the proof of Lemma 6.21, the dummy token handles the case where the reference token does not appear previously in the sequence. While we believe that this extra token is not necessary for our technical arguments, we include it for the sake of simplicity.

We model this dummy token as a *start-of-sequence* token X_0. Concretely, if we employ X_0 in a self-attention $f \in \mathsf{Attn}_m^N$ which takes as input X, we instead treat f as an attention unit in Attn_m^{N+1} that operates on $(X_0, X_1, \ldots, X_N)$. We assume that X_0 is constant-valued, and therefore never both to pay attention to its outputs; it's only relevance is via its key and value embeddings $K_0(X_0), V_0(X_0) \in \mathbb{R}^m$. If X_0 is unmentioned, we assume that it does not exist, or is set such that its key embedding inner products are all zero.

Supplemental chain-of-thought tokens. We periodically (see Theorem 6.12 and the proofs of Corollaries 6.9 and 6.19) consider transformers with supplemental blank "chain-of-thought" tokens appended to the end of the sequence. Unlike the start token, these are only constant *at initialization* and may be used deeper in the model to perform meaningful computations.

Let $\mathsf{Transformer}^{N,M}_{m,L,H,d_{\text{in}},d_{\text{out}}}$ denote transformers with $M - N$ extra blank elements appended to the input sequence. Concretely, we represent $T \in \mathsf{Transformer}^{N,M}_{m,L,H,d_{\text{in}},d_{\text{out}}}$ as some $T' \in \mathsf{Transformer}^{M}_{m,L,H,d_{\text{in}},d_{\text{out}}}$ and define the output $T(X)$ for $X \in \mathbb{R}^{N \times d_{\text{in}}}$ by letting $Y \in \mathbb{R}^{M \times d_{\text{in}}}$ for $Y_{1:N} = X$ and $Y_{N+1:M} = \vec{0}$, and letting $T(X) = T'(Y)$.

6.2.3 Graphs as sequential inputs

When providing a graph $G = (V, E)$ as input to transformers or MPC protocols, we serialize G as a sequence in $[|V|]^{2|E|}$ that encodes each edge as a pair of vertex tokens. The resulting transformer has $N = 2|E|$ and $d_{\text{in}} = 1$, and the resulting MPC protocol has

$n_{\text{in}} = 2|E|$.

6.3 Relating transformers and MPC

We coarsely characterize the computational power of transformers in a certain size regime by establishing a bidirectional relationship between transformers and MPC. Theorems 6.3 and 6.8 show that any MPC protocol can be simulated by a transformer, and vice versa. As corollaries (Corollaries 6.5 and 6.9), we obtain tight upper and lower bounds on the depth of bounded-size transformers for computing connected components in graphs.

6.3.1 Simulation of MPC protocols by transformers

The following theorem shows that any MPC protocol π with sublinear local memory can be simulated by a transformer whose depth L is linear in the number of rounds R of π, and embedding dimension m is polynomial in the local memory size $s = O(N^\delta)$ of machines used by π.

Theorem 6.3. *For constants $0 < \gamma < \delta < 1$ and any deterministic R-round (γ, δ)-MPC protocol π on n_{in} input words and $n_{\text{out}} \leq n_{\text{in}}$ output words, there exists a transformer $T \in \mathsf{Transformer}^N_{m,L,H}$ with $N = n_{\text{in}}, m = O(n_{\text{in}}^{4\delta} \log n_{\text{in}}), L = R + 1, H = O(\log \log n_{\text{in}})$ such that $T(\texttt{Input})_{:n_{\text{out}}} = \pi(\texttt{Input})$ for all $\texttt{Input} \in \mathbb{Z}^N_{2^p}$.*

The theorem provides a non-trivial construction in the strongly sub-linear local memory regime when $s = O(N^{1/4-\epsilon})$ for any $\epsilon > 0$.[2] Whether the simulation can be improved to $m = O(N^{1-\epsilon'})$ for some $\epsilon' > 0$ whenever $s = O(N^{1-\epsilon})$ is an interesting question for future work.

[2]Applying Theorem 6.3 when $\delta \geq \frac{1}{4}$ yields transformers with embedding dimension $m \geq N$, which trivializes the transformer architecture and negates any advantages of depth under our MLP universality assumption. This is due to the fact a transformer with N-dimensional embeddings could aggregate the entire input sequence $X \in \mathbb{R}^N$ in a single embedding and use its output MLP to compute any arbitrary function on that input.

Theorem 6.3 proof overview. At a high level, the proof in Section 6.3.3.2 entails simulating each round of parallel computation with a single-layer transformer and applying those constructions serially to $\texttt{Input}$. The local computation on each machine (represented by $\texttt{MachineOut}_i^{(r)} = \textsc{Local}_{r,i}(\texttt{MachineIn}_i^{(r)})$) is directly encoded using element-wise query/key/value embeddings.

The crux of the proof simulates a *routing protocol* to determine $\texttt{MachineIn}^{(r+1)}$ from $\texttt{MachineOut}^{(r)}$. We construct a self-attention unit that ensures that an encoding of a sequence of addressed messages from each machine are properly routed to their destinations.[3]

For any message size β, message count bound s, and number of tokens N, we say that $(\texttt{Sent}, \texttt{Rcvd}) \in \mathbb{R}^{N\times m} \times \mathbb{R}^{N\times m}$ is a *valid* (β, s)*-routing* if, for each $i \in [N]$, the i-th row of $\texttt{Sent}$ (resp. $\texttt{Rcvd}$) is the vector encoding of some $\texttt{Sent}_i \subset \mathbb{Z}_{2^p}^{\beta} \times [N]$ (resp. $\texttt{Rcvd}_i \subset \mathbb{Z}_{2^p}^{\beta} \times [N]$) such that

$$\texttt{Rcvd}_i = \left\{(\texttt{Msg}, \texttt{Src}) : (\texttt{Msg}, i) \in \texttt{Sent}_{\texttt{Src}}\right\},$$

and each of $\texttt{Rcvd}_i$ and $\texttt{Sent}_i$ has cardinality at most s.[4]

Lemma 6.4. *For any $\beta, s, N \in \mathbb{N}$, there exists a transformer $\text{route}_{\beta,s} \in \mathsf{Transformer}_{m,1,1}^{N}$ with $m = O(s^4 \beta \log N)$ satisfying $\text{route}_{\beta,s}(\texttt{Sent}) = \texttt{Rcvd}$ for any valid (β, s)-routing $(\texttt{Sent}, \texttt{Rcvd})$.*

The proof of Lemma 6.4 appears in Section 6.3.3.1 and combines two key techniques: sparse propagation and multiple hashing. The former is a simple variant of the "sparse averaging" task of Chapter 5, which simultaneously computes N averages over subsets of inputs; this task is solved a single self-attention head with small embedding dimension (Proposition 6.10). Using sparse propagation, we construct a self-attention head that averages the $\leq s$ encodings of each $\texttt{Rcvd}_{\texttt{Src}}$ for every $\texttt{Src} \in \texttt{Rcvd}_i$. In order to ensure that we can decode that average of encodings, we apply error-correction by encoding each $\texttt{Output}_i$ in a sparse and redundant manner, where each outgoing messages appears as multiple copies of the same

[3]This routing between machines uses the all-pairs structure of self-attention and may not admit a subquadratic approximation.

[4]We abuse notation by writing $\texttt{Dest} \in \texttt{Sent}_i$ to mean there exists some $\texttt{Msg}$ such that $(\texttt{Msg}, \texttt{Dest}) \in \texttt{Sent}_i$.

addressed "packet."

Application: connectivity with log-depth transformers. As an immediate consequence of Theorem 6.3, any graph problem solvable with a logarithmic number of rounds of MPC computation (and local memory s) is also computable by a logarithmic depth transformer (and embedding dimension $\tilde{O}(s^4)$). The following result—which bounds transformer depth needed to compute connected components of a graph G—follows from Theorem 6.2 of Coy and Czumaj (2022), which derandomizes an MPC algorithm of Behnezhad et al. (2019), and Theorem 6.3.

Corollary 6.5. *For any constant $\epsilon \in (0,1)$ and any $D \leq N$, there exists a transformer in $\mathsf{Transformer}^N_{m,L,H}$ with $m = O(N^\epsilon)$, $H = O(\log \log N)$, and $L = O(\log D)$ that identifies the connected components of any input graph $G = (V, E)$ with $|V|, |E| = O(N)$ where each connected component has diameter at most D.*

Theorem 8.1 and Corollary 8.2 of Coy and Czumaj (2022) give efficient MPC protocols for other graph problems besides connectivity, and therefore, as corollaries of Theorem 6.3, we also obtain log-depth transformers for these problems.

Corollary 6.6 (Spanning forest construction)**.** *For any constant $\epsilon \in (0,1)$ and any $D \leq N$, there exists a transformer in $\mathsf{Transformer}^N_{m,L,H}$ with $m = O(N^\epsilon)$, $H = O(\log \log N)$, and $L = O(\log D)$ that computes a rooted spanning forest of any input graph $G = (V, E)$ with $|V|, |E| = O(N)$ where each connected component has diameter at most D.*

Corollary 6.7 (Minimum spanning forest construction)**.** *For any constant $\epsilon \in (0,1)$ and any $D_{MSF} \leq N$, there exists a transformer in $\mathsf{Transformer}^N_{m,L,H}$ with $m = O(N^\epsilon)$, $H = O(\log \log N)$, and $L = O(\log D_{MSF})$ that identifies the connected components of any input graph $G = (V, E)$ with $|V|, |E| = O(N)$ and $\mathrm{poly}(N)$-bounded integer weights whose minimum spanning forest has diameter at most D_{MSF}.*

6.3.2 Simulation of transformers by MPC protocols

The following theorem shows that MPC protocols can simulate transformers and prove depth lower bounds on transformers, conditioned on Conjecture 6.1. We get, as a corollary, the conditional optimality of the transformer depth bound in Corollary 6.5.

Theorem 6.8. *For any transformer* $T \in \mathsf{Transformer}^N_{m,L,H}$ *(or* Λ-$\mathsf{Transformer}^N_{m,L,H}$*) with* $mH = O(N^\delta)$ *for* $\delta \in (0,1)$ *and any* $\delta' \in (\delta, 1)$*, there exists a* $O(\frac{L}{\delta'-\delta})$*-round* $(1+\delta', \delta')$*-MPC protocol with* $q = O(N^2)$ *machines with* $s = O(N^{\delta'})$ *local memory for computing* T*.*

Theorem 6.8 demonstrates that the algorithmic capabilities of transformers are no stronger than those of MPC protocols with a quadratic scaling in the number of machines. While Theorems 6.3 and 6.8 do not jointly provide a sharp characterization of the two computational models, the reductions are tight enough to provide strong evidence for the optimality of the connected components construction of Corollary 6.5.

Theorem 6.8 proof overview. At a high-level, the proof constructs an MPC protocol that simulates a self-attention layer by separating the computation of MLPs and attention matrices into three separate categories of machines.

- Each input token is provided to its own *token machine*, responsible for preparing the query/key/value embeddings.
- Each pair of tokens is associated with an *inner product machine* that will compute the inner product between their respective query and key embeddings.
- *Propagation machines* ensure that embeddings are routed to the proper inner product machine and compute outputs of each softmax unit.

The proof gives the communication protocol for these machines, shows how they simulate a layer of self-attention in $O(1/(\delta' - \delta))$ rounds, and establishes the sufficiency of $O(N^2)$ machines with $O(N^{\delta'})$ local memory.

Application: conditional optimality of Corollary 6.5. Assuming the well-established Conjecture 6.1, we prove an $\Omega(\log D)$ lower bound on the depth of parameter-efficient transformers for determining connectivity of graphs where connected components may have diameter up to D.

Corollary 6.9. *Let $\epsilon \in (0,1)$ be any constant, and let $D \geq N^\epsilon$. Assume Conjecture 6.1, and suppose there exists $T \in \mathsf{Transformer}^N_{m,L,H}$ with $mH = O(D^{1-\epsilon})$ that decides connectivity of any input graph with connected components having diameter $\leq D$. Then $L = \Omega(\log D)$.*

6.3.3 Proofs for Section 6.3.1

6.3.3.1 Proof of Lemma 6.4

Lemma 6.4. *For any $\beta, s, N \in \mathbb{N}$, there exists a transformer $\mathrm{route}_{\beta,s} \in \mathsf{Transformer}^N_{m,1,1}$ with $m = O(s^4 \beta \log N)$ satisfying $\mathrm{route}_{\beta,s}(\texttt{Sent}) = \texttt{Rcvd}$ for any valid (β, s)-routing $(\texttt{Sent}, \texttt{Rcvd})$.*

The proof relies on a *sparse propagation* sequential primitive, which complements the sparse averaging primitive of Chapter 5. For any $Q \leq d, N$, on input $X = (X_1, \ldots, X_N) \in \mathbb{R}^{N \times d}$ with $X_i = (z_i, S_i) \in \mathbb{R}^{d-Q} \times [N]^Q$ and $b_i = |\{S_j \ni i : j \in [N]\}| \leq Q$, we define

$$\mathrm{sparsePropagate}_{Q,d}(X)_i = \begin{cases} \frac{1}{b_i} \sum_{S_j \ni i} z_j & \text{if } b_i > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Closely following the argument of Chapter 5, we show in Proposition 6.10 that there is a self-attention unit with embedding dimension $m = \max(d, O(q \log N))$ that computes $\mathrm{sparsePropagate}_{Q,d}$. This construction is a key component of the single-layer transformer used in the proof of Lemma 6.4.

Proposition 6.10. *For any $b \leq N$ and d, there exists a self-attention unit*

$$\mathrm{sparsePropagate}_{Q,d} \in \mathsf{Attn}^N_{m,p}$$

for $m = d + O(Q \log N)$ *and* $p = O(\log N)$*, which, given any input* X *with*

$$X_i = (z_i, S_i, \vec{0}) \in \mathbb{R}^d \times \binom{[N]}{\leq Q} \times \{0\}^{m-Q-d}$$

such that $b_i = |\{S_j \ni i : j \in [N]\}| \leq Q$ *for all* i*, has output* $\text{sparsePropagate}_{Q,d}(X)$ *satisfying*

$$\text{sparsePropagate}_{Q,d}(X)_i = \frac{1}{b_i} \sum_{S_j \ni i} z_j.$$

The proof of Proposition 6.10 appears in Section 6.7.

Proof of Lemma 6.4. We construct a single-layer single-headed transformer with query, key, and value embeddings Q, K, V and output MLP ψ. Q, K, V can be decomposed as $Q = Q' \circ \phi$, $K = K' \circ \phi$, $V = V' \circ \phi$, for some input MLP ϕ and embeddings Q', K', V'. We fix Q', K', V' to be the respective embeddings of the self-attention unit with embedding dimension m from Proposition 6.10 that computes $Y = \text{sparsePropagate}_{s,m}(X)$ for $X_{\texttt{Src}} = (z_{\texttt{Src}}, S_{\texttt{Src}})$ for every $\texttt{Src} \in [N]$ to be determined. Hence, the proof entails designing element-wise encoders $\phi = (\phi_1, \ldots, \phi_N)$ and decoders $\psi = (\psi_1, \ldots, \psi_N)$ that compute `Rcvd` from `Sent`, using $\text{sparsePropagate}_{s,m}$ as an intermediate step. A high-level overview of the proof construction is visualized in Figure 6.2.

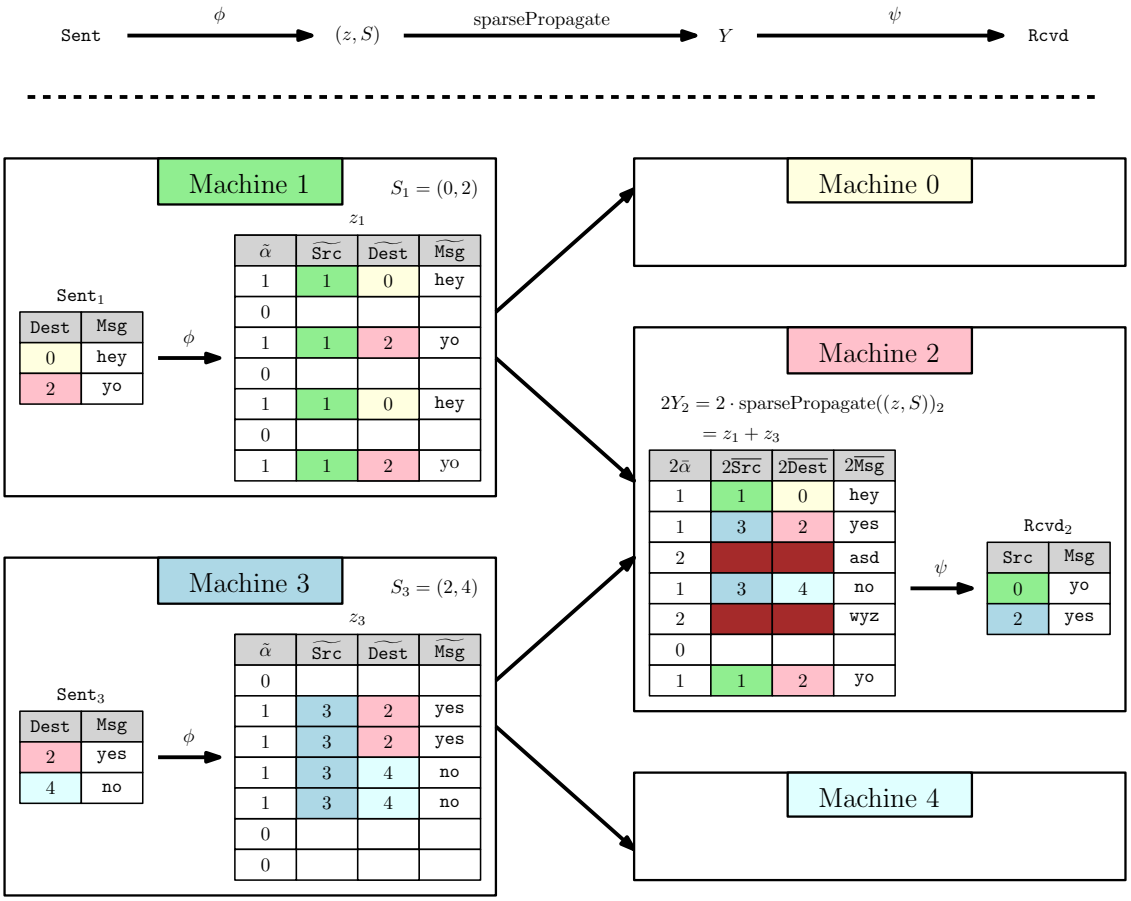


Figure 6.2: A visualization of the construction used to prove Lemma 6.4 in three phases—the encoding of each input $\texttt{Sent}_{\texttt{Src}}$ as embedding $z_{\texttt{Src}}$ and subset $S_{\texttt{Src}}$ with ϕ; the combination of those embeddings into $Y_{\texttt{Dest}}$ via the simulation of $\text{sparsePropagate}_{s,m}((z, S))$; and the decoding of each $Y_{\texttt{Dest}}$ into output $\texttt{Rcvd}_{\texttt{Dest}}$ with ψ. The figure provides an example of the encoding and decoding where machines 1 and 3 transmit messages to machine 2. "Multiple hashing" is used to compute z_1 and z_3 by encoding each message in multiple fixed-location "packets" in embedding space space. This redundancy ensures the possibility of machine 2 decoding $\texttt{Rcvd}_2$ from Y_2, due to each message occurring alone at least once in the encoding.

On input $\texttt{Sent}_{\texttt{Src}}$, we use the encodings $Q_{\texttt{Src}}, K_{\texttt{Src}}, V_{\texttt{Src}}$ to specify that all tokens $\texttt{Dest}$ with $\texttt{Dest} \in \texttt{Sent}_{\texttt{Src}}$ (or equivalently, all $\texttt{Dest}$ with $\texttt{Src} \in \texttt{Rcvd}_{\texttt{Dest}}$) should receive a copy of the encoding of $\texttt{Sent}_{\texttt{Src}}$. That is, we set $S_{\texttt{Src}} := \{\texttt{Dest} \in \texttt{Sent}_{\texttt{Src}}\}$ for each $\texttt{Src} \in [N]$. This ensures that Y satisfies

$$Y_{\texttt{Dest}} = \frac{1}{|\texttt{Rcvd}_{\texttt{Dest}}|} \sum_{\texttt{Src} \in \texttt{Rcvd}_{\texttt{Dest}}} z_{\texttt{Src}}.$$

While it's tempting to simply set each $z_{\texttt{Src}} \in \mathbb{R}^m$ equal to a (βs)-dimensional vectorization of $\texttt{Sent}_{\texttt{Src}}$, it is unclear how to extract $\texttt{Rcvd}_{\texttt{Dest}}$ from each $Y_{\texttt{Dest}}$, since each average performed

by sparsePropagate$_{s,m}$ will combine multiple vector embeddings in a shared space. In order to avoid these troubles, we employ a *multiple hasing-based encoding* that treats messages as "packets" identified by a message, a source, a destination, and a "validity token" that can be used to determine whether a message is uncorrupted. We include multiple copies of each packet in the encoding $z_{\texttt{Src}}$. For notational ease, we represent each $z_{\texttt{Src}} \in \mathbb{R}^m$ as a collection of packets

$$z_{\texttt{Src}} = (\widetilde{\texttt{Msg}}_{\texttt{Src},j}, \widetilde{\texttt{Src}}_{\texttt{Src},j}, \widetilde{\texttt{Dest}}_{\texttt{Src},j}, \alpha_{\texttt{Src},j})_{j\in[m']} \in (\mathbb{Z}_{2^p}^{\beta} \times [N] \times [N] \times \{0,1\})^{m'},$$

where $m = m'(3+\beta)$.

To sparsely and redundantly encode each $\texttt{Sent}_{\texttt{Src}}$ as $z_{\texttt{Src}}$, we encode outgoing messages as packets by utilizing the matrix A guaranteed by the following fact (which we use with $n := N^2$, $b := s^2$, and $m' := d = O(s^4 \log N)$).

Fact 6.11. *For any n, $b \le n$, and $d \ge \lceil 12b^2 \ln n \rceil$, there exists a binary matrix $A \in \{0,1\}^{n\times d}$ such that, for every subset $S \subseteq [n]$ with $|S| \le b$, the columns of the sub-matrix $A_S \in \{0,1\}^{|S|\times d}$ contains all S-dimensional elementary vectors, i.e., $\left\{e_1, \ldots, e_{|S|}\right\}$ is a subset of the columns of A_S.*

The proof of Fact 6.11 is at the end of the section. We use the following rule to determine which (if any) message to encode as a packet at each $\texttt{Src} \in [N]$ and $j \in [m']$. We let $A_{(\texttt{Src},\texttt{Dest}),j} = A_{N(\texttt{Src}-1)+\texttt{Dest},j}$ for notational convenience.

$$z_{\texttt{Src},j} = \begin{cases} (\texttt{Msg}, \texttt{Src}, \texttt{Dest}, 1) & \text{if } (\texttt{Msg}, \texttt{Dest}) \in \texttt{Sent}_{\texttt{Src}} \text{ and } A_{(\texttt{Src},\texttt{Dest}),j} = 1 \\ & \quad \text{and } A_{(\texttt{Src},\texttt{Dest}'),j} = 0,\ \forall\ \texttt{Dest}' \in \texttt{Sent}_{\texttt{Src}} \setminus \{\texttt{Dest}\}, \\ (\vec{0}, 0, 0, 0) & \text{otherwise.} \end{cases}$$

In Figure 6.2, this encoding is visualized in the tables of "Machine 1" and "Machine 3," where the entirety of each message is encoded in two fixed and distinct locations in the

embeddings z_1 and z_3, alongside metadata about the source of message and the validity $\tilde{\alpha}$. Each message is encoded as multiple identical packets in different embedding dimensions and a large fraction of embedding locations are left blank. These features are critical for the proper evaluation of the decoding step ψ.

We analyze the $Y = \text{sparsePropagate}_{\beta,m}(X)$ outputs, letting

$$Y_{\texttt{Dest}} = (Y_{\texttt{Dest},1}, \ldots, Y_{\texttt{Dest},m'}), \quad Y_{\texttt{Dest},j} \in (\mathbb{R}^\beta \times \mathbb{R} \times \mathbb{R} \times \mathbb{R})^{m'},$$

with all numbers represented with p-bit fixed precision. This analysis shows that there exists an element-wise decoder MLP ψ satisfying $\psi_{\texttt{Dest}}(Y_{\texttt{Dest}}) = \texttt{Rcvd}_{\texttt{Dest}}$ for all $\texttt{Dest} \in [N]$. For any $j \in [m']$, observe from the definition of $z_{\texttt{Src}}$ and $\text{sparsePropagate}_{s,m}$ that

$$\begin{aligned} Y_{\texttt{Dest},j} &=: \left(\overline{\texttt{Msg}}_{\texttt{Dest},j}, \overline{\texttt{Src}}_{\texttt{Dest},j}, \overline{\texttt{Dest}}_{\texttt{Dest},j}, \bar{\alpha}_{\texttt{Dest},j}\right) \\ &= \frac{1}{|\texttt{Rcvd}_{\texttt{Dest}}|} \sum_{\texttt{Src} \in \texttt{Rcvd}_{\texttt{Dest}}} \left(\widetilde{\texttt{Msg}}_{\texttt{Src},j}, \widetilde{\texttt{Src}}_{\texttt{Src},j}, \widetilde{\texttt{Dest}}_{\texttt{Src},j}, \alpha_{\texttt{Src},j}\right). \end{aligned}$$

Before formally analyzing this construction, we motivate its utility with Figure 6.2. The encoding $2Y_2$ of Machine 2 contains four "clean" rows j with $2\bar{\alpha}_{2,j} = 1$, two "corrupted" rows with $2\bar{\alpha}_{2,j} = 2$, and one "blank" row with $2\bar{\alpha}_{2,j} = 0$.

- The **blank row** contains no information about any incoming messages, since neither Machine 1 nor Machine 3 encoded messages as packets in these locations. The fact that $2\bar{\alpha}_{2,j} = 0$ certifies the blankness of this row, and hence, the decoder ψ can ignore it.

- The **corrupted rows** correspond to locations where both Machine 1 and Machine 3 saved messages as packets. As a result, the corresponding embedding $Y_{2,j} = \frac{1}{2}(z_{1,j} + z_{3,j})$ is an average of two non-zero embeddings and is hence "corrupted." Because $2\bar{\alpha}_{2,j} = 2$, the decoder ψ detects the corruption and ignores it when computing $\texttt{Rcvd}_2$.

- The **clean rows** are locations where exactly one of Machine 1 and Machine 3 encoded

a message. Hence, these messages can be cleanly understood by the decoder ψ, which simply validates the "cleanliness" of the row with $2\bar{\alpha}_{2,j} = 1$, determines whether Machine 2 is indeed the target recipient of the respective message, and saves all such messages in the decoding $\texttt{Rcvd}_2$.

We prove the validity of this intuition by ensuring that the encoding scheme successfully encodes each incoming message in a clean row and that the category of each row (blank, corrupted, or clean) can be detected by the decoder ψ. We observe the following sequence of facts about every $Y_{\texttt{Dest}}$. Let

$$\texttt{Relevant}_{\texttt{Dest}} := \{(\texttt{Msg}, \texttt{Src}', \texttt{Dest}') : \texttt{Src}' \in \texttt{Rcvd}_{\texttt{Dest}},\ (\texttt{Msg}, \texttt{Dest}') \in \texttt{Sent}_{\texttt{Src}'}\}$$

denote the set of *all* messages sent by sources of messages sent to $\texttt{Dest}$.

1. Consider any outgoing message $(\texttt{Msg}, \texttt{Src}', \texttt{Dest}') \in \texttt{Relevant}_{\texttt{Dest}}$. By the property of A guaranteed by Fact 6.11, there exists some j such that $A_{(\texttt{Src}', \texttt{Dest}'), j} = 1$ and $A_{(\texttt{Src}'', \texttt{Dest}''), j} = 0$ for every $(\texttt{Src}'', \texttt{Dest}'') \in \texttt{Relevant}_{\texttt{Dest}} \setminus \{(\texttt{Src}', \texttt{Dest}')\}$. As a result of the definition of the encoding z and the averaged representation of $Y_{\texttt{Dest}}$:

$$Y_{\texttt{Dest},j} = \frac{1}{|\texttt{Rcvd}_{\texttt{Dest}}|} (\texttt{Msg}, \texttt{Src}', \texttt{Dest}', 1)\,. \tag{6.1}$$

2. Conversely, if $\bar{\alpha}_{\texttt{Dest},j} = 1/|\texttt{Rcvd}_{\texttt{Dest}}|$, then there exists a unique $(\texttt{Msg}, \texttt{Src}', \texttt{Dest}') \in \texttt{Relevant}_{\texttt{Dest}}$ such that (6.1) is satisfied.

3. If at least one message is received, then the minimal nonzero value of $\bar{\alpha}_{\texttt{Dest}}$ is $1/|\texttt{Rcvd}_{\texttt{Dest}}|$.

We design $\psi_{\texttt{Dest}}$ to uniquely identify $\texttt{Rcvd}_{\texttt{Dest}}$ from $Y_{\texttt{Dest}}$ as follows. If at least one message is received, then $1/|\texttt{Rcvd}_{\texttt{Dest}}|$ can be identified by finding the smallest nonzero value of $\bar{\alpha}_{\texttt{Dest}}$. The decoder ψ inspects every $Y_{\texttt{Dest},j}$ satisfying $\bar{\alpha}_{\texttt{Dest},j} = 1/|\texttt{Rcvd}_{\texttt{Dest}}|$, which therefore satisfies

$$|\texttt{Rcvd}_{\texttt{Dest}}| \cdot (\overline{\texttt{Msg}}_{\texttt{Dest},j}, \overline{\texttt{Src}}_{\texttt{Dest},j}, \overline{\texttt{Dest}}_{\texttt{Dest},j}) \in \texttt{Relevant}_{\texttt{Dest}}.$$

Thus, if $|\texttt{Rcvd}_{\texttt{Dest}}| \cdot \overline{\texttt{Dest}}_{\texttt{Dest},j} = \texttt{Dest}$, then $|\texttt{Rcvd}_{\texttt{Dest}}| \cdot (\overline{\texttt{Msg}}_{\texttt{Dest},j}, \overline{\texttt{Src}}_{\texttt{Dest},j}) \in \texttt{Rcvd}_{\texttt{Dest}}$, and ψ encodes it as such.

□

Fact 6.11. *For any n, $b \leq n$, and $d \geq \lceil 12b^2 \ln n \rceil$, there exists a binary matrix $A \in \{0,1\}^{n \times d}$ such that, for every subset $S \subseteq [n]$ with $|S| \leq b$, the columns of the sub-matrix $A_S \in \{0,1\}^{|S| \times d}$ contains all S-dimensional elementary vectors, i.e., $\left\{e_1, \ldots, e_{|S|}\right\}$ is a subset of the columns of A_S.*

Proof. Let $\mathrm{col}(A)$ denote the set of columns of A. We use the probabilistic method and consider A with iid entries $A_{i,j} \sim \text{Bernoulli}(\frac{1}{b+1})$. We bound the probability of failure:

$$\begin{aligned}
\Pr\left[\exists S \in \binom{[n]}{\leq b} \text{ s.t. } \left\{e_1, \ldots, e_{|S|}\right\} \not\subset \mathrm{col}(A_S)\right] &\leq b \cdot n^b \Pr\left[e_i \notin \mathrm{col}(A_S)\right] \\
&\leq n^{b+1} \left(1 - \frac{1}{b+1} \cdot \left(1 - \frac{1}{b+1}\right)^b\right)^d \\
&\leq n^{b+1} \left(1 - \frac{1}{e(b+1)}\right)^d \\
&\leq n^{b+1} \cdot \exp\left(-\frac{d}{e(b+1)}\right) \\
&< \exp\left((b+1)\ln n - \frac{d}{3(b+1)}\right) \leq 1.
\end{aligned}$$

Therefore, there exists a matrix A with the claimed property. □

6.3.3.2 Proof of Theorem 6.3

We give a generalization of Theorem 6.3 that simulates a broader family of MPC protocol, including those with more than n machines (i.e. $\gamma \geq \delta$). We accommodate this generalization by simulating MPC protocols with the generalized transformer family $\mathsf{Transformer}^{N,M}_{m,L,H}$ detailed in Section 6.2.2.2 with supplemental blank "chain-of-thought" tokens.

Theorem 6.12 (Generalization of Theorem 6.3)**.** *For constant $\gamma, \delta > 0$ and any potentially randomized R-round (γ, δ)-MPC protocol π on n_{in} input words and $n_{\text{out}} \leq n_{\text{in}}$ output words,*

there exists a transformer $T \in \mathsf{Transformer}^{N,M}_{m,L,H}$ *with*

$$N = n_{\text{in}}, M = \max(n_{\text{in}}, O(n_{\text{in}}^{1+\gamma-\delta})), m = O(n_{\text{in}}^{4\delta} \log n_{\text{in}}), L = R + 1, H = O(\log \log n_{\text{in}})$$

such that

$$T(\texttt{Input})_{:n_{\text{out}}} = \pi(\texttt{Input}).$$

Theorem 6.3 is an immediate consequence of Theorem 6.12 by noting that $M = N$ for sufficiently large n_{in} when $\gamma < \delta$. Its central construction is summarized in Figure 6.3.

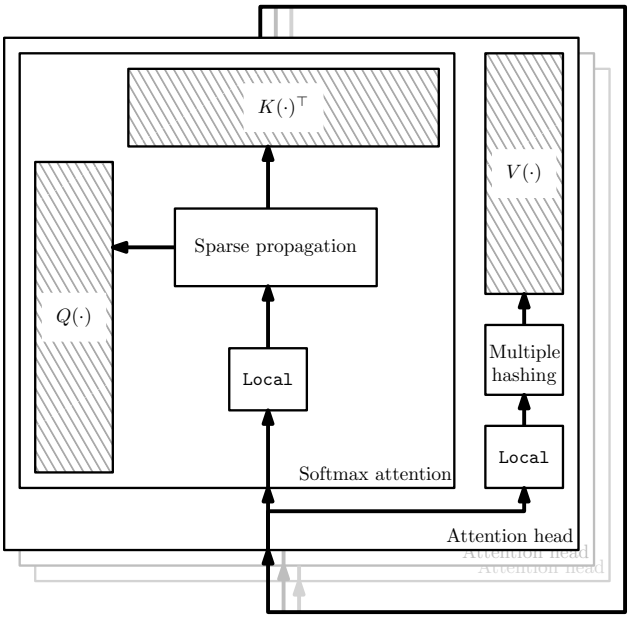


Figure 6.3: To simulate MPC, the local computation within each machine is pushed inside $Q(\cdot), K(\cdot), V(\cdot)$, and then the pairwise attention matrix performs message routing. To ensure proper routing and also that the outputs of $Q(\cdot), K(\cdot), V(\cdot)$ are all tall-and-skinny matrices, the construction carefully utilizes both multiple hashing and sparse propagation.

Proof. Consider any MPC protocol π with $q = O(n_{\text{in}}^{1+\gamma-\delta})$ machines and $s = O(n_{\text{in}}^{\delta})$ local memory that, following the notation of Definition 6.1, maps $\texttt{Input} \in \mathbb{Z}_{2^p}^{n_{\text{in}}}$ to $\texttt{Output} \in \mathbb{Z}_{2^p}^{n_{\text{out}}}$ with intermediates

$$\texttt{MachineIn}^{(1)}, \ldots \texttt{MachineIn}^{(R)} \text{ and } \texttt{MachineOut}^{(1)}, \ldots, \texttt{MachineOut}^{(R)}$$

and deterministic functions $(\textsc{Local}_{r,i})_{r\in[R],i\in[q]}$ with

$$\texttt{MachineOut}_i^{(r)} = \textsc{Local}_{r,i}(\texttt{MachineIn}_i^{(r)}).$$

To simulate the protocol, we let every machine $i \in [q]$ correspond to a particular position in the transformer's context. A transformer that simulates π can then be constructed that consolidates $\texttt{Input}$ onto $\lceil n_{\text{in}}/s \rceil$ machines to match $\texttt{MachineIn}^{(1)}$; computes $\texttt{MachineIn}^{(r+1)}$ from $\texttt{MachineIn}^{(r)}$ for each $r = 1, \ldots, R-1$; and computes and properly distributes $\texttt{Output}$ from $\texttt{MachineIn}^{(r)}$. These three elements of the construction exist due to the following lemmas, which are proved later.

Lemma 6.13. *For any MPC protocol π with local memory s and q machines with n_{in}-word inputs, there exists a transformer* $\text{init} \in \mathsf{Transformer}^{n_{\text{in}},\max(n_{\text{in}},q)}_{s,1,1,d_{\text{in}},d_{\text{out}}}$ *with $d_{\text{in}} = 1$ and $d_{\text{out}} = s$, which, given $\texttt{Input} \in \mathbb{Z}_{2^p}^n$, has output satisfying* $\text{init}(\texttt{Input}) = \texttt{MachineIn}^{(1)}$.

Lemma 6.14. *For any R-round MPC protocol π with local memory s and q machines and any $r \in [R-1]$, there exists a transformer* $\text{round}^{(r)} \in \mathsf{Transformer}^{q}_{m,1,H,d_{\text{in}},d_{\text{out}}}$ *with $H = O(\log\log q)$, $m = O(s^4 \log q)$, and $d_{\text{in}} = d_{\text{out}} = s$ which, given any valid input $X = \texttt{MachineIn}^{(r)} \in \mathbb{Z}_{2^p}^{q\times m}$ under the MPC protocol in vectorized form, has output satisfying* $\text{round}^{(r)}(X) = \texttt{MachineIn}^{(r+1)}$.

Lemma 6.15. *For any R-round MPC protocol π with local memory s and q machines with n_{out}-word output, there exists a transformer* $\text{final} \in \mathsf{Transformer}^{q,\max(n_{\text{out}},q)}_{s,1,1,d_{\text{in}},d_{\text{out}}}$ *for $d_{\text{in}} = s$ and $d_{\text{out}} = 1$, which, given input $X = \texttt{MachineIn}^{(R)}$, has output* $\text{final}(X)$ *with* $\text{final}(X)_{i,1} = \texttt{Output}_i \in \mathbb{Z}_{2^p}$.

The proof immediate from the three lemmas. We construct the final transformer T by stacking the single-layer constructions as a single transformer with embedding dimension m:

$$T = \text{final} \circ \text{round}^{(R-1)} \circ \cdots \circ \text{round}^{(1)} \circ \text{init}.$$

The proofs of Lemmas 6.13 and 6.15 rely on simple constructions with fixed attention matrices and appear in Section 6.7. The proof of Lemma 6.14 relies on Lemma 6.4 and is proved in the following section. □

Proof of $\text{round}^{(r)}$ **construction.** To prove the existence single-layer transformer that simulates $\text{round}^{(r)}$, we separate the computational task into two steps: (i) obtaining $\texttt{MachineOut}^{(r)}$ from $\texttt{MachineIn}^{(r)}$ and (ii) obtaining $\texttt{MachineIn}^{(r+1)}$ from $\texttt{MachineOut}^{(r)}$. Because the former requires no communication between machines, we can encode that conversion in the input MLP to the transformer.

The nontrivial part of the reduction is thus the latter step, which we obtain by utilizing multiple single-headed attention units $\text{route}_{\beta,s}$ of Lemma 6.4 to route messages of different sizes to their recipients. The difficulty in this task is the mismatch in functionality between the two computational models: while the MPC model ensures that each recipient automatically receives its intended messages, transformers must implement this functionality manually, while ensuring that multiple messages do not overwrite one another.

The following lemma implements that routing functionality for all messages, using different attention heads depending on the size of the message. We prove Lemma 6.14 at the end of the section as a simple modification of Lemma 6.16.

Lemma 6.16. *For any R-round MPC protocol π with local memory s and q machines and any $r \in [R-1]$, there exists a transformer* $\text{route}^{(r)} \in \mathsf{Transformer}^{q}_{m,1,H}$ *with $H = O(\log \log q)$ and $m = O(s^4 \log q)$, which, given any valid input $X = \texttt{MachineOut}^{(r)} \in \mathbb{Z}_{2^p}^{q \times m}$ under the MPC protocol in vectorized form, has output satisfying* $\text{route}^{(r)}(X) = \texttt{MachineIn}^{(r+1)}$.

Because at most s messages can be shared and received by each machine, and each message is of size at most s, we can prove an single-headed alternative to Lemma 6.16 with a somewhat suboptimal dependence on embedding dimension. By applying by Lemma 6.4 with message size $\beta = s$, bounded number of messages s, and context length $N = q$, there exists a transformer $\text{route}_{s,s}$ with $H = 1$ and $m = O(s^5 \log q)$ that computes $\texttt{MachineIn}^{(r+1)}$

from $\texttt{MachineOut}^{(r+1)}$ by regarding each outgoing message as belonging to $\mathbb{Z}_{2^p}^s$ by adding padding dimensions as needed.

We improve the embedding dimension to $m = O(s^4 \log q)$ by running in parallel $O(\log \log N)$ transformers guaranteed by Lemma 6.4 that encode differently sized messages. The number of heads H increases at a doubly-logarithmic rate because of a doubling trick employed on the size of message encodings used by constituent part.

Proof. We describe an implementation of $\text{route}^{(r)}$ by considering any fixed input

$$\texttt{MachineOut}^{(r)} \in \mathbb{Z}_{2^p}^{q \times m}.$$

For each $i \in [q]$ and some integer sequence $1 = \beta_0 < \beta_1 < \cdots < \beta_H = s + 1$, we partition $\texttt{MachineOut}_i^{(r)}$ into H disjoint subsets as follows. For any $h \in [H]$, let

$$\begin{aligned}
\texttt{Sent}_i^h &:= \left\{(\texttt{Msg}, \texttt{Dest}) \in \texttt{MachineOut}_i^{(r)} : \dim(\texttt{Msg}) \in [\beta_{h-1}, \beta_h]\right\}, \\
\texttt{Rcvd}_i^h &:= \left\{(\texttt{Msg}, \texttt{Src}) \in \texttt{MachineIn}_i^{(r+1)} : \dim(\texttt{Msg}) \in [\beta_{h-1}, \beta_h]\right\},
\end{aligned}$$

and note that $\texttt{MachineOut}_i^{(r)} = \dot{\bigcup}_{h=1}^H \texttt{Sent}_i^h$ and $\texttt{MachineIn}_i^{(r+1)} = \dot{\bigcup}_{h=1}^H \texttt{Rcvd}_i^h$.

For each $h \in [H]$, note that $\dim(\texttt{Msg}) \le \beta_h$, and $\left|\texttt{Sent}_i^h\right| = \left|\texttt{Rcvd}_i^h\right| \le s/\beta_{h-1}$. As a result, Lemma 6.4 guarantees the existence of a single-headed transformer $\text{route}_h^{(r)}$ such that $\text{route}_h^{(r)}(\texttt{Sent}^h) = \texttt{Rcvd}^h)$ with embedding dimension $m_h \le C s^4 \beta_h \log(q)/\beta_{h-1}^4$ for some sufficiently large universal constant C.

We defined $\text{route}^{(r)}$ as the computation of $\text{route}_1^{(r)}, \ldots, \text{route}_H^{(r)}$ as H parallel heads of self-attention with disjoint embeddings concatenated into in m-dimensional embedding space with $m = \sum_{h=1}^H m_h$. We conclude by letting

$$\beta_h = \begin{cases} 1 & \text{if } h = 0, \\ \min(2\beta_{h-1}^3, q+1) & \text{if } h \in [H], \end{cases}$$

noting that $\beta_H = q + 1$ for $H = O(\log \log q)$, and bounding m:

$$m \leq \sum_{h=1}^{H} \frac{Cs^4 \log(q)\beta_h}{\beta_{h-1}^4} \leq 2Cs^4 \log(q) \cdot \sum_{h=1}^{H} \frac{1}{\beta_{h-1}}$$
$$\leq 2Cs^4 \log(q) \cdot \sum_{h=1}^{H} \frac{1}{2^{h-1}} = O(s^4 \log q). \qquad \square$$

Proof of Lemma 6.14. To simulate a round of MPC protocol π by mapping $\texttt{MachineIn}^{(r)}$ and ρ_r to $\texttt{MachineIn}^{(r+1)}$, the single-layer transformer $\text{round}^{(r)}$ first computes $\texttt{MachineOut}^{(r)}$ element-wise and then properly routes messages in $\texttt{MachineOut}^{(r)}$ to their proper destination. We define $\text{round}^{(r)} = \text{route}^{(r)} \circ \textsc{Local}_r$ for $\text{route}^{(r)}$ in Lemma 6.16 and

$$\textsc{Local}_{r,i}(\texttt{MachineIn}_i^{(r)}, \rho_{r,i}) = \texttt{MachineOut}_i^{(r)}.$$

This can be immediately constructed as a single-layer transformer by prepending the embeddings Q, K, V of the construction of $\text{route}^{(r)}$ with $\textsc{Local}_r$, using $Q \circ \textsc{Local}_r$, $K \circ \textsc{Local}_r$, $V \circ \textsc{Local}_r$ as the embeddings of $\text{round}^{(r)}$. $\square$

6.3.4 Proofs for Section 6.3.2

6.3.4.1 Proof of Theorem 6.8

As in Section 6.3.3.2, we give and prove a generalized version of Theorem 6.8 that broadens the family of considered transformers to include masked models and those that contain extra blank chain-of-thought tokens, using notation from Section 6.2.2.2.

Theorem 6.17 (Generalization of Theorem 6.8)**.** *For any transformer $T \in \mathsf{Transformer}_{m,L,H}^{N,M}$ (or $\mathsf{MaskTransformer}_{m,L,H}^{N,M}$) with $mH = O(N^\delta)$ for $\delta \in (0,1)$ and $M = \Theta(N^{1+\alpha})$ for $\alpha \geq 0$ and for any $\delta' \in (\delta, 1)$, there exists an $O(\frac{L(1+\alpha)}{\delta'-\delta})$-round $(1 + 2\alpha + \delta', \delta')$-MPC protocol with $q = O(M^2)$ machines with $s = O(N^{\delta'})$ local memory that outputs the same sequence as $T(X)$ for all $X \in \mathbb{R}^N$.*

Theorem 6.8 is an immediate consequence by setting $M := N$ and $\alpha := 0$.

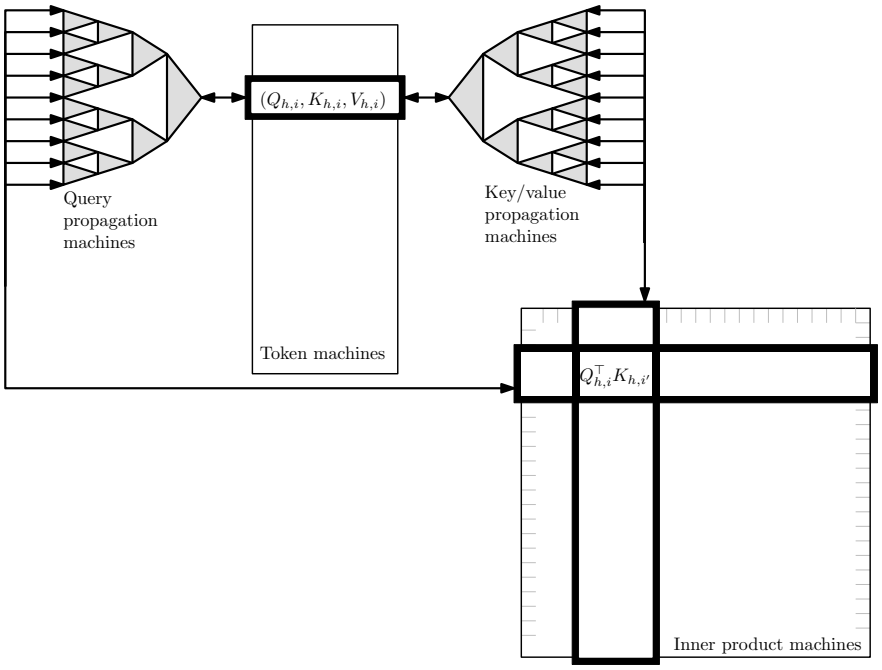


Figure 6.4: This construction employs M^2 *inner product machines* to compute the entries of the softmax matrix, and M *token machines* to compute all values of $Q(\cdot), K(\cdot), V(\cdot)$. What is most complex about the construction are the additional machines and message routing needed to propagate these values efficiently between the inner product machines and the token machines, in particular carefully aggregating the output of the attention mechanism and computing its normalization. To this end, the protocol uses additional machines, organized into a tree with branching factor $b = O(N^{\delta'-\delta})$ and depth $D = O(\frac{1+\alpha}{\delta'-\delta})$.

Proof. It suffices to show that an $O(\frac{1+\alpha}{\delta'-\delta})$-round MPC protocol π that simulates a single-layer transformer $T \in \mathsf{Transformer}^M_{m,m,m,1,H}$ with m-dimensional input and output embeddings since a depth-L transformer can be constructed by applying L such protocols sequentially. Moreover, we can ignore the difference between the input context length N and the context length with padding M by assuming that the input contains M tokens.

Concretely, we consider H heads with embeddings $(Q_h, K_h, V_h)_{h\in[H]}$, element-wise output MLP $\psi = (\psi_1, \ldots, \psi_M)$, and any fixed masks $\Lambda_1, \ldots, \Lambda_H \in \{-\infty, 0\}^{M\times M}$. We show that there exists some π such that for any $\texttt{Input} = X \in \mathbb{R}^{M\times m}$,

$$\pi(X) = \psi\left(X + \sum_{h=1}^{H} \mathrm{softmax}(Q_h(X)K_h(X)^\mathsf{T} + \Lambda_h)V_h(X)\right),$$

where numbers in X and all intermediate products of the transformer computation can be

represented with $p = O(\log M)$ bit precision.

Our MPC protocol π, which will use $q = O(M^2)$ machines and $s = \Theta(N^{\delta'})$ words of local memory per machine, assigns each of the q machines to one of four possible roles: token machine, inner product machine, query propagation machine, and key/value propagation machine. We describe these machines below. For the sake of readability, we identify machines with easily interpretable descriptions and use the bijection $\texttt{ID}$ to map each of those to a token in $[q]$ that is used for routing messages. Our protocol has two important parameters: $b = \lfloor s/(4mH) \rfloor = O(N^{\delta'-\delta})$ is the *branching factor* of the protocol, and $D = \lceil \log_b(M) \rceil = O(\frac{1+\alpha}{\delta'-\delta})$ is the *depth* of the protocol.

At a high level (see Figure 6.4 for a corresponding diagram), the protocol involves computing all intermediate products of the of a transformer unit by performing MLP computations in N *token machines*, computing inner products in N^2 *inner product machines*, and using $O(N^2)$ other *propagation machines* arranged in trees to share information between the two in $O(D)$ rounds. The protocol draws inspiration from Section 5.5.7.1, which uses a similar construction to simulate transformers with CONGEST protocols on fixed graphs. It is also similar to the MPC implementation of the MPI AllReduce functionality (MPICH, 2023) described by Agarwal et al. (2014).

- Machine $i \in [M]$ is a *token machine* that performs all element-wise computation on the ith token embedding, including the computation of $(Q_{h,i}(X_i), K_{h,i}(X_i), V_{h,i}(X_i))_{h\in[H]}$ and the final MLP output ψ_i. Let $\texttt{ID}(i) = i$.

- Machine $(i, i') \in [M]^2$ is an *inner product machine* designed to compute the inner products $(Q_{h,i}(X_i)^\mathsf{T} K_{h,i'}(X_{i'}))_{h\in[H]}$.

- Machine $(\texttt{Q}, i, d, k)$ for token $i \in [M]$, depth $d \in [D-1]$ and position $k \in [b^d]$ is a *query propagation machine*. This machine is responsible for handling communication of query tokens $(Q_{h,i}(X_i))_{h\in[H]}$ and of all partially-computed attention outputs for the

ith token between token machine i and inner product machines (i, i') for

$$i' \in \texttt{Descendants}_{d,k} := \left\{b^{D-d}(k-1), \ldots, b^{D-d}k\right\} \cap [M].$$

Concretely, if $\ell = 1$, then the machine communicates with token machine i and query propagation machines $(\texttt{Q}, i, d+1, k')$ for

$$k' \in \texttt{Children}_k := \{b(k-1)+1, \ldots, bk\}.$$

If $\ell = D-1$, then it communicates with inner product machines (i, i') for $i' \in \texttt{Children}_k \cap [M]$ and query propagation machine $(\texttt{Q}, i, d-1, \lfloor k/b \rfloor)$. Otherwise, it communicates with query propagation machines $(\texttt{Q}, i, d-1, \texttt{Parent}_k)$, for $\texttt{Parent}_k := \lfloor k/b \rfloor$, and $(\texttt{Q}, i, d+1, k')$ for $k' \in \texttt{Children}_k$.

- Machine $(\texttt{KV}, i, d, k)$ is a *key/value propagation machine*. This machine is analogous to a query propagation machine, except that it is responsible for the communication of key and value tokens $(Q_{h,i}(X_i), V_{h,i}(X_i))_{h\in[H]}$ between token machine i and inner product machines (i, i') for $i' \in \texttt{Descendants}_{d,k}$.

Since the total number of machines is $q = M + M^2 + M\sum_{d=1}^{D-1} b^d = O(M^2)$, we conclude that the global memory of the protocol is $qs = O(N^{2+2\alpha+\delta'})$, which means the protocol is $(1 + 2\alpha + \delta', \delta')$-MPC. We simulate the transformer using a four stage protocol using $2D + 3 = O(\frac{1+\alpha}{\delta'-\delta})$ rounds of MPC computation.

Stage 1: Token dispersion. Because the input to an MPC protocol $\texttt{Input} = X$ is divided equally among machines $1, \ldots, \lceil MmH/s \rceil$, the first round of MPC computation routes each input token X_i to its respective token machine. This is completed by setting $(i, X_i) \in \texttt{MachineOut}_{i'}^{(1)}$ if $(i, X_i) \in \texttt{MachineIn}_{i'}^{(1)}$. Thus, $\texttt{MachineIn}_i^{(2)} = \{(\texttt{Src}, X_i)\}$ for all token machines $i \in [M]$.

Stage 2: Embedding propagation. In rounds $2, \ldots, D+1$, π computes the respective key, query, and value embeddings in each token machine and propagate them to respective inner product machines using the query and key/value propagation machines. Concretely:

- In round 2, each token machine i (whose memory contains X_i) computes m-dimensional embeddings embeddings

$$Q_i := (Q_{h,i}(X_i))_{h\in[H]}, K_i := (K_{h,i}(X_i))_{h\in[H]}, V_i := (V_{h,i}(X_i))_{h\in[H]}.$$

  It transmits each embedding to the respective depth-1 query and key/value propagation machine nodes, while also preserving knowledge of its own X_i. (In all further rounds, we assume that $((i, X_i)) \in \texttt{MachineOut}_i^{(r)}$ to ensure that token machine i can compute the skip-level connection at the end.) That is,

$$\begin{aligned}\texttt{MachineOut}_i^{(2)} = &\{(i, X_i)\}\\ &\cup \{(\texttt{ID}(\texttt{Q}, i, 1, k'), Q_i) : k' \in \texttt{Children}_1\}\\ &\cup \{(\texttt{ID}(\texttt{KV}, i, 1, k'), (K_i, V_i)) : k' \in \texttt{Children}_1\}.\end{aligned}$$

  Note that the total amount of messages sent is $b \cdot mH + 2b \cdot mH + m \leq s$ and that the only machines receiving messages are size m-messages by token machines and size $\leq 4mH$ messages by query and key/value propagation machines.

- In rounds $r \in \{3, \ldots, D\}$, each query and key/value propagation machine of depth $d = r - 2$ passes embeddings onto their successors. That is,

$$\begin{aligned}\texttt{MachineOut}^{(r)}_{\texttt{ID}(\texttt{Q},i,d,k)} &= \{(\texttt{ID}(\texttt{Q}, i, d+1, k'), Q_i) : k' \in \texttt{Children}_k\},\\ \texttt{MachineOut}^{(r)}_{\texttt{ID}(\texttt{KV},i,d,k)} &= \{(\texttt{ID}(\texttt{KV}, i, d+1, k'), (K_i, V_i)) : k' \in \texttt{Children}_k\}.\end{aligned}$$

- In round $D+1$, the depth-$(D-1)$ query and key/value propagation machines pass

their embeddings onto their respective inner product machines. That is,

$$\begin{aligned}
\texttt{MachineOut}^{(D+1)}_{\texttt{ID}(\texttt{Q},i,D-1,k)} &= \{(\texttt{ID}(i,k'), Q_i) : k' \in \texttt{Children}_k \cap [M]\}\,, \\
\texttt{MachineOut}^{(D+1)}_{\texttt{ID}(\texttt{KV},i,D-1,k)} &= \{(\texttt{ID}(k',i), (K_i, V_i)) : k' \in k' \in \texttt{Children}_k \cap [M]\}\,.
\end{aligned}$$

Stage 3: Softmax computation. In rounds $D+2, \ldots, 2D+2$, computes each inner product and iteratively builds up each attention output by accumulating partial softmax computations. For each query propagation machine $(\texttt{Q}, i, d, k)$ and $h \in [H]$, we let $S_{i,d,k,h}$ and $Z_{i,d,k,h}$ denote its partial normalization and softmax computations respectively. That is,

$$\begin{aligned}
Z_{i,d,k,h} &= \sum_{i' \in \texttt{Descendants}_{d,k}} \exp(Q_{h,i}(X_i)^\mathsf{T} K_{h,i'}(X_{i'}))\mathbb{1}\left\{\Lambda_{i,i'} = 0\right\} \\
&= \begin{cases}
\sum_{k' \in \texttt{Children}_k} Z_{i,d+1,k',h} & \text{if } d \le D-1, \\
\exp(Q_{h,i}(X_i)^\mathsf{T} K_{h,k}(X_k))\mathbb{1}\left\{\Lambda_{i,k} = 0\right\} & \text{if } d = D.
\end{cases} \\
S_{i,d,k,h} &= \frac{1}{Z_{i,d,k,h}} \sum_{i' \in \texttt{Descendants}_{d,k}} \exp(Q_{h,i}(X_i)^\mathsf{T} K_{h,i'}(X_{i'}))V_{h,i'}(X_{i'})\mathbb{1}\left\{\Lambda_{i,i'} = 0\right\} \\
&= \begin{cases}
\sum_{k' \in \texttt{Children}_k} \frac{Z_{i,d+1,k',h}}{Z_{i,d,k,h}} \cdot S_{i,d+1,k',h} & \text{if } d \le D-1, \\
V_{h,k}(X_k)\mathbb{1}\left\{\Lambda_{i,k} = 0\right\} & \text{if } d = D;
\end{cases}
\end{aligned}$$

Note that $S_{i,0,1,h} = (\text{softmax}(Q_h(X)K_h(X)^\mathsf{T} + \Lambda_h)V_h(X))_i$ and let $S_{i,d,k} = (S_{i,d,k,h})_{h\in[H]} \in \mathbb{R}^{H\times m}$ and $Z_{i,d,k} = (Z_{i,d,k,h})_{h\in[H]} \in \mathbb{R}^H$

- In round $D+2$, each inner product machine computes its respective inner products and passes its partial softmax computations to its parent query propagation machine. As a result of round $D+1$, each inner product machine (i, i') recently received the

embeddings necessary to compute the relevant inner product:

$$\begin{aligned}&\texttt{MachineIn}^{(d+2)}_{\texttt{ID}(i,i')}\\&\quad=\{(\texttt{ID}(\texttt{Q},i,D-1,\texttt{Parent}_i),Q_i),(\texttt{ID}(\texttt{KV},i',D-1,\texttt{Parent}_{i'}),(K_{i'},V_{i'}))\}\,.\end{aligned}$$

It propagates the respective partial computations $S_{i,D,i'}$ and $Z_{i,D,i'}$ as follows:

$$\texttt{MachineOut}^{(D+2)}_{\texttt{ID}(i,i')}=\{(\texttt{ID}(\texttt{Q},i,D-1,\texttt{Parent}_i),(S_{i,D,i'},Z_{i,D,i'}))\}\,.$$

Note that each depth-$(D-1)$ query propagation machine receives messages of size at most $b\cdot(m+1)H\le s$.

- In rounds $r\in\{D+3,\ldots,2D\}$, partial softmax computations are received by query propagation machines of depth $d=2D+1-r$, added together, and passed along to their parent machines. That is, given

$$\texttt{MachineIn}^{(r)}_{\texttt{ID}(\texttt{Q},i,d,k)}=\{(\texttt{ID}(\texttt{Q},i,d+1,k'),(S_{i,d+1,k'},Z_{i,d+1,k'})):k'\in\texttt{Children}_k\}\,,$$

each respective machine computes $S_{i,d,k}$ and $Z_{i,d,k}$ recursively and propagates

$$\texttt{MachineOut}^{(r)}_{\texttt{ID}(\texttt{Q},i,d,k)}=\{(\texttt{ID}(\texttt{Q},i,d-1,\texttt{Parent}_k),(S_{i,d,k},Z_{i,d,k})\}\,.$$

- In round $2D+1$, the top-most query propagation tokens pass their partial sums to the token machines:

$$\texttt{MachineOut}^{(2D+1)}_{\texttt{ID}(\texttt{Q},i,1,k)}=\{(i,(S_{i,1,k},Z_{i,1,k}))\}\,.$$

- In round $2D+2$, the token machines compute their respective output of the transformer,

$T(X)_i$. Given input

$$\texttt{MachineIn}_i^{(2D+2)} = \{(k', (S_{i,1,k'}, Z_{i,1,k'})) : k' \in \texttt{Children}_1\} \cup \{(i, X_i)\},$$

the token machine i computes $S_{i,0,1}$ and $H_{i,0,1}$ and then

$$T(X)_i = \psi_i \left(X_i + \sum_{h=1}^{H} \text{softmax}(Q_h(X) K_h(X)^\mathsf{T} + \Lambda_h)_i^\mathsf{T} V_h(X) \right) = \psi_i \left(X_i + \sum_{h=1}^{H} S_{i,0,1,h} \right).$$

This quantity is used as an intermediate product for the final phase of computation.

Stage 4: Token compression. We invert Stage 1 by properly compressing the MPC output in the final round $2D+3$. That is, we let $\texttt{MachineOut}_i^{(2D+2)} = \{(\lfloor imH/s \rfloor + 1, T(X)_i)\}$ for each token machine $i \in [M]$, which ensures that the outputs are condensed in the proper order in machines $1, \ldots, \lceil MmH/s \rceil$.

Precision analysis. In order for the proof to be fully sound, care must be taken to ensure that the computation of each self-attention output $S_{i,0,1,h}$ is handled with proper numeric precision, as discussed in Section 6.2.2.2. We show that each $S_{i,0,1,h}$ is a *valid implementation* of its corresponding self-attention unit, per Definition 6.4.

To do so, we let $\hat{S}_{i,d,k,h}$ and $\hat{Z}_{i,d,k,h}$ denote the p-bit representations of $S_{i,d,k,h}$ and $Z_{i,d,k,h}$, where scalars of $\hat{S}_{i,d,k,h}$ and $\log(\hat{Z}_{i,d,k,h})$ are represented as discretized rational numbers z satisfying $|z| \leq \frac{1}{2} 2^{p/2}$ and $z \cdot 2^{p/2} \in \mathbb{Z}$. For some sufficiently small $p' = \Theta(p)$, we assume that all embeddings $Q_h(X), K_h(X), V_h(X)$ have scalars z satisfying $|z| \leq \frac{1}{2} 2^{p'/2}$ and $z \cdot 2^{p'/2} \in \mathbb{Z}$. We prove that for each $h \in [H]$,

$$\left\| S_{i,0,1,h} - \hat{S}_{i,d,k,h} \right\|_\infty = O\left(\frac{1}{2^{p'}} \right).$$

Boundedness of intermediate representations is not an issue because

$$\log(Z_{i,d,k,h}) \leq O(\log(N) + \max_{i,i'} |Q(X)_i^\mathsf{T} K(X)_{i'}|) = \exp(O(p')),$$

and

$$\|S_{i,d,k,h}\|_\infty \leq \|V(X)\|_\infty \leq 2^{p'/2}.$$

It remains to show that that all intermediate representations are sufficiently close to their exact counterparts. We prove the following via an inductive argument for $d = D, D-1, \ldots, 0$:

$$\left|\log(Z_{i,d,k,h}) - \log(\hat{Z}_{i,d,k,h})\right| \leq \frac{(2b)^{D-d}}{2^{p/2}}, \tag{6.2}$$

$$\left\|S_{i,d,k,h} - \hat{S}_{i,d,k,h}\right\|_\infty \leq \frac{2^{p'/2}(8b)^{D-d}}{2^{p/2}}. \tag{6.3}$$

If (6.3) holds for $d = 0$, then the claim holds for sufficiently large $p = \Theta(p')$.

For the base case D, we verify (6.3) by

$$\left\|S_{i,D,k,h} - \hat{S}_{i,D,k,h}\right\|_\infty = \left\|V_{h,k}(X_k)\mathbb{1}\left\{\Lambda_{i,k} = 0\right\} - \hat{S}_{i,D,k,h}\right\|_\infty \leq \frac{1}{2^{p/2}},$$

due to the ability to access $V_{h,k}(X_k)$ and round it directly. We verify (6.2) due to the immediate access to and boundedness of $Q_{h,i}(X_i)^\mathsf{T} K_{h,k}(X_k)$:

$$|\log(Z_{i,d,k,h})| \leq \left|Q_{h,i}(X_i)^\mathsf{T} K_{h,k}(X_k)\right| \leq \|Q_{h,i}(X_i)\|_2 \|K_{h,k}(X_k)\|_2 \leq N \cdot 2^{p'/2}.$$

We prove the inductive step for $d-1$, assuming that the inductive hypothesis holds for

d. We first address $\hat{Z}_{i,d-1,k,h}$ by employing the Lipschitzness of the log-sum-exp function.

$$\begin{aligned}
&\left|\log(Z_{i,d-1,k,h}) - \log(\hat{Z}_{i,d-1,k,h})\right| \\
&\quad\leq \frac{1}{2^{p/2}} + \left|\log\left(\sum_{k'}\exp(\log(Z_{i,d,k',h}))\right) - \log\left(\sum_{k'}\exp(\log(\hat{Z}_{i,d,k',h}))\right)\right| \\
&\quad\leq \frac{1}{2^{p/2}} + \sum_{k'}\left|\log(Z_{i,d,k',h}) - \log(\hat{Z}_{i,d,k',h})\right| \\
&\quad\leq \frac{1}{2^{p/2}} + b\cdot\frac{(2b)^{D-d}}{2^{p/2}} \leq \frac{(2b)^{D-d+1}}{2^{p/2}}.
\end{aligned}$$

To obtain (6.3) for $d-1$, we first note that for sufficiently large p:

$$\begin{aligned}
\left|1 - \frac{\hat{Z}_{i,d,k',h}Z_{i,d-1,k',h}}{Z_{i,d,k,h}\hat{Z}_{i,d-1,k',h}}\right| &= \left|1 - \exp\left(\log\left(\frac{\hat{Z}_{i,d,k',h}}{Z_{i,d,k',h}}\right) + \log\left(\frac{Z_{i,d-1,k,h}}{\hat{Z}_{i,d-1,k,h}}\right)\right)\right| \\
&\leq 1 + 2\left(\left|\log\frac{\hat{Z}_{i,d,k',h}}{Z_{i,d,k',h}}\right| + \left|\log\frac{Z_{i,d-1,k,h}}{\hat{Z}_{i,d-1,k,h}}\right|\right) \\
&\leq \frac{4\cdot(2b)^{D-d+1}}{2^{p/2}}.
\end{aligned}$$

We conclude by using the fact that each $S_{i,d-1,k,h}$ is a convex combination of other $S_{i,d,k,h}$.

$$\begin{aligned}
\left\|S_{i,d-1,k,h} - \hat{S}_{i,d-1,k,h}\right\|_\infty &\leq \frac{1}{2^{p/2}} + \sum_{k'}\left\|\frac{Z_{i,d,k',h}}{Z_{i,d-1,k',h}}S_{i,d,k',h} - \frac{\hat{Z}_{i,d,k',h}}{\hat{Z}_{i,d-1,k',h}}\hat{S}_{i,d,k',h}\right\|_\infty \\
&\leq \frac{1}{2^{p/2}} + \sum_{k'}\frac{Z_{i,d,k',h}}{Z_{i,d-1,k',h}}\left\|S_{i,d,k',h} - \frac{\hat{Z}_{i,d,k',h}Z_{i,d-1,k',h}}{Z_{i,d,k,h}\hat{Z}_{i,d-1,k',h}}\hat{S}_{i,d,k',h}\right\|_\infty \\
&\leq \frac{1}{2^{p/2}} + \sum_{k'}\frac{Z_{i,d,k',h}}{Z_{i,d-1,k',h}}\left\|S_{i,d,k',h} - \hat{S}_{i,d,k',h}\right\|_\infty \\
&\quad + \sum_{k'}\frac{Z_{i,d,k',h}}{Z_{i,d-1,k',h}}\left\|\hat{S}_{i,d,k',h}\right\|_\infty\left|1 - \frac{\hat{Z}_{i,d,k',h}Z_{i,d-1,k',h}}{Z_{i,d,k,h}\hat{Z}_{i,d-1,k',h}}\right| \\
&\leq \frac{1}{2^{p/2}} + \frac{2^{p'/2}(8b)^{D-d}}{2^{p/2}} + 2^{p'/2}\sum_{k'}\frac{Z_{i,d,k',h}}{Z_{i,d-1,k',h}}\left|1 - \frac{\hat{Z}_{i,d,k',h}Z_{i,d-1,k',h}}{Z_{i,d,k,h}\hat{Z}_{i,d-1,k',h}}\right| \\
&\leq 2\cdot\frac{2^{p'/2}(8b)^{D-d}}{2^{p/2}} + 2^{p'/2}\cdot\frac{4\cdot(2b)^{D-d+1}}{2^{p/2}} \leq \frac{2^{p'/2}(8b)^{D-d+1}}{2^{p/2}}.
\end{aligned}$$

Owing to the fact that D and p' are constants and $b = N^{O(1)}$, a sufficiently large choice

of p guarantees that the implementation is valid. □

6.3.4.2 Proof of Corollary 6.9

Corollary 6.9. *Let $\epsilon \in (0,1)$ be any constant, and let $D \geq N^{\epsilon}$. Assume Conjecture 6.1, and suppose there exists $T \in \mathsf{Transformer}^{N}_{m,L,H}$ with $mH = O(D^{1-\epsilon})$ that decides connectivity of any input graph with connected components having diameter $\leq D$. Then $L = \Omega(\log D)$.*

We prove Corollary 6.9 by combining Theorem 6.17 and Conjecture 6.1.

Proof. Fix any $D \leq N$ with $D \geq N^{\xi}$ for some $\xi \in (0,1]$. Let C_1 denote a cycle graph on D vertices, and let C_2 denote the union of two cycle graphs each with $D/2$ vertices.

Suppose there is a transformer $T \in \mathsf{Transformer}^{N}_{m,L,H}$ with $mH = O(D^{1-\epsilon})$ that determines the connectivity of graphs with at most N edges and connected components with diameter at most D. We will show that it can be used to design an $\Theta(L)$-round MPC protocol π that distinguishes graphs C_1 and C_2 with $n = D$ edges.

Let π' be an MPC protocol that exactly computes the output of T using taking $R = O(L)$ rounds with local memory $s = O(D^{1-\epsilon/2})$ and $q = O(N^2)$ machines, which is guaranteed to exist by Theorem 6.17.

Let $n := 2\left\lfloor \frac{D}{4} \right\rfloor$ and $k := \left\lfloor \frac{N}{n} \right\rfloor$. We design π with the same local memory and machine count to determine the identity of input graph $G = (V, E) \in \{C_1, C_2\}$ provided as an arbitrary sequence of n edges. Let $u \in V$ be an arbitrary vertex in G.

Using a constant number of MPC rounds, π converts G into a graph $G' = (V', E')$ with $|E'| = kn + k \leq N$ and diameter $n + 2 \leq D$ such that G' is connected if and only if $G = C_1$. We do so by letting G' be composed of k copies $G^1, \ldots, G^k$ of G on separate vertices, along with k extra edges connecting the vertex corresponding to u in each G^j (say $u^j \in G^j$) to $u^1 \in G_1$. This ensures that the connectivity correspondence and edge count diameter bounds are met. Since G' can be produced by simply copying edges from G and adding an additional edge each time an edge containing u is copied, π can produce G' in $O(1)$ rounds.

Then, π simulates π' on G' and returns its output. Since G' is connected if and only if $G = C_1$, this protocol suffices to distinguish C_1 and C_2. Because the protocol uses $s = O(n^{1-\epsilon/2})$ local memory and $q = O(n^{2/\xi})$ machines, Conjecture 6.1 implies that π (and hence T) only exists if $L = \Omega(\log n) = \Omega(\log N)$. □

6.4 Transformers for k-hop induction heads

We complement the generality of Section 6.3 by studying, both empirically and theoretically, a specific toy sequential modeling task which will also serve (in Section 6.6) as a problem to separate the representational capabilities of transformers from that of other neural architectures.

This task, called the *k-hop induction heads* task, draws inspiration from the original *induction heads* task defined and analyzed on trained language models and in synthetic environments by Elhage et al. (2021) (see also Bietti et al., 2023). The standard induction heads task completes bigrams auto-regressively by predicting the token that follows the last previous occurrence of the final token in the sequence. For example, given the input $X =$ baebcabebdea, the standard induction heads task is to complete the final bigram by predicting b for the final token.

The k-hop induction heads tasks generalizes this mechanism by repeatedly using the completion of a bigram to determine the next bigram to complete. In the previous example, the 2-hop induction heads task is to predict c for the final token:

Definition 6.5. For any finite alphabet Σ, define the map $\mathrm{hop}_k \colon \Sigma^N \to (\Sigma \cup \{\perp\})^N$ by

$\text{hop}_k(X)_i = X_{\text{find}_X^k(i)}$ if $\text{find}_X^k(i) \neq 0$ and $\perp$ otherwise, where

$$\text{find}_X^1(i) = \max(\{0\} \cup \{j \in \mathbb{N} : j \leq i,\, X_{j-1} = X_i\});$$
$$\text{find}_X^k(i) = \text{find}_X^1(\text{find}_X^{k-1}(i)) \quad \text{for } k \geq 2.$$

The *k-hop induction heads task* is to compute, for each $i = 1, \ldots, N$, the value of $\text{hop}_k(X)_i$ from $(X_1, \ldots, X_i)$.

We note a similarity to the LEGO tasks of Zhang et al., 2023, who empirically study the ability of transformers to learn sequential operations on Abelian groups and observe the ability to perform more operations than the depth of the network.

6.4.1 Log-depth transformer for k-hop induction heads

Although hop_k appears to requires k steps to solve, we show that it is solved by a transformer of depth $O(\log k)$.

Theorem 6.18. *For any $k \in \mathbb{N}$ and alphabet Σ with $|\Sigma| \leq N$, there exists*

$$T \in \mathsf{MaskTransformer}_{m,L,H}^N$$

that computes $\text{hop}_k \colon \Sigma^N \to (\Sigma \cup \{\perp\})^N$ with $m = O(1)$, $L = \lfloor \log_2 k \rfloor + 2$, and $H = 1$.

In contrast to Corollary 6.5, this construction has constant embedding dimension and is achieved by a causally-masked transformer. As such, its proof in Section 6.4.3.1 depends on other techniques that exploit the simplicity of the problem and build on the induction heads construction of Bietti et al. (2023), rather than simply applying Theorem 6.3.

We give evidence for the optimality of this construction by proving a conditional lower bound using Theorem 6.8, as was done in Corollary 6.9.

Corollary 6.19. *Assuming Conjecture 6.1, for any constants $\xi \in (0, 1/2]$ and $\epsilon \in (0, 1)$,*

and any even $k = \Theta(N^{\xi})$*, every transformer* $T \in \mathsf{MaskTransformer}^{N}_{m,L,H}$ *with* $mH = O(k^{1-\epsilon})$ *that computes* hop_k *has depth* $L = \Omega(\log k)$*.*

6.4.2 Log-depth transformer learned from data

We empirically assess whether the representational trade-offs elucidated by tasks efficiently solved by parallelizable algorithms have implications for optimization and generalization properties of transformers. To that end, we trained auto-regressive transformer architectures of varying sizes to solve $\mathrm{hop}_k(X)$ for a variety of values of k in order to understand how changing depth impacted the performance of the learned models, the goal being to verify the sufficiency of logarithmic depth, just as in our theory.

In brief, we trained transformers with 500K to 5M parameters and depths $\{2, 3, 4, 5, 6\}$ with Adam to solve $\mathrm{hop}_k(X)$ for $k \in \{0, \ldots, 16\}$ with context length $|N| = 100$ and alphabet size $|\Sigma| = 4$. We trained the transformers in a multi-task setting, where a single model was trained to predict the sequence $\mathrm{hop}_k(X)$ auto-regressively when provided with X and k drawn at random. Further experimental details can be found in Section 6.5.1, and the experimental code is available at `https://github.com/chsanford/hop-induction-heads`.

We found that transformers are indeed capable of learning hop_k given sufficient training time, and that the largest learnable k grows exponentially with the depth. As can be seen in Figure 6.5, a six-layer neural network performs well on all $k \leq 16$, a five-layer on $k \leq 8$, a four-layer on $k \leq 4$, and so forth. We further explore these experimental results in Section 6.5.2 and observe a performance threshold appears to specifically lie at $\lfloor \log_2 k \rfloor + 2$ that coincides with Theorem 6.18. This logarithmic dependence of the depth on k persists in a larger-width regime, which is explored in Section 6.5.3. In the finite sample regime where neural networks are prone to overfit, our investigations in Section 6.5.5 note improved generalization in deeper models, which suggests that deeper models have a favorable inductive bias for tasks like hop_k.

Moreover, the learned models are surprisingly interpretable. We examined the activation patterns of attention matrices, and found close correspondences to useful intermediate prod-

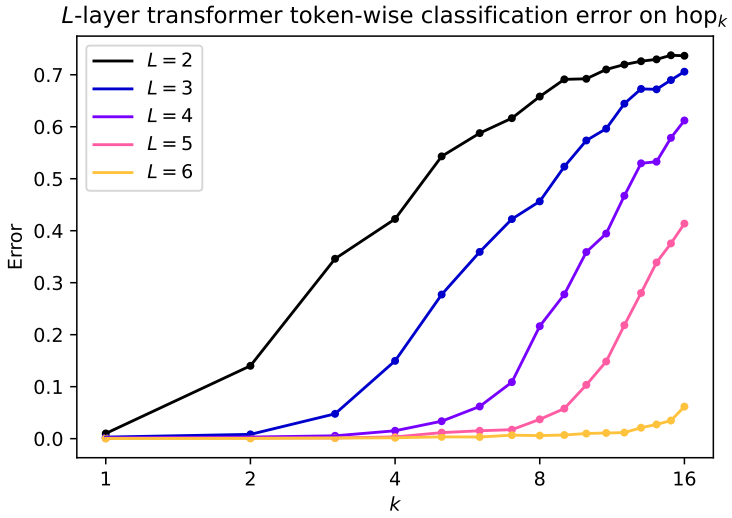


Figure 6.5: Evaluation of transformers of depths $L \in \{2, 3, 4, 5, 6\}$ trained on a mixture of hop_k for $k \in \{0, \ldots, 16\}$ evaluated on $n = 100$ samples of size $N = 100$ from each hop_k. Incrementing depth approximately doubles the largest k such that hop_k is learnable with small error.

ucts such as find_X^j. Taken together, these indicate that the learned models mechanistically resemble the construction employed in the proof of Theorem 6.18. See Section 6.5.4 for our investigation of model interpretability.

6.4.3 Proofs for Section 6.4.1

6.4.3.1 Proof of Theorem 6.18

Theorem 6.18. *For any $k \in \mathbb{N}$ and alphabet Σ with $|\Sigma| \leq N$, there exists*

$$T \in \mathsf{MaskTransformer}^N_{m,L,H}$$

that computes $\text{hop}_k \colon \Sigma^N \to (\Sigma \cup \{\perp\})^N$ with $m = O(1)$, $L = \lfloor \log_2 k \rfloor + 2$, and $H = 1$.

Proof. We design a masked transformer that implements hop_k in two phases. The first two layers compute $\text{find}_X^1(i)$ for each $i \in [N]$ using a similar approach to the induction heads construction of Bietti et al., 2023. The subsequent layers employ a doubling trick to compute each $\text{find}_X^{2^{\ell-2}}(i)$ after ℓ layers.

To do so we employ two technical lemmas (which are proved in Section 6.7.4) that describe the implementation of masked self-attention units that copy .

Lemma 6.20. *For some $m \geq d+2$, $\tau : [N] \times \mathbb{R}^m \to [N]$, and $\rho : \mathbb{R}^m \to \mathbb{R}^d$, there exists an attention head* $\text{lookUp}_{\tau,\rho} \in \mathsf{MaskAttn}_m^N$ *with precision $p = O(\log N)$ and $m \geq d+2$ satisfying* $\text{lookUp}_{\tau,\rho}(X)_{i,:d} = \rho(X_{\tau(i,X_i)})$.

Lemma 6.21. *For finite alphabet Σ, $m \geq d+2$, $\mu_1, \mu_2 : \mathbb{R}^m \to \Sigma$, and $\rho : \mathbb{R}^m \to \mathbb{R}^d$, there exists an attention head* $\text{lastOccurrence}_{\mu,\rho} \in \mathsf{MaskAttn}_m^N$ *with precision $p = O(\log(N\,|\Sigma|))$ such that,*

$$\text{lastOccurrence}(X)_{i,:d} = \begin{cases} \rho(\vec{0}) & \text{if } \forall\ i' < i : \mu_1(X_{i'}) \neq \mu_2(X_i), \\ \rho(X_{i'}) & \text{if } i' = \max\left\{i' < i : \mu_1(X_{i'}) = \mu_2(X_i)\right\}. \end{cases}$$

The first layer obtains the previous token X_{i-1} from each X_i. This is accomplished via the self-attention head $\text{lookUp}_{\tau,\rho}$ with $\tau(i, X_i) = i - 1$ and $\rho(X_i) = X_i$.

The second layer retrieves $(\text{find}_X^1(i), X_{\text{find}_X^1(i)})$ for each $i \in [N]$ by finding the most recent token whose *preceding* token is X_i. It does so by employing the $\text{lastOccurrence}_{\mu_1,\mu_2,\rho}$ primitive on the intermediate state $X_i^1 = (X_i, X_{i-1})$ with $\mu_1(X_i^1) = X_{i-1}$, $\mu_2(X_i^1) = X_i$, and $\rho(X_i^1) = (i, X_i)$.

- If $\text{find}_X^1(i) > 0$, then $\text{lastOccurrence}_{\mu_1,\mu_2,\rho}(X_i^1) = (\text{find}_X^1(i), X_{\text{find}_X^1(i)})$.
- Otherwise, it obtains $\vec{0}$ and performs no further passing, returning $\perp$ after all L layers.

If $k = 1$, the transformer returns $T(X)_i = X_{\text{find}_X^1(i)} = \text{hop}_k(X)_i$.

Otherwise, let $k := \sum_{j=0}^{\lfloor \log_2 k \rfloor} k_j 2^j$ for some $k_j \in \{0, 1\}$, and let $k_{:\ell} = \sum_{j=0}^{\ell} k_j 2^j$. Construct a transformer inductively to ensure that the ith output of the ℓth layer $X_i^\ell \in \mathbb{R}^m$ for $\ell \geq 2$ contains an encoding of

$$\left(X_i, \text{find}_X^{2^{\ell-2}}(i), X_{\text{find}_X^{2^{\ell-2}}(i)}, \text{find}_X^{k_{:\ell-2}}(i), X_{\text{find}_X^{k_{:\ell-2}}(i)}\right).$$

Note that the base case holds for $\ell = 2$, since $\text{find}_X^{k;0}(0) = \text{find}_X^1(0)$ if $k_0 = 0$ and is i otherwise.

For each $\ell = 1, \ldots, \lfloor \log_2 k \rfloor + 1$, we assume that the inductive hypothesis holds up to layer ℓ and prove that it also holds for layer $\ell+1$. To do so, we use a $\text{lookUp}_{\tau,\rho}$ self-attention head with $\tau(i, X_i^\ell) = \text{find}_X^{2^{\ell-2}}(i)$ and

$$\rho(X_i^\ell) = (\text{find}_X^{2^{\ell-2}}(i), X_{\text{find}_X^{2^{\ell-2}}(i)}, \text{find}_X^{k;\ell-2}(i), X_{\text{find}_X^{k;\ell-2}(i)}),$$

which ensures that $X_i^{\ell+1}$ can encode

$$\begin{aligned}
\text{find}_X^{2^{\ell-1}}(i) &= \text{find}_X^{2^{\ell-2}}(\text{find}_X^{2^{\ell-2}}(i)) \\
X_{\text{find}_X^{2^{\ell-1}}(i)} &= X_{\text{find}_X^{2^{\ell-2}}(\text{find}_X^{2^{\ell-2}}(i))} \\
\text{find}_X^{k;\ell-1}(i) &= \begin{cases} \text{find}_X^{k;\ell-2}(\text{find}_X^{2^{\ell-2}}(i)) & \text{if } k_{\ell-1} = 1 \\ \text{find}_X^{k;\ell-2}(i) & \text{if } k_{\ell-1} = 0 \end{cases} \\
X_{\text{find}_X^{k;\ell-1}(i)} &= \begin{cases} X_{\text{find}_X^{k;\ell-2}(\text{find}_X^{2^{\ell-2}}(i))} & \text{if } k_{\ell-1} = 1 \\ X_{\text{find}_X^{k;\ell-2}(i)} & \text{if } k_{\ell-1} = 0. \end{cases}
\end{aligned}$$

As a result, the output of layer $L = \lfloor \log_2 k \rfloor + 2$ contains an encoding of

$$X_{\text{find}_X^{k;L-2}(i)} = X_{\text{find}_X^k(i)} = \text{hop}_k(X)_i$$

for each $i \in [N]$. This is returned as the output of $T(X)$.

□

6.4.3.2 Proof of Corollary 6.19

Corollary 6.19. *Assuming Conjecture 6.1, for any constants $\xi \in (0, 1/2]$ and $\epsilon \in (0, 1)$, and any even $k = \Theta(N^\xi)$, every transformer $T \in \mathsf{MaskTransformer}_{m,L,H}^N$ with $mH = O(k^{1-\epsilon})$*

that computes hop_k *has depth* $L = \Omega(\log k)$.

Proof. The proof is analogous to that of Corollary 6.9. Let C_1 be a cycle on k vertices, and C_2 be the union of two cycles each on $k/2$ vertices. So both C_1 and C_2 have k edges. We show that the existence of $T \in \mathsf{Transformer}^N_{m,L,H}$ with $mH = O(k^{1-\epsilon})$ such that $T(X) = \mathrm{hop}_k(X)$ can be used to design an $\Theta(L)$-round MPC protocol π to solve the task.

As a result of Theorem 6.17, there exists an MPC protocol π' that exactly computes T with $R = \Theta(L)$ rounds with local memory $s = O(D^{1-\epsilon/2})$ and $q = O(N^2)$ machines. On input $G = (V, E) \in \{C_1, C_2\}$, we design a constant-round protocol that computes an sequence $X \in \Sigma^N$ such that $\mathrm{hop}_k(X)_N$ exactly determines the identity of G.

Since the k edges are passed to π in an unknown ordering with unknown labelings, we let $V = [k]$ and denote the edges as $e_1 = \{u_1, v_1\}, \ldots, e_k = \{u_k, v_k\}$. We define an operator next over the domain $\{(u, v), (v, u) : \{u, v\} \in E\}$ as follows: for $\{u, v\} \in E$, let $\mathrm{next}(u, v) := (v', u)$ where $v' \in V$ is the unique vertex $v' \neq v$ such that $\{u, v'\} \in E$. Notice that next is well-defined because all vertices in a cycle have degree 2. If $G = C_2$, then $\mathrm{next}^{k/2}(u_i, v_i) = (u_i, v_i)$ for any $i \in [k]$.

To set up our encoding of G as a sequence X, we first construct a gadget for each edge e_i that will be used to compute a single $\mathrm{next}(u_i, v_i)$. Under the alphabet $\Sigma = [k] \cup \{\dagger, \star, \_\}$, we define the nine-token sequence

$$\mathbf{e}_i = \ \star\ u_i\ \dagger\ v_i\ u_i\ \dagger\ v_i\ \star\ \_.$$

This gadget ensures that two hops will swap the values of u_i and v_i. That is

$$\mathrm{find}^2_{\mathbf{e}_i \circ u_i}(10) = \mathrm{find}^1_{\mathbf{e}_i \circ u_i}(6) = 4, \qquad X_{\mathrm{find}^2_{\mathbf{e}_i \circ u_i}(10)} = v_i,$$
$$\mathrm{find}^2_{\mathbf{e}_i \circ v_i}(10) = \mathrm{find}^1_{\mathbf{e}_i \circ v_i}(8) = 2, \qquad X_{\mathrm{find}^2_{\mathbf{e}_i \circ v_i}(10)} = u_i.$$

Likewise, concatenating sequences corresponding to overlapping edges facilitates multiple

hops. For example, if $e_1 = (1,2), e_2 = (3,4), e_3 = (2,3)$, then

$$\begin{aligned} &\mathrm{find}^2_{\mathbf{e}_1\circ\mathbf{e}_2\circ\mathbf{e}_3\circ 2}(28) = 22, && X_{\mathrm{find}^2_{\mathbf{e}_1\circ\mathbf{e}_2\circ\mathbf{e}_3\circ 2}(28)} = 3, \\ &\mathrm{find}^4_{\mathbf{e}_1\circ\mathbf{e}_2\circ\mathbf{e}_3\circ 2}(28) = 13, && X_{\mathrm{find}^4_{\mathbf{e}_1\circ\mathbf{e}_2\circ\mathbf{e}_3\circ 2}(28)} = 4, \\ &\mathrm{find}^4_{\mathbf{e}_1\circ\mathbf{e}_2\circ\mathbf{e}_3\circ 3}(28) = 2, && X_{\mathrm{find}^4_{\mathbf{e}_1\circ\mathbf{e}_2\circ\mathbf{e}_3\circ 3}(28)} = 1. \end{aligned}$$

Let

$$\mathbf{E} := (\mathbf{e}_1 \circ \mathbf{e}_2 \circ \cdots \circ \mathbf{e}_k)^{k/2} \circ 1$$

be a length $N_k := 9k \cdot \frac{k}{2} + 1$ sequence and let $X = (\_)^{N-N_k} \circ \mathbf{E}$. We show that $\mathrm{hop}_k(X)_N = \mathrm{hop}_k(\mathbf{E})_{N_k} = 1$ if and only if $G = C_2$.

Without loss of generality, let $\{j, j+1\} = e_{i_j} \in E$ for all $j \in [\frac{k}{2} - 1]$. Let $e_{i_0} = \{1, v^*\}$, where $v^* = \frac{k}{2}$ if $G = C_2$ and $v^* = k$ if $G = C_1$. Assume without loss of generality that $i_1 > i_0$. We argue inductively that for any $j \in [\frac{k}{2}]$:

1. Every two hops simulates a single step of next:

$$\mathrm{hop}_{2j}(\mathbf{E})_{N_k} = \mathrm{next}^j(1, v^*)_1 = \begin{cases} j & \text{if } j+1 < \frac{k}{2} \text{ or } G = C_1, \\ 1 & \text{if } j = \frac{k}{2},\ G = C_2; \end{cases}$$

2. Every two hops never "jumps" by more than one repetition of all edges gadgets:

$$\mathrm{find}^{2j}_{\mathbf{E}}(N_k) \geq \mathrm{find}^{2j-2}_{\mathbf{E}}(N_k) - 9(k-1);$$

3. The executed gadget corresponds to the correct edge and the gadget is executed correctly:

$$\mathrm{find}^{2j}_{\mathbf{E}}(N_k) \in \{9kj' + 9i_j + \iota : j' \in \mathbb{N}, \iota \in \{2, 4\}\}.$$

If all three conditions are met, then $\mathrm{hop}_k(X)_N = 1$ if and only if $G = C_1$ from condition

1.

We first show that the base case holds for $j = 1$. Since $i_1 > i_0$, the second-last time 1 appears in the $\mathbf{E}$ is in the final encoding $\mathbf{e}_{i_1}$. By the two-case analysis of the $\mathbf{e}_{i_1}$ gadget, we validate that $\text{hop}_2(\mathbf{E})_{N_k} = 2$ and conditions (1) and (3) hold. Since $\mathbf{e}_{i_1}$ cannot be the first edge encoding appearing in $\mathbf{e}_1 \circ \mathbf{e}_2 \circ \cdots \circ \mathbf{e}_k$, owing to it following $\mathbf{e}_{i_0}$), condition (2) is satisfied.

Suppose that the inductive hypotheses holds up to $j < \frac{k}{2}$. Then, we argue that it holds for $j + 1$. Since $\text{hop}_{2j}(\mathbf{E})_{N_k} = j + 1$ (from condition (1)) and $\text{find}_{\mathbf{E}}^{2j}(N_k)$ resides at the left-most side of the gadget for $\mathbf{e}_{i_j}$ (from condition (3)), the two subsequent $\text{find}_{\mathbf{E}}$ iterations must occur in the gadget $\mathbf{e}_{i_{j+1}}$. Because $\text{find}_{\mathbf{E}}^{2j}(N_k) \geq 9k(k-j)$ (from condition (2)), all edges appear in the k gadgets to the left of $\text{find}_{\mathbf{E}}^{2j}(N_k)$, and all other edges (including $\mathbf{e}_{i_{j+1}}$) must occur before the next occurrence of $\mathbf{e}_{i_j}$. Thus, the two hops occur in the $\mathbf{e}_{i_{j+1}}$ gadget (within distance $9(k-1)$) and results in a properly positioned $\text{find}_{\mathbf{E}}^{2j+2}(N_k)$ with $\text{hop}_{2j+2}(\mathbf{E})_{N_k} = \text{next}^{j+1}(1, v^*)_1$.

Since an MPC protocol can convert G to X using a constant number of layers, and because π' outputs $T(X)_N = 1$ if and only if $G = C_1$, we can construct a protocol of π by simulating π'. Because the protocol π uses $s = O(k^{1-\epsilon/2})$ local memory and $q = O(k^{2/\xi})$ machines, Conjecture 6.1 implies that the existence of T requires $L = \Omega(\log k)$. □

6.5 Detailed empirical analysis of k-hop induction heads

This section presents in-depth explanations of the empirical results of Section 6.4.2, along with further experiments. Taken together, these results suggest that the relationship between the number of hops k and the depth L of transformers trained on the task is well-characterized by the representational thresholds of Theorem 6.18 and Corollary 6.19; that the construction described in the proof of Theorem 6.18 is attainable by trained models; and deep models likely exhibit an inductive bias that favors compositional learning rules in the finite sample regime.

We define our experimental methodology precisely in Section 6.5.1 and provide supporting evidence for our claims in the subsequent sections.

Exponential powers of depth. Our principal empirical claim is that incrementing the depth L of a transformer exponentially increases the model's capabilities to learn k-hop induction heads tasks. We explore this claim primarily in Section 6.5.2, where we compare this empirical claim with the relevant theoretical results (Theorem 6.18 and Corollary 6.19), which suggest a similar dependence. We further study the impacts of increasing the embedding dimension m of the transformer in Section 6.5.3 and find that doubling the width is roughly equivalent in performance to incrementing the depth by one.

Empirical Claim 6.22. *A transformer* $T \in \mathsf{MaskTransformer}^N_{m,L,H}$ *trained with Adam to solve* hop_k *has small token-wise classification error if* $L \log(m) = \Omega(\log k)$ *and large error if* $L \log m = O(\log k)$.

Mechanistic alignment with theoretical construction. We further demonstrate the empirical salience of our theoretical construction by conducting a study of the interpretability of learned transformers in Section 6.5.4. This investigation reveals that the attention matrices of sufficiently deep transformers exhibit an implementation of a circuit that relies on the same "doubling" principle of the construction in the proof of Theorem 6.18. The resulting circuit is comprised of the same intermediate products that are used in that hop_k construction.

Empirical Claim 6.23. *The outputs of individual attention matrices of a transformer* $T \in \mathsf{MaskTransformer}^N_{m,L,H}$ *trained with Adam to solve* hop_k *with* $L = \Omega(\log k)$ *and evaluated on input* $X \in \Sigma^N$ *(i) correspond to the* find^j_X *intermediate products of the Theorem 6.18 construction and (ii) demonstrate a "doubling" phenomenon where the each head layer* ℓ *corresponds to* find^j_X *for some* $j = O(2^\ell)$.

Beneficial inductive biases of depth. While most of our experiments belong to the "infinite-sample" regime where new samples are randomly generated on each training step,

we also evaluate our models in two finite-sample regimes in Section 6.5.5. We find that a small number of samples is sufficient to approach the performance of the infinite-sample regime. When the amount of training data is small, we find that deeper models perform better than shallower models, possibly due to an inductive bias that favors compositional hypotheses.

Empirical Claim 6.24. hop_k *can be learned in a sample-efficient manner by transformers* $T \in \mathsf{MaskTransformer}^N_{m,L,H}$ *trained with Adam with* $L = \Omega(\log k)$*. If* T *overfits to* hop_k *tasks for some* k*, then increasing the depth* L *while holding* k *fixed leads superior performance.*

The experiments detailed here were conducted under limited computational resources. The authors are interested in future work that would evaluate whether these scaling rules persist on larger architectures and more complex tasks.

6.5.1 Experimental details

Task details. We study a multi-task variant of k-hop induction heads that predicts

$$\mathrm{hop}_k(X) = (0, \mathrm{hop}_k(X'))$$

from input $X = (k, X')$ for $k \in \{0, 1, \ldots, k_{\max}\}$[5] and $X' \in \Sigma^{N-1}$. We refer to this task as *multi-hop* and provide the task hyperparameters in Table 6.1.

Hyperparameter	Value
Context length N	100
Alphabet size $\lvert\Sigma\rvert$	4
Max hops $k_{\max}$	16

Table 6.1: Multi-hop task hyperparameters

We define the distribution $\mathcal{D}_{\mathrm{multi-hop}}$ over labeled samples for the multi-hop task and $\mathcal{D}_{\mathcal{X}}$ over input sequences $X \in \Sigma^{N-1}$. We draw a labeled sample $(X, \mathrm{hop}_k(X)) \sim \mathcal{D}_{\mathrm{multi-hop}}$

[5]The task hop_0 is simply the identity mapping: $\mathrm{hop}_0(X') = X'$.

by independently sampling $k \sim \text{Unif}(\{0, 1, \ldots, k_{\max}\})$ and $X' \sim \mathcal{D}_{\mathcal{X}}$. Input sequences $X' \sim \mathcal{D}_{\mathcal{X}}$ are drawn uniformly from inputs *with no repeating elements*. That is, we sample $X'_1 \sim \text{Unif}(\Sigma)$ and each $X'_{j+1} \sim \text{Unif}(\Sigma \setminus \left\{X'_j\right\})$. For each $k \in [k_{\max}]$, let $\mathcal{D}_{\text{hop}_k}$ denote the conditional distribution $((k', X'), (0, \text{hop}_{k'}(X'))) \sim \mathcal{D}_{\text{multi-hop}} \mid (k = k')$. Also, let $\text{dom}(\text{hop}_k) = \{(k, X') : \Pr[X' \sim \mathcal{D}_{\mathcal{X}}] > 0\}$.

For $\overline{\Sigma} := \Sigma \cup [k_{\max}]$, we define the n-sample *empirical token-wise classification error* of a transformer $T : \overline{\Sigma}^N \to \overline{\Sigma}^N$ on a task hop_k as

$$\texttt{err}_k^n(T) = \frac{1}{n} \sum_{\iota=1}^{n} \frac{1}{|\{i : \text{hop}_k(X^\iota)_i \neq \perp\}|} \sum_{i=1}^{N} \mathbb{1}\{T(X^\iota)_i \neq \text{hop}_k(X^\iota)_i \neq \perp\},$$

for iid samples $(X^1, \text{hop}_k(X^1)), \ldots, (X^n, \text{hop}_k(X^n)) \sim \mathcal{D}_{\text{hop}_k}$. We ignore null $\perp$ outputs of hop_k when no k-hop induction head exists in order to avoid inadvertently over-estimating the performance of transformers on large k tasks, which have a large fraction of null outputs.

Training details. We trained a variety of causally-masked GPT-2 transformers (Radford et al., 2019) from HuggingFace to solve the multi-hop task. The model has an absolute positional encoding.

The transformers are trained with Adam (Kingma and Ba, 2014) on the cross-entropy loss. In the infinite-sample regime, we draw 32 new iid samples from $\mathcal{D}_{\text{multi-hop}}$ on each training step. Otherwise, n_{train} samples are drawn before training commences and all samples are rotated through batches, before repeating. We use the hyperparameters in Table 6.2 to train all of the models identified in Table 6.3.

Computational resources. All experiments were run on a 2021 Macbook Pro with an M1 chip.

Hyperparameter	Value
Embedding dimension m	$\{128, 256\}$
Depth L	$\{2, 3, 4, 5, 6\}$
Number of heads H	$\{4, 8\}$
Vocabulary size	30
Activation function	GeLU
Layer norm ϵ	10^{-5}
Training samples n_{train}	$\{10^3, 3 \cdot 10^3, \infty\}$
Learning rate	10^{-4}
Training steps	10^5
Batch size	32

Table 6.2: Model and training hyperparameters

Identifier	Heads H	Embed. dim. m	Depth L	Train samples n_{train}	# parameters
$T^{\infty}_{4,2}$	4	128	2	∞	413,440
$T^{\infty}_{4,3}$	4	128	3	∞	611,712
$T^{\infty}_{4,4}$	4	128	4	∞	809,984
$T^{\infty}_{4,5}$	4	128	5	∞	1,008,256
$T^{\infty}_{4,6}$	4	128	6	∞	1,206,528
$T^{\infty}_{8,2}$	8	256	2	∞	1,613,312
$T^{\infty}_{8,3}$	8	256	3	∞	2,403,072
$T^{\infty}_{8,4}$	8	256	4	∞	3,192,832
$T^{\infty}_{8,5}$	8	256	5	∞	3,982,592
$T^{\infty}_{8,6}$	8	256	6	∞	4,772,352
$T^{3000}_{4,2}$	4	128	2	3000	413,440
$T^{3000}_{4,3}$	4	128	3	3000	611,712
$T^{3000}_{4,4}$	4	128	4	3000	809,984
$T^{3000}_{4,5}$	4	128	5	3000	1,008,256
$T^{3000}_{4,6}$	4	128	6	3000	1,206,528
$T^{1000}_{4,2}$	4	128	2	1000	413,440
$T^{1000}_{4,3}$	4	128	3	1000	611,712
$T^{1000}_{4,4}$	4	128	4	1000	809,984
$T^{1000}_{4,5}$	4	128	5	1000	1,008,256
$T^{1000}_{4,6}$	4	128	6	1000	1,206,528

Table 6.3: Hyperparameters of all $\mathsf{MaskTransformer}^{N}_{m,L,H}$ trained for the empirical analysis.

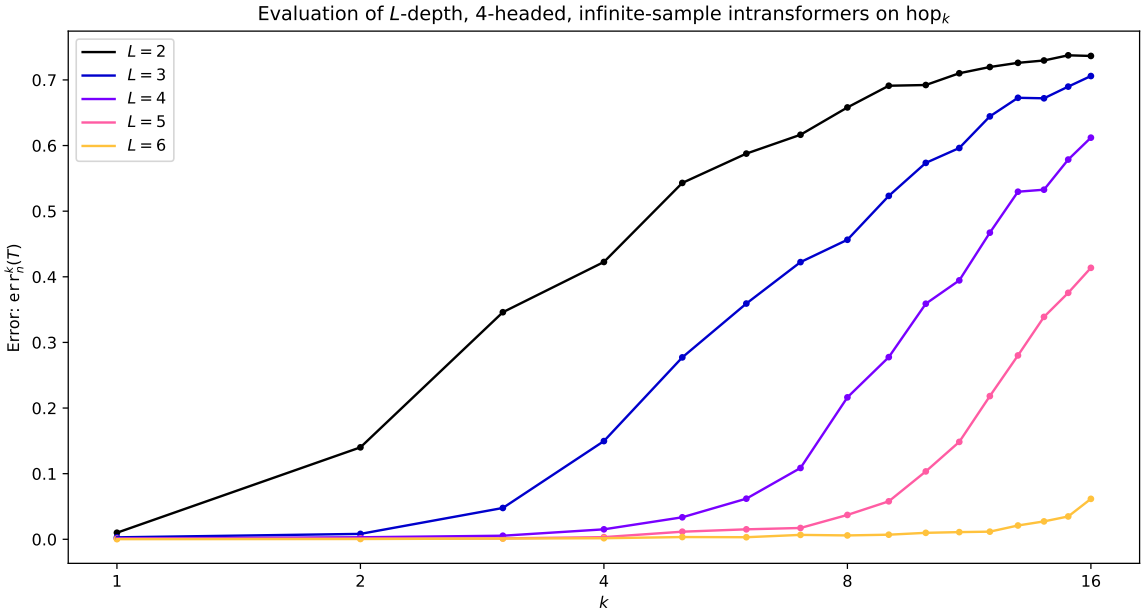


Figure 6.6: Zoomed in version of Figure 6.5. Evaluation of transformers $\mathtt{err}_k^n(T_{4,L}^\infty)$ with depths $L \in \{2, 3, 4, 5, 6\}$, heads $H = 4$, and embedding dimension $m = 128$ trained on the multi-hop task. This figure plots $\mathtt{err}_k^n(T_{4,L}^\infty)$ on $n = 100$ samples as a function of k for each choice of L.

6.5.2 Exponential increases in k-hop capacity with depth (Empirical Claim 6.22; Figures 6.6 to 6.8)

We visualize the relationship between the depth L of a transformer and the largest k such that $\mathtt{err}_k^n(T)$ is small in Figure 6.6, Figure 6.7, and Figure 6.8. We exhibit the relationship in its simplest form by considering transformers with heads $H = 4$, embedding dimension $m = 128$, and new training samples on every epoch. The figures provide alternate views of $\mathtt{err}_k^n(T_{4,L}^\infty)$ for each $L \in \{2, 3, 4, 5, 6\}$ with $n = 100$ samples for each $k \in [k_{\max}]$.

Together, these plots illustrate a sharp phase transition when $D = \lfloor \log_2 k \rfloor + 2$, which identically matches the depth scaling in Theorem 6.18. Increasing the depth of a transformer by one approximately doubles the number of values $k \in [k_{\max}]$ with bounded error. For instance, following the theoretical and empirical intuition of Bietti et al., 2023, the depth $L = 2$ transformer $T_{4,2}^\infty$ succeeds in solving the standard induction heads task, but attains at least 10% error on all other tasks. Likewise, a depth $L = 3$ model has error bounded by 1%

for $k \in \{1, 2\}$, which increases rapidly for larger values of k.

This doubling phenomenon suggests that simple compositional tasks with a larger number of compositions than the depth of the model are easily learnable if the model can employ a doubling trick, similar to the one used in the proof of Theorem 6.18. This relationship between compositionality and depth reflects the results of Zhang et al. (2023), where the learnable task complexity also scales super-linearly in depth.

Given the lower bounds of Corollary 6.19, one may ask why models with depth $L < \lfloor \log_2 k \rfloor$ achieve non-trivial success on hop_k tasks that cannot be represented in a compositional manner. There are several relevant explanations:

1. In these experiments, the embedding dimension $m = 128$ is actually larger than the context $N = 100$, which may enable the model to memorize more of its preceding samples and offload logical work to the MLP, rather than executing a pointer-doubling strategy. While practical models regularly have the opposite (and our theoretical results are oriented around that parametric scaling), we used a larger m than is necessary for representational purpose to improve the optimization landscape and speed convergence.

2. This is made further plausible by the small alphabet size $|\Sigma|$ and randomly drawn sequences X', which place effective bounds on how much look-back from each token i is necessary to compute $\mathrm{hop}_k(X)_i$.

Nonetheless, these results provide strong support that models are substantially easier to train to low classification error in the regime where the depth is sufficient to implement a pointer-doubling construction. In the following subsection, we further investigate this phenomenon by examining the intermediate attention matrices produced by trained models.

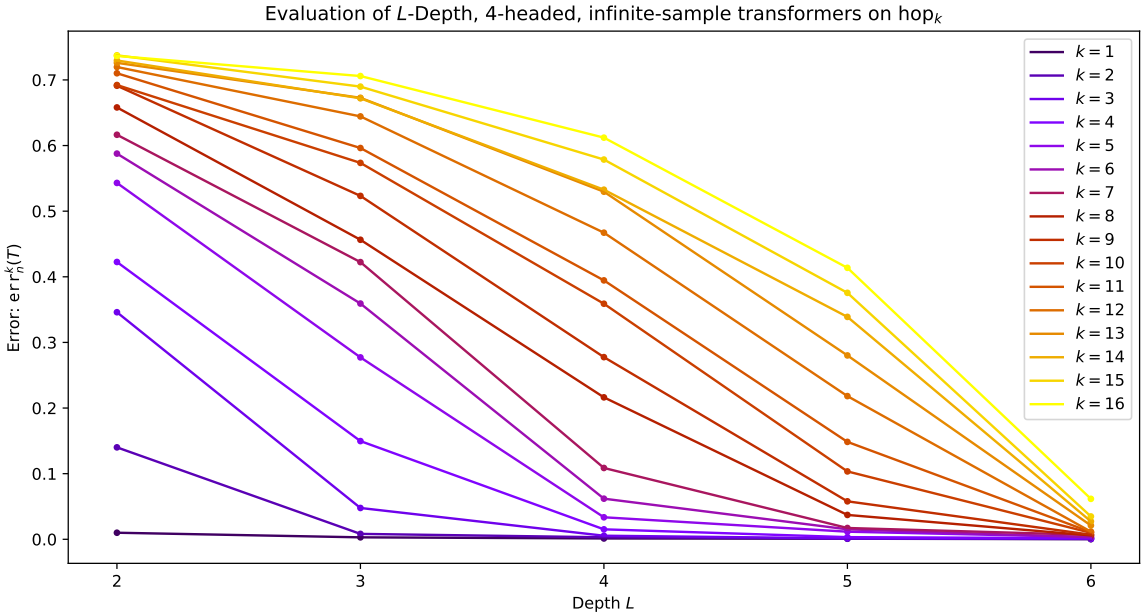


Figure 6.7: Alternate view of Figure 6.6 including $\mathtt{err}_k^n(T_{4,L}^\infty)$ plotted as a function of L for each k.

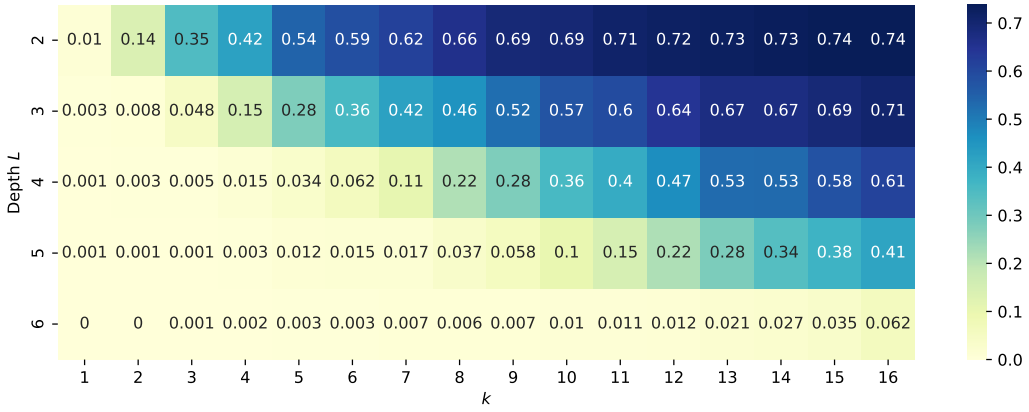


Figure 6.8: Alternate views of Figure 6.6 including $\mathtt{err}_k^n(T_{4,L}^\infty)$ as a table with one cell for each (L, k) pair.

6.5.3 Width variation (Empirical Claim 6.22; Figure 6.9)

While the primary focus of these empirical results and the paper as a whole is on the role of depth in the ability of transformer to learn parallelizable and compositional tasks, we also aim to understand the interplay of depth and width in learning the multi-hop task. Here, we contrast the previous transformers $T^{\infty}_{4,L}$ with models $T^{\infty}_{8,L}$ that have more heads ($H = 8$) and larger embedding dimensions ($m = 256$). We plot the classification errors of all 10 architectures over 16 hop_k sub-tasks in Figure 6.9.

Here, we observe a rough correspondence in performance between the transformers $T^{\infty}_{H,L}$ and $T_{2H,L-1}$ and the same doubling phenomenon as is evident models with $H = 4$ heads. That is, while increasing the width improves the classification error of learned models, it does so in a far less parameter-efficient manner than incrementing the depth. As mentioned before, the relative success of wide and shallow transformers is likely contingent on the relatively short context length N and alphabet size $|\Sigma|$. However, these results still suggest an important role for wider models to play beyond representational capabilities of transformers.

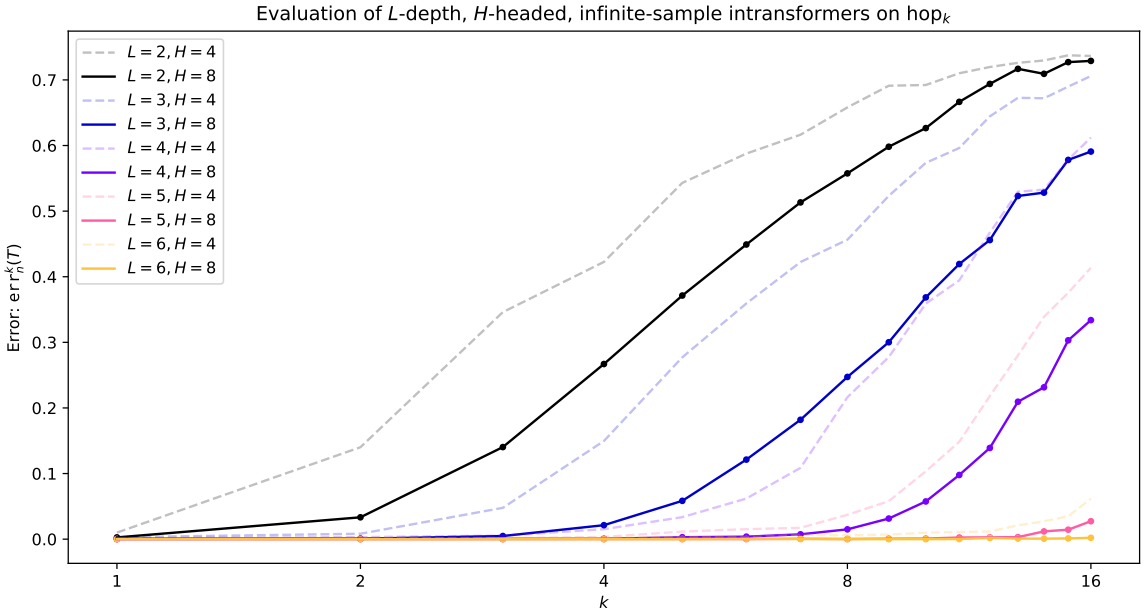


Figure 6.9: Comparison between the errors $\texttt{err}_k^n(T_{H,L}^\infty)$ of transformers with embedding dimension and heads $(m, H) = (4, 128)$ (dashed line, same plots as Figure 6.6) and $(m, H) = (8, 256)$ (solid line) trained on the multi-hop task, evaluated on $n = 100$ samples per hop_k task.

6.5.4 Mechanistic alignment with construction (Empirical Claim 6.23, Figures 6.10 to 6.15)

We use standard attention-based interpretability techniques to better understand what particular logical circuits are implemented by transformers trained to solve the multi-hop task. By qualitatively inspecting the attention matrices produced by trained models and by measuring the alignment between those inner products and partial solutions find^j of hop_k, we uncover a striking correspondence between the behaviors of the trained models and the transformer construction designed in the proof of Theorem 6.18. We further observe that trained transformers with high accuracy have "decisive" self-attention units with particularly strong correlations to some find^j intermediate, while poorly performing models have less predictable attention activations.

For a fixed trained model $T \in \mathsf{Transformer}^N_{m,L,H}$, we let $A^{\ell,h}[T](X)$ represent the output of the hth self-self attention matrix in the ℓth layer for $h \in [H]$ and $\ell \in [L]$, evaluated at some input $X \in \text{dom}(\text{hop}_k)$. That is, we let

$$A^{\ell,h}[T](X) = \text{softmax}\left(Q^{\ell,h}(X^{\ell-1})K^{\ell,h}(X^{\ell-1})^{\mathsf{T}} + \Gamma\right) \in \mathbb{R}^{N\times N},$$

where $X^{\ell-1}$ is the intermediate state representing the output of layer $\ell - 1$ of T on input X and Γ is the causal masking matrix. Each row i in the matrix represents the coefficients of the convex combination of value vectors affiliated with each query, which can be used as a signifier of which embeddings i receives information from.

Visualization of find^j alignment for hop_{16} and depth $L = 6$ (Figure 6.10). The outputs of self-attention matrices are often highly structured matrices that reveal which relationships between tokens are encoded and how information is shared within the model (Li and McClelland, 2022; Clark et al., 2019; Rogers, Kovaleva, and Rumshisky, 2020). We plot several self-attention matrices associated with a depth $L = 6$, heads $H = 4$ transformer trained in the infinite-sample regime and evaluated on a single sample $X \in \text{dom}(\text{hop}_{16})$ in

Figure 6.10.

By looking at the six self-attention matrices, one can infer that all heads are "decisive" and obtain nearly all of their relevant information from a single value embedding, rather than averages of a large number of embeddings. The top-left self-attention matrix, which belongs to the first self-attention head, clearly associates elements with their predecessors, which is identical the to the function of our lookUp attention head in the first layer of the hop_k construction of Theorem 6.18.

While the roles of the other heads are not immediately obvious, they can be understood by overlaying colored matrices with non-zero cells at $(i, \text{find}_X^j(i))$ for some $j \leq k$. For instance, the top-right attention matrix in layer $\ell = 2$ corresponds almost exactly with find_X^1 (as suggested by the second-layer of our construction), and the others are closely associated with find_X^1, find_X^2, find_X^3, and find_X^8 for layers $\ell = 3, 4, 5, 6$ respectively. This is a remarkably close correspondence to our construction, which includes a self-attention matrix in the ℓth layer whose activations correspond to $\text{find}_X^{2^{\ell-2}}$.

While not conclusive, this experiment suggests a strong alignment between the behaviors of this particular transformer and our theoretical construction. This suggests a high likelihood that the transformer successfully learns to solve hop_{16} by employing a pointer-doubling primitive. However, these results apply to only a single model, a single task, and a single input; in the subsequent section, we generalize this interpretability analysis.

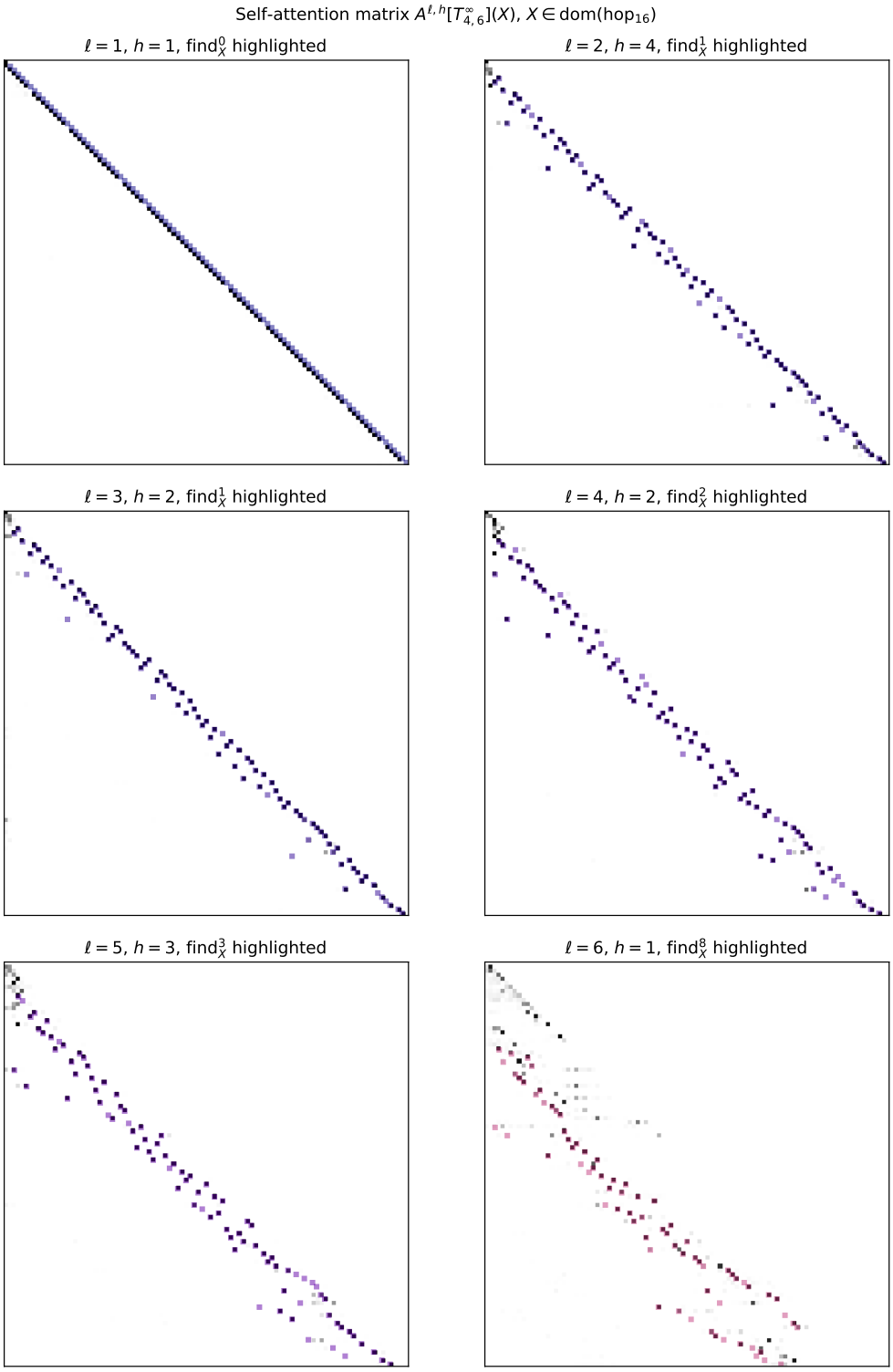


Figure 6.10: The outputs of several internal self-attention matrices $A^{\ell,h}[T^{\infty}_{4,6}](X) \in \mathbb{R}^{100\times 100}$ of a trained multi-task transformer of depth $D = 6$ evaluated on a single sample $X \sim \mathcal{D}_{\mathrm{hop}_{16}}$ are plotted in grayscale. In each cell, the matrix with non-zero entries $(\mathrm{find}^j_X(i), i)_{i\in[N]}$ for some j is included in transparent color to visualize the function of each self-attention unit.

Alignment between attention heads and find^j **for a single** hop_k **sub-task (Figures 6.11 to 6.13).** To broaden and quantify the analysis of the previous section, we measure the extent to which each self-attention head mimics the functionality of find^j, which are partial computations of hop_k that are employed in the proof of Theorem 6.18. We use cell-wise matrix inner products to quantify the strength of correlation between a self-attention matrix and a fixed function potentially relevant to interpretability.

For two matrices $A, B \in \mathbb{R}^{N \times N}$, let

$$\langle A, B \rangle = \frac{\|A \odot B\|_F^2}{\|A\|_F \|B\|_F}$$

be their normalized element-wise inner-product, where $\|\cdot\|_F$ is the Frobenius norm and $\odot$ denotes element-wise multiplication. For some function $g : [N] \to \{0\} \cup [N]$, we let $\langle g, B \rangle := \langle A^g, B \rangle$, where

$$A^g_{i,j} = \begin{cases} 1 & \text{if } g(j) = i, \\ 0 & \text{otherwise.} \end{cases}$$

We use this notation to analyze experimentally how closely the self-attention matrices $A^{\ell,h}$ encode the intermediate products of the proof of Theorem 6.18, find^j_X. For n iid samples $X^1, \ldots, X^n \in\sim \mathcal{D}_{\text{hop}_k}$, let

$$\left\langle A^{\ell,h}, \text{find}^j \right\rangle_{n,k} := \frac{1}{n} \sum_{\iota=1}^{n} \left\langle \text{find}^j_{X^\iota}, A^{\ell,h}(X^\iota) \right\rangle.$$

Due to the non-negativity of $A^{\ell,h}$ and find^j, $\left\langle A^{\ell,h}, \text{find}^j \right\rangle_{n,k} \in [0,1]$, and $\left\langle A^{\ell,h}, \text{find}^j \right\rangle_{n,k} = 1$ only if $\forall \iota \in [n]$:

$$A^{\ell,h}(X^\iota)_{i,i'} = 1 \iff \text{find}^j_{X^\iota}(i) = i'.$$

These inner products make it possible to visualize the strength of correlations of all heads in a particular model $T \in \mathsf{MaskTransformer}^N_{m,L,H}$ with all target functions find^j on a collection of random samples drawn from some $\mathcal{D}_{\text{hop}_k}$. Figure 6.11 visualizes the functionality of all

attention units in the 4-layer, 4-head transformer $T^{\infty}_{4,4}$ when evaluated on the sub-task hop_4. The figure gives several clues about how hop_4 is successfully computed by the trained model: the second layer and third layer both utilize find^1 to determined find^2 jointly by the end of the third layer. The fourth layer uses the ability to create a stable find^2 construction to obtain find^4 and hence hop_4.

This plot also indicates the relative stability of this circuit interpretation of the procedure: a large number of heads are very strongly correlated with find^1 or find^2 across the 10 samples, which indicates they are likely utilized consistently to compute those intermediates regardless of input.

Figure 6.12 is a similar plot for the transformer $T^{\infty}_{4,6}$ with depth $L = 6$, evaluated on the task hop_{16}. The functionalities of the heads visualized in Figure 6.10 can be observed in the corresponding inner products. The collection of all inner products presents further evidence that the pointer-doubling phenomenon occurs in the trained models, due to the increase in compositions present in the largest inner products of deeper attention units.

While Figures 6.11 and 6.12 showcase the decisive alignment between self-attention heads and particular partial computations find^j in successfully trained models, Figure 6.13 demonstrates the loss of that decisiveness in poorly performing transformers. There, we visualize the alignments of the trained depth-4 transformer $T^{\infty}_{4,4}$ evaluated on hop_{16}, in which it attains a 61% token error. While a self-attention units in the second layer coincides with find^1, no strong correlations emerge deeper in the model. Unlike the other figures, the deeper self-attention units are "indecisive," lacking any large inner products and failing in particular to correlate with any highly compositional targets. This provides a visual explanation of the transformer's failure, since it lacked the effective representational capacity needed to learn a circuit with consistent and highly-compositional outputs.[6]

[6]Since these experiments are in the small alphabet size $|\Sigma| = 4$ regime, this task performs better than random guessing due to inferential capabilities that are are powered by the high embedding dimension and do not require implementing a pointer-chasing algorithm. We suspect that the "checkerboard" patterns are powered by this inference.

Alignment between attention heads and find^j for all hop_k sub-tasks (Figures 6.14 and 6.15). For an even more global lens on the mechanistic interpretability of these trained models, we visualize how the maximum inner products of each self-attention unit change for a fixed transformer for different sub-tasks hop_k. Figures 6.14 and 6.15 do so for the depth-4 and depth-6 networks respectively. The hue of each cell (and its numerical label) corresponds to the j^* with the most correlated inner product with corresponding attention unit $A^{\ell,h}$ in samples from $\text{dom}(\text{hop}_k)$, and the opacity corresponds to the magnitude of that inner product.

The takeaways of the previous inner product figures are apparent in these: the approximate doubling for the depth $L = 6$ transformer can be visualized by the vertically changing opaque colors. Conversely, a separation can be observed between the tasks where the depth $L = 4$ transformer performs well and has "decisive" self-attention units deeper in the network and those where it does not.

Moreover, the figures (especially Figure 6.15) demonstrate that several self-attention units have a consistent function among samples from the same task, while adapting in function to different hop_k tasks. This is most apparent in head $h = 4$ of layer $\ell = 6$, where the self-attention head functions as $\text{find}^1, \text{find}^3, \text{find}^5$ or find^7 depending on the complexity of the task.

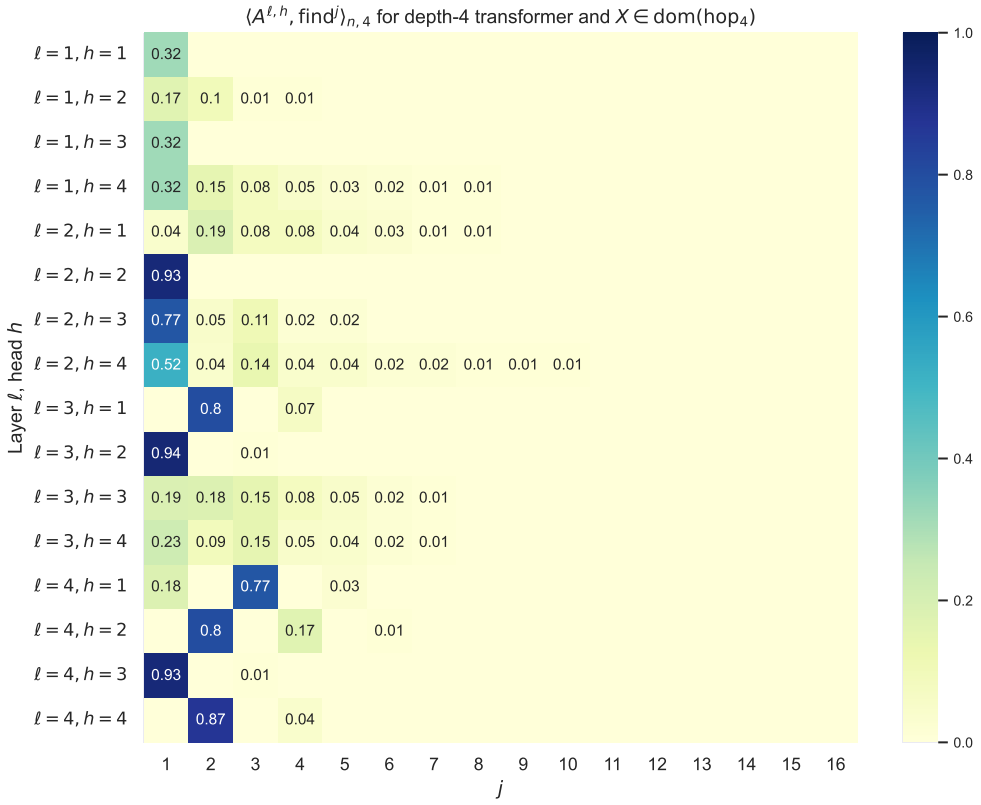


Figure 6.11: Plots of all inner products $\left\langle A^{\ell,h}[T^{\infty}_{4,4}], \text{find}^j \right\rangle_{10,4}$ for $n = 10$ samples $X^1, \ldots, X^{10} \in \text{dom}(\text{hop}_4)$ for the 4-layer transformer $T^{\infty}_{4,4}$.

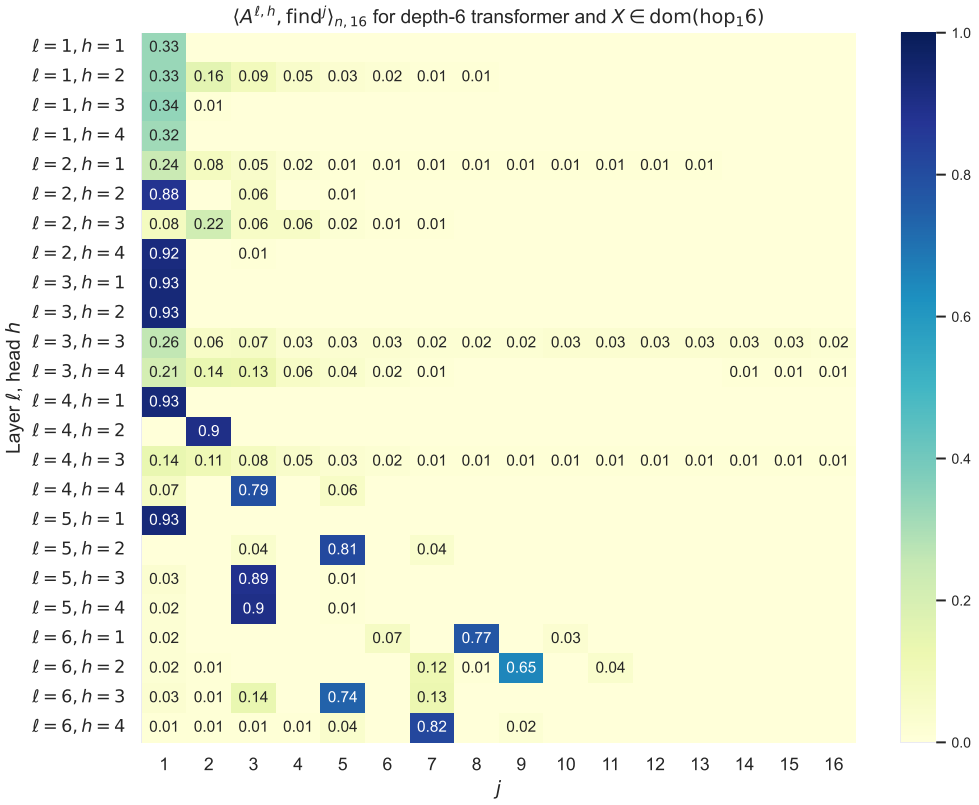


Figure 6.12: Plots of all inner products $\left\langle A^{\ell,h}[T^{\infty}_{4,6}], \text{find}^j \right\rangle_{10,16}$ for $n = 10$ samples $X^1, \ldots, X^{10} \in \text{dom}(\text{hop}_{16})$ for the 6-layer transformer $T^{\infty}_{4,6}$.

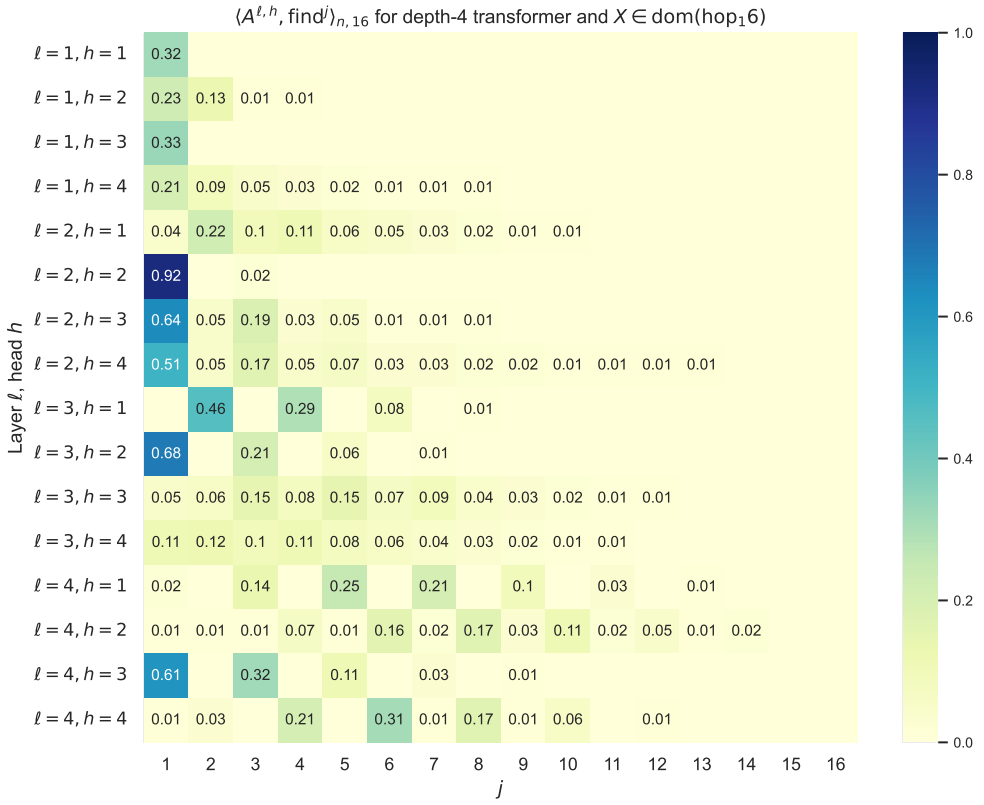


Figure 6.13: Plots of all inner products $\left\langle A^{\ell,h}[T^{\infty}_{4,4}], \text{find}^j \right\rangle_{10,16}$ for $n = 10$ samples $X^1, \ldots, X^{10} \in \text{dom}(\text{hop}_{16})$ for the 4-layer transformer $T^{\infty}_{4,4}$.

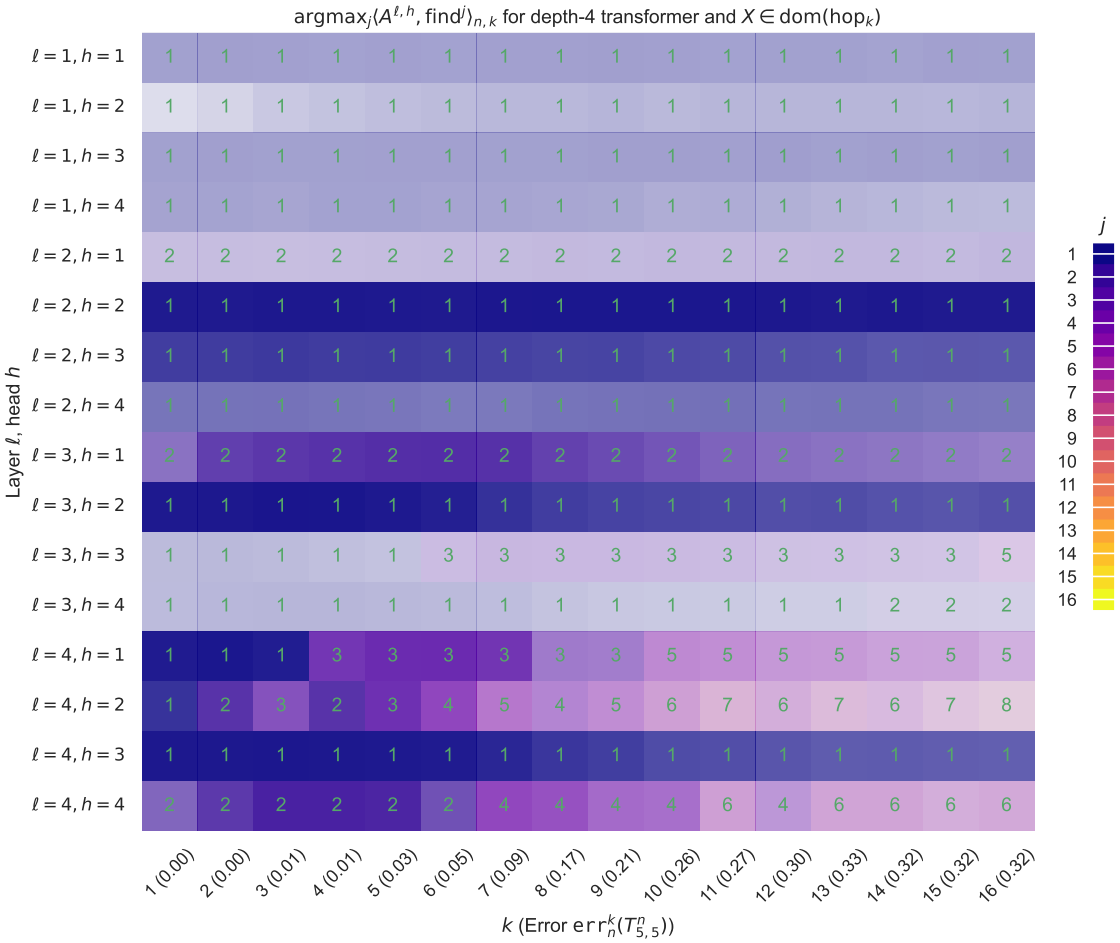


Figure 6.14: Plots of all the maximum inner products $\left\langle A^{\ell,h}[T^{\infty}_{4,4}], \text{find}^j \right\rangle_{n,k}$ for $n = 10$ fixed samples $X^1, \ldots, X^{10} \in \text{dom}(\text{hop}_k)$ for each $k \in [16]$ for the 4-layer transformer $T^{\infty}_{4,4}$. The hue corresponds to the index of the largest inner product $j^* = \arg\max_j \left\langle A^{\ell,h}[T^{\infty}_{4,4}], \text{find}^j \right\rangle_{n,k}$, while the opacity is determined by the magnitude of the correlation.

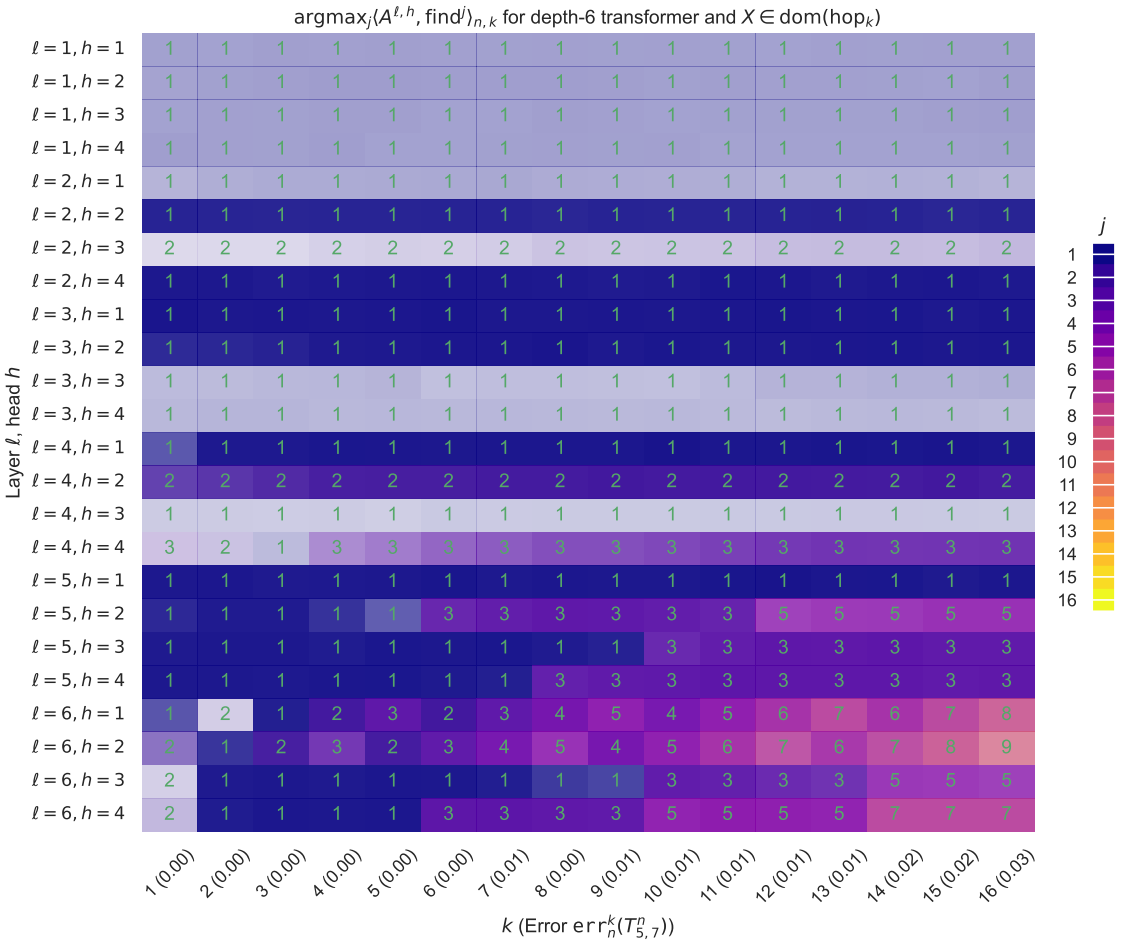


Figure 6.15: Plots of all the maximum inner products $\left\langle A^{\ell,h}[T^{\infty}_{4,6}], \text{find}^j \right\rangle_{n,k}$ for $n = 10$ fixed samples $X^1, \ldots, X^{10} \in \text{dom}(\text{hop}_k)$ for each $k \in [16]$ for the 6-layer transformer $T^{\infty}_{4,6}$.

6.5.5 Finite-sample experiments (Empirical Claim 6.24; Figures 6.16 to 6.19)

While most of our multi-hop experiments reside in the infinite-sample regime (where new samples are generated for every batch), we also trained several transformers on $n_{\text{train}} \in \{1000, 3000\}$ samples to evaluate whether generalization is possible in this domain, especially when the number of model parameters far exceeds the number of training samples. The two training set sizes expose a sharp threshold between two different generalization modes: low accuracy due to overfitting for most models on most tasks when $n_{\text{train}} = 1000$ and high accuracy approaching the infinite-sample regime when $n_{\text{train}} = 3000$.

Figure 6.16 compares the infinite-sample transformers $T^{\infty}_{4,L}$ with the 3000-sample models $T^{3000}_{4,L}$. 3000 training samples are sufficient to obtain comparable (if slightly worse) generalization error rates across model depths L and task complexities k. This supports a hypothesis that the existence of a small transformer that perfectly fits the data enables larger transformers to actually realize such architectures in the over-parameterized regime.

On the other hand, Figure 6.17 demonstrates that transformers trained on $n_{\text{train}} = 1000$ samples suffer poor performance on most tasks due to overfitting. While all models perform poorly on hop_k sub-tasks for large k, a depth-separation exists for simpler sub-tasks like hop_3. This suggests a positive inductive bias of deep transformers for simple compositional decision rules, which enables far better performance than other models in the overfitting regime.

To investigate this gap in performance, we contrast the self-attention inner products of depth-4 $T^{1000}_{4,4}$ and depth-6 $T^{1000}_{4,6}$ on the task hop_3 in Figures 6.18 and 6.19. The 6-layer model obtains a far superior classification error on the sub-task, and the interpretability plot establishes a plausible circuit it implements: It uses self-attention heads with find^1 functionality consecutively in layers 4, 5, and 6, which enables the robust retrieval of find^3 and hop_3. On the other hand, the 4-layer plot exhibits poor performance and only has two layers with find^1 functionality; this justifies the relatively strong performance of $T^{1000}_{4,4}$ on hop_2 and its poor performance on hop_3.

While neither model learns any kind of pointer-doubling construction, the 6-layer model is still able to learn a simple construction of hop_3 that the 4-layer model misses. The representational suitability of deeper models to compositional reasoning may thus provide a favorable inductive bias for learning the task in a setting with little data.

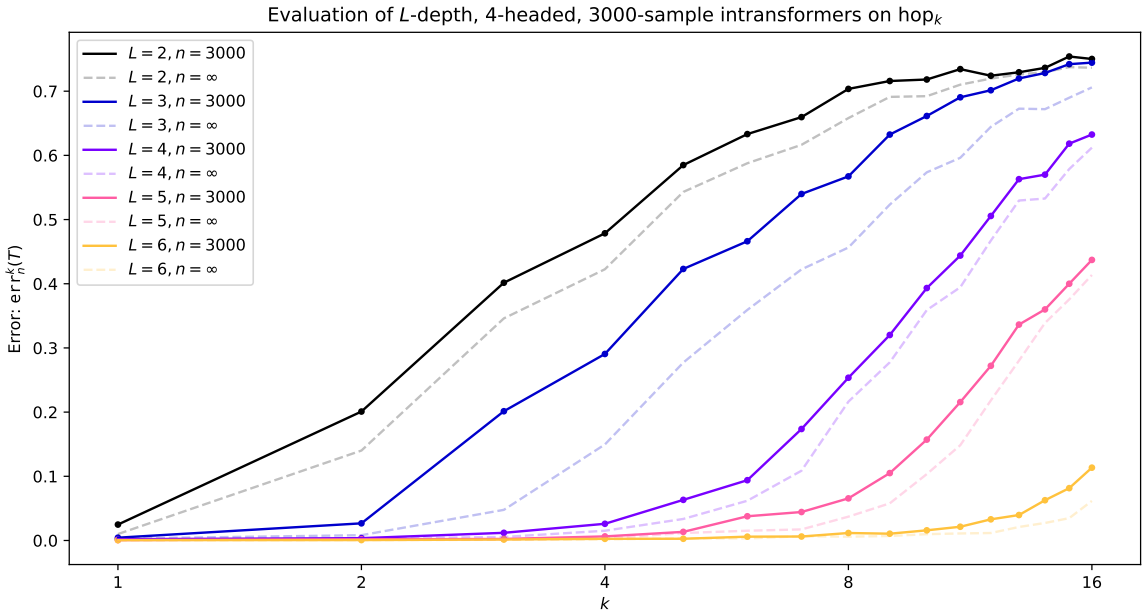


Figure 6.16: Comparison between the errors $\texttt{err}_k^n(T_{4,L}^n)$ of transformers trained in the infinite sample regime (dashed line) and on $n_{\text{train}} = 3000$ samples (solid line) on the multi-hop task, evaluated on $n = 100$ samples per hop_k task.

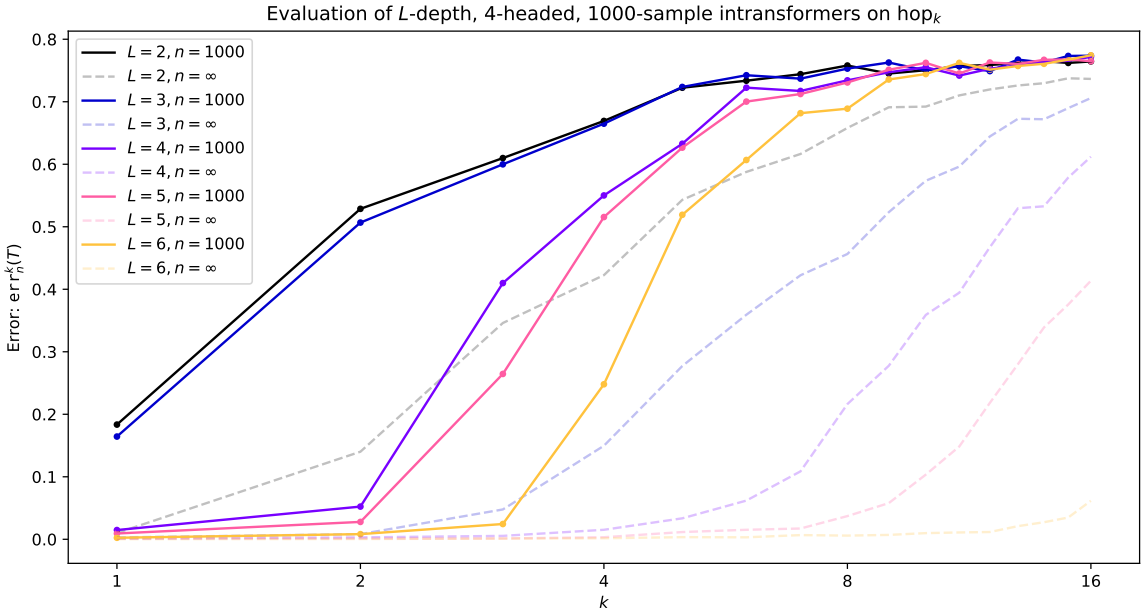


Figure 6.17: Comparison between the errors $\mathtt{err}_k^n(T_{4,L}^n)$ of transformers trained in the infinite sample regime (dashed line) and on $n_{\text{train}} = 1000$ samples (solid line) on the multi-hop task, evaluated on $n = 100$ samples per hop$_k$ task.

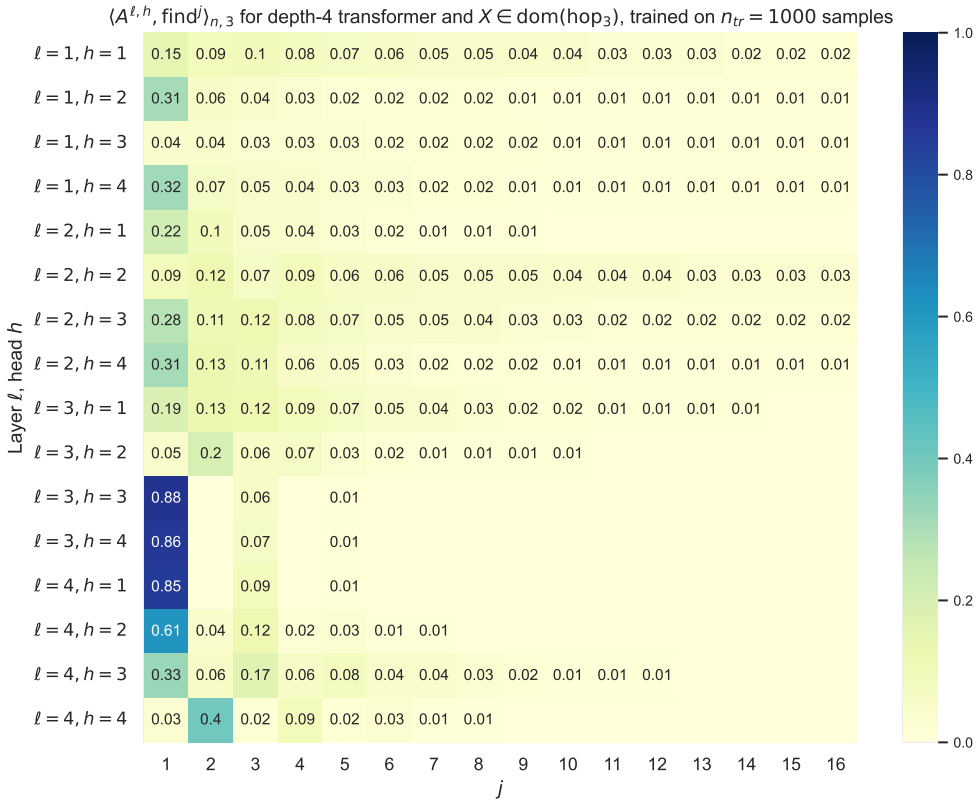


Figure 6.18: Plots of all inner products $\left\langle A^{\ell,h}[T^{1000}_{4,4}], \text{find}^j \right\rangle_{10,3}$ for $n = 10$ samples $X^1, \ldots, X^{10} \in \text{dom}(\text{hop}_3)$ for the 4-layer transformer $T^{1000}_{4,4}$.

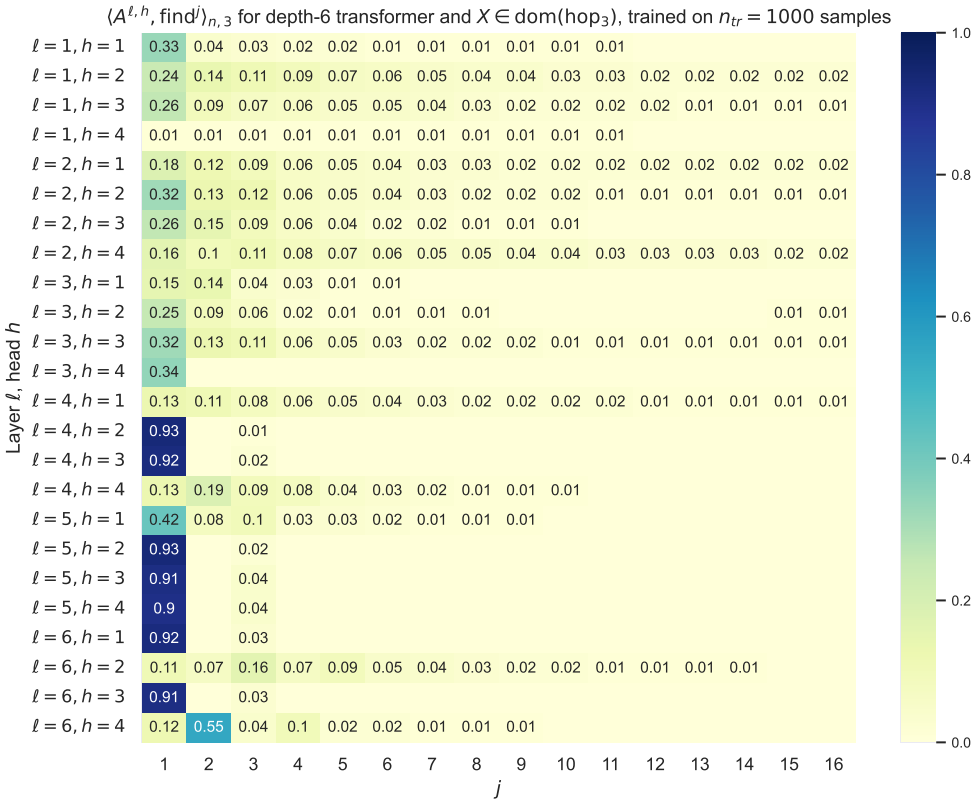


Figure 6.19: Plots of all inner products $\left\langle A^{\ell,h}[T^{1000}_{4,6}], \text{find}^j \right\rangle_{10,3}$ for $n = 10$ samples $X^1, \ldots, X^{10} \in \text{dom}(\text{hop}_3)$ for the 6-layer transformer $T^{1000}_{4,6}$.

6.6 Separations between transformers and alternative architectures

Sections 6.3 and 6.4 characterize the representational capability of transformers by providing algorithmic problems they can solve with logarithmic depth and small polynomial or constant width. In contrast, other well-known architectures are unable to solve those same problems in a parameter-efficient manner. This section provides lower bounds on the parameter complexity of graph neural networks (GNNs), recurrent neural architectures, transformers with computationally efficient alternatives to softmax self-attention, and single-layer transformers with autoregressive chain-of-thought tokens needed to solve graph connectivity and the k-hop task.

6.6.1 GNNs need polynomial depth for graph connectivity

The bidirectional relationship between transformers and MPC draws inspiration from past work drawing a similar connection between message passing graph neural networks (GNN_{mp}) and the CONGEST distributed computing model Loukas, 2019. Their computation model of GNN_{mp} for width m and depth L closely resembles our $\mathsf{Transformer}^N_{m,L,H}$ in providing a general framework for the analysis of graph neural networks by allowing unbounded computation in each vertex with bounded communication on edges. On some input graph G, vertices send neighbors messages of size at most m—which are aggregated and crafted into new messages with MLPs—over L rounds of communication.

By restating Corollary 4.2 of Loukas, 2019, we demonstrate a sharp contrast in the abilities of GNNs and transformers to solve graph algorithmic tasks.

Theorem 6.25 (Corollary 4.2 of Loukas, 2019)**.** *There exists a graph G with N edges such that any GNN_{mp} with width m and depth L that determines whether an input subgraph H either (1) is connected or (2) forms a spanning tree of G requires $L\sqrt{m} = \tilde{\Omega}(N^{1/4})$.*

While Corollaries 6.5 and 6.6 demonstrate the ability of transformers to determine whether any input graph is connected[7] or to identify a spanning tree with logarithmic depth and small polynomial width (i.e. $m = O(N^\epsilon)$), GNNs require depth $L = \tilde{\Omega}(N^{1/4-\epsilon/2})$ in the same regime. This gap is explainable by the fact that transformers on graph inputs G are not bound to pass messages exclusively along the edges of G. By "rewiring" the graphical structure in each layer, transformers can perform aggregation and "pointer passing" tasks with greater parametric ease than GNNs.

[7]While the problem of subgraph connectivity for GNNs may at first glance appear more difficult than general graph connectivity for transformers, an implementation of this exact task can be implemented by modifying the protocol Corollary 6.5 to remove all edges from the graph that do not belong to H.

6.6.2 Suboptimality of recurrent architectures for hop_k

The logarithmic-depth and constant-width transformer implementation of hop_k in Theorem 6.18 cannot be replicated by recurrent neural architectures (Chung et al., 2014; Bengio, Simard, and Frasconi, 1994; Turkoglu et al., 2021), including not just multi-layer recurrent neural networks (RNNs) but any sequential prediction procedure equivalent to them at inference time, which includes state space models such as Mamba (Gu and Dao, 2023).

We first consider a family of multi-layer RNNs of depth L and width m, consisting of arbitrary MLP units $g_\ell : \mathbb{R}^{m\times m} \to \mathbb{R}^{m\times m}$, which on input $X \in \mathbb{R}^{N\times d_{\text{in}}}$ produce output $Y \in \mathbb{R}^{N\times d_{\text{out}}}$ as follows using intermediates $X = Z^0, Z^1, \ldots, Z^{L-1}, Z^L = Y \in \mathbb{R}^{N\times m}$[8]and hidden states $H^1, \ldots, H^L \in \{0,1\}^{N\times m}$ with $H_0^\ell = \vec{0}$:

$$(Z_i^\ell, H_i^\ell) = g_\ell(Z_i^{\ell-1}, H_{i-1}^\ell),\ \forall i \in [N], \ell \in [L].$$

We provide a polynomial bound on the width and depth of a multi-layer RNN solving hop_k.

Corollary 6.26. *A multi-layer RNN of depth L and width m as above with $Y_N = \text{hop}_k(X)_N$ satisfies either $L \geq k$ or $m = \Omega(\frac{N}{k^6})$.*

In contrast to Theorem 6.18, which demonstrates that depth $O(\log k)$ transformers with constant width suffice to solve hop_k for any k, Corollary 6.26 demonstrates that all multi-layer RNNs with width $O(N^{1/7})$ require depth k when $k = O(N^{1/7})$.

Mamba (Gu and Dao, 2023) can be seen as the combination of three ideas: (1) a continuous-time dynamics model of sequential prediction, powerful enough to model Kalman filters, hidden markov models, and many others; (2) a family of time-discretization schemes; (3) an unrolling technique to enable efficient linear-time training, using ideas similar to FlashAttention (Dao et al., 2022). Ultimately, at inference time, the time-discretization step results in an RNN (see Gu and Dao, 2023, Algorithm 2 and Theorem 1), and is therefore directly handled by Corollary 6.26.

[8]We assume that $d_{\text{in}}, d_{\text{out}} \leq m$ and treat X and Y as if they are padded with zeros.

This corollary is a near immediate application of a communication complexity fact about the hardness of solving multi-player *pointer-chasing* problems with limited communication among players (Guha and McGregor, 2009; Assadi and N, 2021). We provide the communication model and this result in Section 6.6.5.1, and the reductions necessary to prove the above hardness results in Section 6.6.5.2.

6.6.3 Suboptimality of sub-quadratic attention transformers for hop_k

Due to the quadratic computational cost of computing the attention matrix

$$\text{softmax}(Q(X)K(X)^T) \in \mathbb{R}^{N \times N}$$

and the continued desire for ever-larger context lengths, there is substantial interest in improving the computational complexity of the transformer architecture while preserving its expressive capabilities and inductive biases. As a result, a rich literature has emerged that proposes computationally-efficient alternatives to standard softmax attention. In this section, we demonstrate how several representative examples of sub-quadratic attention mechanisms lose the ability to perform efficient parallel computation under a logarithmic-depth scaling.

Kernel-based sub-quadratic attention. One approach to computationally-efficient approximation of transformers are *kernel-based sub-quadratic attention* mechanisms such as Performer (Choromanski et al., 2022), and Poly-Sketchformer (Kacham, Mirrokni, and Zhong, 2023). Both approximate the attention matrix $\text{softmax}(Q(X)K(X)^\mathsf{T})$ with a low-rank matrix $Q'(X)K'(X)^\mathsf{T}$ where $Q', K' : \mathbb{R}^m \to \mathbb{R}^{m'}$ are applied element-wise. For sufficiently small $m' \ll N$, $Q'(X)K'(X)^\mathsf{T}V(X)$ can be computed efficiently by first computing $K'(X)^\mathsf{T}V(X) \in \mathbb{R}^{m' \times m}$, bounding the total runtime as $O(Nmm')$, rather than $O(N^2 m)$.

Let $\mathsf{KernelFormer}^N_{m,m',L,H}$ denote all H-headed L-layer transformer whose softmax attention modules are replaced by kernel-based sub-quadratic attention. We demonstrate the

limitations of $\mathsf{KernelFormer}^N_{m,m',L,H}$ by showing that, unlike $\mathsf{Transformer}^N_{m,L,H}$, they have no depth-efficient implementation of hop_k.

Corollary 6.27. *Any $T \in \mathsf{KernelFormer}^N_{m,m',L,H}$ with $T(X)_N = \mathrm{hop}_k(X)_N$ satisfies either $L \geq k$ or $mm'Hp = \Omega(\frac{N}{k^6})$.*

Under a parameter-efficient regime where $mpHL = O(N^\epsilon)$, solving hop_k for $k = \Theta(N^\epsilon)$ necessitates kernel feature dimension $m' = \Omega(N^{1-9\epsilon})$, which forces each attention unit to compute an $N \times N^{1-9\epsilon}$ matrix, yielding a nearly quadratic runtime. We prove Corollary 6.27 in Section 6.6.5.3 using a similar pointer chasing reduction.

Masking-based sub-quadratic attention. Another method that reduces the computational cost of transformers is to used masked models of Λ-$\mathsf{Transformer}^N_{m,L,H}$ for a sparse mask Λ. The Longformer architecture (Beltagy, Peters, and Cohan, 2020) introduces a particular masked architecture that combines sliding windows with sparse unmasked global tokens. Put concretely, for window radius w and global frequency g, let $\Lambda^{w,g} \in \{-\infty, 0\}^{N \times N}$ be masking matrix with

$$\Lambda^{w,g}_{i,j} = \begin{cases} 0 & \text{if } |i-j| \leq w \text{ or } j \equiv 0 \pmod{g}, \\ -\infty & \text{otherwise.} \end{cases}$$

Then, the output of a single unit of $\Lambda^{w,g}$-masked attention is computable in time $O((w + \frac{N}{g})Nm)$.

Corollary 6.28. *Any $T \in \Lambda^{w,g}$-$\mathsf{Attn}^N_{m,L,H}$ with $T(X)_N = \mathrm{hop}_k(X)_N$ satisfies either $L \geq k$ or $(w + \frac{N}{gk})mHp = \Omega(\frac{N}{k^6})$.*

Like kernel-based attention, sparsely-masked attention models fail to efficiently compute hop_k. Similarly, in the same parameter-efficient regime, a Longformer must have either $w = \Omega(N^{1-9\epsilon})$ or $g = O(N^{9\epsilon})$, which jointly ensures that the masked matrix has at least $\Omega(N^{2-9\epsilon})$ entries and diminishes any computational advantages. This proof also appears in Section 6.6.5.3.

6.6.4 Limitations of 1-layer transformers with chain-of-thought

While most of the paper considers transformers as sequence-to-sequence models, we can also frame them as auto-regressive models performing next-token-prediction with chain-of-thought prompting. In this regime, a single causally-masked transformer aims to compute a function of its input by repeatedly predicting the next token, appending previously predicted tokens to the end of the input. In doing so, a function is computable if there exists an intermediate *chain-of-thought* produced by the model that eventually reaches the answer.

Definition 6.6. We say that $T \in \mathsf{MaskTransformer}_{m,L,H}^{N+N_{\mathrm{CoT}}}$ computes $f : \Sigma^{N+N_{\mathrm{CoT}}} \to \Sigma^N$, where the additional N tokens denote chain-of-thought, if for every $X \in \mathrm{dom}(f)$, there exists $X_{\mathrm{CoT}} \in \Sigma^{N_{\mathrm{CoT}}}$ such that $T(X \circ X_{\mathrm{CoT}})_{N:N+N_{\mathrm{CoT}}} = (X_{\mathrm{CoT}} \circ f(X))$.

The theoretical capabilities of chain-of-thought augmented transformers to simulate finite-state automata and Turing machines have been studied (Malach, 2023; Merrill and Sabharwal, 2023a), but the comparative capabilities of shallow models with chain-of-thought prompting and deep sequential models are unknown. In contrast to the fact that any transformer with N_{CoT} tokens can be simulated by a sequential model with depth scaled by N_{CoT}, we show that deep transformers cannot necessarily be efficiently simulated by shallow chain-of-thought models. We do so by demonstrating that a linear amount of chain-of-thought prompting in k is necessary to solve $\mathrm{hop}_k(X)_N$, and also sufficient.

Corollary 6.29. *Any* $T \in \mathsf{MaskTransformer}_{m,1,H}^{N+N_{\mathrm{CoT}}}$ *that computes* $\mathrm{hop}_k(X)_N$ *with* N_{CoT} *tokens of chain-of-thought requires either* $N_{\mathrm{CoT}} \geq k$ *or* $mHp = \Omega(\frac{N}{k^6})$.

The proof appears in Section 6.6.5.4. For future work, it remains to consider the comparative powers of chain-of-thought models of depths greater than one.

6.6.5 Proofs for Section 6.6

6.6.5.1 Multi-player pointer chasing communication complexity

We introduce the multi-pass multi-player blackboard communication model studied by Guha and McGregor (2009) and Assadi and N (2021) to prove lower bounds for multi-pass streaming algorithms. A protocol in this model specifies how k players, each possessing a portion of a shared input, can jointly compute a function on the input over the course of R rounds of communication. In each round, all players take turns to broadcast an s-bit message to all other players. We provide a formal definition of the model as described in Section 6 of Assadi and N (2021).

Definition 6.7. A *k-player R-round s-space sequential blackboard communication protocol* includes k players $P_1, \ldots, P_k$. On input Z that can be partitioned into $(Z_1, \ldots, Z_k)$, each player P_j is provided with its respective Z_j. In each round, players communicate via a shared blackboard. That is, in round r and in order $P_k, \ldots, P_1$, each player P_j writes a message $\Pi_j^r \in \{0,1\}^s$ on the blackboard (which can be viewed by all players) as a potentially randomized function of input Z_j and all information on the blackboard. After the conclusion of R rounds, the final message Π_1^R is the output of the protocol.

Assadi and N (2021) proves a lower bound on the round complexity necessary to solve the well-studied *multi-party pointer chasing problem* of Nisan and Wigderson (1993). We present the problem as defined by Assadi and N (2021).

Definition 6.8. For $q, k \in \mathbb{Z}_+$, let an *(q,k)-layered graph* $G = (V, E)$ have disjoint vertex layers $V_1, \ldots, V_{k+1}$ with $V = V_1 \cup \cdots \cup V_{k+1}$ and each $|V_j| = q$ and edge layers $E_1, \ldots, E_k$ with $E = E_1 \cup \cdots \cup E_k$ and each E_j being a perfect matching between V_j and V_{j+1}. The *pointer chasing* task is provides a (q,k)-layered graph G, an arbitrary $v \in V_1$, and an arbitrary equipartition V_{k+1}^1 and V_{k+1}^2 of V_{k+1} as input and asks whether v is connected to a vertex in V_{k+1}^1 or V_{k+1}^2.

Assadi and N (2021) give the following lower bound.

Proposition 6.30 (Proposition 4.12 of Assadi and N, 2021)**.** *Consider a k-player R-round s-space sequential blackboard protocol that solves the (q, k)-pointer chasing task where each player P_j is provided with the matching E_j and v and V_{k+1}^1, V_{k+1}^2 are globally known. Then, the protocol succeeds with probability at least $\frac{2}{3}$ only if $R \geq k$ or $s = \Omega(\frac{q}{k^5})$.*

All of the lower bounds in Section 6.6 are most naturally proved by reducing from hop_k, rather than pointer chasing. So we first prove a lower bound for hop_k using the lower bound for pointer chasing from Proposition 6.30.

Proposition 6.31. *Consider a k-player R-round s-space sequential blackboard protocol that computes $\text{hop}_k(X)_N$ on any $X \in \Sigma^N$ for $\Sigma = [2q + 2]$ with $q = \left\lfloor \frac{N}{2k} \right\rfloor$ where each player P_j is provided with $X^j := (X_{2(k-j)q+1}, \ldots, X_{2(k-j+1)q})$, except for P_1, who is given $X^1 := (X_{2(k-1)q+1}, \ldots, X_N)$. Then, the protocol succeeds with probability at least $\frac{2}{3}$ only if $R \geq k$ or $s = \Omega(\frac{N}{k^6})$.*

Proof. Assuming the existence of a k-player R-round s-space sequential blackboard protocol for $\text{hop}_k(X)_N$ as described above, we design a protocol for solving (q, k)-pointer chasing with R rounds and s-size messages. The claimed lower bound will then follow by Proposition 6.30.

Consider any pointer chasing input with universally known $V_1, \ldots, V_{k+1}$, $v \in V_1$, and V_{k+1}^1 and V_{k+1}^2, and each player P_j knowing matching E_j. We recursively define $v_1, \ldots, v_k + 1$ such that $v_1 = v$ and $(v_j, v_{j+1}) \in E_j$, noting that the output hinges on whether $v_{k+1} \in V_{k+1}^1$.

Without loss of generality, let $v = 1$ and

$$V_j = \begin{cases} \{1, \ldots, q\} & \text{if } j \text{ is odd,} \\ \{q + 1, \ldots, 2q\} & \text{if } j \text{ is even.} \end{cases}$$

Each player independently determines their substring X^j of a input X to hop_k before running the aforementioned protocol:

- Player P_1 encodes X^1 by letting $X_N = s = 1$ and for any $i \in 1, \ldots, 2q$, letting

$$X_i^1 = \begin{cases} \frac{i+1}{2} \in V_1 & \text{if } i \text{ is odd,} \\ i' \in V_2 & \text{if } i \text{ is even, } (\frac{i}{2}, i') \in E_1. \end{cases}$$

  This ensures that that every integer in $\{1, \ldots, 2q\}$ appears exactly once in $X_1^1, \ldots, X_{2q}^1$, which in turn guarantees that $\text{find}_X^1(N) = (k-1+1)q+2$ and that $X_{\text{find}_X^1(N)} = v_2$ where $(1, i') \in E_1$.

- For any $j \in \{2, \ldots, k-1\}$, player P_j encodes E_j as X^j as follows. If j is odd, then for every $i \in \{1, \ldots, 2q\}$,

$$X_i^j = \begin{cases} \frac{i+1}{2} \in V_j & \text{if } i \text{ is odd,} \\ i' \in V_{j+1} & \text{if } i \text{ is even, } (\frac{i}{2}, i') \in E_j. \end{cases}$$

  Alternatively, if j is even,

$$X_i^j = \begin{cases} q + \frac{i+1}{2} \in V_j & \text{if } i \text{ is odd,} \\ i' \in V_{j+1} & \text{if } i \text{ is even, } (\frac{i}{2}, i') \in E_j. \end{cases}$$

  Since every odd token corresponds to a vertex in V_j and each subsequent token corresponds to the vertex it's connected to by E_j, we can ensure that for every $i \in [2q]$:

$$(X_{2(k-j+1)+i}, X_{\text{find}_X^1(2(k-j+1)+i)}) \in E_j.$$

  Hence, it follows inductively that $X_{\text{find}_X^j(N)} = v_{j+1}$.

- Player P_k encodes X^k if k is odd by letting

$$X_i^k = X_i = \begin{cases} \frac{i+1}{2} \in V_k & \text{if } i \text{ is odd,} \\ 2q+1 & \text{if } i \text{ is even, } (\frac{i}{2}, v) \in E_k, \text{ and } v \in V_{k+1}^1, \\ 2q+2 & \text{if } i \text{ is even, } (\frac{i}{2}, v) \in E_k, \text{ and } v \in V_{k+1}^2. \end{cases}$$

  Likewise, if k is even,

$$X_i^k = X_i = \begin{cases} q + \frac{i+1}{2} \in V_k & \text{if } i \text{ is odd,} \\ 2q+1 & \text{if } i \text{ is even, } (\frac{i}{2}, v) \in E_k, \text{ and } v \in V_{k+1}^1, \\ 2q+2 & \text{if } i \text{ is even, } (\frac{i}{2}, v) \in E_k, \text{ and } v \in V_{k+1}^2. \end{cases}$$

These jointly ensure that

$$\mathrm{hop}_k(X)_N = X_{\mathrm{find}_X^k(N)} = \begin{cases} 2q+1 & \text{if } v_{k+1} \in V_{k+1}^1, \\ 2q+2 & \text{if } v_{k+1} \in V_{k+1}^2. \end{cases}$$

Therefore, by formatting $E_1, \ldots, E_k$ appropriately as X, running the protocol for $\mathrm{hop}_k(X)_N$, and observing that the final output of player P^1 is $2q+1$ if and only if $v_{k+1} \in V_{k+1}^1$, there exists a k-player R-round s-space protocol for pointer chasing. Hence, by Proposition 6.30, the protocol for $\mathrm{hop}_k(X)_N$ must use $R \geq k$ rounds or $s = \Omega(\frac{N}{k^6})$ space. □

6.6.5.2 Proofs for Section 6.6.2

Corollary 6.26. *A multi-layer RNN of depth L and width m as above with $Y_N = \mathrm{hop}_k(X)_N$ satisfies either $L \geq k$ or $m = \Omega(\frac{N}{k^6})$.*

Proof. Suppose there exists a multi-layer RNN computing output Y with $Y_{N,1} = \mathrm{hop}_k(X)_N$ from input X with intermediate states $Z_1, \ldots, Z_{L-1}$ and hidden states $H^1, \ldots, H^L$. For any $\ell \in [L]$ and $i \leq i'$, note that $Z_i^\ell, \ldots, Z_{i'}^\ell$ can be determined exactly from H_{i-1}^ℓ and

$Z_i^{\ell-1}, \ldots, Z_{i'}^{\ell-1}$. Given this RNN, we provide a multi-player blackboard communication protocol for solving $\mathrm{hop}_k(X)_N$ under the input model of Proposition 6.31.

In round r, we assume inductively that each player P_j knows

$$Z^{\ell-1,j} = (Z^{\ell-1}_{2(k-j)q+1}, \ldots, Z^{\ell-1}_{2(k-j+1)q}),$$

except for P_1, who knows

$$Z^{\ell-1,1} = (Z^{\ell-1}_{2(k-1)q+1}, \ldots, Z^{\ell-1}_{N}).$$

In descending order, each player P_j computes $Z^{\ell,j}$ and $H^{\ell}_{2(k-j+1)q}$—writing the latter on the blackboard—from $Z^{\ell-1,j}$ and $H^{\ell}_{2(k-j)q}$,which was written on the blackboard by the previous player. Thus, player P^1 after round L knows and outputs $Z^L_{N,1} = Y_{N,1} = \mathrm{hop}_k(X)_N$, which provides an L-round protocol m-space protocol.

So the claimed lower bounds on width and depth follow from Proposition 6.31. □

6.6.5.3 Proofs for Section 6.6.3

Corollary 6.27. *Any $T \in \mathsf{KernelFormer}^N_{m,m',L,H}$ with $T(X)_N = \mathrm{hop}_k(X)_N$ satisfies either $L \geq k$ or $mm'Hp = \Omega(\frac{N}{k^6})$.*

Proof. Under the distribution of input $X = (X^1, \ldots, X^k)$ to players $P_1, \ldots, P_k$ stipulated in the statement of Proposition 6.31, we explain how the players can all compute the outcome of a single layer of H-headed kernelized attention in a single round of a blackboard protocol. It is immediate that a depth L network can be simulated in L rounds.

On input X, consider H kernelized self-attention units with embeddings

$$(Q'_1, K'_1, V_1), \ldots, (Q'_H, K'_H, V_H)$$

and output MLP ψ. Each player P_j immediately computes its embeddings

$$(Q'_h(X^j), K'_h(X^j), V_h(X^j))_{h\in[H]},$$

followed by

$$(K'_h(X^j)^\mathsf{T} V_h(X^j)) \in \mathbb{R}^{m'\times m}$$

for each $h \in [H]$. Because the object is to compute for each h

$$\psi(Q'_h(X)K'_h(X)^\mathsf{T}V_h(X)) = \psi(Q'_h(X)\sum_{j=1}^{k} K'_h(X^j)^\mathsf{T}V_h(X^j)),$$

each player writes their $(K'_h(X^j)^\mathsf{T}V_h(X^j))_{h\in[H]}$ using message size $s = \Theta(mm'Hp)$. Each can then construct $K'_h(X)^\mathsf{T}V_h(X))$ by reading the board, and use it to compute its respective outputs without requiring supplemental communication.

Hence, T (and thus $\mathrm{hop}_k(X)_N$) can be simulated using an L-round blackboard protocol with message size $s = \Theta(mm'Hp)$, and the corollary follows from Proposition 6.31. □

Corollary 6.28. *Any $T \in \Lambda^{w,g}\text{-}\mathsf{Attn}^N_{m,L,H}$ with $T(X)_N = \mathrm{hop}_k(X)_N$ satisfies either $L \geq k$ or $(w + \frac{N}{gk})mHp = \Omega(\frac{N}{k^6})$.*

Proof. As in the proof of Corollary 6.27, we explain how each player can compute their respective outputs of a single unit of self-attention masked by $\Lambda^{w,g}$.

To compute the output corresponding to X_i, note that it is necessary to only know the embeddings corresponding to $X_{i-w}, X_{i-w+1}, \ldots, X_{i+w}$ and $X_g, X_{2g}, \ldots, X_{\lfloor N/g\rfloor g}$. Thus, player X^j can compute the outputs of all of their inputs $X^j = (X_{2(k-j)q+1}, \ldots, X_{2(k-j+1)q})$ given access to

$$X_{2(k-j)q+1-w}, \ldots, X_{2(k-j)q}, X_{2(k-j+1)q+1}, \ldots, X_{2(k-j+1)q+w},$$

as well as $X_g, X_{2g}, \ldots, X_{\lfloor N/g\rfloor g}$.

Therefore, the protocol can be simulated if each player X^j writes inputs

$$X_{2(k-j)q+1}, \ldots, X_{2(k-j)q+w}, X_{2(k-j+1)q-w+1}, \ldots, X_{2(k-j+1)q} \in \mathbb{R}^m,$$

in addition to all $X_i \in X^j$ such that $i \equiv 0 \pmod{g}$. This can be accomplished by a protocol where each player writes $s = O((w + \frac{N}{gk})mp)$ bits of information on the blackboard.

By repeating this protocol in parallel for every head and sequentially for every layer, T and $\mathrm{hop}_k(X)_N$ can be simulated, and hence the claim follows from Proposition 6.31. □

6.6.5.4 Proofs for Section 6.6.4

Corollary 6.29. *Any* $T \in \mathsf{MaskTransformer}_{m,1,H}^{N+N_{\mathrm{CoT}}}$ *that computes* $\mathrm{hop}_k(X)_N$ *with* N_{CoT} *tokens of chain-of-thought requires either* $N_{\mathrm{CoT}} \geq k$ *or* $mHp = \Omega(\frac{N}{k^6})$.

Proof. We reduce to Proposition 6.31. Consider some input $X \in \mathbb{R}^N$ partitioned into $X^1, \ldots, X^j$ as specified by the proof of Proposition 6.31 with chain-of-thought X_{CoT} and $\mathrm{hop}_k(X)_N$ determined by some masked transformer T.[9] Suppose T has embeddings

$$Q_h, K_h, V_h)_{h \in [H]}$$

and output MLP ψ. We provide an $(N_{\mathrm{CoT}} + 1)$-round blackboard protocol to compute $\mathrm{hop}_k(X)_N$ from X.

Suppose in the rth round of the protocol, all players know $X_{\mathrm{CoT},1}, \ldots, X_{\mathrm{CoT},r-1}$ and aim

[9] We abuse notation to index $X_{N+i} = X_{\mathrm{CoT},i}$ and let $X_i \in X^j$ be true if $i \in \{2(k-j)q+1, \ldots, w(k-j+1)q\}$.

to compute

$$
\begin{aligned}
&T(X \circ X_{\mathrm{CoT}})_{N+r-1} \\
&\quad = \begin{cases} X_{\mathrm{CoT},r} & \text{if } r \le N_{\mathrm{CoT}} \\ \mathrm{hop}_k(X)_N & \text{if } r = N_{\mathrm{CoT}} + 1 \end{cases} \\
&\quad = \psi_{N+r-1}\left(X_{N+r-1} + \sum_{h=1}^{H} \frac{\sum_{i=1}^{N+r-1} \exp(Q^h_{N+r-1}(X_{N+r-1})^\mathsf{T} K^h_i(X_i)^\mathsf{T}) V^h_i(X_i)}{\sum_{i=1}^{N+r-1} \exp(Q^h_{N+r-1}(X_{N+r-1})^\mathsf{T} K^h_i(X_i))} \right).
\end{aligned}
$$

If we let

$$
\begin{aligned}
S_{r,h,j} &= \sum_{X_i \in X^j} \exp(Q^h_{N+r-1}(X_{N+r-1})^\mathsf{T} K^h_i(X_i)^\mathsf{T}) V^h_i(X_i) \in \mathbb{R}^m, \\
S_{r,h,\mathrm{CoT}} &= \sum_{i=N+1}^{N+r-1} \exp(Q^h_{N+r-1}(X_{N+r-1})^\mathsf{T} K^h_i(X_i)^\mathsf{T}) V^h_i(X_i) \in \mathbb{R}^m, \\
Z_{r,h,j} &= \sum_{X_i \in X^j} \exp(Q^h_{N+r-1}(X_{N+r-1})^\mathsf{T} K^h_i(X_i)^\mathsf{T}) \in \mathbb{R}, \\
Z_{r,h,\mathrm{CoT}} &= \sum_{i=N+1}^{N+r-1} \exp(Q^h_{N+r-1}(X_{N+r-1})^\mathsf{T} K^h_i(X_i)^\mathsf{T}) \in \mathbb{R},
\end{aligned}
$$

then we observe that

$$
T(X \circ X_{\mathrm{CoT}})_{N+r-1} = \psi_{N+r-1}\left(X_{N+r-1} + \sum_{h=1}^{H} \frac{\sum_{j=1}^{k} S_{r,h,j} + S_{r,h,\mathrm{CoT}}}{\sum_{j=1}^{k} Z_{r,h,j} + Z_{r,h,\mathrm{CoT}}} \right).
$$

Each player P_k computes $(S_{r,h,j}, Z_{r,h,j})_{h \in [H]}$ and writes them on the blackboard with $O(mHp)$-bit messages. Since $S_{r,h,\mathrm{CoT}}$ and $Z_{r,h,\mathrm{CoT}}$ are known by all players, every player can individually $T(X \circ X_{\mathrm{CoT}})_{N+r-1}$.

By induction, all players know $\mathrm{hop}_k(X)_N$ after $N_{\mathrm{CoT}} + 1$ rounds. The claim now follows from Proposition 6.31. □

6.7 Proofs of low-level attention constructions

This section provides the proofs of "low-level" transformer constructions, which are used to prove the main results throughout the chapter. These results talk directly about the embeddings utilized in various self-attention units. The separation of these proofs from the main text is intended to make the main text more readable and to allow the reader to focus on the high-level ideas of the main results.

6.7.1 Hardmax simulation proof of Section 6.2.2.2

Lemma 6.2. *Let $f \in \mathsf{Attn}_m^N$ be a self-attention unit with precision $p = \Theta(\log N)$ and embedding functions Q, K, V such that for some fixed $1 \geq \xi = N^{-O(1)}$ and every $X \in \mathbb{R}^{N \times m}$ and $i \in [N]$:*

$$A(X)_{i,i'} \leq \max_{i''} A(X)_{i,i''} - \xi, \ \forall i' \notin I_{\max}(A(X)_i),$$

where $A(X) = Q(X)K(X)^\mathsf{T}$. Then there exists a self-attention unit $f' \in \mathsf{Attn}_m^N$ with a valid p'-bit implementation with $p' = O(p)$ satisfying

$$f'(X) = \text{hardmax}(A(X))V(X).$$

Proof. For some $p' = \Theta(p + \log \frac{1}{\xi})$ and $c = \Theta(\frac{p'+\zeta}{\xi} \cdot \log N)$ where ζ is as in Section 6.2.2.2), let f' have query embedding $Q'(X) = cQ(X)$ and identical key K and value V embeddings as f. Therefore, by construction, these embeddings can be written with precision $p' = O(\ln(c) + p) = O(\log \frac{1}{\xi} + \log\log N + p) = O(p)$.

Let $\hat{f}'$ be a valid p'-bit implementation of f', meaning that the two $\|\hat{f}' - f'\|_\infty = O(1/2^{p+1})$ (thus $\hat{f}'$ rounds f' to p' bits of precision), and fix some X. We first show that the softmax matrix is sufficiently close to that of the hardmax and is also a valid p'-bit implementation

of the hardmax. Without loss of generality, let $1 \in I_{\max}(A(X)_i)$. First, note that

$$\sum_{i' \notin I_{\max}(A(X)_i)} \exp(cA(X)_{i,i'}) \leq \frac{N}{\exp(c\xi)} \exp(cA(X)_{i,1}) = \frac{1}{N^{O(p'+\zeta)}} \exp(cA(X)_{i,1}).$$

Then,

$$\begin{aligned} |\text{softmax}(cA(X))_{i,1} - \text{hardmax}(A(X))_{i,1}| &= \frac{1}{|I_{\max}(A(X)_i)|} - \frac{\exp(cA(X)_{i,1})}{\sum_{i'=1}^{N} \exp(cA(X)_{i,i'})} \\ &\leq \frac{\sum_{i' \notin I_{\max}(A(X)_i)} \exp(cA(X)_{i,i'})}{|I_{\max}(A(X)_i)| \exp(cA(X)_{i,1})} = \frac{1}{N^{\Omega(p'+\zeta)}}. \end{aligned}$$

Likewise, for any $i'' \notin I_{\max}(A(X)_i)$:

$$|\text{softmax}(cA(X))_{i,i''} - \text{hardmax}(A(X))_{i,i''}| \leq \frac{\exp(cA(X)_{i,i''})}{\sum_{i'=1}^{N} \exp(cA(X)_{i,i'})} = \frac{1}{N^{\Omega(p'+\zeta)}}.$$

Therefore,

$$\begin{aligned} &\|\text{softmax}(cA(X))_i - \text{hardmax}(cA(X))_i\|_2 \\ &\leq \sqrt{N} \cdot \max_{i''} |\text{softmax}(cA(X))_{i,i''} - \text{hardmax}(cA(X))_{i,i''}| \\ &= \frac{1}{N^{\Omega(p'+\zeta)}}. \end{aligned}$$

We conclude that the approximation is sufficiently close, meaning it is $O(1/2^{p'})$, whereby

it is exact after rounding:

$$\begin{aligned}
&\left\|\hat{f}'(X) - \text{hardmax}(Q(X)K(X)^\mathsf{T})V(X)\right\|_\infty \\
&\quad\leq \left\|f'(X) - \text{hardmax}(Q(X)K(X)^\mathsf{T})V(X)\right\|_\infty + \left\|\hat{f}'(X) - f'(X)\right\|_\infty \\
&\quad\leq \max_{i,j} \left|\text{softmax}(cA(X))_i^\mathsf{T} V(X)_{\cdot,j} - \text{hardmax}(A(X))_i^\mathsf{T} V(X)_{\cdot,j}\right| + O\left(\frac{1}{2^{p'}}\right) \\
&\quad\leq \max_{i,j} \left\|\text{softmax}(cA(X))_i^\mathsf{T} - \text{hardmax}(A(X))_i^\mathsf{T}\right\|_2 \left\|V(X)_{\cdot,j}\right\|_2 + O\left(\frac{1}{2^{p'}}\right) \\
&\quad\leq \frac{1}{N^{\Omega(p'+\zeta)}} \cdot \sqrt{N} \cdot N^\zeta + O\left(\frac{1}{2^{p'}}\right) \\
&\quad= O\left(\frac{1}{2^{p'}}\right).
\end{aligned}$$

Therefore, $\hat{f}'$ is a valid p'-bit implementation of $\text{hardmax}(Q(X)K(X)^\mathsf{T})V(X)$. □

6.7.2 Constructions for Section 6.3.3.1

Proposition 6.10. *For any $b \leq N$ and d, there exists a self-attention unit*

$$\text{sparsePropagate}_{Q,d} \in \mathsf{Attn}_{m,p}^N$$

for $m = d + O(Q \log N)$ and $p = O(\log N)$, which, given any input X with

$$X_i = (z_i, S_i, \vec{0}) \in \mathbb{R}^d \times \binom{[N]}{\leq Q} \times \{0\}^{m-Q-d}$$

such that $b_i = |\{S_j \ni i : j \in [N]\}| \leq Q$ for all i, has output $\text{sparsePropagate}_{Q,d}(X)$ satisfying

$$\text{sparsePropagate}_{Q,d}(X)_i = \frac{1}{b_i} \sum_{S_j \ni i} z_j.$$

Proof. Following the proof of Theorem 5.4, there exist p-bit precision vectors $u_1, \ldots, u_N \in$

$\{\pm 1/\sqrt{m}\}^m$ and w_S with $w_S \leq 2\sqrt{Q}$ for all $S \in \binom{N}{\leq Q}$ such that

$$u_i^\mathsf{T} w_S = 1, \text{ for all } i \in S$$
$$u_i^\mathsf{T} w_S \leq \frac{1}{2}, \text{ for all } i \notin S.$$

We then design the embeddings of sparsePropagate$_{Q,d}$ with

$$Q(X)_i = (u_i, 1),$$
$$K(X)_i = \begin{cases} (w_{S_i}, 0) & \text{if } i > 0, \\ (\vec{0}, \frac{3}{4}) & \text{if } i = 0, \end{cases}$$
$$V(X)_i = \begin{cases} z_i & \text{if } i > 0, \\ \vec{0} & \text{if } i = 0. \end{cases}$$

As a result,

$$Q(X)_i^\mathsf{T} K(X)_{i'} = 1 \qquad \text{if } i \in S_{i'}, i' > 0,$$
$$Q(X)_i^\mathsf{T} K(X)_{i'} \leq \frac{1}{2} \qquad \text{if } i \notin S_{i'}, i' > 0,$$
$$Q(X)_i^\mathsf{T} K(X)_0 = \frac{3}{4}.$$

Hence, the largest inner products for query i correspond to i' for all $S_{i'} \ni i$ if any exist, and 0 otherwise. There exists a margin of at least $\frac{1}{4}$ between the largest inner product in each row and all others. By applying Lemma 6.2, we conclude that there exists a self attention unit f' with embedding dimension $p = \Theta(\log N)$ that computes

$$f'(X) = \text{hardmax}(Q(X)K(X)^\mathsf{T})V(X) = \text{sparsePropagate}(X). \qquad \square$$

6.7.3 Constructions for Section 6.3.3.2

Lemma 6.13. *For any MPC protocol π with local memory s and q machines with n_{in}-word inputs, there exists a transformer* $\text{init} \in \mathsf{Transformer}^{n_{\text{in}},\max(n_{\text{in}},q)}_{s,1,1,d_{\text{in}},d_{\text{out}}}$ *with $d_{\text{in}} = 1$ and $d_{\text{out}} = s$, which, given $\texttt{Input} \in \mathbb{Z}^n_{2^p}$, has output satisfying* $\text{init}(\texttt{Input}) = \texttt{MachineIn}^{(1)}$.

Proof. Let $M = \max(n_{\text{in}}, q)$ and $Q, K, V : \mathbb{Z}^M_{2^p} \to \mathbb{R}^{M\times s}$ be the query, key, and value embeddings of the attention unit f in init, and let $\psi : \mathbb{R}^{M\times s} \to \mathbb{Z}^s_{2^p} \times [N]$ be its output MLP. Let $q_{\text{in}} = \left\lceil \frac{n_{\text{in}}}{s} \right\rceil$ denote the number of machines used to store the inputs.

Let $\texttt{Dest}_{i'} = \left\lceil \frac{i'}{s} \right\rceil \in [q_{\text{in}}]$ denote the machine that stores the input token index $i' \in [n_{\text{in}}]$ in the MPC protocol, and let

$$\texttt{Rcvd}_i = \{(s-1)i + 1, \ldots, \min(si, n_{\text{in}})\}$$

denote the set of all input tokens indices belonging to $\texttt{MachineIn}^{(1)}_i$ for machine $i \in [q_{\text{in}}]$.

For each machine $i \in [q_{\text{in}}]$, we define the query embedding as

$$Q(\texttt{Input})_i = \left(\cos\left(\frac{2\pi i}{M}\right), \sin\left(\frac{2\pi i}{M}\right), \ldots, \cos\left(\frac{2\pi i}{M}\right), \sin\left(\frac{2\pi i}{M}\right)\right).$$

Likewise, for each token index $i' \in [n_{\text{in}}]$, the key and value vectors are

$$K(\texttt{Input})_{i',(2\iota-1,2\iota)} = \begin{cases} \left(\cos\left(\frac{2\pi\cdot\texttt{Dest}_{i'}}{M}\right), \sin\left(\frac{2\pi\cdot\texttt{Dest}_{i'}}{M}\right)\right) & \text{if } i' \le n_{\text{in}},\ i' \equiv \iota \pmod s, \\ (0,0) & \text{otherwise,} \end{cases}$$

$$V(\texttt{Input})_{i',(2\iota-1,2\iota)} = \begin{cases} (\texttt{Input}_{i'}, i') & \text{if } i' \le n_{\text{in}},\ i' \equiv \iota \pmod s, \\ (0, i') & \text{otherwise.} \end{cases}$$

These definitions guarantee that large inner products only occur between machine queries

$Q(\texttt{Input})_i$ and tokens keys $K(\texttt{Input})_{i'}$ when $\texttt{Input}_{i'}$ is allocated to $\texttt{MachineIn}_i^{(1)}$. That is,

$$\begin{aligned} Q(\texttt{Input})_i^\mathsf{T} K(\texttt{Input})_{i'} &= 1, && \text{if } i' \in \texttt{Rcvd}_i \\ Q(\texttt{Input})_i^\mathsf{T} K(\texttt{Input})_{i'} &\leq 1 - \Omega\left(\frac{1}{M^2}\right), && \text{otherwise.} \end{aligned}$$

By applying Lemma 6.2 with $\xi = \Omega(\frac{1}{N^2})$, there exists some self-attention unit f' such that

$$f'(\texttt{Input})_i = \text{hardmax}(Q(\texttt{Input})K(\texttt{Input})^\mathsf{T}) = \frac{(\texttt{Input}_{i'}, i')_{i' \in \texttt{Rcvd}_i}}{|\texttt{Rcvd}_i|}.$$

A proper choice of ψ and an invocation of the definition of $\texttt{MachineIn}^{(1)}$ ensures that $\text{init}(\texttt{Input})_i = \psi(f(\texttt{Input}))_i = \texttt{MachineIn}_i^{(1)}$. □

Lemma 6.15. *For any R-round MPC protocol π with local memory s and q machines with n_{out}-word output, there exists a transformer* $\text{final} \in \mathsf{Transformer}^{q,\max(n_{\text{out}},q)}_{s,1,1,d_{\text{in}},d_{\text{out}}}$ *for $d_{\text{in}} = s$ and $d_{\text{out}} = 1$, which, given input $X = \texttt{MachineIn}^{(R)}$, has output* $\text{final}(X)$ *with* $\text{final}(X)_{i,1} = \texttt{Output}_i \in \mathbb{Z}_{2^p}$.

Proof. This argument inverts that of Lemma 6.13, after applying the $\textsc{Local}_R$ to transform $\texttt{MachineIn}^{(R)}$ to $\texttt{MachineOut}^{(R)}$. Let $Q, K, V : \mathbb{Z}_{2^p}^M \to \mathbb{R}^{M \times s}$ be the query, key, and value embeddings of the only attention unit f in final, and let $\psi : \mathbb{R}^{M \times s} \to \mathbb{Z}_{2^p}^s \times [N]$ be its output MLP. Let $q_{\text{out}} = \left\lceil \frac{n_{\text{out}}}{s} \right\rceil$ denote the number of machines storing relevant information for the output of the MPC protocol.

For each machine $i' \in [q_{\text{out}}]$, let

$$\texttt{Sent}_{i'} = \{(s-1)i' + 1, \ldots, \min(si', n_{\text{out}})\}$$

denote the set of all token indices receiving its output. Likewise, for each token index $i \in [n_{\text{out}}]$, let $\texttt{Src}_i = \lceil i/s \rceil$ be the machine containing its relevant token. We define $Q =$

$Q' \circ \textsc{Local}_R, K = K' \circ \textsc{Local}_R, V = V' \circ \textsc{Local}_R$ as follows.

$$Q'(\texttt{MachineOut}^{(R)})_{i,(2\iota-1,2\iota)} = \begin{cases} \left(\cos\left(\frac{2\pi\lfloor \texttt{Src}_i \rfloor}{M}\right), \sin\left(\frac{2\pi\lfloor \texttt{Src}_i \rfloor}{M}\right)\right) & \text{if } i \le n_{\text{out}},\ i \equiv \iota \pmod{s} \\ (0,0) & \text{otherwise.} \end{cases}$$

$$K'(\texttt{MachineOut}^{(R)})_{i'} = \left(\cos\left(\frac{2\pi i'}{M}\right), \sin\left(\frac{2\pi i'}{M}\right), \ldots, \cos\left(\frac{2\pi i'}{M}\right), \sin\left(\frac{2\pi i'}{M}\right)\right).$$

$$V'(\texttt{MachineOut}^{(R)})_{i'} = \texttt{MsgOut}^{(R)}_{i'}.$$

Applying Lemma 6.2 as before yields

$$f(\texttt{MachineIn}^{(R)})_i = \begin{cases} \texttt{MachineOut}^{(R)}_{i'} & \text{if } i \in \texttt{Sent}_{i'}, \\ 0 & \text{otherwise.} \end{cases}$$

A properly chosen ψ ensures that $\text{final}(\texttt{MachineIn}^{(R)})_i = \psi(f(\texttt{MachineIn}^{(R)}))_i = \texttt{Output}_i$.

□

6.7.4 Constructions for Section 6.4.3.1

Lemma 6.20. *For some $m \ge d+2$, $\tau : [N] \times \mathbb{R}^m \to [N]$, and $\rho : \mathbb{R}^m \to \mathbb{R}^d$, there exists an attention head $\text{lookUp}_{\tau,\rho} \in \mathsf{MaskAttn}^N_m$ with precision $p = O(\log N)$ and $m \ge d+2$ satisfying $\text{lookUp}_{\tau,\rho}(X)_{i,:d} = \rho(X_{\tau(i,X_i)})$.*

Proof. We let $V(X_i) = (\rho(X_i), \vec{0})$ and define sinusoidal embeddings Q and K with

$$Q(X)_i = \left(\cos\left(\frac{2\pi\tau(i,X_i)}{N}\right), \sin\left(\frac{2\pi\tau(i,X_i)}{N}\right), \vec{0}\right),$$
$$K(X)_i = \left(\cos\left(\frac{2\pi i}{N}\right), \sin\left(\frac{2\pi i)}{N}\right), \vec{0}\right).$$

Note that

$$\begin{aligned} Q(X)_i^\mathsf{T} K(X)_{i'} &= 1, && \text{if } \tau(i, X_i) = i', \\ Q(X)_i^\mathsf{T} K(X)_{i'} &\leq \cos\left(\frac{2\pi}{N}\right) = 1 - \Omega\left(\frac{1}{N^2}\right), && \text{otherwise.} \end{aligned}$$

By applying Lemma 6.2 with $\xi = \Omega(\frac{1}{N^2})$, we conclude that a satisfactory self-attention unit exists. □

Lemma 6.21. *For finite alphabet Σ, $m \geq d + 2$, $\mu_1, \mu_2 : \mathbb{R}^m \to \Sigma$, and $\rho : \mathbb{R}^m \to \mathbb{R}^d$, there exists an attention head* $\text{lastOccurrence}_{\mu,\rho} \in \mathsf{MaskAttn}_m^N$ *with precision $p = O(\log(N\,|\Sigma|))$ such that,*

$$\text{lastOccurrence}(X)_{i,:d} = \begin{cases} \rho(\vec{0}) & \text{if } \forall\ i' < i : \mu_1(X_{i'}) \neq \mu_2(X_i), \\ \rho(X_{i'}) & \text{if } i' = \max\left\{i' < i : \mu_1(X_{i'}) = \mu_2(X_i)\right\}. \end{cases}$$

Proof. Let $N' = N|\Sigma|$. We define token embeddings as follows, including start token "dummy embeddings" as discussed in Section 6.2.2.2.

$$\begin{aligned} Q(X)_i &= \left(\cos\left(\frac{2\pi(N\mu_2(X_i) + i)}{N|\Sigma|}\right), \sin\left(\frac{2\pi(N\mu_2(X_i) + i)}{N|\Sigma|}\right), 1, \vec{0}\right), \\ K(X)_i &= \left(\cos\left(\frac{2\pi(N\mu_1(X_i) + i)}{N|\Sigma|}\right), \sin\left(\frac{2\pi(N\mu_1(X_i) + i)}{N|\Sigma|}\right), 0, \vec{0}\right), \\ K(X)_0 &= \left(0, 0, \cos\left(\frac{2\pi(N - \frac{1}{2})}{N|\Sigma|}\right), \vec{0}\right), \\ V(X)_i &= (\rho(X_i), \vec{0}), \\ V(X)_0 &= \vec{0}. \end{aligned}$$

Taken together, these embeddings provide the following characterization of the inner prod-

ucts (with causal masking matrix Γ):

$$
\begin{aligned}
Q(X)_0^\intercal K(X)_{i'} + \Gamma_{i,i'} &= \cos\left(\frac{2\pi(i-i')}{N|\Sigma|}\right) && \text{if } i \geq i' > 0,\ \mu_1(X_{i'}) = \mu_2(X_i),\\
Q(X)_i^\intercal K(X)_{i'} + \Gamma_{i,i'} &\leq \cos\left(\frac{2\pi}{N}\right) && \text{if } i \geq i' > 0,\ \mu_1(X_{i'}) \neq \mu_2(X_i),\\
Q(X)_i^\intercal K(X)_{i'} + \Gamma_{i,i'} &= -\infty && \text{if } i < i',\\
Q(X)_i^\intercal K(X)_i + \Gamma_{i,0} &= \cos\left(\frac{2\pi(N-\frac{1}{2})}{N|\Sigma|}\right).
\end{aligned}
$$

As a result, the largest inner product $Q(X)_i^\intercal K(X)_{i'}$ for some i is the largest i' with $\mu_1(X_{i'}) = \mu_2(X_i)$ if one exists and $i' = 0$ otherwise. Furthermore, there exists a margin of $\Omega(\frac{1}{N^2|\Sigma|^2})$ between this inner product and all others. We conclude by applying Lemma 6.2. □

6.8 Conclusion and future work

This work highlights parallelism as a central feature of transformers that sets them apart from other neural architectures. The focus on log-depth and sublinear-width transformers applied to specific computational tasks accentuates the benefits of parallelism, even for tasks like k-hop that appear inherently serial at first glance. There is some efficiency loss in the "compilation" of MPC protocols to transformers that we hope to understand better in future work. Furthermore, although we have empirically demonstrated the learnability of transformers that exploit parallelism in crucial ways, a theoretical understanding of learning such solutions remains an open question.

As discussed previously, this work is a direct follow-up to the previous chapter, which extends the communication complexity lens on transformers to a variable-depth regime. In doing so, these results suggest that modeling a transformer as a restricted multi-round communication protocol between tokens provides insight into the strengths and limitations of the architecture. By establishing that transformers can simulate parallelizable algorithms, while alternative architectures are akin to a bounded-size blackboard model, we apply this

communication lens to quantify the advantages of the transformer over state-space models and sub-quadratic-attention models. The tasks under consideration are *compositional* in nature, and the results suggest that the transformer's ability to exploit parallelism is crucial for efficiently learning such tasks. The empirical results provide evidence that these representational benefits are realizable by practical learning algorithms and that this "pointer-passing" primitive may be a key subroutine of trained transformers.

Epilogue

Throughout this dissertation, we have explored the representational capabilities of neural networks and used a wide range of theoretical tools to derive sharp separations between design choices. Beyond the worst-case framing of the universal approximation theorem, we have developed a more precise and prescriptive understanding of the fundamental limitations of neural architectures.

Underlying this dissertation—and the field of neural network theory writ large—is a fundamental tension between the principled rigor of theoretical computer science and the shifting landscape of empirical machine learning. The former demands abstractions and generalizations that are often too coarse to capture the complexities of modern deep learning, which leaves the study of state-of-the-art neural networks to practitioners and empirical researchers. The author's Ph.D. research has aimed to bridge that gap, and the body of work herein is a testament to the author's struggle to find a middle ground between these two fields.

The works in this dissertation attempt to provide a beyond-worst-case formulation of neural networks that incorporates architectural complexities and scaling regimes of practical interest; however, the focus on approximation and expressivity leaves numerous research questions unanswered. While negative representational results provide a hard limitation on the capabilities of neural architectures, *positive* representational results leave open the question of whether the target function can be learned by a practical algorithm with a feasible number of samples. Indeed, the author's "mid-Ph.D." body of work includes several papers

(Ardeshir, Sanford, and Hsu, 2021; Bietti, Bruna, Sanford, and Song, 2022; Chatziafratis, Panageas, Sanford, and Stavroulakis, 2022) excluded from the thesis that focuses more on optimization and generalization, as an attempt to move beyond the representational focus of the works herein. However, a rigorous analysis of gradient descent and generalization have proven elusive for all but the simplest of settings, which prompted the author to return to the representational focus for the final years of his Ph.D.

Unlike the earlier chapters of the dissertation, the final two chapters are inspired by the rapid innovation in the transformer architecture. Numerous variants of and alternatives to the transformer have been proposed in recent years, and the motivation of the author's work on transformers is to answer concrete questions about how to decide between these architectures, which benchmark tasks can measure their success, and how to adapt these architectures to new tasks. In the context of transformer architecture research in early 2024, these questions pertain to the effectiveness of various sub-quadratic attention mechanisms, the learnability of compositional tasks, and whether state-space models are a viable alternative to transformers. In particular, the final chapter was inspired by extensive experimentation on toy tasks, which clarified that the studied targets may be not only representable but also learnable by transformers.

As the theory of transformers matures, their primary theoretical research focus may shift from representation to optimization and generalization, just as was the case for feed-forward neural networks. The author hopes that the mathematical connections drawn in this dissertation will provide a foundation for future research. However, as long as the space of neural architectures is being explored, novel representational results will remain relevant and informative to theoreticians and practitioners alike. Indeed, the core contribution of representational results since the XOR construction of Minsky and Papert (1969) has been to distill in as simple a form as possible the differences in capacities of different architectures and to use these differences to guide the design of new architectures. The targets developed in this work—the sinusoidal single-index model, the iterated logistic mapping, the parity

dataset, three-wise matching, and the k-hop compositionality problem—are all inspired by this tradition, and the author hopes that they will inspire future work in the same vein.

References

Aamand, Anders et al. (2022). "Exponentially Improving the Complexity of Simulating the Weisfeiler-Lehman Test with Graph Neural Networks". In: *Advances in Neural Information Processing Systems 35*.

Abbe, Emmanuel and Colin Sandon (2020). "Poly-time universality and limitations of deep learning". In: *arXiv preprint arXiv:2001.02992*. arXiv: 2001.02992 [cs.LG].

Agarwal, Alekh et al. (2014). "A reliable effective terascale linear learning system". In: *Journal of Machine Learning Research* 15.1, pp. 1111–1133.

Alman, Josh and Zhao Song (2023). "How to Capture Higher-order Correlations? Generalizing Matrix Softmax Attention to Kronecker Computation". In: *CoRR* abs/2310.04064. arXiv: 2310.04064.

Alsedà, Lluís, Jaume Llibre, and Michal Misiurewicz (2000). *Combinatorial Dynamics and Entropy in Dimension One*. 2nd. WORLD SCIENTIFIC. eprint: https://www.worldscientific.com/doi/pdf/10.1142/4205.

Andoni, Alexandr et al. (2014a). "Learning Polynomials with Neural Networks". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, pp. II–1908–II–1916.

Andoni, Alexandr et al. (2014b). "Parallel algorithms for geometric graph problems". In: *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 574–583.

Andoni, Alexandr et al. (Oct. 2018). "Parallel Graph Connectivity in Log Diameter Rounds". In: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE.

Angluin, Dana (1980). "Local and global properties in networks of processors". In: *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*.

Angluin, Dana, David Chiang, and Andy Yang (2023). *Masked Hard-Attention Transformers and Boolean RASP Recognize Exactly the Star-Free Languages*. arXiv: 2310.13897 [cs.FL].

Anthony, Martin and Peter L Bartlett (1999). *Neural network learning: Theoretical foundations*. Vol. 9. cambridge university press Cambridge.

Ardeshir, Navid, Daniel J. Hsu, and Clayton Hendrick Sanford (2023). "Intrinsic dimensionality and generalization properties of the R-norm inductive bias". In: *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*. Ed. by Gergely Neu and Lorenzo Rosasco. Vol. 195. Proceedings of Machine Learning Research. PMLR, pp. 3264–3303.

Ardeshir, Navid, Clayton Sanford, and Daniel J. Hsu (2021). "Support vector machines and linear regression coincide with very high-dimensional features". In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc'Aurelio Ranzato et al., pp. 4907–4918.

Arora, Raman et al. (2016). "Understanding deep neural networks with rectified linear units". In: *arXiv preprint arXiv:1611.01491*.

Assadi, Sepehr and Vishvajeet N (June 2021). "Graph streaming lower bounds for parameter estimation and property testing via a streaming XOR lemma". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. STOC '21. ACM.

Bach, Francis (2017). "Breaking the curse of dimensionality with convex neural networks". In: *Journal of Machine Learning Research* 18.1, pp. 629–681. arXiv: 1412.8690 [cs.LG].

Bach, Francis and Lenaïc Chizat (2021). "Gradient Descent on Infinitely Wide Neural Networks: Global Convergence and Generalization". In: *arXiv preprint arXiv:2110.08084*.

Baldi, Pierre and Peter J Sadowski (2013). "Understanding dropout". In: *Advances in Neural Information Processing Systems 26*.

Barak, Boaz et al. (2022). *Hidden Progress in Deep Learning: SGD Learns Parities Near the Computational Limit*.

Barron, Andrew R (1993). "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3, pp. 930–945.

Bartlett, Peter L (1996). "For valid generalization the size of the weights is more important than the size of the network". In: *Advances in Neural Information Processing Systems 9*.

Bartlett, Peter L. et al. (2019). "Benign Overfitting in Linear Regression". In: *CoRR* abs/1906.11300. arXiv: 1906.11300.

Bauer, Benedikt and Michael Kohler (2019). "On deep learning as a remedy for the curse of dimensionality in nonparametric regression". In: *The Annals of Statistics* 47.4, pp. 2261–2285.

Beame, Paul, Paraschos Koutris, and Dan Suciu (2017). "Communication steps for parallel query processing". In: *Journal of the ACM (JACM)* 64.6, pp. 1–58.

Behnezhad, Soheil et al. (2019). "Massively parallel computation of matching and MIS in sparse graphs". In: *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pp. 481–490.

Belkin, Mikhail et al. (2018). "Reconciling modern machine learning and the bias-variance trade-off". In: *CoRR* abs/1812.11118. arXiv: 1812.11118.

Bellman, Richard (1944). "Almost orthogonal series". In: *Bulletin of the American Mathematical Society* 50, pp. 517–519.

Beltagy, Iz, Matthew E. Peters, and Arman Cohan (2020). *Longformer: The Long-Document Transformer*. arXiv: 2004.05150 [cs.CL].

Ben-David, Shai, Nadav Eiron, and Hans Ulrich Simon (2002). "Limitations of learning via embeddings in Euclidean half spaces". In: *Journal of Machine Learning Research* 3.Nov, pp. 441–461.

Bengio, Y., P. Simard, and P. Frasconi (1994). "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2, pp. 157–166.

Bhattamishra, Satwik, Kabir Ahuja, and Navin Goyal (2020). "On the Ability and Limitations of Transformers to Recognize Formal Languages". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Bhattamishra, Satwik et al. (2022). "Simplicity Bias in Transformers and their Ability to Learn Sparse Boolean Functions". In: *arXiv preprint arXiv:2211.12316*.

Bhojanapalli, Srinadh, Behnam Neyshabur, and Nati Srebro (2016). "Global optimality of local search for low rank matrix recovery". In: *Advances in Neural Information Processing Systems* 29.

Bietti, Alberto et al. (2022). "Learning Single-Index Models with Shallow Neural Networks". In: *arXiv preprint arXiv:2210.15651*.

Bietti, Alberto et al. (2023). *Birth of a Transformer: A Memory Viewpoint*. arXiv: 2306.00802 [stat.ML].

Boas, R. P. jun. (1941). "A general moment problem." In: *American Journal of Mathematics* 63, pp. 361–370.

Boucheron, Stéphane, Gábor Lugosi, and Pascal Massart (2013). *Concentration Inequalities - A Nonasymptotic Theory of Independence*. Oxford University Press.

Bresler, Guy and Dheeraj Nagaraj (2020). *Sharp Representation Theorems for ReLU Networks with Precise Dependence on Depth*. arXiv: 2006.04048 [stat.ML].

Brown, Tom B. et al. (2020). "Language Models are Few-Shot Learners". In: *arXiv preprint arXiv:2005.14165*.

Bu, Kaifeng, Yaobo Zhang, and Qingxian Luo (2020). *Depth-Width Trade-offs for Neural Networks via Topological Entropy*. arXiv: 2010.07587 [cs.LG].

Bubeck, Sébastien, Yuanzhi Li, and Dheeraj M Nagaraj (2021). "A law of robustness for two-layers neural networks". In: *Conference on Learning Theory*.

Candès, Emmanuel J. (1999). "Harmonic analysis of neural networks". In: *Applied and Computational Harmonic Analysis* 6.2, pp. 197–218.

Candès, Emmanuel J and Benjamin Recht (2009). "Exact Matrix Completion via Convex Optimization". In: *Foundations of Computational Mathematics* 9.6, pp. 717–772.

Candès, Emmanuel J, Justin Romberg, and Terence Tao (2006). "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". In: *IEEE Transactions on information theory* 52.2, pp. 489–509.

Candes, Emmanuel J and Terence Tao (2005). "Decoding by linear programming". In: *IEEE transactions on information theory* 51.12, pp. 4203–4215.

Charikar, Moses, Weiyun Ma, and Li-Yang Tan (2020). *New lower bounds for Massively Parallel Computation from query complexity*. arXiv: 2001.01146 [cs.DS].

Chatziafratis, Vaggos, Sai Ganesh Nagarajan, and Ioannis Panageas (2020). "Better depth-width trade-offs for neural networks through the lens of dynamical systems". In: *International Conference on Machine Learning*. PMLR, pp. 1469–1478.

Chatziafratis, Vaggos et al. (2019). "Depth-width trade-offs for relu networks via sharkovsky's theorem". In: *arXiv preprint arXiv:1912.04378*. arXiv: 1912.04378 [cs.LG].

Chatziafratis, Vaggos et al. (2022). "On Scrambling Phenomena for Randomly Initialized Recurrent Networks". In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. Ed. by Sanmi Koyejo et al.

Chen, Nuo et al. (2022). "CAT-probing: A Metric-based Approach to Interpret How Pretrained Models for Programming Language Attend Code Structure". In: *arXiv preprint arXiv:2210.04633*.

Chen, Zhengdao et al. (2019). "On the equivalence between graph isomorphism testing and function approximation with GNNs". In: *Advances in Neural Information Processing Systems 32*.

Cho, Youngmin and Lawrence K. Saul (2009). "Kernel Methods for Deep Learning". In: *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*. Ed. by Yoshua Bengio et al. Curran Associates, Inc., pp. 342–350.

Choromanski, Krzysztof et al. (2022). *Rethinking Attention with Performers*. arXiv: 2009.14794 [cs.LG].

Chung, Junyoung et al. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555*.

Clark, Kevin et al. (2019). "What does bert look at? an analysis of bert's attention". In: *arXiv preprint arXiv:1906.04341*.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine Learning* 20.3, pp. 273–297.

Coy, Sam and Artur Czumaj (2022). "Deterministic Massively Parallel Connectivity". In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2022. Rome, Italy: Association for Computing Machinery, 162–175. ISBN: 9781450392648.

Cybenko, G. (Dec. 1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4, pp. 303–314.

Damian, Alexandru, Jason Lee, and Mahdi Soltanolkotabi (2022). "Neural networks can learn representations with gradient descent". In: *Conference on Learning Theory*.

Daniely, Amit (July 2017a). "Depth Separation for Neural Networks". In: *Proceedings of the 2017 Conference on Learning Theory*. Ed. by Satyen Kale and Ohad Shamir. Vol. 65. Proceedings of Machine Learning Research. PMLR, pp. 690–696.

— (2017b). "SGD Learns the Conjugate Kernel Class of the Network". In.

Daniely, Amit and Eran Malach (2020). "Learning Parities with Neural Networks". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al.

Dao, Tri et al. (2022). "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness". In: *NeurIPS*.

Dean, Jeffrey and Sanjay Ghemawat (2004). "MapReduce: Simplified Data Processing on Large Clusters". In: *OSDI*, pp. 137–150.

Debarre, Thomas et al. (2022). "Sparsest piecewise-linear regression of one-dimensional data". In: *Journal of Computational and Applied Mathematics* 406, p. 114044.

Dettmers, Tim et al. (2022). "LLM.int8(): 8-bit matrix multiplication for transformers at scale". In: *Advances in Neural Information Processing Systems*. Vol. 35.

Donoho, David L (2006). "Compressed sensing". In: *IEEE Transactions on Information Theory* 52.4, pp. 1289–1306.

Dosovitskiy, Alexey et al. (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *arXiv preprint arXiv:2010.11929*. arXiv: `2010.11929 [cs.CV]`.

Duris, Pavol, Zvi Galil, and Georg Schnitger (1984). "Lower bounds on communication complexity". In: *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, 81–91.

Dym, H. and H. P. McKean (1972). *Fourier series and integrals*. Probability and Mathematical Statistics. Vol. 14. New York-London: Academic Press. X,295 p. $ 18.50 (1972).

E, Weinan, Chao Ma, and Lei Wu (2019). "The Barron Space and the Flow-induced Function Spaces for Neural Network Models". In: *arXiv preprint arXiv:1906.08039*.

Edelman, Benjamin L. et al. (2022). "Inductive Biases and Variable Creation in Self-Attention Mechanisms". In: *International Conference on Machine Learning*.

Eldan, Ronen and Ohad Shamir (June 2016). "The Power of Depth for Feedforward Neural Networks". In: *CoRR*. Proceedings of Machine Learning Research abs/1512.03965. Ed. by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, pp. 907–940. arXiv: `1512.03965`.

Elhage, Nelson et al. (2021). "A Mathematical Framework for Transformer Circuits". In: *Transformer Circuits Thread*. https://transformer-circuits.pub/2021/framework/index.html.

Ergen, Tolga and Mert Pilanci (2021). "Convex geometry and duality of over-parameterized neural networks". In: *Journal of Machine Learning Research* 22.212, pp. 1–63.

Fischer, Paul and Hans-Ulrich Simon (1992). "On learning ring-sum-expansions". In: *SIAM Journal on Computing* 21.1, pp. 181–192.

Folland, Gerald B. (1999). *Real analysis. Modern techniques and their applications.* 2nd ed. Pure Appl. Math., Wiley-Intersci. Ser. Texts Monogr. Tracts. New York, NY: Wiley. ISBN: 0-471-31716-0.

Frei, Spencer, Niladri S Chatterji, and Peter L Bartlett (2022). "Random feature amplification: Feature learning and generalization in neural networks". In: *arXiv preprint arXiv:2202.07626*.

Funahashi, Ken-ichi (1989). "On the approximate realization of continuous mappings by neural networks". In: *Neural Networks* 2.3, pp. 183–192.

Furst, Merrick, James B Saxe, and Michael Sipser (1984). "Parity, circuits, and the polynomial-time hierarchy". In: *Mathematical systems theory* 17.1, pp. 13–27.

Gal, Yarin and Zoubin Ghahramani (2016). "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning". In: *International Conference on Machine Learning*.

Galanti, Tomer et al. (2022). "SGD and Weight Decay Provably Induce a Low-Rank Bias in Neural Networks". In: *arXiv preprint arXiv:2206.05794*.

Gale, David (1963). "Neighborly and cyclic polytopes". In: *Proc. Sympos. Pure Math.* Vol. 7, pp. 225–232.

Ghaffari, Mohsen, Fabian Kuhn, and Jara Uitto (Nov. 2019). "Conditional Hardness Results for Massively Parallel Computation from Distributed Lower Bounds". In: *IEEE 60th Annual Symposium on Foundations of Computer Science*, pp. 1650–1663.

Goodrich, Michael T, Nodari Sitchinava, and Qin Zhang (2011). "Sorting, searching, and simulation in the mapreduce framework". In: *International Symposium on Algorithms and Computation*. Springer, pp. 374–383.

Gu, Albert and Tri Dao (2023). *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. arXiv: `2312.00752 [cs.LG]`.

Guha, Sudipto and Andrew McGregor (2009). "Stream Order and Order Statistics: Quantile Estimation in Random-Order Streams". In: *SIAM Journal on Computing* 38.5, pp. 2044–2059. eprint: `https://doi.org/10.1137/07069328X`.

Györfi, László et al. (2002). *A distribution-free theory of nonparametric regression*. Vol. 1. Springer.

Hahn, Michael (2020). "Theoretical Limitations of Self-Attention in Neural Sequence Models". In: *Trans. Assoc. Comput. Linguistics* 8, pp. 156–171.

Hanin, Boris (2021). "Ridgeless Interpolation with Shallow ReLU Networks in $1D$ is Nearest Neighbor Curvature Extrapolation and Provably Generalizes on Lipschitz Functions". In: *arXiv preprint arXiv:2109.12960*.

Hanin, Boris and David Rolnick (2019). "Deep relu networks have surprisingly few activation patterns". In: *Advances in Neural Information Processing Systems*, pp. 359–368.

Hanson, Stephen and Lorien Pratt (1988). "Comparing biases for minimal network construction with back-propagation". In: *Advances in Neural Information Processing Systems 1*.

Hao, Yiding, Dana Angluin, and Robert Frank (2022). "Formal Language Recognition by Hard Attention Transformers: Perspectives from Circuit Complexity". In: *Trans. Assoc. Comput. Linguistics* 10, pp. 800–810.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Helmbold, David, Robert Sloan, and Manfred K Warmuth (1992). "Learning integer lattices". In: *SIAM Journal on Computing* 21.2, pp. 240–266.

Hewitt, John and Christopher D Manning (2019). "A structural probe for finding syntax in word representations". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Hinton, Geoffrey E (1987). "Learning translation invariant recognition in a massively parallel networks". In: *International Conference on Parallel Architectures and Languages Europe*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation* 9, pp. 1735–1780.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (July 1989). "Multilayer Feedforward Networks Are Universal Approximators". In: *Neural Netw.* 2.5, pp. 359–366.

Hsu, Daniel et al. (2021). "On the Approximation Power of Two-Layer Networks of Random ReLUs". In: *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*. Ed. by Mikhail Belkin and Samory Kpotufe. Vol. 134. Proceedings of Machine Learning Research. PMLR, pp. 2423–2461.

Im, Sungjin et al. (2023). "Massively Parallel Computation: Algorithms and Applications". In: *Foundations and Trends® in Optimization* 5.4, pp. 340–417.

Jacot, Arthur, Franck Gabriel, and Clément Hongler (2018). "Neural Tangent Kernel: Convergence and Generalization in Neural Networks". In: *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., pp. 8580–8589.

Ji, Ziwei, Matus Telgarsky, and Ruicheng Xian (2019). *Neural tangent kernels, transportation mappings, and universal approximation*. arXiv: 1910.06956 [cs.LG].

Jin, Hui and Guido Montúfar (2020). "Implicit bias of gradient descent for mean squared error regression with wide neural networks". In: *arXiv preprint arXiv:2006.07356*.

Jumper, John et al. (2021). "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873, pp. 583–589.

Kacham, Praneeth, Vahab Mirrokni, and Peilin Zhong (2023). *PolySketchFormer: Fast Transformers via Sketches for Polynomial Kernels*. arXiv: 2310.01655 [cs.LG].

Kakade, Sham M, Karthik Sridharan, and Ambuj Tewari (2008). "On the complexity of linear prediction: Risk bounds, margin bounds, and regularization". In: *Advances in Neural Information Processing Systems 21*.

Kamath, Pritish, Omar Montasser, and Nathan Srebro (2020). "Approximate is good enough: Probabilistic variants of dimensional and margin complexity". In: *Conference on Learning Theory*. arXiv: 2003.04180 [cs.LG].

Karchmer, Mauricio and Avi Wigderson (1988). "Monotone circuits for connectivity require super-logarithmic depth". In: *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*.

Karloff, Howard, Siddharth Suri, and Sergei Vassilvitskii (Dec. 2010). "A Model of Computation for MapReduce". In: *Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 938–948.

Keriven, Nicolas and Gabriel Peyré (2019). "Universal invariant and equivariant graph neural networks". In: *Advances in Neural Information Processing Systems 32*.

Kileel, Joe, Matthew Trager, and Joan Bruna (2019). "On the expressive power of deep polynomial neural networks". In: *Advances in Neural Information Processing Systems*, pp. 10310–10319.

Kim, Jinwoo et al. (2022). *Pure Transformers are Powerful Graph Learners*. arXiv: 2207.02505 [cs.LG].

Kimeldorf, George and Grace Wahba (1971). "Some results on Tchebycheffian spline functions". In: *Journal of mathematical analysis and applications* 33.1, pp. 82–95.

Kingma, Diederik P. and Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG].

Klusowski, Jason M and Andrew R Barron (2016). "Risk bounds for high-dimensional ridge function combinations including neural networks". In: *arXiv preprint arXiv:1607.01434*.

Klusowski, Jason M. and Andrew R. Barron (Dec. 2018). "Approximation by Combinations of ReLU and Squared ReLU Ridge Functions With L1 and L0 Controls". In: *IEEE Transactions on Information Theory* 64.12.

Kohler, Michael and Adam Krzyżak (2005). "Adaptive regression estimation with multilayer feedforward neural networks". In: *Nonparametric Statistics* 17.8, pp. 891–913.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012a). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. Ed. by Peter L. Bartlett et al., pp. 1106–1114.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012b). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25, pp. 1097–1105.

Kurková, Vera and Marcello Sanguineti (2001). "Bounds on rates of variable-basis and neural-network approximation". In: *IEEE Transactions on Information Theory* 47.6, pp. 2659–2665.

Lecun, Y. et al. (Nov. 1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86 (11), pp. 2278–2324.

Lee, Holden et al. (2017). "On the ability of neural nets to express distributions". In: *Conference on Learning Theory*. PMLR, pp. 1271–1296.

Lefkowitz, Melanie (Sept. 2019). "Professor's perceptron paved the way for AI – 60 years too soon". In: *Cornell Chronicle*.

Leoni, Giovanni (2017). *A first course in Sobolev spaces*. 2nd edition. Vol. 181. Grad. Stud. Math. Providence, RI: American Mathematical Society (AMS). ISBN: 978-1-4704-2921-8; 978-1-4704-4226-2.

Li, Husheng (2018). *Analysis on the Nonlinear Dynamics of Deep Neural Networks: Topological Entropy and Chaos*. arXiv: `1804.03987 [cs.LG]`.

Li, Tien-Yien and James A Yorke (1975). "Period three implies chaos". In: *The American Mathematical Monthly* 82.10, pp. 985–992.

Li, Yuxuan and James L. McClelland (2022). *Systematic Generalization and Emergent Structures in Transformers Trained on Structured Tasks*. arXiv: `2210.00400 [cs.LG]`.

Likhosherstov, Valerii, Krzysztof Choromanski, and Adrian Weller (2021). "On the expressive power of self-attention matrices". In: *arXiv preprint arXiv:2106.03764*.

Liu, Bingbin et al. (2022). *Transformers Learn Shortcuts to Automata*. arXiv: 2210.10749 [cs.LG].

Loukas, Andreas (2019). "What graph neural networks cannot learn: depth vs width". In: *arXiv preprint arXiv:1907.03199*.

Maennel, Hartmut, Olivier Bousquet, and Sylvain Gelly (2018). "Gradient descent quantizes ReLU network features". In: *arXiv preprint arXiv:1803.08367*.

Maiorov, V.E (1999). "On Best Approximation by Ridge Functions". In: *Journal of Approximation Theory* 99.1, pp. 68 –94.

Malach, Eran (2023). *Auto-Regressive Next-Token Predictors are Universal Learners*. arXiv: 2309.06979 [cs.LG].

Malach, Eran and Shai Shalev-Shwartz (2019). "Is Deeper Better only when Shallow is Good?" In: *arXiv preprint arXiv:1903.03488* abs/1903.03488. arXiv: 1903.03488.

Malach, Eran et al. (2021a). "Quantifying the Benefit of Using Differentiable Learning over Tangent Kernels". In: *arXiv preprint arXiv:2103.01210*.

Malach, Eran et al. (2021b). "The Connection Between Approximation, Depth Separation and Learnability in Neural Networks". In: *arXiv preprint 2102.00434*.

Maron, Haggai et al. (2019). "On the universality of invariant networks". In: *International Conference on Machine Learning*.

Martens, James et al. (2013). "On the representational efficiency of restricted boltzmann machines". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., pp. 2877–2885.

McCulloch, Warren S. and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The Bulletin of Mathematical Biophysics* 5, pp. 115–133.

Meir, Ron and Tong Zhang (2003). "Generalization error bounds for Bayesian mixture algorithms". In: *Journal of Machine Learning Research* 4.Oct, pp. 839–860.

Mendelson, Shahar, Alain Pajor, and Nicole Tomczak-Jaegermann (2007). "Reconstruction and subgaussian operators in asymptotic geometric analysis". In: *Geometric and Functional Analysis* 17.4, pp. 1248–1282.

Merrill, William and Ashish Sabharwal (2022). *A Logic for Expressing Log-Precision Transformers*. arXiv: 2210.02671 [cs.LG].

Merrill, William and Ashish Sabharwal (2023a). *The Expressive Power of Transformers with Chain of Thought*. arXiv: 2310.07923 [cs.LG].

— (2023b). "The Parallelism Tradeoff: Limitations of Log-Precision Transformers". In: *Transactions of the Association for Computational Linguistics* 11, 531–545.

Merrill, William, Ashish Sabharwal, and Noah A. Smith (2022). "Saturated Transformers are Constant-Depth Threshold Circuits". In: *Transactions of the Association for Computational Linguistics* 10, 843–856.

Metropolis, N, M.L Stein, and P.R Stein (1973). "On finite limit sets for transformations on the unit interval". In: *Journal of Combinatorial Theory, Series A* 15.1, pp. 25 –44.

Mhaskar, Hrushikesh Narhar (2004). "On the tractability of multivariate integration and approximation by neural networks". In: *Journal of Complexity* 20.4, pp. 561–590.

Minsky, Marvin and Seymour A Papert (1969). *Perceptrons: An introduction to computational geometry*. MIT press.

Misiurewicz, Michal and Wieslaw Szlenk (1980). "Entropy of piecewise monotone mappings". In: *Studia Mathematica* 67, pp. 45–63.

Mitzenmacher, Michael and Eli Upfal (2017). *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge University Press.

Montufar, Guido F et al. (2014). "On the number of linear regions of deep neural networks". In: *Advances in neural information processing systems*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., pp. 2924–2932.

Morris, Christopher et al. (2019). "Weisfeiler and leman go neural: Higher-order graph neural networks". In: *AAAI Conference on Artificial Intelligence*.

Mousavi-Hosseini, Alireza et al. (2022). "Neural Networks Efficiently Learn Low-Dimensional Representations with SGD". In: *arXiv preprint arXiv:2209.14863*.

MPICH (2023). *MPI Allreduce*.

Murata, Noboru (1996). "An Integral Representation of Functions Using Three-layered Networks and Their Approximation Bounds". In: *Neural Networks* 9.6, pp. 947–956.

Neal, Radford M. (1996). *Bayesian learning for neural networks*. Vol. 118. Lect. Notes Stat. New York, NY: Springer. ISBN: 0-387-94724-8.

Neyshabur, Behnam, Ryota Tomioka, and Nathan Srebro (2015). "In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning." In: *ICLR Workshop*.

Nisan, Noam and Avi Wigderson (1993). "Rounds in Communication Complexity Revisited". In: *SIAM Journal on Computing* 22.1, pp. 211–219. eprint: `https://doi.org/10.1137/0222016`.

Olson, Matthew, Abraham Wyner, and Richard Berk (2018). "Modern neural networks generalize on small data sets". In: *Advances in Neural Information Processing Systems 31*.

O'Neil, Patrick E. (1971). "Hyperplane cuts of an n-cube". In: *Discrete Mathematics* 1.2, pp. 193–195.

Ongie, Greg et al. (2019). "A Function Space View of Bounded Norm Infinite Width ReLU Nets: The Multivariate Case". In: *International Conference on Learning Representations*. arXiv: `1910.01635 [cs.LG]`.

OpenAI (2023). *GPT-4 Technical Report*. arXiv: `2303.08774 [cs.CL]`.

Oren, Matanel et al. (2024). *Transformers are Multi-State RNNs*. arXiv: `2401.06104 [cs.CL]`.

Papadimitriou, Christos H. and Michael Sipser (1982). "Communication complexity". In: *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, 196–200.

Parhi, Rahul and Robert D Nowak (2021a). "Banach Space Representer Theorems for Neural Networks and Ridge Splines." In: *Journal of Machine Learning Research* 22.43, pp. 1–40.

— (2021b). "Near-Minimax Optimal Estimation With Shallow ReLU Neural Networks". In: *arXiv preprint arXiv:2109.08844*.

Peleg, David (2000). *Distributed computing: a locality-sensitive approach*. SIAM.

Pérez, Jorge, Pablo Barceló, and Javier Marinkovic (2021). "Attention is turing complete". In: *Journal of Machine Learning Research* 22.1, pp. 3463–3497.

Pérez, Jorge, Javier Marinković, and Pablo Barceló (2019). "On the turing completeness of modern neural network architectures". In: *arXiv preprint arXiv:1901.03429*.

Pinkus, Allan (1999). "Approximation theory of the MLP model in neural networks". In: *Acta Numerica Vol. 8, 1999*. Cambridge: Cambridge University Press, pp. 143–195. ISBN: 0-521-77088-2.

Poole, Ben et al. (June 2016). "Exponential expressivity in deep neural networks through transient chaos". In: *arXiv e-prints*, arXiv:1606.05340, arXiv:1606.05340. arXiv: 1606.05340 [stat.ML].

Qi, Charles R et al. (2017). "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Radford, Alec et al. (2019). "Language Models are Unsupervised Multitask Learners". In: *OpenAI blog* 1.8, p. 9.

Raghu, Maithra et al. (Aug. 2017). "On the expressive power of deep neural networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. JMLR. org. International Convention Centre, Sydney, Australia: PMLR, pp. 2847–2854.

Rahimi, Ali and Benjamin Recht (2008). "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems 20*. Ed. by J. C. Platt et al. Curran Associates, Inc., pp. 1177–1184.

— (2009). "Weighted Sums of Random Kitchen Sinks: Replacing minimization with randomization in learning". In: *Advances in Neural Information Processing Systems 21*. Ed. by D. Koller et al. Curran Associates, Inc., pp. 1313–1320.

Rogers, Anna, Olga Kovaleva, and Anna Rumshisky (Dec. 2020). "A Primer in BERTology: What We Know About How BERT Works". In: *Transactions of the Association for Computational Linguistics* 8, 842–866.

Rosenblatt, Frank (1958). "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65 6, pp. 386–408.

Rosser, Barkley (1941). "Explicit Bounds for Some Functions of Prime Numbers". In: *American Journal of Mathematics* 63.1, pp. 211–232.

Rosset, Saharon et al. (2007). "ℓ_1 regularization in infinite dimensional feature spaces". In: *Conference on Learning Theory*.

Roughgarden, Tim, Sergei Vassilvitskii, and Joshua Wang (Nov. 2018). "Shuffles and Circuits (On Lower Bounds for Modern Parallel Computation)". In: *Journal of the ACM* 65, pp. 1–24.

Roweis, Sam and Lawrence Saul (Dec. 2000). "Nonlinear Dimensionality Reduction by Locally Linear Embedding". In: *Science* 290.5500, pp. 2323–2326.

Rubin, Boris (1998). "The Calderón reproducing formula, windowed X-ray transforms, and Radon transforms in L^p-spaces". In: *The Journal of Fourier Analysis and Applications* 4.2, pp. 175–197.

Rudin, Walter (1987). *Real and complex analysis.* 3rd ed. New York, NY: McGraw-Hill. ISBN: 0-07-054234-1.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams (1986). "Learning representations by back-propagating errors". In: *Nature, London* 323.6088, pp. 533–536.

Safran, Itay, Ronen Eldan, and Ohad Shamir (2019). "Depth separations in neural networks: what is actually being separated?" In: *Conference on Learning Theory*. PMLR, pp. 2664–2666.

Safran, Itay and Ohad Shamir (2017). "Depth-Width Tradeoffs in Approximating Natural Functions with Neural Networks". In: *International Conference on Machine Learning*. arXiv: 1610.09887 [cs.LG].

Sanford, Clayton, Daniel Hsu, and Matus Telgarsky (2023). *Representational Strengths and Limitations of Transformers*. arXiv: 2306.02896 [cs.LG].

— (2024). "Transformers, parallel computation, and logarithmic depth". In: *CoRR* abs/2402.09268. arXiv: 2402.09268.

Sanford, Clayton Hendrick and Vaggos Chatziafratis (2022). "Expressivity of Neural Networks via Chaotic Itineraries beyond Sharkovsky's Theorem". In: *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*. Ed. by Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera. Vol. 151. Proceedings of Machine Learning Research. PMLR, pp. 9505–9549.

Santoro, Adam et al. (2017). "A simple neural network module for relational reasoning". In: *Advances in Neural Information Processing Systems 30*.

Sauer, Norbert (1972). "On the density of families of sets". In: *Journal of Combinatorial Theory, Series A* 13.1, pp. 145–147.

Savarese, Pedro et al. (2019). "How do infinite width bounded norm networks look in function space?" In: *Conference on Learning Theory*.

Schmidt-Hieber, Johannes (2020). "Nonparametric regression using deep neural networks with ReLU activation function". In: *The Annals of Statistics* 48.4, pp. 1875–1897.

Schmitt, Michael (2000). "Lower bounds on the complexity of approximating continuous functions by sigmoidal neural networks". In: *Advances in neural information processing systems*, pp. 328–334.

Sharkovsky, OM (1964). "Coexistence of the cycles of a continuous mapping of the line into itself". In: *Ukrainskij matematicheskij zhurnal* 16.01, pp. 61–71.

— (1965). "On cycles and structure of continuous mapping". In: *Ukrainskij matematicheskij zhurnal* 17.03, pp. 104–111.

Shelah, Saharon (1972). "A combinatorial problem; stability and order for models and theories in infinitary languages". In: *Pacific Journal of Mathematics* 41.1, pp. 247–261.

Shevchenko, Alexander, Vyacheslav Kungurtsev, and Marco Mondelli (2021). "Mean-field Analysis of Piecewise Linear Solutions for Wide ReLU Networks". In: *arXiv preprint arXiv:2111.02278*.

Siegel, Jonathan W and Jinchao Xu (2021). "Characterization of the variation spaces corresponding to shallow neural networks". In: *arXiv preprint arXiv:2106.15002*.

Silver, David et al. (2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nat.* 529.7587, pp. 484–489.

Sonoda, Sho et al. (2020). *On the Approximation Lower Bound for Neural Nets with Random Weights*. arXiv: `2008.08427 [cs.LG]`.

Strobl, Lena (2023). *Average-Hard Attention Transformers are Constant-Depth Uniform Threshold Circuits*. arXiv: `2308.03212 [cs.CL]`.

Strobl, Lena et al. (2023). *Transformers as Recognizers of Formal Languages: A Survey on Expressivity*. arXiv: `2311.00208 [cs.LG]`.

Sun, Yitong, Anna Gilbert, and Ambuj Tewari (2018). *On the Approximation Properties of Random ReLU Features*. arXiv: `1810.04374 [stat.ML]`.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014). "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Zoubin Ghahramani et al., pp. 3104–3112.

Telgarsky, Matus (2015). "Representation benefits of deep feedforward networks". In: *arXiv preprint arXiv:1509.08101*.

— (June 2016). "Benefits of Depth in Neural Networks". In: *29th Annual Conference on Learning Theory*. Ed. by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir. Vol. 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, pp. 1517–1539.

Telgarsky, Matus (2022). "Feature selection with gradient descent on two-layer networks in low-rotation regimes". In: *arXiv preprint arXiv:2208.02789*.

Turkoglu, Mehmet Ozgur et al. (2021). "Gating revisited: Deep multi-layer RNNs that can be trained". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.8, pp. 4081–4092.

Vapnik, Vladimir Naumovich and Aleksei Yakovlevich Chervonenkis (1968). "The uniform convergence of frequencies of the appearance of events to their probabilities". In: *Doklady Akademii Nauk* 181.4, pp. 781–783.

Vardi, Gal et al. (2021). "Size and depth separation in approximating benign functions with neural networks". In: *Conference on Learning Theory*.

Vaswani, Ashish et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*.

Vershynin, Roman (2018). *High-dimensional probability: An introduction with applications in data science*. Cambridge University Press.

Wang, Huiyuan and Wei Lin (2021). "Harmless Overparametrization in Two-layer Neural Networks". In: *arXiv preprint arXiv:2106.04795*.

Wang, Ziwei et al. (2022). "Quantformer: Learning extremely low-precision vision transformers". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Warren, Hugh E (1968). "Lower bounds for approximation by nonlinear manifolds". In: *Transactions of the American Mathematical Society* 133.1, pp. 167–178.

Wei, Colin, Yining Chen, and Tengyu Ma (2022). *Statistically Meaningful Approximation: a Case Study on Approximating Turing Machines with Transformers*. arXiv: `2107.13163 [cs.LG]`.

Wei, Colin et al. (2019). "Regularization matters: Generalization and optimization of neural nets vs their induced kernel". In: *Advances in Neural Information Processing Systems* 32.

Williams, Francis et al. (2019). "Gradient dynamics of shallow univariate ReLU networks". In: *Advances in Neural Information Processing Systems 32*.

Xu, Keyulu et al. (2018). "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826*.

Yao, Andrew Chi-Chih (1979). "Some complexity questions related to distributive computing (preliminary report)". In: *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*.

Yao, Shunyu et al. (2021). "Self-Attention Networks Can Process Bounded Hierarchical Languages". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Yehudai, Gilad and Ohad Shamir (2019). "On the Power and Limitations of Random Features for Understanding Neural Networks". In: *Advances in Neural Information Processing Systems 32*. arXiv: 1904.00687 [cs.LG].

Young, Lai-Sang (1981). "On the prevalence of horseshoes". In: *Transactions of the American Mathematical Society* 263, pp. 75–88.

Yun, Chulhee et al. (2020). "Are Transformers universal approximators of sequence-to-sequence functions?" In: *International Conference on Learning Representations*.

Yurinskii, V. V. (1976). "Exponential inequalities for sums of random vectors". In: *Journal of Multivariate Analysis* 6, pp. 473–499.

Zaheer, Manzil et al. (2017). "Deep sets". In: *Advances in Neural Information Processing Systems 30*.

Zhang, Chiyuan et al. (2017). "Understanding deep learning requires rethinking generalization". In: *ICLR*.

— (2021). "Understanding deep learning (still) requires rethinking generalization". In: *Communications of the ACM* 64.3, pp. 107–115.

Zhang, Kaiqi and Yu-Xiang Wang (2022). "Deep Learning meets Nonparametric Regression: Are Weight-Decayed DNNs Locally Adaptive?" In: *arXiv preprint arXiv:2204.09664*.

Zhang, Yi et al. (2023). *Unveiling Transformers with LEGO: a synthetic reasoning task*. arXiv: 2206.04301 [cs.LG].

Ziegler, Günter M (2006). "Lectures on Polytopes". In: *Graduate Texts in Mathematics* 152.

Zweig, Aaron and Joan Bruna (2022). "Exponential Separations in Symmetric Neural Networks". In: *CoRR* abs/2206.01266. arXiv: 2206.01266.