UNIVERSALITY OF SCALING: PERSPECTIVES IN ARTIFICIAL INTELLIGENCE AND PHYSICS

by
Utkarsh Sharma
उत्कर्ष शर्मा

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of
Doctor of Philosophy

Baltimore, Maryland
July 2021

Abstract

The presence of universal phenomena both hints towards deep underlying principles and can also serve as a tool to uncover them. Often, the scaling behavior of systems shows such universality. An example of this is artificial neural networks (ANNs), which are ubiquitously employed in artificial intelligence (AI) technology today. The performance of an ANN, measured by the loss L, scales with the size of the network N and with the quantity of training data D as simple power laws in N or D. We explain these laws theoretically. Additionally, our theory also explains the persistence over many orders of magnitude of the scaling with model size.

When both the amount of data D and the model size N are finite, the loss scales as $L \propto D^{-1}$ and $L \propto N^{-1/2}$. The scaling in the regime where either N or D is effectively infinite is more non-trivial, being tied to the intrinsic dimension d of the training dataset by the simple relations $L \propto N^{-4/d}$ and $L \propto D^{-4/d}$. We test our theoretical predictions in a teacher/student framework, and on several datasets and with GPT-type language models. These measurements yield intrinsic dimensions for several image datasets and set bounds on the dimension of the English language that these were trained on.

Scaling behaviors also act as a tool to probe fundamental phenomena in nature—in this case the theory of quantum gravity. We use holography to probe spacetime by using the physics on its boundary. Specifically, previous work has employed the scaling properties of operators on the boundary to construct a scalar field in the bulk. Our construction extends this procedure to allow for arbitrary choice of gravitational dressing of the field. Apart from yielding a more comprehensive understanding of the quantum properties of gravity, our construction is suitable to test the non-locality of quantum gravity.

Advisor: Professor Jared Kaplan

Acknowledgments

When I first came across Jared Kaplan's research website, I was instantly and magnetically drawn. Under his areas of interest, this man made a list of the most interesting problems in fundamental physics, whether they relate to cosmology, particle physics, or condensed matter physics. I was afraid though that coming to Hopkins solely to work with Jared was risky since there was a chance that it wouldn't work out. Four years later, I can happily declare that my fears were unfounded.

It is hard for me to overstate how much I've enjoyed working with Jared. He is unconstrained by the boundaries of any discipline. And yet, despite always being on full throttle, he has been incredibly patient with me. I have always felt at home during my time in grad school, and a large part of the credit for that goes to Jared. I hope that the last four years have been only the start of a lifelong friendship.

I would like to thank my collaborators, Hongbin Chen, Ethan Dyer, Yasaman Bahri, and Jaehoon Lee. I learned a great deal from working with them. My special thanks to Hongbin, who patiently helped me get started when I was a young graduate student. I also thank Ethan for giving me the opportunity to work at Google with his team.

I would like to thank my family, particularly my sister, mom, and dad. I can trace my intrigue about the world to my early formative years, which were spent surrounded by my family. My parents managed to give me a great education despite hardships. I feel especially grateful for the presence of my grandpa during my childhood. He stimulated my interest in science and even helped me set up my own little laboratory at home.

I have been very fortunate to form many close friendships over the course of my graduate studies. In terms of personal growth, the last few years have felt like a decade, and this is largely due to the presence of friends and well wishers in my life. Of course, all of this growth would have been a hundred times harder without the presence of my partner Nicole, who is acceptance personified. She hasn't let her sharp intellect prevent her from experiencing the fragrance of life. In being herself she has shown me the power of openness. On a more technical front, I also thank her for editing this thesis.

My heartfelt thanks to Satyanarayana Dasa Babaji for leading by example and being a constant presence over the last many years. Lastly, special thanks to the couple from Vṛndāvana, Rādhā and Kṛṣṇa, who, by hook or by crook, refuse to be forgotten.

In memory of grandpa, who joyfully nurtured my interest in science.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Human civilization has shown rapid and almost monotonic growth since 10,000 BCE, as measured by the economic output of the world. This is seen in figure 1.1. Usually, we understand fast growth as being exponential. For example, for a while during 2020 the number of people carrying the coronavirus doubled every fixed number of days. The growth I'm talking about is not that. Economic and population estimates from the last many millenia suggest that they have been growing super-exponentially, meaning that they keep doubling faster and faster.

This kind of growth predicts something alarming—that the economic output of the world will become infinite at a finite time in the future. Open Philanthropy's models estimate this to happen before the turn of the century, and possibly even by 2050 [134]. This prediction is, of course, not literally true. As the human population or economic output gets unsustainably large, some factors will put breaks on it, such as climate change, depletion of natural resources, or human conflict originating from the previous two.

One of the main drivers of economic and social transformation today is Artificial Intelligence (AI). The World Economic Forum calls the AI revolution the 4$^{\text{th}}$ industrial revolution [139]. Some communities are facing its impact before others, and not always positive. A joint study done in 2017 by four prominent European transport groups estimated that automated trucks could reduce the demand for drivers by 50-70% in the US and Europe by 2030 [51].

Clearly, there is an urgent need for critical changes to our social and economic institutions to account for the AI revolution. The next few decades could be a great opportunity to bring about large-scale human well-being, but without proper steps it could just as easily implode human society. An essential ingredient to making sociopolitical changes is to have a timeline for the development of AI. Therefore, it is imperative to have a more comprehensive understanding of how fast AI is improving, and which directions will persist into socioeconomically significant developments.

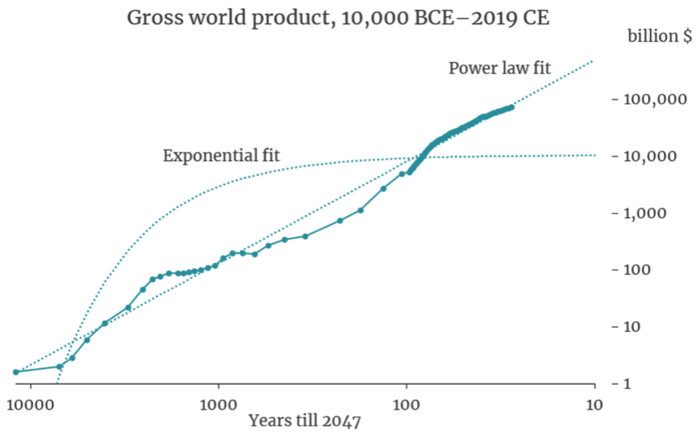

Figure 1.1: The economic growth of human civilization from 10,000 BCE to 2019 CE, as measured by the gross world product (GWP). According to this trend, the GWP would become infinite in the year 2047. Source: [134]

The brains of AI technology today are mathematical models called artificial neural networks (ANNs)[1]. Artificial neural networks are inspired by the human brain, which somehow interconnects billions of neuron cells to process information. Correspondingly, an artificial neural network is composed of a web of artificial "neurons". Similar to how human beings learn from their mistakes, ANNs learn by repeated trial and error. For example, the autopilot neural network that self-drives a car will be trained by letting it drive for thousands of hours. The longer it drives the less likely it is, for instance, to misidentify a stop sign or fail to brake in time.

ANNs are getting bigger (in terms of the number of neurons) every year and are training on more and more data. Typically, we expect an ANN with more neurons to perform better on complicated tasks. Likewise, an ANN that is trained for longer is expected to perform better.[2] Thus, it should be possible to predict the rate of improvement in AI performance for the foreseeable future.

And yet, a comprehensive quantitative understanding of how performance scales with the above factors was missing until now. Our work bridges this gap. We find which factors govern how fast AI performance improves with the size of the network and with the amount of training data. I describe this in more detail in section 1.1 below, as well as in chapters 2 and 3. Our scaling laws are stated mathematically in relations 1.1.2 and 1.1.3, and relation 1.1.4.

[1] Most modern deep learning models are based on ANNs. The McKinsey Global Institute expects ANNs to be even more dominant in the near future [30]

[2] The performance of neural networks may not always get better with size or longer training time, particularly for simpler tasks, due to a phenomenon called overfitting. Most real world tasks, e.g. language modelling, are complicated enough to prevent this from happening. See also footnote 6.

Our theory also explains a fundamental result—the scaling of the performance of an ANN with its size, or with the amount of training data, is often largely independent of the choice of architecture [81]. An architecture is the way the neurons are interconnected together to form the network. It is true that the performance of a network on a task depends on the choice of architecture. Yet, it turns out that how the performance *improves* with the above mentioned parameters is independent of this choice. Moreover, the scaling of performance is dependent on the nature of the training data. For example, Alexa's voice recognition AI [83] would have a different scaling than Facebook's face recognition AI [147], since one trains on the English language while the other trains on photographs.

The *universality* in the scaling of AI performance indicates that it is governed by the intrinsic properties of the training data, while the microscopic details of the architecture get washed out. Specifically, we have found that the scaling is governed by the *intrinsic dimension* of the data, which can be thought of as a measure of complexity of that dataset. Thus, by observing the scaling behaviour of ANNs on various datasets, we can infer the intrinsic dimensions of those datasets. In particular, we have found the dimension of CIFAR100, which is a collection of images of airplanes, cars, cats, dogs etc., and have also put bounds on the dimension of the English language. I elaborate more on this in section 1.1.2 below and in chapters 2 and 3.

The existence of universal scaling behaviors and their utility in revealing deep phenomena extend beyond the realm of AI. Fundamental processes in nature often result in universal scalings. When water is subjected to a temperature of about 374°C and about 218 atm pressure [153], it acquires unusual properties. Near this critical point, its heat capacity blows up. Specifically the numerical exponent of this divergence is about 0.11008 [87, 150].

Remarkably, the same numerical exponent is observed for almost any material's liquid-gas phase transition at the critical point, not just water. This intriguing universality doesn't end there—even magnets when heated show the same exponent. A piece of metallic iron is ferromagnetic below 770°C [19], but paramagnetic above. In other words, when a piece of iron is cold enough, it can be made into a permanent magnet, but if it is sufficiently heated, it loses its magnetism altogether. In the vicinity of the critical point, we again observe that the heat capacity diverges, and diverges with that same exponent. The microscopic details of the material's makeup don't seem to matter when it comes to the behavior near the critical point.

To understand the universal nature of the critical point, let's look at the example of a magnet in more detail. We can imagine a magnetic material, such as a piece of iron, as being a lattice of tiny molecular magnets [3]. The nature of these molecular magnets is such that they want to align in the same direction as their neighbors, whether it be up or down. When the molecules are all aligned in the same direction, the piece of iron becomes a strong magnet. However when the iron is heated, the added heat energy tends to scramble

[3] In literature, this model is referred to as the Ising model. For details see [150]

any alignments and patterns that are formed by the molecules. Thus, at high enough temperatures—more precisely at T above the critical temperature T_c—we observe that on an average there is no net magnetization of the iron, since nearby molecules are aligned randomly rather than uniformly. On the other hand, at low temperatures, molecules align uniformly to form large regions where their magnetizations points in the same direction. Consequently, the piece of iron then acts as a magnet. The behavior of a hot magnetic material is demonstrated by the top two figures in figure 1.2, while the behavior of cold magnetic material is shown in the bottom two figures. In these figures blue and yellow colors denote upward and downward pointing molecular magnetizations respectively.

As can be observed visually in figure 1.2 below, if we zoom out of the $T > T_c$ picture, it gets even more randomly scrambled. Similarly, if we zoom out of the $T < T_c$ picture, it gets more uniformly aligned. But, as demonstrated by figure 1.3 below, if $T = T_c$, then zooming out does nothing, meaning that it is impossible to tell what length scale of the material one is looking at. Hence, at the critical point, the system becomes independent of the length scale [159, 160].

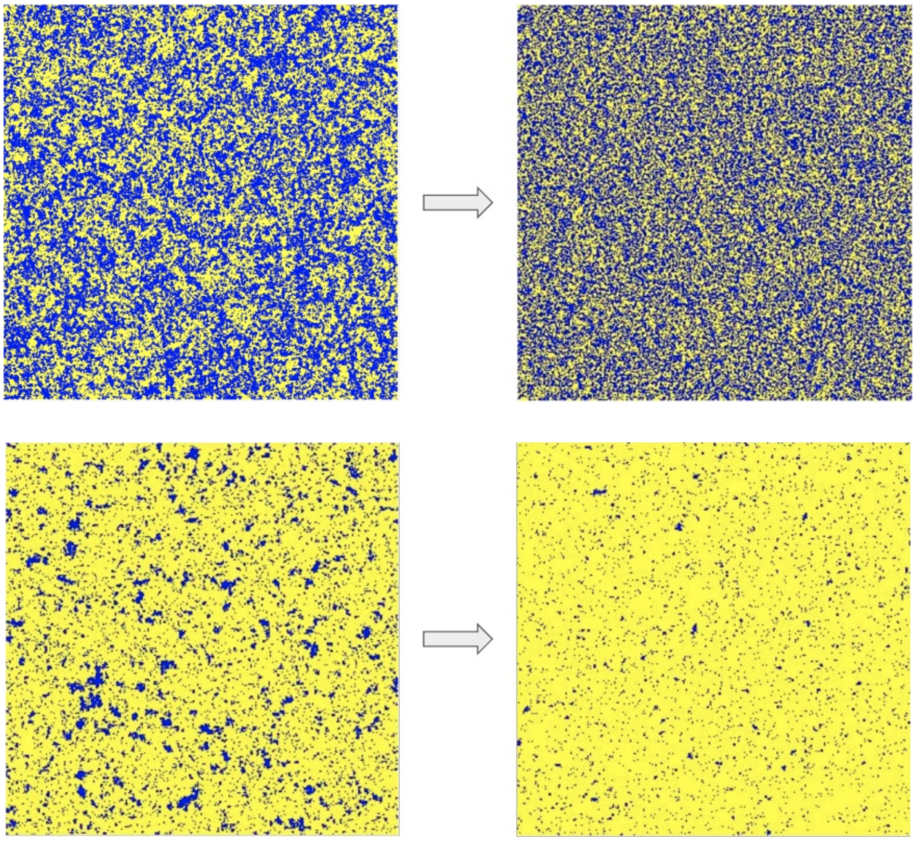

Figure 1.2: These pictures depict a (Ising) magnetic lattice. Zooming out of the system when $T > T_c$ (top) leads to more randomly aligned regions, i.e. the size of yellow and blue patches becomes even smaller. Zooming out of the system when $T < T_c$ (bottom) leads to more yellow regions. Source: [150]

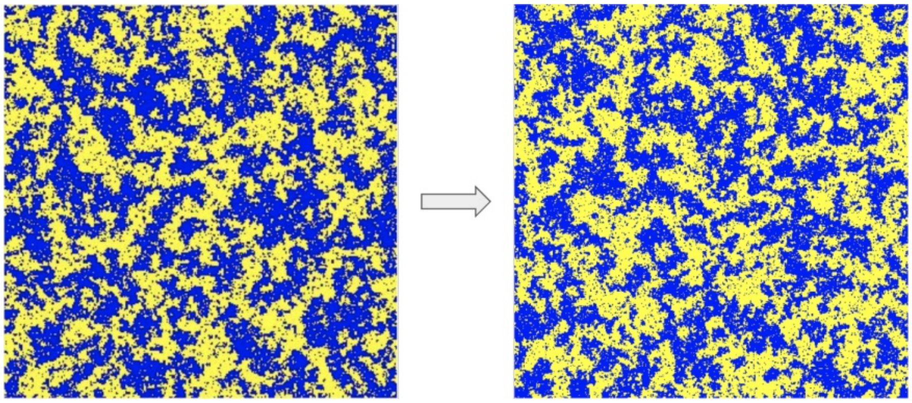

Figure 1.3: These pictures depict a (Ising) magnetic lattice at critical point $T = T_c$. Since the system is at the critical point, it is scale invariant. Consequently, there is no difference in the geometrical properties of the two figures. Source: [150]

Scale invariance explains the universality of phase transitions near the critical point. There exist only a handful of scale invariant mathematical models.[4] Thus, a large number of systems need to be described by relatively few mathematical models. Consequently, every mathematical model ends up describing a plurality of systems [159, 160]. This is why almost all materials that undergo liquid-gas phase transitions at the critical point have the same heat capacity exponent of 0.11008 [87, 150].

These mathematical models that describe the critical point are called *conformal field theories*, or CFTs in short.[5]. Remarkably, it has been found that CFTs are also useful in making progress in one of the biggest unsolved problems in fundamental physics today: the problem of quantum gravity. The four fundamental forces in nature—gravitational, electromagnetic, strong nuclear, and weak nuclear—are not all described by the same theory. The latter three are described by a quantum theory called the Standard Model, while gravity is described by Einstein's general theory of relativity. This separate treatment works well in most everyday scenarios, but it falls short when it comes to understanding the fundamental properties of nature, for example the events at the birth of the universe. Quantum gravity is an attempt to unify our understanding of all four fundamental forces of nature under one theory. Unfortunately, previous approaches have had limited success with this problem. In section 1.2 and chapter 4, we will see that the scaling behaviors of CFTs allow us to gain insight into this difficult problem.

This introduction is organized as follows. I elaborate on the scaling behaviors of artificial neural networks in section 1.1. There, I explain the mathematical formulation of an

[4] The system also has locality, which restricts the class of allowed models. I'm also assuming that scale invariant models are conformal field theories. See footnote 5 below

[5] Technically scale invariant models don't have to be CFTs, since CFTs have extra symmetries as well. In practice scale invariant models are almost always CFTs [114]. I will use the two terms interchangeably.

ANN, as well as our results and their significance in more detail. More information on this can be found in chapters 2 and 3. Next, I describe in more detail the problem of quantum gravity, its brief history, as well as our contributions in section 1.2. Remaining details and a technical treatment can be found in chapter 4. Lastly, I end with a discussion of further interesting universal scaling phenomena in other disciplines.

1.1 Universal Scaling Behavior of Neural Networks

Almost exclusively, the brain of any artificial intelligence (AI) technology today is an object called *artificial neural network* (ANN) or neural network in short. ANNs, which are inspired by the human brain, are a complex interconnected web of artificial neurons. Artificial neurons are described by simple mathematical functions, but together they can perform a staggering range of tasks.

In the next subsection I will describe the types of tasks that ANNs need to perform. Then we will see concretely what an artificial neuron is, and the structure of a simple ANN. I will then describe what it means to train a network. In later subsections I will describe mathematically the scaling behavior of ANNs, followed by our results.

1.1.1 Background

Artificial neurons

An artificial neuron is a mathematical function that takes a bunch of numbers as input, and churns out one number as the output. To be precise, the churning involves taking a weighted sum of the inputs followed by a non linear operation on the weighted sum

$$y = \phi\left(\sum_{i=1}^{m} w_i x_i + b\right) \tag{1.1.1}$$

where x_i's are a collection of m inputs, and y is the output. The w_i's and b are the weights and bias that act on the inputs. The function ϕ is a nonlinear function that makes the operation non-trivial.

By tweaking any individual weight w_i, we can control how much the input x_i contributes to the output. For example, if w_1 is much larger than the other w's then x_1 contributes more to the output.

The function ϕ can be chosen from a wide variety of nonlinear functions, a very common one being the piecewise linear function ReLU (Rectified Linear Unit) activation defined by

$$\phi(x) = \begin{cases} x & \text{if x>0} \\ 0 & \text{otherwise} \end{cases}$$

In chapters 2 and 3, we will almost exclusively use ReLUs.

Artificial neural networks

It is indeed remarkable that neurons, the building blocks of complex AI systems, are such simple mathematical functions. The power of AI comes from having neurons work together. Now, I will describe arguably the simplest ANN architecture—the fully connected network.

Fully connected networks consist of layers of neurons. The input is fed into the first layer. The neurons in the first layer process the input according to the mathematical function described above. The output of the first layer is then fed into the second layer, and so on. We pull the final output from the last layer. The layers between the input and the output layers are called hidden layers, because their outputs are normally hidden from the outside. This process can be visualized as in figure 1.4.

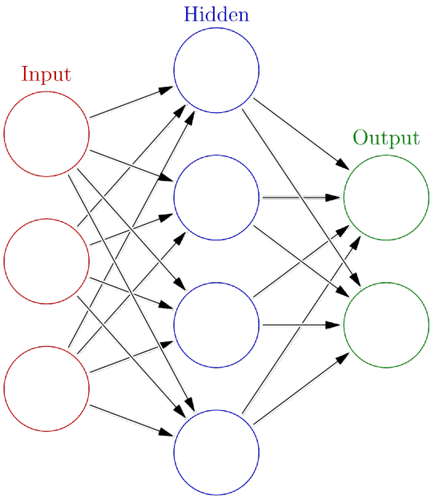

Figure 1.4: A fully connected artificial neural network with one hidden layer. There are 3 inputs, 2 outputs, and the hidden layer has 4 artificial neurons. The inputs are fed to the hidden layer, where the neurons process it and then feed it forward. In general there can be any number of hidden layers, any number of neurons in each hidden layer, and any number of inputs and outputs. Source: [34]

Different architectures are suited to different problems. For example, convolutional neural networks are more suited to tasks related to images, while recurrent neural networks work better on time series data, like human languages etc. Despite their differences, typically these architectures can be visualized as having a layered structure.

Training

Since each neuron has w_i and b, an ANN has a whole collection of weights and biases coming from the different neurons that comprise the network. By tuning each weight, we can modify how the neurons process the input. This process of adjusting the weights to get a desired result is called training. Of course, this is not done manually. There are several algorithms to do this, but we shall not get into the details of those algorithms.

Supervised, unsupervised and reinforcement learning

Machine learning tasks can be of many varieties. I will describe three common types.

Imagine you want to train an ANN to recognize handwritten digits. A straightforward way to do this is to write a bunch of digits by hand on a piece of paper and then take pictures of them. This forms our training data. Now, we can feed these images into the network and tweak the weights of the network until it outputs the correct answer indicating which digit is on the picture. In practice this is done automatically by having a set of training data, along with a set of labels that indicate the correct answer, like the correct digit on each image. We feed the neural network with the training data and training labels and train it until it achieves a desired accuracy. This is called *supervised learning*.

Some other tasks do not need explicit labels. For example, let's say that most of the customers of a product are either in their twenties or fifties. To find this pattern we just need to feed the customer information into a network and let it cluster them into categories. Since there is a natural classification, the ANN would hopefully pick up on that. This is a trivial example, but in general non-trivial patterns can be uncovered through this process, called *unsupervised learning*.

ANNs can also be trained to play games. One method to do that is to let the network play the game, and then reward it for a correct move and punish it for wrong ones. This is called *reinforcement learning*.

1.1.2 Intrinsic dimension of data and universal scaling of performance

We saw above that an ANN has a bunch of parameters that need to be trained to yield the desired outcome. The size of a neural network is measured in terms of the number of these trainable parameters. As of 2021, ANNs typically have millions of parameters. Some big ones have tens of billions of parameters. Open AI's GPT-3 has 175 billion parameters [20].

The sizes of ANNs are growing every year. Similarly, the amount of training data available is also increasing. In general a network that is bigger, or more trained, does better at complicated tasks.[6]

When we try to study how ANN performance improves (i.e. scales) with their size, an intriguing observation emerges. Even though there exist a plethora of vastly different neural network architectures, the scaling of performance of most of these architectures is the same. This is not to say that they perform the same. Some are still better than others, but how quickly the performance gets better with increasing size is largely independent of which architecture is chosen.

Moreover, the scaling behavior changes when the task is changed, e.g. recognizing faces in photographs will scale differently than playing the game of Go. This suggests that scaling behaviors have something to do with the intrinsic properties of the task that the ANN needs to perform. I describe this next, while more details are contained in chapter 2.

Intrinsic dimension of data

The data that is fed to a neural network is in the form of arrays of numbers. For example, imagine a black and white picture. Each pixel of the image can be assigned a number, based on how bright it is. Let's say that the color white is 1, black is 0, and every shade of grey in between is a number between 0 and 1. Thus, every such image can be fully described by an array or numbers, each denoting one pixel.

Suppose the image that we started with was 10 pixels high and 10 pixels wide. Then, there are $10 \times 10 = 100$ pixels in the image, and so the image will be described by an array of 100 numbers. Now, imagine starting out with a whole set of such images for the purpose of training the network.

This description of the images is likely to be redundant. Imagine a bunch of photographs of landscapes. In almost all these photographs, the top half is the image of the sky, which is blue. Thus, the numbers corresponding to those pixels hardly change between different images. One can drop these redundant numbers and still be able to fully describe each image. Let's say that after getting rid of all the redundancies, we are left with 25 numbers to describe each image. Then we say that the *intrinsic dimension* of the set of images is 25. As the name suggests, this is an intrinsic property of the set of images. We can also look at the intrinsic dimension of the data as being a measure of complexity. Small dimensional data is typically less complex compared to large dimensional data.

More technically, we're viewing data as a being on a manifold. The data naively lies in a large manifold, but truly occupies only a submanifold within the larger space. In the above example, each image is a point in a 100 dimensional manifold. But, the image dataset really

[6] When an ANN has too many parameters, it essentially memorizes the training dataset, resulting in poor performance outside the training dataset. Most real world tasks, e.g. language modelling, are complicated enough to prevent this from happening. In chapters 2 and 3 we take measures where needed to avoid overfitting.

lies on a 25 dimensional submanifold embedded within the bigger manifold. The dimension of the submanifold is related to the inherent information content and compressibility of the dataset. The dimension of this submanifold is the intrinsic dimension of the data.

The scaling law, mathematically

We have found that the intrinsic dimension of the data is the driving force behind the universal scaling behavior. Essentially, when the data is large dimensional the performance improves only slowly as we increase the size of the neural network, and vice versa. Mathematically, neural network performance is measured by loss. Larger loss means worse performance. When the amount of training data is plentiful, i.e. when the network size is the bottleneck, then we can write down the scaling relation between loss and model size-

$$\text{Loss} \propto \frac{1}{(\text{Network Size})^{4/d}} \tag{1.1.2}$$

where d is the intrinsic dimension of the data, that we talked about above.

Thus, for any data and network choice, a bigger network means better performance (smaller loss), but how fast the performance improves depends on d, the dimensionality of data. For large d, we need a much bigger network to get an improvement in the performance.

Figure 2.1 in chapter 2 demonstrates this law. We measure the scaling exponents, and independently measure the intrinsic dimension of various datasets. The plot between the two establishes relation 1.1.2. More details, including a simple theory behind relation 1.1.2 can be found in section 2.2 in chapter 2.

1.1.3 Computing the dimension of the English language and other data

We saw above that the intrinsic dimension of the data governs the scaling of performance of ANNs. So, by looking at the scaling behavior of neural networks on various datasets, we can infer their intrinsic dimensions.[7]

In particular, we have found that the MNIST [91], fashion MNIST [162] and SVHN [120] datasets, which consist of 10 classes of images, have a dimension of around 10. More non-trivially, the dataset CIFAR100, which consists of images from 100 classes like trucks, airplanes, cats, dogs etc., has a dimension around 25. The details of these measurements can be found in chapters 2 and 3.

From the scaling of OpenAI's language models [81], we have been able to bound the dimension of the training data in English. Since the training data is both vast and chosen

[7] Technically, the scaling exponent only puts a bound on the intrinsic dimension. In many cases the bound is tight, but exceptions do occur. For details about the inequality and exceptions see section 2.2.2.

from a variety of sources, this can be interpreted as being a bound on the dimension of the English language, as seen by neural networks. We expect the dimension to be about $d \geq 53$ from measurement of the scaling exponent, and $d > 90$ from independent methods of measurements. A detailed discussion can be found in chapter 2 in section 2.3.4 .

1.1.4 Scaling in other regimes

We discussed above the scaling of neural network performance with the size of the network. The relation 1.1.2 holds when size is the limiting factor, i.e. when training data is plentiful.

On the other hand, if network size is very large (effectively infinite), but the amount of training data is limited, we find a similar scaling behavior-

$$\text{Loss} \propto \frac{1}{(\text{Dataset Size})^{4/d}} \tag{1.1.3}$$

This can be understood rigorously for the case of a linear network, as explained in section 3.2 of chapter 3. The top right plot in figure 3.1 supports this law. Relation 1.1.2 is also verified by the bottom left plot in figure 3.1.

Apart from the two regimes that we have discussed so far, the top left and bottom right plots in figure 3.1 demonstrate two other regimes. These show the scaling behavior with network size and training data size, when neither of them is very large. In this case, the scaling is very simple-

$$\begin{aligned} \text{Loss} &\propto \frac{1}{\text{Model Width}} \\ \text{Loss} &\propto \frac{1}{\text{Training Data Size}} \end{aligned} \tag{1.1.4}$$

We refer to the first two regimes, where either data or model size is effectively infinite, as *resolution limited regimes.* The remaining two regimes, where both model and data are finite are referred to as *variance limited regimes.* A detailed analysis of all four regimes is found in chapter 3.

1.1.5 Limitations and future directions

While our theory of scaling exponents described above has been formulated for supervised learning, our superclassing experiments suggest that unsupervised learning behaves similar to supervised learning (see section 3.4) in the way that it treats data. This could hint towards a deeper theoretical understanding of scaling in unsupervised learning. It is less clear how the theory extends to reinforcement learning. A theory of sample efficiency scaling would be more likely to be relevant to reinforcement learning.

Each layer in a neural network is expected to see a different level of abstraction of the data. Therefore, to test our theory we choose the minimum intrinsic dimension out of all hidden layers, since that represents the most compact representation of the data. In this regard, it is less trivial to test our theory on architectures that have a residual (ResNet) structure, for example OpenAI's language models [128], since the input itself is added after each layer. For these architectures, we still expect the inequality to hold (see section 2.2.2), but the bound may not be saturated.

1.2 Scale Invariance and Quantum Gravity

1.2.1 Background: what is quantum gravity?

Is Einstein's relativity wrong?

Albert Einstein discovered the general theory of relativity in 1915. His theory provided a beautiful geometric description of gravity. Gravity is one of the four fundamental forces in nature— the other three being electromagnetic, weak nuclear, and strong nuclear forces. A unified description of the latter three, called the Standard Model, had to wait until the second half of 20th century. The Standard Model is a quantum theory, while general relativity is classical, meaning that it can be understood without quantum mechanics.

Around the same time that the Standard Model was taking its final form, Stephen Hawking combined some properties of quantum mechanics with general relativity and discovered that black holes aren't so black, meaning that they can radiate [65]. Black holes are objects that pack an incredibly high density of matter in a very small volume. They are usually formed when a gas of matter collapses due to the strength of its own gravitational field. According to general relativity, black holes are supposed to be black, meaning that they absorb any matter or radiation that falls on them. Nothing can escape from a black hole.

What Hawking found was startling—that even black holes can radiate. This is an indication that general relativity may not be entirely correct. At least, quantum mechanics and general relativity are at odds with one another.

The separation of quantum and gravity

General relativity is not a complete description of the universe because it explains only gravity, while leaving the other three for the Standard Model. In most situations this division of labor works. For example, to describe how the Earth affects flying birds, we can consider separately the effects of gravitational and magnetic fields of the Earth. In other words, electromagnetic and gravitational phenomena don't usually mix with each other.

We can understand this separation from the scales of the forces. Gravity is a large scale force, in the sense that its effects are relevant on the macroscopic scales. For example, the dynamics of planetary motion are governed by gravity. The other three forces in nature are microscopic. Electromagnetic force holds atoms and molecules together, and gives solids their rigidity. The two nuclear forces act at the scale of protons and neutrons. Similarly, there is a separation of scale in terms of strength of the forces. The strong nuclear force is about a hundred times stronger than electromagnetic force, about a million times stronger than weak force, but about 10^{38} times stronger than gravity.

Black holes mix quantum and gravity

The neat separation of scales between quantum forces and gravity breaks down when too much matter starts accumulating in a very small region of space. The gravitational field produced by highly dense matter can no longer be ignored in comparison to the other three forces. This happened at the beginning of the universe when all the matter and energy of the universe was crunched up in a tiny volume. Since black holes have extremely dense matter in a tiny volume, they also become a playground for the mixing of quantum theory and gravity.

The information paradox

That quantum mechanics and general relativity are at odds with one another can also be seen in a another related phenomenon—the information paradox. Quantum mechanics is unitary, meaning that as a quantum system evolves in time, it cannot gain or lose information. Black holes, on the other hand, lose information. Black holes are completely characterized by a handful of quantities, namely their mass, momentum, angular momentum and electric charge. For example, imagine two different black holes, both formed by crunching together objects of different shapes and sizes. If we can make the total mass etc. of the falling matter the same, then we get identical black holes, even though the starting objects were very different in shapes and sizes. Thus, a large amount of information about the matter that fell into the black hole is lost. This is called the no-hair theorem—that the surface of a black hole is smooth, and contains no additional information [112, 71].

The resolution: a theory of quantum gravity

Hawking made his startling prediction of black hole radiation by combining some properties of quantum theory with general relativity. But, his approach only scratched the surface (pun intended) of a complete, unified quantum theory of gravity. The full theory of *quantum gravity* would explain all four forces of nature. It would be the fruit of thousands of years of progress towards a fundamental understanding of nature.

Unfortunately, our efforts so far have fallen short. Combining gravity with other forces has presented both technical as well as conceptual difficulties. In fact, to even think about a unified theory of gravity, we're having to rethink what we took for granted, for example what it even means for an experimental apparatus to make a measurement.

For a while in late 20th century, string theory was thought to be a contender for a theory of quantum gravity. Gradually, it became clear that string theory was not just incredibly hard to solve, but also that it is experimentally unverifiable. Essentially, whatever properties of the universe we find through experiments, we can find a way in string theory to justify them [161]. However, through string theory we found a property that turned out to be very robust. Robust to the extent that we're willing to forget about string theory and use this instead as our fundamental principle. This property is called *holography*.

1.2.2 Background: holography

A photograph is a two dimensional representation of an object, but a hologram is the three dimensional representation. Therefore, a hologram faithfully captures the entirety of the object.

Einstein treats space and time as one combined object, called spacetime, and gravity is just a geometrical property of spacetime. For technical reasons, we will deal with a hypothetical spacetime that does not exactly resemble our universe, but nevertheless gives important physical insight. This is called the *anti-de Sitter* (AdS) spacetime.[8]

A surprising relationship was discovered about AdS spacetime in the late 1990's–that the physics on the boundary is enough to tell us everything that's happening in the entire spacetime [103]. This is not at all obvious. Imagine a city where you can tell exactly what is happening inside by just watching the events on the outskirts! This duality is called holography. In particular, the complete quantum gravity in this spacetime is described in terms of the physics on the boundary. To use this approach to study quantum gravity, we need to understand the boundary physics better.

The boundary of AdS

Let's say we have a (AdS-)spacetime. Now, say we travel infinitely far out towards the boundary of space. We find something interesting about the spacetime close to the boundary—depending on the details of how we approach the boundary, it looks either stretched or compressed. Thus, the same physical phenomena happening at the boundary may appear stretched or compressed from various points of view [80]. The phenomena themselves don't change. They just look different. This is tantalizingly pointing towards the fact that the physics on the boundary is independent of stretching or compressing, i.e. it is scale invariant.

[8] AdS is a spacetime of constant negative curvature. It is unlikely that our universe resembles AdS, but our understanding of quantum gravity so far is is largely restricted to AdS.

The physics on the boundary is thus described by a conformal field theory (CFT).[9]

Holography is a general principle, and the correspondence between AdS physics and CFT on the boundary is a prime example. In this introduction, as well as in 4, we shall use the terms holography and AdS/CFT correspondence interchangeably.

Promoting holography

The discovery of holography was done within the framework of string theory, but it was soon realized that the duality seems to be much more robust. In fact, evidence from fundamental principles in physics, e.g. thermodynamics, also points towards holography, without relying on the heavily complicated machinery of string theory. Thus, we make a bold move—we turn the problem on its head. We assume that holography is true, then we can use it to find the properties of the yet undiscovered theory of quantum gravity.

1.2.3 Bulk reconstruction

In order to use holography to find the properties of quantum gravity, we start with a CFT on the boundary and use that to reconstruct the events in the bulk. The procedure is called *bulk reconstruction*.

Bulk reconstruction is relatively straightforward if we assume that the AdS space in the bulk has no gravitational fluctuations, i.e. all gravity is just a static force that does not mix with any other forces. This old-fashioned gravity takes the interesting physics out of the picture. In particular, it takes away the quantum properties of gravity.

A more involved procedure for bulk reconstruction was recently implemented, where explicit calculations can be done without stripping gravity of all its quantum properties [8]. This 'proto-field' approach makes interesting predictions, but, as we discuss next, one problem with this approach is that it keeps some essential properties hidden under the rug.

1.2.4 Dressings in quantum theories

Forces are produced from fields, e.g. an electric field produces an attraction or repulsion between charged particles. In particular, the three quantum forces (electromagnetism, strong nuclear, and weak nuclear) are mathematically described as quantum fields. This mathematical description explains a host of phenomena that we take for granted. One of them is of particular interest to us. An electron always has to come surrounded with an electric

[9] If we ignore the dynamics of gravity in the bulk then the boundary theory lacks a stress tensor, hence is not a CFT [80]. See also footnote 5.

field.[10] Naked electrons don't exist in nature. Naively, one could imagine that an electron is its own particle, while electric fields are related to electromagnetic fields, and so they should be allowed to exist independently. Indeed, the mathematics of field theory prohibits that. An electron has to always come *dressed* in electric fields in order to keep the theory consistent.

Gravity also has the same constraint. Therefore, a correct theory of quantum gravity should yield dressed fields, rather than bare fields. Since the proto-field constructed in [8] is interacting with gravity, it must be dressed. But, it is not clear what the nature of the dressing is. Besides, the exact form of the dressing should be a choice, but the construction of the proto-field seems to take the choice away from us.

In chapter 4, this problem is rectified by explicitly constructing a bulk field dressed with a dressing of choice. This new construction is not pedantic. It allows an exploration of some fundamental properties of quantum gravity, which was previously not possible. One such property is locality.

1.2.5 Locality in quantum gravity

We expect quantum gravity to be non-local at very short length scales, meaning that the position of any object cannot be specified precisely. This is because quantum observables are fuzzy. Since in quantum gravity the geometry of spacetime itself is a quantum entity at very short length scales, geometrical quantities like distance become fuzzy.

Any theory of quantum gravity should be able to test this property. Indeed, the original proto-field construction has been shown to have non-locality [26]. But, since the proto-field comes attached (dressed) to a line that extends all the way to the boundary, we don't know if the non-locality that we detect is due to genuine non-locality of quantum gravity, or an artifact of this construction.

Having the ability to dress the proto-field with a dressing of our choice allows one to extract the genuine non-localities of quantum gravity, as I now explain. The procedure to tell if something is present at a point in space is to collide it with a second object. If a collision happens then there was indeed an object at that spot. Now, if a collision happens at multiple points, then the original object was present at multiple points, i.e. it was non local. In the original proto-field construction, when we try to collide two objects together, their dressings themselves also start colliding with each other. Thus, it is not possible to tell what is causing the collision, the collision of the dressings or the fundamental non-localities of quantum gravity. With the new construction, we can arrange the dressings of two colliding objects such that the dressings themselves stay out of the way, i.e. they themselves don't collide with anything. Thus, the only non-local collisions that can potentially happen are

[10] The property mentioned here is called gauge invariance, and the dressings are a generalization of Wilson lines [41]

due to the genuine non-locality of quantum gravity.

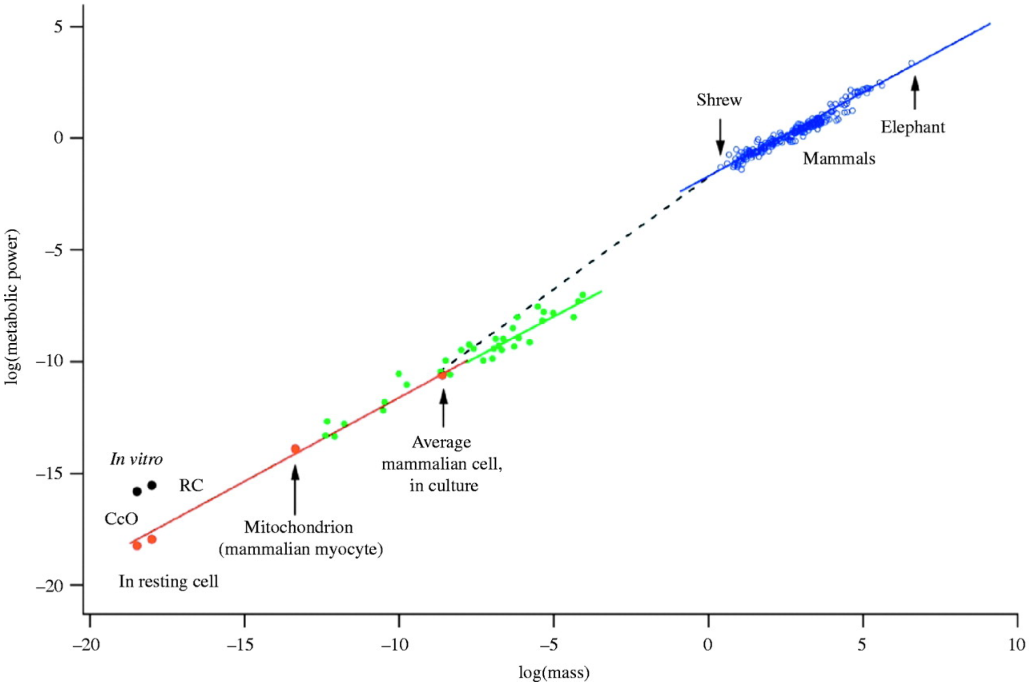

Figure 1.5: Basal metabolic rate of organisms plotted against their body mass. A power law with exponent ≈ 0.75 holds across 27 orders of magnitude in mass

1.3 Discussion

Apart from machine learning and physics, universal scaling laws are found in a variety of other areas. For example, if the basal metabolic rate of organisms is plotted against their mass M, a clear power law extending over 27 orders of magnitude is found (figure 1.5). Various other 'allometric' scaling laws have been found that are summarized in table 1.1. Interestingly, the exponents are almost always in multiples of 1/4 or 1/8. The universality of these 'quarter power' scalings has been explained in terms of the efficient geometry of distribution systems, like the circulatory system [156, 155].

A remarkable consequence of these scaling laws is the emergence of invariant quantities. For example, since mammalian lifespan increases as $M^{1/4}$, while the heart rate decreases as $M^{-1/4}$, the number of heartbeats per lifetime ($\approx 1.5 \times 10^9$) is approximately independent of body size [155]. This means that whether it is giant animals like elephants or tiny ones like mice, their hearts beat about a billion times in their lives. Similarly, the population density in forest communities ($M^{-3/4}$) and the the individual power ($M^{3/4}$) together imply that the power used by individuals in any size class is roughly constant [43, 155].

Parameter	Scaling with Mass
Mammalian life span	$M^{1/4}$
Mammalian heart rate	$M^{-1/4}$
Radii of aortas and tree trunks	$M^{3/8}$
Unicellular genome lengths	$M^{1/4}$
RNA concentration	$M^{-1/4}$
Population density in forest communities	$M^{-3/4}$

Table 1.1:: Allometric scaling laws. Source of data: [155]

Similarly, a number of properties of cities, like total wages, patent production, and electricity consumption scale with the population size. The scaling exponents (β) fall into universality classes—$\beta \approx 1.2 > 1$, $\beta \approx 1$ and $\beta \approx 0.8 < 1$. As expected, quantities like household water consumption scale linearly with population ($\beta \approx 1$), since an individual's domestic water needs are roughly the same irrespective of the size of the city they live in. $\beta < 1$ corresponds to economies of scale, i.e. quantities that are easier to produce on a larger scale, for example the length of electric cables. In contrast, innovation and wealth creation is supported by larger size. Thus, the quantities that depend on them scale with $\beta > 1$, like the number of new patents [14]. The population dynamics based on these scalings [14] provides quantitative insight into the super exponential growth of human population and economic output since 10,000 BCE which we encountered in figure 1.1.

The exponents $\beta > 1$ are qualitatively distinct from scaling exponents found in biological systems, which are all smaller than 1. Innovation in urban economies now happens on time scales much shorter than individual life spans, and will only becomes more rapid as population increases (since $\beta > 1$). On the other hand, evolution in biological systems happens on the evolutionary time scale, which vastly exceeds individual life spans. Urban "metabolism" is likely fueled by uniquely human social dynamics that transcend biology [14].

Universality of quantitative processes indicates the unity of the machinery behind them, be it in natural phenomena in physics or biology, or human engineered systems like artificial intelligence and urban centers. On one hand, in case of artificial neural networks, the universality of scaling behaviors lead us to uncover the simple mechanisms that underlie these complex machines. On the other hand, in case of quantum gravity, scaling behaviors themselves become tools to understand the elusive secrets of the universe.

Chapter 2

A Neural Scaling Law from the Dimension of the Data Manifold

This thesis chapter originally appeared in the literature as:

U. Sharma and J. Kaplan, "A Neural Scaling Law from the Dimension of the Data Manifold," *arXiv*: 2004.10802

Abstract

When data is plentiful, the loss achieved by well-trained neural networks scales as a power-law $L \propto N^{-\alpha}$ in the number of network parameters N. This empirical scaling law holds for a wide variety of data modalities, and may persist over many orders of magnitude. The scaling law can be explained if neural models are effectively just performing regression on a data manifold of intrinsic dimension d. This simple theory predicts that the scaling exponents $\alpha \approx 4/d$ for cross-entropy and mean-squared error losses. We confirm the theory by independently measuring the intrinsic dimension and the scaling exponents in a teacher/student framework, where we can study a variety of d and α by dialing the properties of random teacher networks. We also test the theory with CNN image classifiers on several datasets and with GPT-type language models.

2.1 Introduction

Neural Network based Machine Learning has made enormous progress in a wide variety of domains. Scale has been a key ingredient in this success: large amounts of computation, large datasets, and large models with millions or billions of parameters.

Not only is scale beneficial to performance, but the benefits from scale can be predicted precisely. Recent works [69, 68, 135, 81] studying a variety of data modalities and model

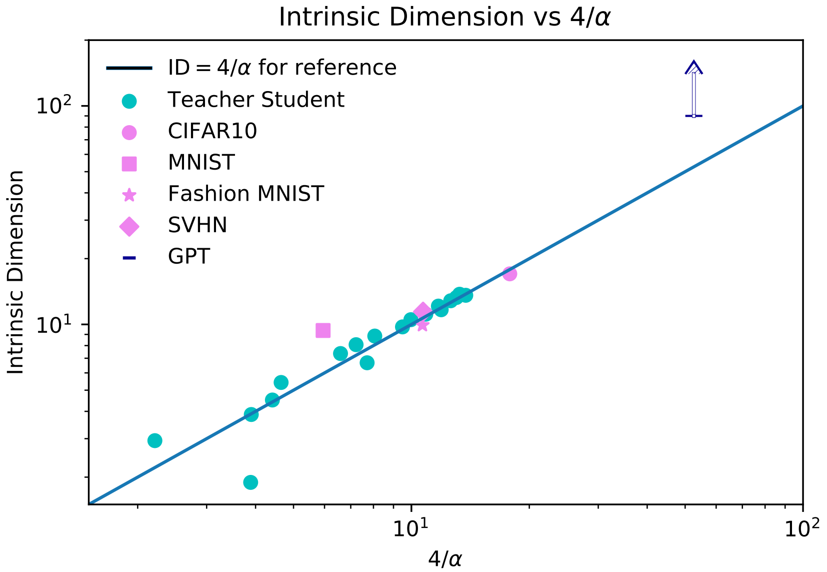

Figure 2.1: This figure shows the relationship between the measured intrinsic dimension (ID) of the data manifold and $\frac{4}{\alpha}$, where α is the model size scaling exponent. We include data from fully-connected teacher/student experiments, simple CNNs, and GPT-type [127, 128] language models (represented as a lower-bound due to large uncertainties with large IDs).

architectures all find the same scaling relation in the underfitting regime. In particular, the dependence of the loss on the number of model parameters N has the following properties, and each suggests a corresponding question:

- As the number of model parameters N is increased, the cross-entropy loss of well-trained and well-tuned models scales with N as a power-law

  $$L(N) \propto \frac{1}{N^{\alpha}} \tag{2.1.1}$$

  with observed values such as $\alpha \approx 0.076$ for language modeling [81], and much larger $\alpha \approx 0.5$ observed for image classification [135]. Why do we encounter this simple functional form, and what determines the value of the exponent α?
- Scaling holds very accurately across a wide range of N, sometimes spanning many orders of magnitude [69, 68, 81]. Why does scaling persist over a large range of model sizes, and what determines the N_{max} where it eventually breaks down?
- Empirically, the scaling exponent α may not depend greatly on model architecture. For example, LSTMs and Transformers scale similarly over a large range of N [81], with losses differing only by an overall, N-independent factor. Why would scaling exponents be roughly independent of model architecture?

We will argue that a simple conjectural theory can address these questions while making a number of testable predictions.

2.1.1 Main Ideas

The key idea is that neural models map the data to a manifold with intrinsic dimension d, and then use added capacity to carve up this manifold into ever smaller sub-regions. If the underlying data varies continuously on the manifold, then the size of these sub-regions (rather than their number) determines the model's loss. To shrink the size of the sub-regions by a factor of 2 requires increasing the parameter count by a factor of 2^d, and so the inverse of the scaling exponent $1/\alpha$ will be proportional to the intrinsic dimension d of the data manifold. We develop these ideas in detail in section 2.2.

The scaling exponent α can be measured by training a succession of models of varying size. We measure the intrinsic dimension d within the final layer[1] activations of trained networks, using the distances among nearest neighbor activation vectors [95, 44].

We test the theory in a student/teacher framework, which makes it possible to scan over a large range of α and d and test more idiosyncratic features of the theory (see figure

[1] It was shown in [10] that the final hidden layer activations have the smallest intrinsic dimension in image classifiers. Our findings are largely consistent with this.

2.4). We also perform tests using CNNs for image classification, and by measuring the intrinsic dimension of GPT-type models [127, 128], where scaling exponent have already been documented [81].

2.1.2 Contributions: Predictions and Results

In what follows we list the concrete predictions made by our theory, and their status based on our results[2] and information in the literature. Throughout we use L to denote the loss, N to denote the number of parameters in a neural network (often referred to informally as 'model size'), α as the power-law scaling exponent, and d as the intrinsic dimension of the data manifold.

1. **Prediction:** In the range of N where the loss scales as $L(N) \propto \frac{1}{N^\alpha}$, we predict $\alpha \propto \frac{1}{d}$, where d is the intrinsic dimension of the data manifold for the dataset and task in question. If the network is composed of ReLU non-linearities and the loss is mean squared error or cross-entropy (or KL divergence), we predict

   $$\alpha \gtrsim \frac{4}{d} \tag{2.1.2}$$

   with equality expected in the generic case.

   Results: See figure 2.1 for the summary combining all datasets. We find a variety of evidence supporting this prediction, and the factor of '4' fits quite well. We show in figure 2.8 that this factor can be modified if we use other loss functions. For language modeling with GPT [127, 128], we know $\frac{4}{\alpha} \approx 53$ while we measure the intrinsic dimension as $d \geq 90$ (figure 2.10), in accord with the inequality, but quite far from equality.

2. **Prediction:** The maximum network size $N_{\max}$ where we obtain power-law scaling grows with d via $\log N_{\max} \propto d$. Larger d should correspond with much larger $N_{\max}$.

   Results: We have confirmed the approximate relation $\log N_{\max} \propto d$ (see figure 2.2) with teacher/student experiments by identifying when $L(N_{\max})$ reaches a fixed value.

3. **Prediction:** The exponent α will not depend significantly on model architecture except through the intrinsic dimension d. Since larger α and smaller d lead to improved performance with scale, the best architectures will tend to have the smallest d.

   Results: In [10] it was discovered empirically that better performing image classifiers have smaller d, and [81] showed that LSTMs and Transformers have very similar exponents. We leave the measurement of both α and d across distinct architectures to future work.

[2] Code for our experiments will be available at: `https://github.com/U-Sharma/NeuralScaleID`

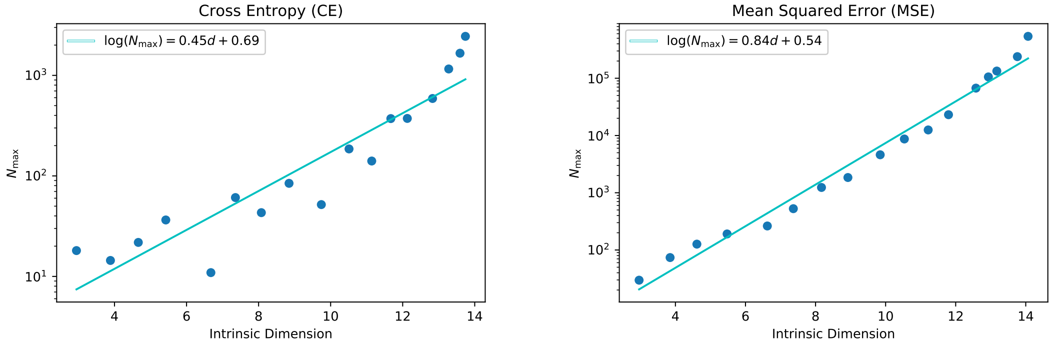

Figure 2.2: This figure estimates the behavior of $N_{\max}$, the maximum network size where we find power-law scaling, as a function of the intrinsic dimension in student/teacher experiments. We determine $N_{\max}$ as the model size where the loss reaches an arbitrarily chosen small value of 0.006, as a stand-in for the entropy of real data. We discuss this procedure in section 2.3.1.

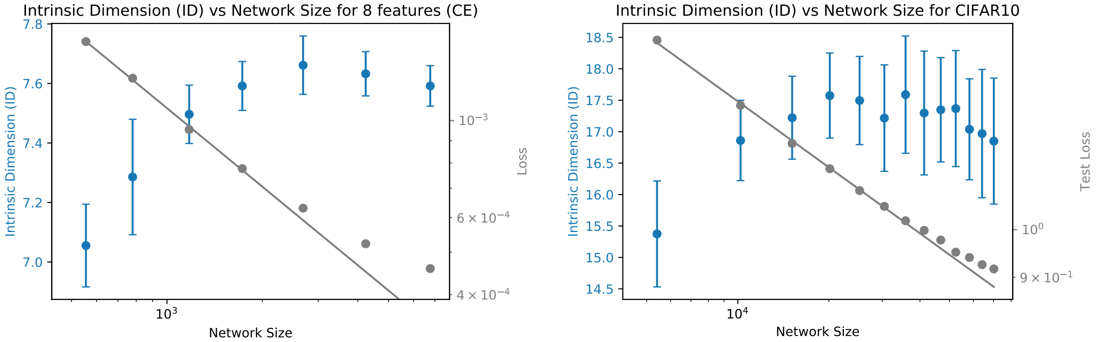

Figure 2.3: We show how ID measurements vary among different student network sizes N trained from the same teacher (left), and for CNNs on CIFAR10 (right). We display the test loss $L(N)$ for reference. The ID does not depend significantly on N, though it increases by about 10% among the various model sizes tested as N increases.

4. **Prediction:** Models with size $N \in [N_{min}, N_{max}]$ where the loss scales as a power-law in N all map the data to a manifold with the same intrinsic dimension d.

   Results: We verify this for teacher/student experiments in figure 2.3 and for CIFAR10 in figure 2.9. This prediction holds to about 10% for these models.

5. **Prediction:** If the data manifold $M = X_1 \times X_2 \cdots \times X_n$ and the loss $L(x) = \sum_i L_i(x_i)$, then we should replace the dimension of M with the maximum dimension of X_i when estimating α, as the network can behave as an ensemble, modeling each X_i independently (see the right of figure 2.4).

   Results: We confirm this prediction in section 2.3.2, see figure 2.7.

2.2 A Simple Theory for Scaling in the Underfitting Regime

In this section we explain our theory, beginning with a toy model in section 2.2.1. Then in section 2.2.2 we argue[3] that the toy model can be applied to realistic neural networks with only a few small modifications. In section 2.2.3 we explain how we measure the dimension of the data manifold, a necessary step in validating the theory.

2.2.1 A Toy Model

Consider one of the simplest scenarios for multidimensional regression. We are given a Lipschitz function $f : [0,1]^d \to \mathbb{R}$, and we would like to approximate it as a piecewise constant function $c(x)$, by cutting $[0,1]^d$ into smaller hypercubes. If these hypercubes have a side length s, then we will have

$$N = s^{-d} \tag{2.2.1}$$

cubes, and so our approximation will depend on the N constant values $c(x)$ takes within each hypercube. If the loss is mean-squared error (MSE), then it will be bounded by

$$L = \int_0^1 d^d x |f(x) - c(x)|^2 \lesssim \lambda^2 \left(s^2 d\right) \tag{2.2.2}$$

where λ is the Lipschitz bound $|f(x+y) - f(x)| < \lambda |y|$, and we have ignored overall numerical factors. Translating the s-dependence into N, this means that $L(N) \lesssim \frac{1}{N^{2/d}}$ up to a constant factor.

If the model is piecewise linear instead of piecewise constant and $f(x)$ is smooth with

[3] one might say conjecture; for a more sophisticated perspective in a simpler context see [16]

bounded derivatives, then the deviation $|f(x) - c(x)| \propto s^2$, and so the L^2 loss will scale[4] as s^4. We would predict

$$L(N) \propto \frac{1}{N^{4/d}} \tag{2.2.4}$$

This will be important later, since networks with ReLU activations produce piecewise linear functions.

Finally, consider the case where $f_i(x)$ encode a smooth probability distribution over $i = 1, \cdots, k$ possibilities, and we replace the MSE loss with the KL divergence. If the $c_i(x)$ are a piecewise linear model for the logits, then we also find that $L \propto s^4$. So the KL and MSE losses will scale with the same exponent in N at a given value of d. We demonstrate this in appendix A.1.5; it is a simple consequence of the fact that the expansion of $D_{KL}(p||q)$ in $(q - p)$ begins at second order. Note that if we use a cross-entropy instead of the KL divergence, the loss will scale in the same way towards a fixed constant value, the entropy of the true distribution.

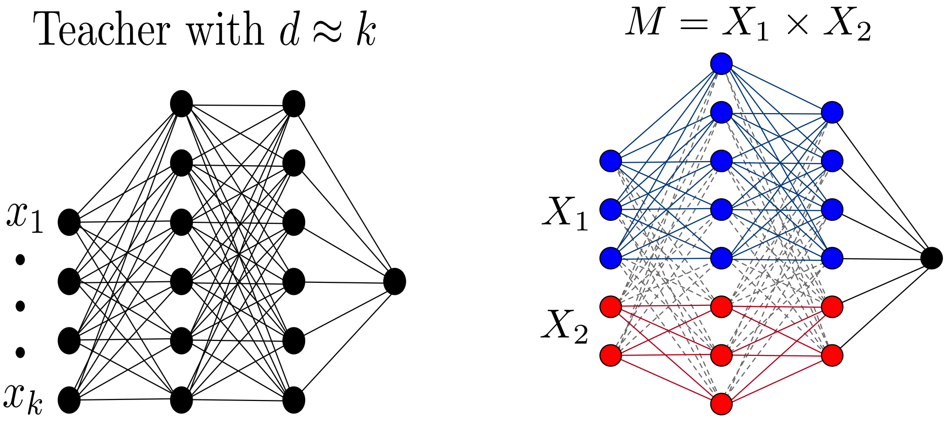

Figure 2.4: **Left**: This shows the setup of a teacher network, emphasizing how we can control the data manifold dimension via the number of input features k. **Right**: When the data manifold is a product and the teacher $T(X) = T_1(X_1)+T_2(X_2)$, then student networks can learn T by combining sub-networks and behaving, in effect, like an ensemble. Then we predict $4/\alpha \approx d_{\max}$, the maximum d among the components.

[4] A straightforward generalization suggests that if $c(x)$ is composed of piece-wise k-degree polynomials, and we use a loss $|f - c|^p$, then

$$L(s) \propto s^{(k+1)p} \tag{2.2.3}$$

in the infinite data limit. But if p is large then $c(x)$ within each hypercube will utilize many parameters. We test the p-dependence of this prediction in figure 2.8.

2.2.2 A Conjectural Theory for Neural Networks

Neural Networks perform well on data with thousands or even millions of dimensions. It is widely believed that this is possible because neural networks map the data into a much lower-dimensional 'data manifold', preserving and focusing on the features that are relevant for the task.

We emphasize that the data manifold is a feature of both the dataset and the task or loss function that has been optimized. Classifiers need only attend to features relevant for classification. Similarly, in the case of autoregressive models the data manifold would consist only of the features necessary to predict the next token in a sequence. So the data manifold for such a model (as we are defining it) may have many fewer dimensions than the space of full sequences, such as complete images or text samples. Properties of the data manifold may also depend on the model that is learning it, such as its architecture and activation functions.

We can explain the observed scaling relations for NNs by applying our toy theory while replacing the ambient dimension of the dataset with the intrinsic dimension of the data manifold. If we perform regression with a neural network with ReLU activations and a mean-squared error or KL divergence loss, the analysis of section 2.2.1 implies[5]

$$L(N) \propto \frac{1}{N^\alpha} \quad \text{with} \quad \alpha \approx \frac{4}{d} \tag{2.2.5}$$

In the case where the function $f(x)$ depends in a generic way on d independent variables, we will confirm this prediction empirically in section 2.3.1 (see figure 2.1). We also explore some special data manifolds and other loss functions in section 2.3.2.

This theory also largely explains why the scaling relation holds over such a large range of N. To double the resolution with which the model differentiates different points on the data manifold, we need 2^d times more parameters. It's reasonable to expect that model performance improves smoothly when we change the resolution by an order-one factor. But this seemingly natural assumption implies that if $d \gg 1$, we will see smooth scaling with N over many orders of magnitude. We would predict that the range in ΔN over which smooth scaling holds satisfies $\log(\Delta N) \propto d$. This also strongly suggests $\log N_{\max} \propto d$, where $N_{\max}$ is the largest network size exhibiting power-law scaling, as we do not expect $N_{\min}$, the beginning of the power-law region, to increase with d. We discuss some reasons why power-law scaling may cease in section 2.2.2.

Finally, the theory suggests an interpretation for the fact that different NN architectures tend to have similar scaling exponents when applied to the same dataset. It would appear

[5] Depending on the network architecture and parameter values, the network could represent a piecewise linear function with $C \gg N$ piecewise components [113]. However, these C components cannot be independently configured to optimize the loss. Since there are only N independent degrees of freedom available, we expect N, rather than C, to determine the effective capacity.

that a given dataset and task are associated with a data manifold of fixed dimension, and improvements in architecture do not greatly alter its properties. Network architectures that can achieve smaller d on the same dataset can be scaled up to achieve larger gains, and so we would expect smaller d to correlate with better performance.

The interpretation of $4/\alpha$ as the dimension of the data manifold has a close connection with the notion of fractal dimensions. Typically fractal dimensions measure how the number of components needed to approximate a fractal scales as the components shrink. But we can reinterpret this definition by asking how many components are needed to obtain a certain quality of approximation to the underlying fractal. When we use the loss itself to measure the quality of the approximation, then $4/\alpha$ is proportional to the corresponding fractal dimension.

Before moving on, let us discuss a few subtleties.

A Bound, Not an Equality

The classic analysis we reviewed in section 2.2.1 provides an upper bound on the loss for function approximation (regression in the infinite data limit) using piecewise constant or piecewise linear approximators. This bound becomes an estimate when the function being approximated is a generic Lipschitz function in d-dimensions. However, if the function has a simple, non-generic structure then the loss may decrease much more quickly with increasing model size. So we should expect that

$$\alpha \gtrsim \frac{4}{d} \tag{2.2.6}$$

In special cases where the true underlying function or distribution is non-generically simple, we may find that this inequality is far from saturation.

As a concrete example, consider a data manifold $M = X_1 \times X_2 \times \cdots \times X_n$ with loss $L(x) = \sum_i L_i(x_i)$, as suggested on the right of figure 2.4. In this case a fully connected neural network may learn[6] this decomposition, computing each $L_i(X_i)$ using a separate path through the network, and only combining these paths in the last layer. This would result in a scaling exponent determined by the maximum of the dimensions d_i of the manifolds X_i. We test $L(N)$ for product data manifolds in section 2.3.2 and verify these predictions.

We may end up finding $d > \frac{4}{\alpha}$ for other reasons. We will attempt to measure d among neural activations, but there may not be any single layer where the model compresses all of the data onto the data manifold. For example, one might imagine a scenario where different components of the manifold are processed or compressed in different layers of the network. And networks with non-ReLU activations (eg Transformers and ResNets) may mix and

[6] If the total loss does not decompose as a sum, it is less clear that the network can learn an effective decomposition, but it may still be possible.

superimpose different data manifolds upon each other, obscuring the manifold structure and causing the measured dimension to exceed the true dimension.

Why Does Power-Law Scaling Break Down?

If the dataset size is finite, then power-law scaling with model size N will cease when we begin to overfit the data. Overfitting dominates performance on many real-world datasets, obscuring potentially clean scalings with N. We encounter it with CIFAR10 in figure 2.9 and on other datasets in appendix A.1.4.

Even in the infinite data limit, if the data contains any entropy or noise then the power-law scaling must eventually end with the loss reaching a final plateau. Scaling could also end for other, more interesting reasons. For example, perhaps beyond a certain point the loss can only improve by exploring a higher dimensional data manifold. This is possible if the data manifold has a pancake-like structure, with a small width that can only be dissected by models with very large capacity. We will explore the simplest possibility, where the data has entropy, with mock teacher/student experiments; see figure 2.2 for the result.

2.2.3 Measuring the Intrinsic Dimension of the Data Manifold

In section 2.2.2 we extended the toy model in order to make a variety of predictions relating the scaling of the loss with model size to d, the intrinsic dimension (ID) of the data manifold. In some of our experiments, we will control d by constructing generic functions of d inputs and then measuring α. But the theory would be tautological for real-world data if we could not independently measure the data manifold's ID.

We will define d by measuring the ID of neural activations as the network processes data from the distribution on which it was trained. There is an extensive literature on intrinsic dimension estimation (for a review see [22]). In most cases we use the simple two-nearest neighbors (TwoNN) method [44], though we also compare to the MLE estimation [95] method on which TwoNN was based.

To summarize the method, let r_k be the distance from a given datapoint to its kth nearest neighbor, and define $\mu_k \equiv r_k/r_1$. Then the cumulative distribution $C(\mu_k)$ takes the form

$$C(\mu_k) = \left(1 - \frac{1}{\mu_k^d}\right)^{k-1} \tag{2.2.7}$$

and so we can measure the intrinsic dimension d by using the relation

$$d = \frac{\log\left(1 - C(\mu_k)^{\frac{1}{k-1}}\right)}{\log \mu_k} \tag{2.2.8}$$

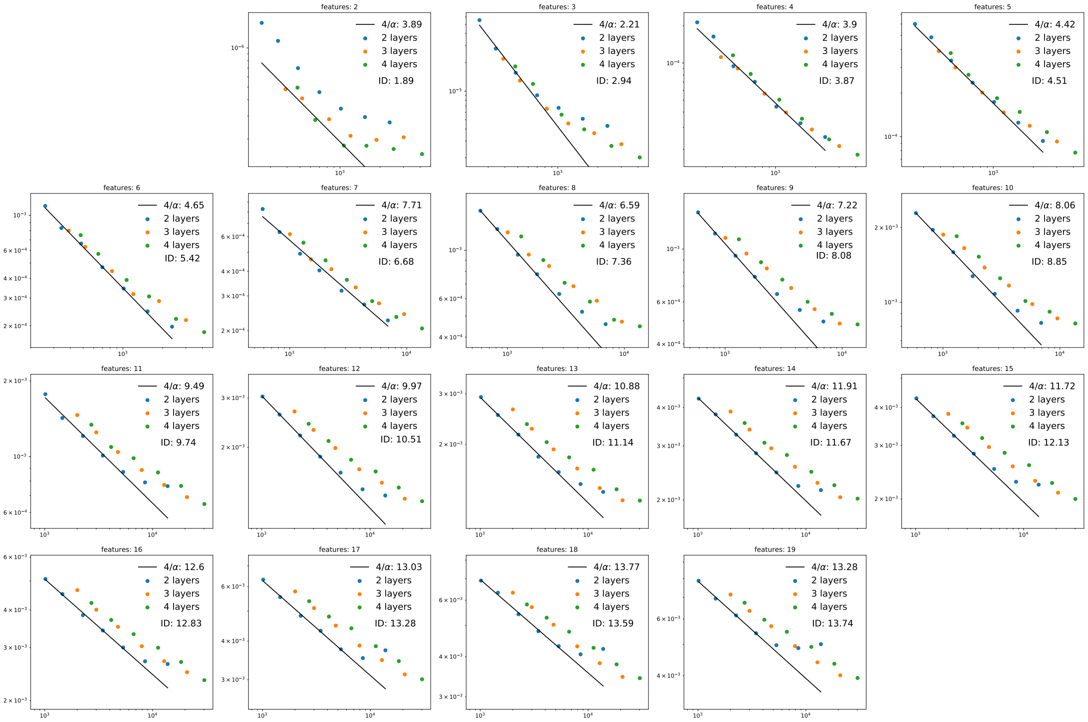

Figure 2.5: This figure shows $L(N)$ along with power-law fits for teacher/student experiments. The students learn from a randomly initialized 2-layer teacher with 2-19 features and use a cross-entropy loss. The students have 2,3, or 4 layers, but for $k > 5$ input features the 2-layer students perform best and determine the model-size scaling. The measured $4/\alpha$ increases linearly with the number of features, as shown in figure 2.6.

Practically speaking, we evaluate μ_k for every point on the manifold, and then apply linear regression to measure the slope d. We measure d using various k and verify that different values of k give consistent results. We also verify that the MLE method [95] agrees with the TwoNN method. Fortunately, nearest neighbors can be efficiently identified [21].

The TwoNN method (the case $k = 2$) has already been applied to neural networks [10]. There it was found that the dimension is smallest when measured using the activations of the final hidden layer of the network (immediately before the logits or output, so sometimes we refer to this as 'prefinal'). We will use these activations to measure d and compare to $1/\alpha$. For the GPT-type models (and for some others as a test in appendix A.3) we show ID measurements for every layer.

For convenience we provide a self-contained derivation of these ID measurement algorithms and a minor extension ($k > 2$) in appendix A.2. We also provide several tests of the method in appendix A.3, using both synthetic and neural activation data. We find that the method is fairly accurate for $d \lesssim 20$, while for larger dimensions it's less reliable, and typically (but not always) underestimates the true dimension. Statistical errors from these methods are often fairly small (particularly from TwoNN), but we expect there may be larger systematic errors, as discussed in the appendices.

2.3 Experiments and Results

In this section we discuss results from teacher/student experiments and various extensions, and also some tests using image classification and language modeling. We relegate a variety of technical details and a few minor observations to appendix A.1. We discuss potential errors in the ID measurement, along with several examples, in appendix A.3.

2.3.1 Teacher/Student with Random Teachers

We generate functions of $k = 2, 3, \cdots, 19$ input features using a randomly initialized, fully connected 'teacher' neural network with a 20-dimensional input space. To achieve $k < 20$ we simply zero out all other inputs to this single teacher. We refer to k as the number of features, and distinguish it from d, the intrinsic dimension, which we measure using the activations of trained student networks.

For each value of k, we train fully connected student networks of various widths and depths to imitate the outputs of the teacher. We work in the online setting, generating random inputs in $[-\frac{1}{2}, \frac{1}{2}]^k$ so the dataset size is effectively infinite. Details of the network topologies, training procedure, fits, errors, and ID measurements are documented in appendix A.1.2.

After training the students, we evaluate the loss $L_k(N)$ for each number of features k.

Then we fit

$$L_k(N) = \frac{c}{N^\alpha} \tag{2.3.1}$$

to measure c, α for each k. The results of this process (with cross-entropy loss) are shown in figure 2.5.

Next we measure the intrinsic dimension from the activations of the final hidden layer of each trained student. We use $12,000$ activation vectors for each ID measurement. In all cases we find that using more nearest neighbors, as discussed in section 2.2.3, does not change the result significantly. In figure 2.3 we show the measured ID of the final layer of a student network with various sizes N, along with a plot of the loss $L(N)$. We see that the ID is approximately constant for these networks, though it does slowly grow by about 10% from the smallest to the largest student network.

We plot the relationship between $4/\alpha$ and either the number of features or the measured ID d. The result, along with linear fits, are shown in figure 2.6. For both the cross-entropy and MSE loss functions, $\frac{4}{\alpha} \approx d$. The inverse exponent $1/\alpha$ is linearly related to the number of input features k, but the multiplier is larger than 4.

In section 2.2.2 we argued that scaling should end at an N_{max} that grows as $\log N_{\text{max}} \propto d$. We would like to test this prediction with teacher/student experiments, but in this case the data has no entropy. So instead we will introduce an artificial threshold for the loss, as a fictitious stand-in for the entropy of real data. Then we simply ask at what N_{max} the loss $L(N)$ reaches this fixed, arbitrary value.

We chose $L = 6 \times 10^{-3}$ as an arbitrary threshold in figure 2.2. Note that for the teacher networks with fewer features we used the power-law fit for $L(N)$ to estimate N_{max}, as it was smaller than any network tested. This means we had to extrapolate $L(N)$, so these results are not purely empirical. We also compare $\log N_{\text{max}}$ and d by defining N_{max} as the end of the purely empirical power-law scaling region for 2-layer students (due to a failure of optimization or numerical precision issues); these results are relegated to figure A.1 in the appendix.

The ID is typically a bit smaller than the number of input features. This may arise from a combination of two factors: the ID measurement may be underestimating the data manifold dimension, and randomly initialized networks may not provide sufficiently generic or non-linear functions of their inputs. We explore the second hypothesis in appendix A.1.3, where we show that by vetting the teacher networks we can improve agreement between ID and the number of input features. Figure A.7 provides some idea of the potential errors in the ID measurements. Since the inputs themselves are drawn from a uniform distribution it is plausible that the ID is somewhat of an underestimate due to boundary effects.

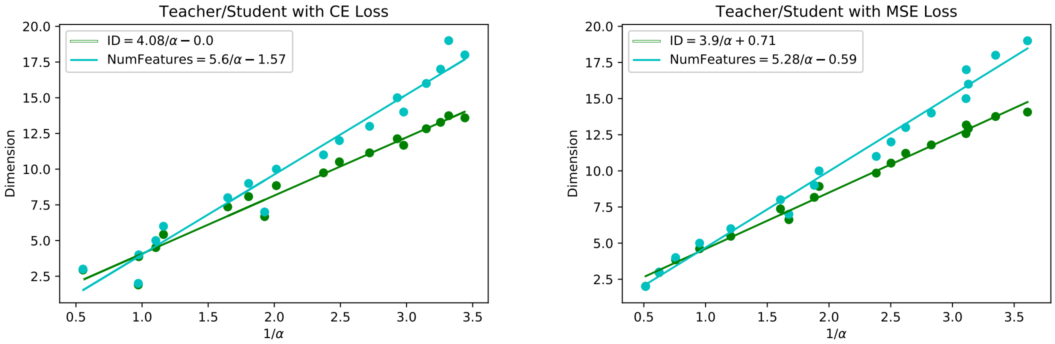

Figure 2.6: These figures show the correlation between the inverse scaling exponent $4/\alpha$ and both the measured intrinsic dimension and the number of input features (dimensions) in the teacher network. Both notions of dimension are linearly correlated with $1/\alpha$, and the intrinsic dimension scales almost exactly as $4/\alpha$, as predicted in section 2.2.2.

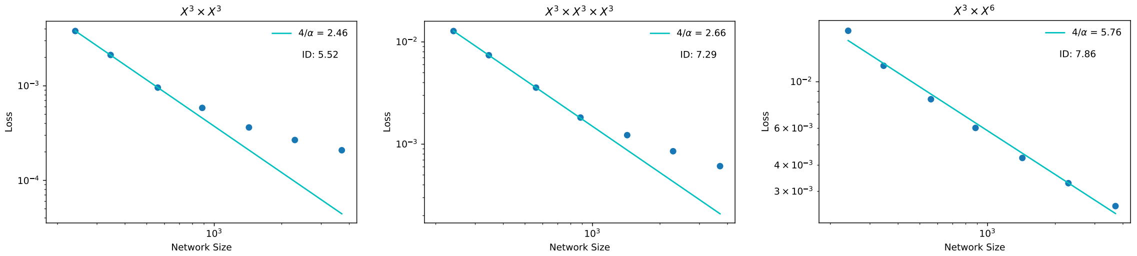

Figure 2.7: This figure shows results for α and d for product data manifolds with teachers T_{3+3} (left), T_{3+3+3} (middle), and T_{3+6} (right). We see that in all cases $\frac{4}{\alpha} \approx \max(d_i)$ among the product factor manifolds. The total measured IDs are approximately equal to the sum of the dimensions of the product factors, as expected.

2.3.2 Product Data Manifolds and Other Loss Functions

Product Data Manifolds $M = X_1 \times \cdots \times X_n$

If the data manifold takes the form $M = X_1 \times X_2 \times \cdots \times X_n$, with the underlying function of $x \in M$ decomposing as $F(x) = \sum_i f_i(x_i)$, then we expect that a neural network should be capable of separately modeling each f_i within separate blocks of activations, and then combining them in the final layer to compute the full F. This means that although the ID of M will be measured as $d_M = \sum_i d_{X_i}$, we should expect

$$\alpha \;=\; \frac{4}{\max(d_{X_i})} \tag{2.3.2}$$

as we discussed briefly in section 2.2.2, and demonstrate diagrammatically on the right of figure 2.4.

To test this prediction we use a vetted teacher network with 3 real inputs $T_3(x_1, x_2, x_3)$ and another vetted teacher taking 6 real inputs $T_6(x_1, \cdots x_6)$. Individually, these had ID $d_3 = 2.98$ and $d_6 = 5.31$ and their $L(N)$ exponents satisfied $\frac{4}{\alpha_3} = 3.3$ and $\frac{4}{\alpha_6} = 4.9$. These teachers each produce a pair of logits. We then constructed the new teacher functions with logits

$$\begin{aligned}
T_{3+3}(x) &= T_3(x_1, x_2, x_3) + T_3(x_4, x_5, x_6) \\
T_{3+3+3}(x) &= T_3(x_1, x_2, x_3) + T_3(x_4, x_5, x_6) + T_3(x_7, x_8, x_9) \\
T_{3+6}(x) &= T_3(x_1, x_2, x_3) + T_6(x_4, x_5, \cdots, x_9)
\end{aligned} \tag{2.3.3}$$

and trained students to imitate these teachers using the cross-entropy loss. We then measured the resulting ID and α for these three product-manifold teachers. For the T_{3+3} and T_{3+3+3} cases we used two or three different teachers to make sure the network could not take advantage of the exact repetition of a single teacher.

As shown in figure 2.7, the results confirm our predictions. This provides a concrete example where we may find that $\alpha > \frac{4}{d}$ for reasons that the theory precisely anticipates. More importantly, it provides a very detailed test of our theoretical picture relating scaling exponents to properties of the data manifold.

Other Loss Functions

The factor of '4' in the relation $d \approx \frac{4}{\alpha}$ is derived from the behavior of the loss function and the expectation that networks with ReLU activations form piecewise linear functions. If we use a loss function such as $L(y, y^*) = |y - y^*|^p$ for regression, from the argument of section 2.2.1 we would expect

$$\alpha \approx \frac{2p}{d} \tag{2.3.4}$$

where the MSE case corresponds to $p = 2$. We verify this in figure 2.8 using a fixed teacher with intrinsic dimension $d \approx 7$, as measured in the usual student/teacher context.

2.3.3 Image Classification with Simple CNNs

Our goal with these experiments was to study a simple, all ReLU architecture that could scale down to a small enough size to avoid overfitting CIFAR10 [88]. So we used a version of the default tutorial CNN in tensorflow [107], which we modified only by scaling the number of channels (ie the width). Figure 2.9 shows the scaling of the test loss with number of parameters N. Our only regularization was early stopping. The results match $4/\alpha = d$ quite well.

In an ideal test of the theory, we would measure α fully in the underfitting regime, with no distinction between train and test performance. But there is a train/test gap even for the smallest network sizes, so its unclear how to model the error in the α measurement. In addition to the test loss, we also measured the scaling of the training loss for these models, recording it at the early-stopping step, and found that it also scales similarly. Furthermore, note that on the right of figure 2.9 we record the error rate ($\equiv 1-$ accuracy), and find that it scales very similarly to the loss.

We performed a very similar analysis on the MNIST [91], fashion MNIST [162], and SVHN [120] datasets using slightly smaller networks (see section A.1.4). We plot $L(N)$ in figure A.4, which we have relegated to the appendix, as the power-law trends on these datasets are less clear than on CIFAR10.

Power-law exponents and IDs for CIFAR10 have been measured elsewhere using more powerful architectures, finding both a larger value of $\alpha \approx 0.5$ (for the error rate) [135] and a smaller ID ≈ 8 [10]. We cannot make a clean comparison, but given that we find that the exponent for error-rate and loss scaling seem to be similar, these results appear to match our predictions.

2.3.4 Language Modeling with GPT-type Models

The GPT-type language models display power-law scaling of $L(N)$ over at least five orders of magnitude in N, with exponent $\alpha \approx 0.076$ [81]. This value of α is much smaller than those observed for many other datasets [135], meaning that it allows us to probe a rather different regime, where we predict the quite large value $d \gtrsim 53$.

We generated activation vectors from the 'small' 117M parameter GPT-2 model using test data drawn from the same distribution as the training data [127, 128], and measured the IDs. Decoder-only [99] Transformers [152] have a residual structure with blocks including an attention mechanism and a fully-connected component. For each layer of blocks, one can measure the ID from the output of the attention mechanism, the fully-connected layer,

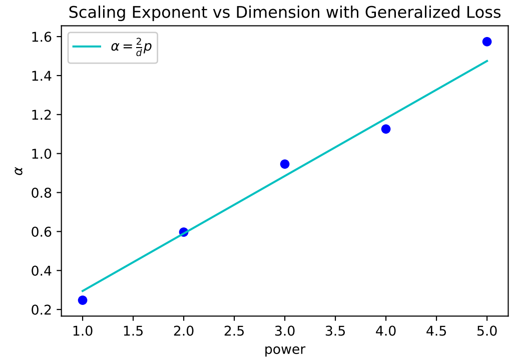

Figure 2.8: This figure shows the relationship between α and the power p when we use the generalized loss $|y - y^*|^p$. As expected from section 2.2.1, we find $\alpha = \frac{2}{d}p$. This is a student/teacher experiment with $d \approx 7$.

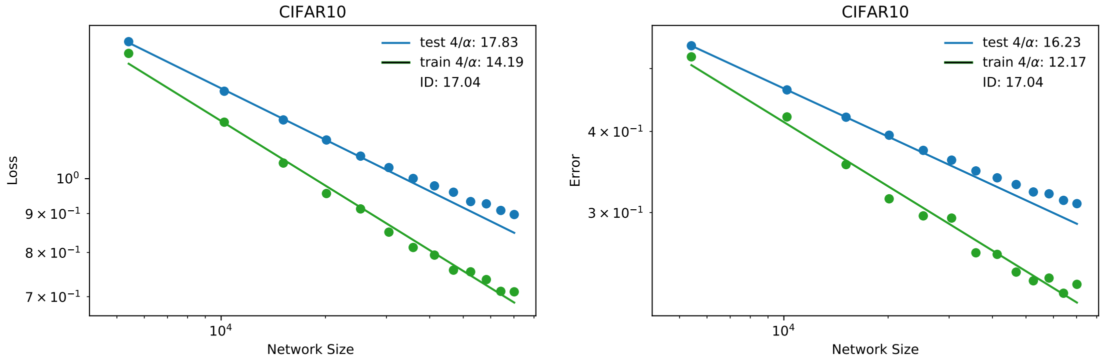

Figure 2.9: The left figure shows the test and training loss $L(N)$ for various sizes of CNN trained on CIFAR10, while the right figure shows error ($1-$ accuracy). All results are evaluated at the early stopping step, where the test loss is minimized. We report test loss results in figure 2.1, but note that the exponents for accuracy are very close to those for loss.

or from the output of the residual re-combination.

The activations that contribute to the Transformer's outputs at any given token-position depend on all activations from earlier in the sequence, except for the case of the final layer (before multiplying by the unembedding matrix). Thus it is only the final layer activations that can be said to capture the data manifold associated with the model's prediction for a single token. The mean loss over tokens has scaling exponent $\alpha \approx 0.076$, and from figure 21 of [81] we see that α is roughly constant for tokens that occur late in any text sequence. So we use the activations from the last token in each sequence to measure the ID, though the ID does not vary significantly across token positions (see figure 2.11).

In figure 2.10 we plot the measured ID for the attention output, the fully connected output, and the combined output of the residual blocks for all layers. For these measurements we used 10,000 activation vectors, each from the last token in a different text sequence (for more details see appendix A.3.2). We see that unlike the case of image classifiers [10], the ID is roughly constant across layers, with the exception of the first layer, where it is significantly smaller. If instead we measure the ID from the 1024 tokens in a single contiguous passage of text, we instead find an ID ≈ 7. This strongly suggests that the data manifold has a scale-dependent structure, and may not be well-characterized by a single intrinsic dimension.

It is tempting to observe that the intrinsic dimension of activations from the first attention layer is of order 50-80, which matches well with $4/\alpha$ for these models. One might argue that this bounds the total data manifold dimensionality entering the model through its input tokens. But as discussed above, this reasoning seems untrustworthy as an estimate of the data manifold dimensionality relevant for next-token predictions. So we take a conservative attitude and do not use early layer IDs as an estimate of the relevant ID for scaling.

We conclude that since $d > 90$, we have that $d \geq 4/\alpha \approx 53$, which accords with our expectations (see 2.2.2). Given the very small value of α in language modeling, it is satisfying to observe that the corresponding ID is very large. But it would have been more exciting to discover $\alpha \approx 4/d$ for language modeling. We do not know if the discepancy is due to added complexities from the structure of the Transformer, special structure on the data manifold itself, a scrambling of data manifolds due to the residual structure and attention mechanism, or some other oversimplification in our theory.

2.4 Related Work

The theory of scaling we have advocated applies basic, 'textbook' [154] ideas from regression and density estimation. Our work was also partly inspired by similar scaling relations in random forest models; with some added assumptions, it is possible to prove them [15]. As one passes from classical techniques, to random forests, and then to neural networks,

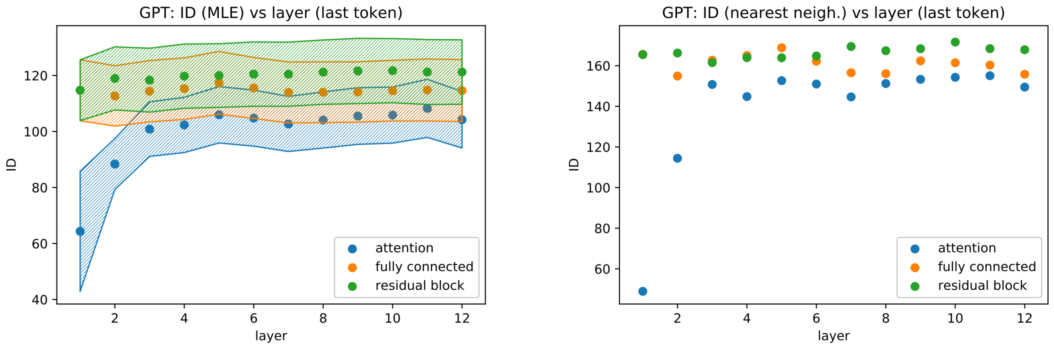

Figure 2.10: These figures show the ID estimates for the attention and fully-connected outputs of a 117M parameter GPT-type model, where $4/\alpha \approx 53$. The left figure shows results from the nearest neighbor method, with 2,3, and 4 neighbors, while the right plot shows results from the MLE method. The results roughly agree for the first layer, but the MLE method gives smaller IDs for later layers, and is likely an under-estimate.

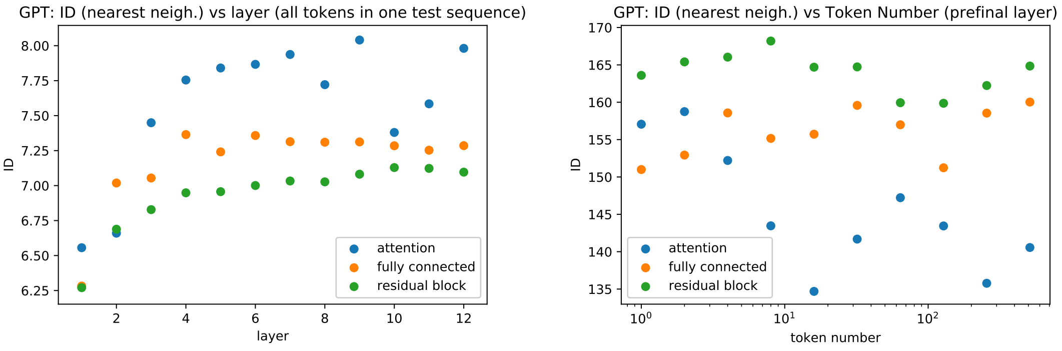

Figure 2.11: ID estimates from a single 1024-token text sequence (left) and the final layer ID as measured using tokens with fixed positions within distinct sequences (right). The data manifold associated with a single sequence has a much, much smaller dimension than the full manifold.

the models become increasingly powerful but less and less amenable to a direct analysis. Nevertheless, we argue that similar principles apply and underly their scaling behavior. A similar overall perspective has been discussed by Bickel and collaborators [16].

There is a large literature on dimensionality estimation; for a nice overview see [22]. We have primarily used the two nearest neighbor method [44], which was based on the MLE method [95] for distances among points in a local neighborhood. In neural image classifiers, the intrinsic dimension of the data manifold was studied [10] using the TwoNN method. They demonstrated that the ID is much smaller than the dimension estimated via linear methods such as PCA, among other interesting results. Other authors have established a connection between ID and noisy labels [102], and demonstrated that neural models can effectively identify a low-dimensional manifold in a larger ambient space [13]. It would be interesting to understand the relationship between the data manifold and neural circuits [123], and how the manifold changes when non-robust features are eliminated [74]. Recent work [144] relates data dimensionality and dataset size scaling exponents for kernel methods. The intrinsic dimension of the neural network parameter space has also been discussed [98].

Neural scaling laws have been studied in a number of papers. Perhaps the first work on the subject was [69]. The more recent work [135] studies scaling with model size and dataset size, both independently and simultaneously. Language models were studied in [81], where scaling relations with model size, dataset size, training compute, and training steps were identified. EfficientNet [149] displays near power-law scaling with model size, though these models are not in the underfitting regime.

2.5 Discussion

We have proposed a theory connecting the model-size scaling exponent with the intrinsic dimension of the data manifold. Many other neural scaling laws have been identified [69, 135, 81], including scalings with dataset size and compute budget, and fairly accurate power-law fits to learning curves. We have focused on scaling with model size in the infinite data limit because we expect it to be the simplest and most theoretically tractable scaling relation. Scaling with dataset size may involve issues of regularization, requiring a balance between bias and variance, while understanding the scaling with compute would require that we contend with optimization.

Nevertheless, neural scaling exponents with dataset size are often very similar[7] to model size exponents. One might argue that dataset size scaling can be understood as a consequence of interpolation between points on the data manifold, and so should have a similar relationship to the data manifold dimension. Recent works have made this case [144].

[7] Though in almost all cases [135, 81] dataset exponents are slightly larger. This runs somewhat counter to classical expectations [154], where the number of parameters determines a tradeoff between bias and variance, and dataset size exponents are smaller than the bias-scaling exponents that depend on model size.

Compute scaling exponents [81] are also not far from model-size exponents, but combine optimization and model scaling. It seems most natural to interpret them by modeling learning curves, but perhaps optimization can be re-interpreted as the identification and dissection of the data manifold. Something like this will be necessary in order to explain the fact that larger models are much more sample efficient [81] than small models. This may be the most impactful direction for future work.

It will be interesting to test this theory with a wider variety of models and datasets. Generative modeling may be the ideal setting, since the abundance of unlabeled text, image, and video data provides many opportunities to train large models on nearly unlimited datasets. In this context, it may be interesting to explore what the theory suggests for finetuning pre-trained generative models on downstream tasks. We would expect that these tasks benefit from the pre-established existence of the data manifold; perhaps finetuning can be understood as a process of zooming-in and refining performance in a small region of this manifold. It would also be interesting to understand how scaling relations for the loss compare to those for quantities that are not directly optimized, such as prediction accuracies. In the case of CIFAR10 we saw that accuracy and loss exhibit similar exponents. Finally, it's worth thinking about the extent to which larger models perform better in reinforcement learning [31]. Due to the non-stationary distribution in RL it may be difficult to understand model-size scaling quantitatively, and it's less clear how to apply our theory in that context. A theory of sample efficiency scaling would be more likely to be relevant to RL.

Acknowledgements

We thank Yasaman Bahri, Ethan Dyer, Tom Henighan, Danny Hernandez, Jaehoon Lee, and Sam McCandlish for interesting discussions and feedback. We especially thank Ethan for sharing his notes on linear models and Yasaman for emphasizing that our theory of model size scaling might be re-purposed as a theory of dataset size scaling. JK has been supported in part by NSF grant PHY-1454083. This work was also supported in part by Open Philanthropy.

Chapter 3

Explaining Neural Scaling Laws

This thesis chapter originally appeared in the literature as:

Y. Bahri, E. Dyer, J, Kaplan, J. Lee, U. Sharma, “Explaining Neural Scaling Laws,” *arXiv*: 2102.06701

Abstract

The test loss of well-trained neural networks often follows precise power-law scaling relations with either the size of the training dataset or the number of parameters in the network. We propose a theory that explains and connects these scaling laws. We identify *variance-limited* and *resolution-limited* scaling behavior for both dataset and model size, for a total of four scaling regimes. The variance-limited scaling follows simply from the existence of a well-behaved infinite data or infinite width limit, while the resolution-limited regime can be explained by positing that models are effectively resolving a smooth data manifold. In the large width limit, this can be equivalently obtained from the spectrum of certain kernels, and we present evidence that large width and large dataset resolution-limited scaling exponents are related by a duality. We exhibit all four scaling regimes in the controlled setting of large random feature and pretrained models and test the predictions empirically on a range of standard architectures and datasets. We also observe several empirical relationships between datasets and scaling exponents: super-classing image tasks does not change exponents, while changing input distribution (via changing datasets or adding noise) has a strong effect. We further explore the effect of architecture aspect ratio on scaling exponents.

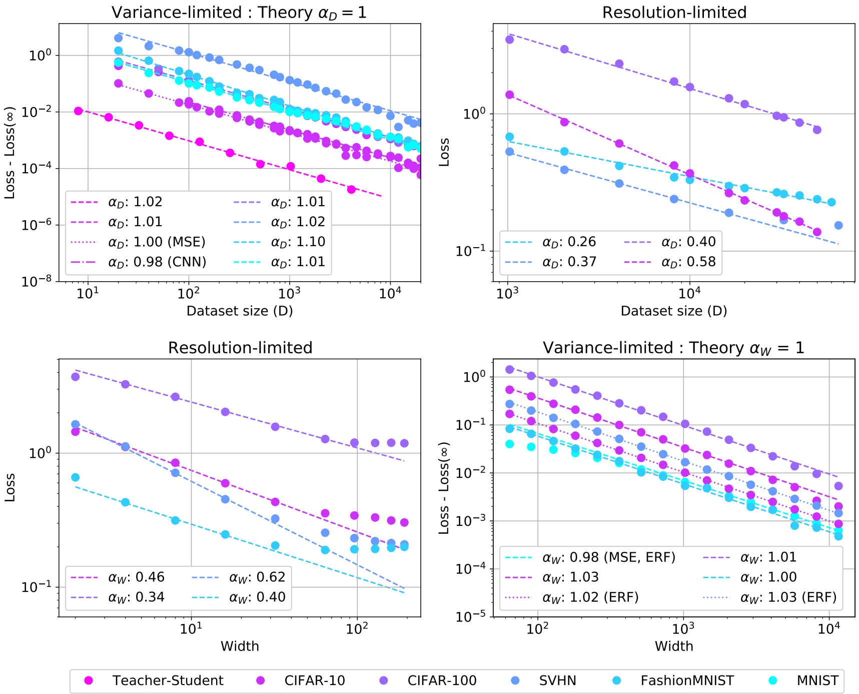

Figure 3.1: **Four scaling regimes** Here we exhibit the four regimes we focus on in this work. **(top-left, bottom-right)** *Variance-limited* scaling of under-parameterized models with dataset size and over-parameterized models with number of parameters (width) exhibit universal scaling ($\alpha_D = \alpha_W = 1$) independent of the architecture or underlying dataset. **(top-right, bottom-left)** *Resolution-limited* over-parameterized models with dataset or under-parameterized models with model size exhibit scaling with exponents that depend on the details of the data distribution. These four regimes are also found in random feature (Figure 3.3) and pretrained models (see appendices).

3.1 Scaling Laws for Neural Networks

For a large variety of models and datasets, neural network performance has been empirically observed to scale as a power-law with model size and dataset size [70, 81, 136, 67]. We would like to understand why these power laws emerge, and what features of the data and models determine the values of the power-law exponents. Since these exponents determine how quickly performance improves with more data and larger models, they are of great importance when considering whether to scale up existing models.

In this work, we present a theoretical framework for explaining scaling laws in trained neural networks. We identify four related scaling regimes with respect to the number of model parameters P and the dataset size D. With respect to each of D, P, there is both a *resolution-limited* regime and a *variance-limited* regime.

Variance-Limited Regime In the limit of infinite data or an arbitrarily wide model, some aspects of neural network training simplify. Specifically, if we fix one of D, P and

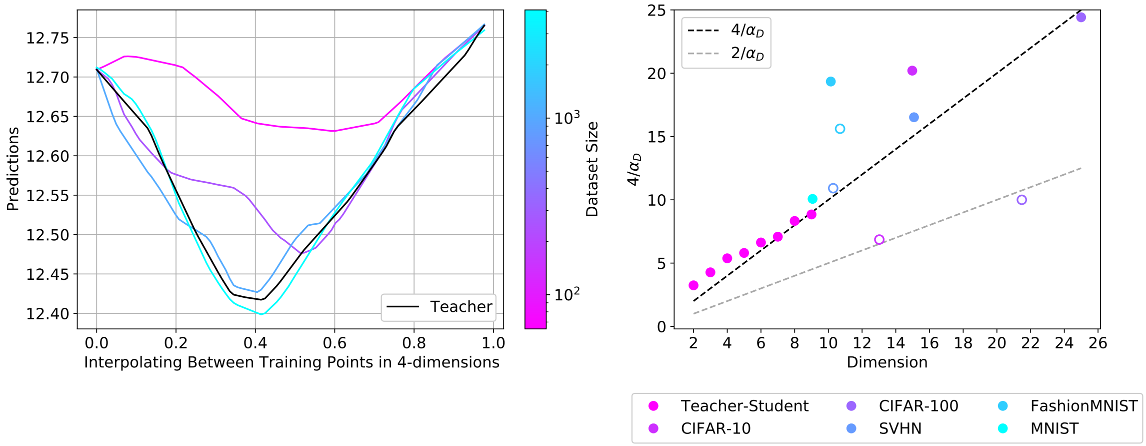

Figure 3.2: **Resolution-limited models interpolate the data manifold** Linear interpolation between two training points in a four-dimensional input space **(left)**. We show a teacher model and four student models, each trained on different sized datasets. In all cases teacher and student approximately agree on the training endpoints, but as the training set size increases they increasingly match everywhere. **(right)** We show $4/\alpha_D$ versus the data manifold dimension (input dimension for teacher-student models, intrinsic dimension for standard datasets). We find that the teacher-student models follow the $4/\alpha_D$ (dark dashed line), while the relationship for a four layer CNN (solid) and WRN (hollow) on standard datasets is less clear.

study scaling with respect to the other parameter as it becomes arbitrarily large, then the loss scales as $1/x$, i.e. as a power-law with exponent 1, with $x = D$ or $\sqrt{P} \propto$ width in deep networks and $x = D$ or P in linear models. In essence, this *variance-limited* regime is amenable to analysis because model predictions can be series expanded in either inverse width or inverse dataset size. To demonstrate these variance-limited scalings, it is sufficient to argue that the infinite data or width limit exists and is smooth; this guarantees that an expansion in simple integer powers exists.

Resolution-Limited Regime In this regime, one of D or P is effectively infinite, and we study scaling as the *other* parameter increases. In this case, a variety of works have empirically observed power-law scalings $1/x^\alpha$, typically with $0 < \alpha < 1$ for both $x = P$ or D.

We can provide a very general argument for power-law scalings if we assume that trained models map the data into a d-dimensional data manifold. The key idea is then that additional data (in the infinite model-size limit) or added model parameters (in the infinite data limit) are used by the model to carve up the data manifold into smaller components. The model then makes independent predictions in each component of the data manifold in order to optimize the training loss.

If the underlying data varies continuously on the manifold, then the size of the subregions into which we can divide the manifold (rather than the number of regions) determines the model's loss. To shrink the size of the sub-regions by a factor of 2 requires increasing the parameter count or dataset size by a factor of 2^d, and so the inverse of the scaling exponent will be proportional to the intrinsic dimension d of the data manifold, so that $\alpha \propto 1/d$. A visualization of this successively better approximation with dataset size is shown in Figure 3.2 for models trained to predict data generated by a random fully-connected network.

Explicit Realization These regimes can be realized in linear models, and this includes linearized versions of neural networks via the large width limit. In these limits, we can solve for the test error directly in terms of the feature covariance (kernel). The scaling of the test loss then follows from the asymptotic decay of the spectrum of the covariance matrix. Furthermore, well-known theorems provide bounds on the spectra associated with continuous kernels on a d-dimensional manifold. Since otherwise generic kernels saturate these bounds, we find a tight connection between the dimension of the data manifold, kernel spectra, and scaling laws for the test loss. We emphasize, this analysis relies on an implicit model of realistic data only through the assumption of a generic, power law kernel spectrum.

Summary of Contributions:

1. We identify four scaling regions of neural networks and provide empirical support for all four regions for deep models on standard datasets. To our knowledge, the variance-limited dataset scaling has not been exhibited previously for deep networks on realistic data.

2. We present simple yet general theoretical assumptions under which we can derive this scaling behavior. In particular, we relate the scaling exponent in the resolution-limited regime to the *intrinsic dimension* of the *data-manifold* realized by trained networks representations.

3. We present a concrete solvable example where all four scaling behaviors can be observed and understood: linear, random-feature teacher-student models.

4. We empirically investigate the dependence of the scaling exponent on changes in architecture and data. We find that changing the input distribution via switching datasets, or the addition of noise has a strong effect on the exponent, while changing the target distribution via superclassing does not.

3.1.1 Related Works

There have been a number of recent works demonstrating empirical scaling laws [70, 81, 136, 67, 137] in deep neural networks, including scaling laws with model size, dataset size, compute, and other observables such as mutual information and pruning. Some precursors [5, 33] can be found in earlier literature.

There has been comparatively little work on theoretical ideas [141] that match and explain empirical findings in generic deep neural networks across a range of settings. In the particular case of large width, deep neural networks behave as random feature models [119, 92, 108, 75, 94, 42], and known results on the loss scaling of kernel methods can be applied [145, 17]. During the completion of this work [73] presented a solvable model of learning exhibiting non-trivial power-law scaling for power-law (Zipf) distributed features.

In the variance-limited regime, scaling laws in the context of random feature models [129, 64, 37], deep linear models [3, 4], one-hidden-layer networks [111, 1, 2], and wide neural networks treated as Gaussian processes or trained in the NTK regime [94, 42, 9, 53] have been studied. In particular, this behavior was used in [81] to motivate a particular ansatz for simultaneous scaling with data and model size.

This work also makes use of classic results connecting the spectrum of a smooth kernel to the geometry it is defined over [157, 130, 89, 46] and on the scaling of iteratively refined approximations to smooth manifolds [146, 16, 90].

Recently, scaling laws have also played a significant role in motivating work on the largest models that have yet been developed [20, 45].

3.2 Theory

Throughout this work we will be interested in how the average test loss $L(D, P)$ depends on the dataset size D and the number of model parameters P. Unless otherwise noted, L denotes the test loss averaged over model initializations and draws of a size D training set. Some of our results only pertain directly to the scaling with width $w \propto \sqrt{P}$, but we expect many of the intuitions apply more generally. We use the notation α_D, α_P, and α_W to indicate scaling exponents with respect to dataset size, parameter count, and width.

3.2.1 Variance-Limited Exponents

In the limit of large D the outputs of an appropriately trained network approach a limiting form with corrections which scale as D^{-1}. Similarly, recent work shows that wide networks have a smooth large P limit, [75], where fluctuations scale as $1/\sqrt{P}$. If the loss is analytic about this limiting model then its value will approach the asymptotic loss with corrections proportional to the variance, ($1/D$ or $1/\sqrt{P}$). Let us discuss this in a bit more detail for both cases.

Dataset scaling

Consider a neural network, and its associated training loss $L_{\text{train}}(\theta)$. For every value of the weights, the training loss, thought of as a random variable over draws of a training set of size D, concentrates around the population loss, with a variance which scales as $\mathcal{O}\left(D^{-1}\right)$. Thus, if the optimization procedure is sufficiently smooth, the trained weights, network output, and test loss will approach their infinite D values plus an $\mathcal{O}\left(D^{-1}\right)$ contribution.

As a concrete example, consider training a network via full-batch optimization. In the limit that $D \to \infty$, the gradients will become exactly equal to the gradient of the population loss. When D is large but finite, the gradient will include a term proportional to the $\mathcal{O}(D^{-1})$ variance of the loss over the dataset. This means that the final parameters will be equal to the parameters from the $D \to \infty$ limit of training plus some term proportional to D^{-1}. This also carries over to the test loss.

Since this argument applies to any specific initialization of the parameters, it also applies when we take the expectation of the test loss over the distribution of initializations. We do not prove the result rigorously at finite batch size. We expect it to hold however, in expectation over instances of stochastic optimization, provided hyper-parameters (such as batch size) are fixed as D is taken large.

Large Width Scaling

We can make a very similar argument in the $w \to \infty$ or large width limit. It has been shown that the predictions from an infinitely wide network, either at initialization [119, 92], or when trained via gradient descent [75, 94] approach a limiting distribution equivalent to training a linear model. Furthermore, corrections to the infinite width behavior are controlled by the variance of the full model around the linear model predictions. This variance has been shown to scale as $1/w$ [42, 164, 9]. As the loss is a smooth function of these predictions, it will differ from its $w = \infty$ limit by a term proportional to $1/w$.

We note that there has also been work studying the combined large depth and large width limit, where [62] found a well-defined infinite size limit with controlled fluctuations. In any such context where the model predictions concentrate, we expect the loss to scale with the variance of the model output. In the case of linear models, studied below, the variance is $\mathcal{O}(P^{-1})$ rather than $\mathcal{O}(\sqrt{P})$ and we see the associated variance scaling in this case.

3.2.2 Resolution-Limited Exponents

In this section we consider training and test data drawn uniformly from a compact d-dimensional manifold, $x \in \mathcal{M}_d$ and targets given by some smooth function $y = \mathcal{F}(x)$ on this manifold.

Over-parameterized dataset scaling

Consider the double limit of an over-parameterized model with large training set size, $P \gg D \gg 1$. We further consider *well trained* models, i.e. models that interpolate all training data. The goal is to understand $L(D)$. If we assume that the learned model f is sufficiently smooth, then the dependence of the loss on D can be bounded in terms of the dimension of the data manifold $\mathcal{M}_d$.

Informally, if our train and test data are drawn i.i.d. from the same manifold, then the distance from a test point to the closest training data point decreases as we add more and more training data points. In particular, this distance scales as $\mathcal{O}(D^{-1/d})$ [95]. Furthermore, if f, $\mathcal{F}$ are both sufficiently smooth, they cannot differ too much over this distance. If in addition the loss function, L, is a smooth function vanishing when $f = \mathcal{F}$, we have $L = \mathcal{O}(D^{-1/d})$. This is summarized in the following theorem.

Theorem 1. *Let $L(f)$, f and $\mathcal{F}$ be Lipschitz with constants K_L, K_f, and $K_{\mathcal{F}}$. Further let $\mathcal{D}$ be a training dataset of size D sampled i.i.d from $\mathcal{M}_d$ and let $f(x) = \mathcal{F}(x)$, $\forall x \in \mathcal{D}$ then $L(D) = \mathcal{O}\left(K_L max(K_f, K_{\mathcal{F}}) D^{-1/d}\right)$.*

Under-Parameterized Parameter Scaling

We will again assume that $\mathcal{F}$ varies smoothly on an underlying compact d-dimensional manifold $\mathcal{M}_d$. We can obtain a bound on $L(P)$ if we imagine that f approximates $\mathcal{F}$ as a piecewise linear function with roughly P regions (see [141]). Here, we instead make use of the argument from the over-parameterized, resolution-limited regime above. If we construct a sufficiently smooth estimator for $\mathcal{F}$ by interpolating among P randomly chosen points from the (arbitrarily large) training set, then by the argument above the loss will be bounded by $\mathcal{O}(P^{-1/d})$.

Theorem 2. *Let $L(f)$, f and $\mathcal{F}$ be Lipschitz with constants K_L, K_f, and $K_{\mathcal{F}}$. Further let $f(x) = \mathcal{F}(x)$ for P points sampled i.i.d from $\mathcal{M}_d$ then $L(P) = \mathcal{O}\left(K_L max(K_f, K_{\mathcal{F}}) P^{-1/d}\right)$.*

We provide the proof of Theorem 1 and 2 in the appendices.

From Bounds to Estimates

Theorems 1 and 2 are phrased as bounds, but we expect the stronger statement that these bounds also generically serve as estimates, so that eg $L(D) = \Omega(D^{-c/d})$ for $c \geq 2$, and similarly for parameter scaling. If we assume that $\mathcal{F}$ and f are analytic functions on $\mathcal{M}_d$ and that the loss function $L(f, \mathcal{F})$ is analytic in $f - \mathcal{F}$ and minimized at $f = \mathcal{F}$, then the loss at a given test input, x_{test}, can be expanded around the nearest training point, $\hat{x}_{\text{train}}$.[1]

$$L(x_{\text{test}}) = \sum_{m=n\geq 2}^{\infty} a_m(\hat{x}_{\text{train}})(x_{\text{test}} - \hat{x}_{\text{train}})^m\,, \tag{3.2.1}$$

where the first term is of finite order $n \geq 2$ because the loss vanishes at the training point. As the typical distance between nearest neighbor points scales as $D^{-1/d}$ on a d-dimensional manifold, the loss will be dominated by the leading term, $L \propto D^{-n/d}$, at large D. Note that if the model provides an accurate piecewise linear approximation, we will generically find $n \geq 4$.

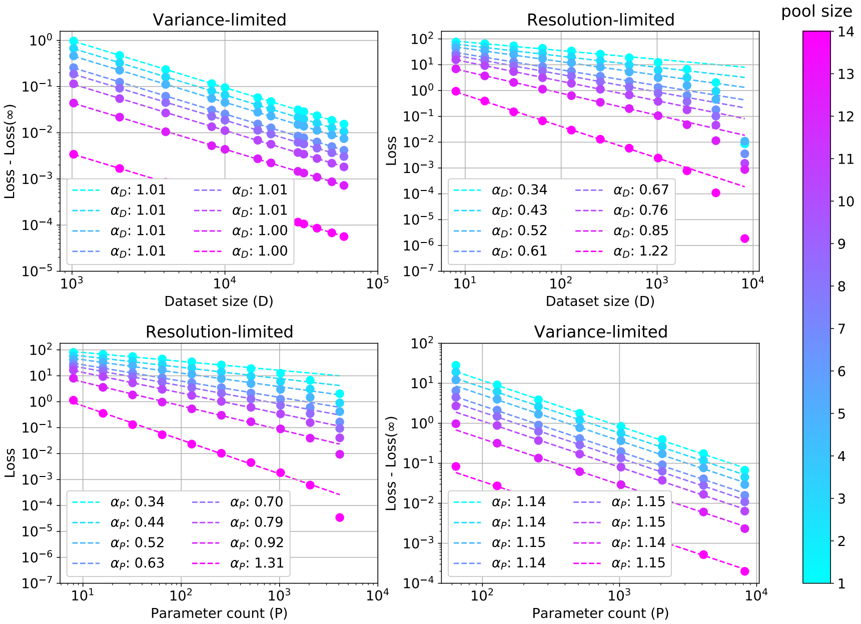

Figure 3.3: **Random feature models exhibit all four scaling regimes** Here we consider linear teacher-student models with random features trained with MSE loss to convergence. We see both *variance-limited* scaling **(top-left, bottom-right)** and *resolution-limited* scaling **(top-right, bottom-left)**. Data is varied by downsampling MNIST by the specified pool size.

3.2.3 Kernel realization

In the proceeding sections we have conjectured typical case scaling relations for a model's test loss. We have further given intuitive arguments for this behavior which relied on smoothness assumptions about the loss and training procedure. In this section, we provide a concrete realization of all four scaling regimes within the context of linear models. Of particular interest is the resolution-limited regime, where the scaling of the loss is a consequence of the linear model kernel spectrum – the scaling of over-parameterized models

[1] For simplicity we have used a very compressed notation for multi-tensor contractions in higher order terms

with dataset size and under-parameterized models with parameters is a consequence of a classic result, originally due to [157], bounding the spectrum of sufficiently smooth kernel functions by the dimension of the manifold they act on.

Linear predictors serve as a model system for learning. Such models are used frequently in practice when more expressive models are unnecessary or infeasible [110, 131, 63] and also serve as an instructive test bed to study training dynamics [4, 56, 64, 117, 58]. Furthermore, in the large width limit, randomly initialized neural networks become Gaussian Processes [119, 92, 108, 121, 52, 165], and in the low-learning rate regime [94, 97, 72] neural networks train as linear models at infinite width [75, 94, 28].

Here we discuss linear models in general terms, though the results immediately hold for the special cases of wide neural networks. In this section we focus on teacher-student models with weights initialized to zero and trained with mean squared error (MSE) loss to their global optimum.

We consider a linear teacher, F, and student f.

$$F(x) = \sum_{M=1}^{S} \omega_M F_M(x), \quad f(x) = \sum_{\mu=1}^{P} \theta_\mu f_\mu(x)\,. \tag{3.2.2}$$

Here $\{F_M\}$ are a (potentially infinite) pool of features and the teacher weights, ω_M are taken to be normal distributed, $\omega \sim \mathcal{N}(0, 1/S)$.

The student model is built out of a subset of the teacher features. To vary the number of parameters in this simple model, we construct P features, $f_{\mu=1,\ldots,P}$, by introducing a projector $\mathcal{P}$ onto a P-dimensional subspace of the teacher features, $f_\mu = \sum_M \mathcal{P}_{\mu M} F_M$.

We train this model by sampling a training set of size D and minimizing the MSE training loss,

$$L_{\text{train}} = \frac{1}{2D} \sum_{a=1}^{D} \left(f(x_a) - F(x_a)\right)^2\,. \tag{3.2.3}$$

We are interested in the test loss averaged over draws of our teacher and training dataset. In the limit of infinite data, the test loss, $L(P) := \lim_{D\to\infty} L(D, P)$, takes the form.

$$L(P) = \frac{1}{2S} \mathrm{Tr}\left[\mathcal{C} - \mathcal{C}\mathcal{P}^T \left(\mathcal{P}\mathcal{C}\mathcal{P}^T\right)^{-1} \mathcal{P}\mathcal{C}\right]\,. \tag{3.2.4}$$

Here we have introduced the feature-feature second moment-matrix, $\mathcal{C} = \mathbb{E}_x\left[F(x)F^T(x)\right]$.

If the teacher and student features had the same span, this would vanish, but as a result of the mismatch the loss is non-zero. On the other hand, if we keep a finite number of training points, but allow the student to use all of the teacher features, the test loss, $L(D) := \lim_{P\to S} L(D, P)$, takes the form,

$$L(D) = \frac{1}{2}\mathbb{E}_x\left[\mathcal{K}(x,x) - \vec{\mathcal{K}}(x)\bar{\mathcal{K}}^{-1}\vec{\mathcal{K}}(x)\right]\,. \tag{3.2.5}$$

Here, $\mathcal{K}(x, x')$ is the data-data second moment matrix, $\vec{\mathcal{K}}$ indicates restricting one argument to the D training points, while $\bar{\mathcal{K}}$ indicates restricting both. This test loss vanishes as the number of training points becomes infinite but is non-zero for finite training size.

We present a full derivation of these expressions in the appendices. In the remainder of this section, we explore the scaling of the test loss with dataset and model size.

Kernels: Variance-Limited exponents

To derive the limiting expressions (3.2.4) and (3.2.5) for the loss one makes use of the fact that the sample expectation of the second moment matrix over the finite dataset, and finite feature set is close to the full covariance.

$$\frac{1}{D}\sum_{a=1}^{D} F(x_a)F^T(x_a) \,=\, \mathcal{C} + \delta\mathcal{C}\,, \quad \frac{1}{P} f^T(x)f(x'), = \,\mathcal{K} + \delta\mathcal{K}\,,$$

with the fluctuations satisfying $\mathbb{E}_D\left[\delta C^2\right] = \mathcal{O}(D^{-1})$ and $\mathbb{E}_P\left[\delta K^2\right] = \mathcal{O}(P^{-1})$, where expectations are taken over draws of a dataset of size D and over feature sets.

Using these expansions yields the variance-limited scaling, $L(D, P) - L(P) = \mathcal{O}(D^{-1})$, $L(D, P) - L(D) = \mathcal{O}(P^{-1})$ in the under-parameterized and over-parameterized settings respectively.

In Figure 3.3 we see evidence of these scaling relations for features built from randomly initialized ReLU networks on pooled MNIST independent of the pool size. In the appendices we provide an in depth derivation of this behavior and expressions for the leading contributions to $L(D, P) - L(P)$ and $L(D, P) - L(D)$.

Kernels: Resolution-limited exponents

We now would like to analyze the scaling behavior of our linear model in the resolution-limited regimes, that is the scaling with P when $1 \ll P \ll D$ and the scaling with D when $1 \ll D \ll P$. In these cases, the scaling is controlled by the shared spectrum of $\mathcal{C}$ or $\mathcal{K}$. This spectrum is often well described by a power-law, where eigenvalues λ_i satisfy

$$\lambda_i = \frac{1}{i^{1+\alpha_K}}\,. \tag{3.2.6}$$

See Figure 3.4 for example spectra on pooled MNIST.

In this case, we will argue that the losses also obey a power law scaling, with the exponents controlled by the spectral decay factor, $1 + \alpha_K$.

$$L(D) \propto D^{-\alpha_K}\,, \quad L(P) \propto P^{-\alpha_K}\,. \tag{3.2.7}$$

In other words, in this setting, $\alpha_P = \alpha_D = \alpha_K$.

This is supported empirically in Figure 3.4. We then argue that when the kernel function, $\mathcal{K}$ is sufficiently smooth on a manifold of dimension d, $\alpha_K \propto d^{-1}$, thus realizing the more general resolution-limited picture described above.

From spectra to scaling laws for the loss To be concrete let us focus on the over-parameterized loss. If we introduce the notation e_i for the eigenvectors of $\mathcal{C}$ and $\bar{e}_i$ for the eignvectors of $\frac{1}{D}\sum_{a=1}^{D} F(x_a)F^T(x_a)$, the loss becomes,

$$L(D) = \frac{1}{2}\sum_{i=1}^{S} \lambda_i (1 - \sum_{j=1}^{D} (e_i \cdot \bar{e}_j)^2)\,. \tag{3.2.8}$$

Before discussing the general asymptotic behavior of (3.2.8), we can gain some intuition by considering the case of large α_K. In this case, $\bar{e}_j \approx e_j$ (see e.g. [101]), we can simplify (3.2.8) to,

$$L(D) \propto \sum_{D+1}^{\infty} \frac{1}{i^{1+\alpha_K}} = \alpha_K D^{-\alpha_K} + \mathcal{O}(D^{-\alpha_K - 1})\,. \tag{3.2.9}$$

More generally in the appendices, following [17, 23], we use replica theory methods to derive, $L(D) \propto D^{-\alpha_K}$ and $L(P) \propto P^{-\alpha_K}$, without requiring the large α_K limit.

Data Manifolds and Kernels In Section 3.2.2, we discussed a simple argument that resolution-limited exponents $\alpha \propto 1/d$, where d is the dimension of the data manifold. Our goal now is to explain how this connects with the linearized models and kernels discussed above: how does the spectrum of eigenvalues of a kernel relate to the dimension of the data manifold?

The key point is that sufficiently *smooth* kernels must have an eigenvalue spectrum with a bounded tail. Specifically, a C^t kernel on a d-dimensional space must have eigenvalues $\lambda_n \lesssim \frac{1}{n^{1+t/d}}$ [89]. In the generic case where the covariance matrices we have discussed can be interpreted as kernels on a manifold, and they have spectra *saturating* the bound, linearized models will inherit scaling exponents given by the dimension of the manifold.

As a simple example, consider a d-torus. In this case we can study the Fourier series decomposition, and examine the case of a kernel $K(x - y)$. This must take the form

$$K = \sum_{n_I} [a_{n_I} \sin(n_I \cdot (x - y)) + b_{n_I} \cos(n_I \cdot (x - y))]$$

where $n_I = (n_1, \cdots, n_d)$ is a list of integer indices, and a_{n_I}, b_{n_I} are the overall Fourier coefficients. To guarantee that K is a C^t function, we must have $a_{n_I}, b_{n_I} \lesssim \frac{1}{n^{d+t}}$ where $n^d = N$ indexes the number of a_{n_I} in decreasing order. But this means that in this simple case, the tail eigenvalues of the kernel must be bounded by $\frac{1}{N^{1+t/d}}$ as $N \to \infty$.

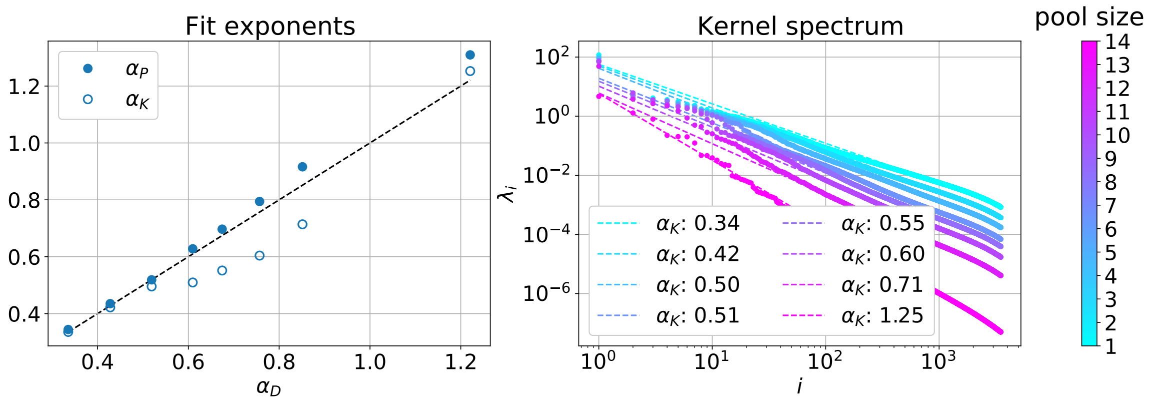

Figure 3.4: **Duality and spectra in random feature models** Here we show the relation between the decay of the kernel spectra, α_K, and the scaling of the loss with number of data points, α_D, and with number of parameters, α_P **(left)**. The theoretical relation $\alpha_D = \alpha_P = \alpha_K$ is given by the black dashed line. **(right)** The spectra of random FC kernels on pooled MNIST. The spectra appear well described by a power law decay.

3.2.4 Duality

We argued above that for kernels with pure power law spectra, the asymptotic scaling of the under-parameterized loss with respect to model size and the over-parameterized loss with respect to dataset size share a common exponent. In the linear setup at hand, the relation between the under-parameterized parameter dependence and over-parameterized dataset dependence is even stronger. The under-parameterized and over-parameterized losses are directly related by exchanging the projection onto random features with the projection onto random training points. Note, sample-wise double descent observed in [117] is a concrete realization of this duality for a simple data distribution. In the appendices, we present examples exhibiting the duality of the loss dependence on model and dataset size outside of the asymptotic regime.

3.3 Experiments

3.3.1 Deep teacher-student models

Our theory can be tested very directly in the teacher-student framework, in which a *teacher* deep neural network generates synthetic data used to train a *student* network. Here, it is possible to generate unlimited training samples and, crucially, controllably tune the dimension of the data manifold. We accomplish the latter by scanning over the dimension of the inputs to the teacher. We have found that when scanning over both model size and dataset size, the interpolation exponents closely match the prediction of $4/d$. The dataset size scaling is shown in Figure 3.2, while model size scaling experiments appear in the appendices

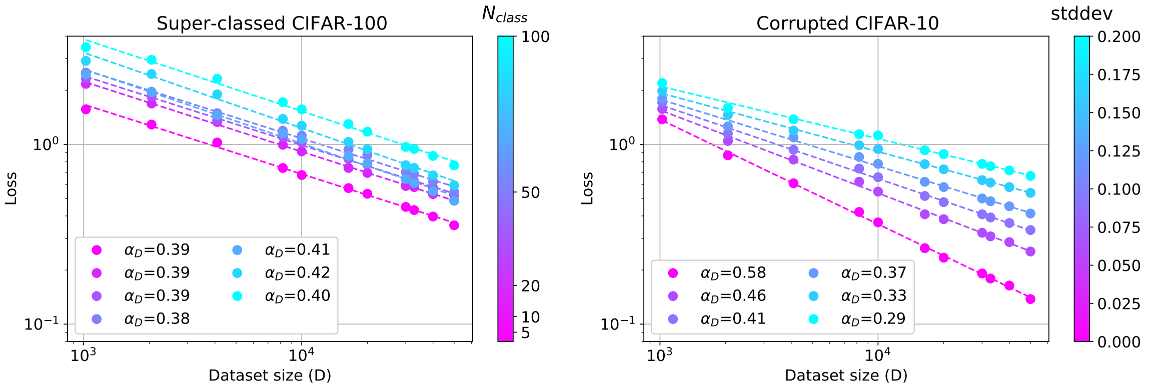

Figure 3.5: **Effect of data distribution on scaling exponents** For CIFAR-100 superclassed to N classes **(left)**, we find that the number of target classes does not have a visible effect on the scaling exponent. **(right)** For CIFAR-10 with the addition of Gaussian noise to inputs, we find the strength of the noise has a strong effect on performance scaling with dataset size. All models are WRN-28-10.

and have previously been observed in [141].

3.3.2 Variance-limited scaling in the wild

Variance-limited scaling can be universally observed in real datasets. The theory describing the variance scaling in Section 3.2.1 does not make any particular assumptions about data, model or loss type, beyond smoothness. Figure 3.1 (top-left, bottom-right) measures the variance-limited dataset scaling exponent α_D and width scaling exponent α_W. In both cases, we find striking agreement with the theoretically predicted values $\alpha_D, \alpha_W = 1$ across a variety of dataset, network architecture, and loss type combinations.

Our testbed includes deep fully-connected and convolutional networks with Relu or Erf nonlinearities and MSE or softmax-cross-entropy losses. Experiments in Figure 3.1 (top-left) utilize relatively small models, with the number of trainable parameteters $P \sim \mathcal{O}(1000)$, trained with full-batch gradient descent (GD) and small learning rate on datasets of size $D \gg P$. Each data point in the figure represents an average over subsets of size D sampled from the full dataset. Conversely, experiments in Figure 3.1 (bottom-right) utilize a small, fixed dataset $D \sim \mathcal{O}(100)$, trained with full-batch GD and small learning rate using deep networks with widths $w \gg D$. As detailed in the appendices, each data point is an average over random initializations, where the infinite-width contribution to the loss has been computed and subtracted off prior to averaging.

3.3.3 Resolution-limited scaling in the wild

In addition to teacher-student models, we explored resolution-limited scaling behavior in the context of standard classification datasets. Experiments were performed with the Wide ResNet (WRN) architecture [166] and trained with cosine decay for a number of steps equal to 200 epochs on the full dataset. In Figure 3.2 we also include data from a four hidden layer CNN detailed in the appendices. As detailed above, we find dataset dependent scaling behavior in this context.

We further investigated the effect of the data distribution on the resolution-limited exponent, α_D by tuning the number of target classes and input noise (Figure 3.5).

To probe the effect of the number of target classes, we constructed tasks derived from CIFAR-100 by grouping classes into broader semantic categories. We found that performance depends on the number of categories, but α_D is insensitive to this number. In contrast, the addition of Gaussian noise had a more pronounced effect on α_D. These results suggest a picture in which the network learns to model the input data manifold, independent of the classification task, consistent with observations in [118, 57].

We also explored the effect of network aspect ratio on the dataset scaling exponent. We found that the exponent magnitude increases with width up to a critical width, while the dependence on depth is more mild (see the appendices).

3.4 Discussion

We have presented a framework for categorizing neural scaling laws, along with derivations that help to explain their very general origins. Crucially, our predictions agree with empirical findings in settings which have often proven challenging for theory – deep neural networks on real datasets.

The variance-scaling regime yields, for smooth test losses, a universal prediction of $\alpha_D = 1$ (for $D \gg P$) and $\alpha_W = 1$ (for $w \gg D$). The resolution-limited regime – more closely tied to the regime in which real neural networks are trained in practice – yields exponents α_D, α_P whose numerical value is variable, but we have traced their origins back to a single simple quantity: the intrinsic dimension of the data manifold d, which in a general setting is significantly smaller than the input dimension. In linear models, this is also closely related to α_K, the exponent governing the power-law spectral decay of certain kernels. Neural scaling laws depend on the data distribution, but perhaps they only depend on 'macroscopic' properties such as spectra or a notion of intrinsic dimensionality.

Along the way, our empirical investigations have revealed some additional intriguing observations. The invariance of the dataset scaling exponent to superclassing (Figure 3.5) suggests that commonly-used deep networks may be largely learning properties of the input data manifold – akin to unsupervised learning – rather than significant task-specific structure, which may shed light on the versatility of learned deep network representations for different downstream tasks.

In our experiments, models with larger exponents do indeed tend to perform better, due to increased sample or model efficiency. We see this in the teacher-student setting for models trained on real datasets and in the appendices find that trained features scale noticeably better than random features. This suggests the scaling exponents and intrinsic dimension as possible targets for meta-learning and neural architecture search.

On a broader level, we think work on neural scaling laws provides an opportunity for discussion in the community on how to define and measure progress in machine learning. The values of the exponents allow us to concretely estimate expected gains that come from increases in scale of dataset, model, and compute, albeit with orders of magnitude more scale for constant-factor improvements. On the other hand, one may require that truly non-trivial progress in machine learning be progress that occurs *modulo scale*: namely, improvements in performance across different tasks that are not simple extrapolations of existing behavior. And perhaps the right combinations of algorithmic, model, and dataset improvements can lead to *emergent* behavior at new scales. Large language models such as GPT-3 (Fig. 1.2 in [20]) have exhibited this in the context of few-shot learning. We hope our work spurs further research in understanding and controlling neural scaling laws.

Acknowledgements

The authors would like to thank Guy Gur-Ari, Boris Hanin, Tom Henighan, Danny Hernandez, Aitor Lewkowycz, Sam McCandlish, Preetum Nakkiran, Behnam Neyshabur, Jeffrey Pennington, Vinay Ramasesh, Dan Roberts, Jonathan Rosenfeld, Jascha Sohl-Dickstein, and Lechao Xiao for useful conversations during the completion of this work. US completed a portion of this work during an internship at Google. JK and US were supported in part by Open Philanthropy.

Chapter 4

AdS$_3$ Reconstruction with General Gravitational Dressings

This thesis chapter originally appeared in the literature as:

H. Chen, J. Kaplan and U. Sharma , "AdS$_3$ reconstruction with general gravitational dressings" *J. High Energ. Phys.* **2019**, 141 (2019), *arXiv*: 1905.00015

Abstract

The gauge redundancy of quantum gravity makes the definition of local operators ambiguous, as they depend on the choice of gauge or on a 'gravitational dressing' analogous to a choice of Wilson line attachments. Recent work identified exact AdS$_3$ proto-fields by fixing to a Fefferman-Graham gauge. Here we extend that work and define proto-fields with general gravitational dressing. We first study bulk fields charged under a $U(1)$ Chern-Simons gauge theory as an illustrative warm-up, and then generalize the results to gravity. As an application, we compute a gravitational loop correction to the bulk-boundary correlator in the background of a black hole microstate, and then verify this calculation using a newly adapted recursion relation. Branch points at the Euclidean horizon are present in the $1/c$ corrections to semiclassical correlators.

4.1 Introduction and Summary

A complete description of AdS/CFT requires an exact prescription for bulk reconstruction, which would ideally provide a quantitative guide to its own limitations. This problem may decomposed into two (overlapping) sub-problems:

- Reconstruction of interacting bulk fields from dual boundary 'CFT' operators in the absence of AdS gravity. It's easy to solve this problem for free bulk fields and generalized free theory (GFT) duals, and it can also be addressed order-by-order in bulk perturbation theory [79, 76, 77, 78]. This problem is very similar [126] to the question of how to relate the operators in a CFT which ends at a boundary to the BCFT operators living on that boundary.

- Bulk reconstruction in the presence of gravity. This problem is qualitatively different, because we do not expect local bulk operators to be uniquely defined – they must be associated with a 'gravitational Wilson line' or 'gravitational dressing'. These complications arise because of the gauge redundancy of bulk diffeomorphisms and the universality of the gravitational force.

Both gravitational and non-gravitational interactions seem to require bulk field operators $\Phi(X)$ to include mixtures of infinitely many CFT operators.

In AdS$_3$ the purely gravitational component of bulk reconstruction can be more precisely specified by taking advantage of the relation between bulk gravity and Virasoro symmetry. This makes it possible to solve one aspect of reconstruction exactly. Prior work [8] defined bulk operators by first fixing to Fefferman-Graham gauge, thereby assuming a specific and arbitrarily chosen gravitational dressing. The purpose of this paper is to define bulk proto-fields with much more general gravitational dressings, or equivalently, by defining the bulk field in a more general gauge.

Bulk Operators from Symmetry

The AdS/CFT dictionary specifies that

$$\lim_{y\to 0} y^{-2h}\Phi(y,z,\bar{z}) = \mathcal{O}(z,\bar{z}) \tag{4.1.1}$$

for a bulk scalar field $\Phi(Y)$ and a dual boundary CFT primary $\mathcal{O}$. However, at finite y the bulk field operator $\Phi(Y)$ will include an infinite sum of contributions from other primaries. Ultra-schematically, we may write [79]

$$\begin{aligned}\Phi &= \mathcal{O} + \sqrt{G_N}[T\mathcal{O}] + G_N[TT\mathcal{O}] + \cdots \\ &\quad + g[\mathcal{O}_i\mathcal{O}_j] + g^2[\mathcal{O}_i\mathcal{O}_j\mathcal{O}_k] + \cdots\end{aligned} \tag{4.1.2}$$

to indicate that Φ includes a mixture of multi-trace operators made from the stress-tensor T and $\mathcal{O}$, as well as multi-trace operators made from other primaries $\mathcal{O}_i$, with perturbative coefficients that can be computed [79, 76, 77, 78] when such a description applies.

We will be studying the terms in Φ involving $\mathcal{O}$ and any number of stress tensors, as these are determined by the Virasoro symmetry[1] in AdS$_3$. Just as conformal symmetry

[1] There has been much recent work on AdS$_3$ reconstruction [8, 109, 115, 26, 24, 60, 59, 25, 38, 27]. Our

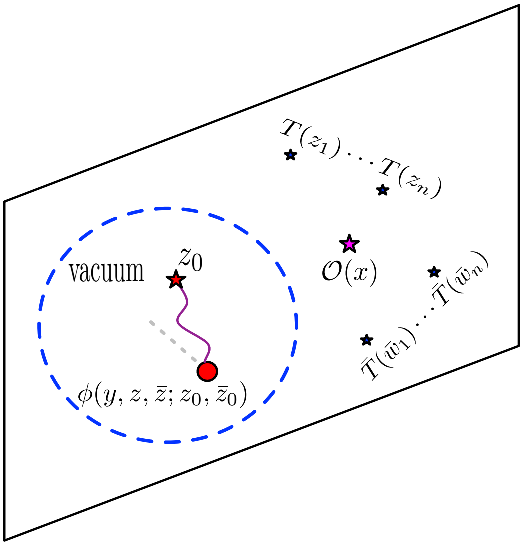
$T(z_1)\cdots T(z_n)$
vacuum
z_0
$\mathcal{O}(x)$
$\bar{T}(\bar{w}_1)\cdots\bar{T}(\bar{w}_n)$
$\phi(y, z, \bar{z}; z_0, \bar{z}_0)$

Figure 4.1: This figure suggests many aspects of our reconstruction strategy. We reconstruct a bulk operator ϕ connected by a Wilson line or 'gravitational dressing' that attaches to the boundary at z_0. Correlators of ϕ with stress tensors will only have singularities when the stress tensors approach z_0. When ϕ acts on the vacuum, it creates a state that we expand in radial quantization, and so when we define ϕ we assume it is surrounded by empty space. As our methods are ultimately based on symmetry, we only compute the proto-field ϕ as a linear combination of a CFT$_2$ primary $\mathcal{O}$ and its Virasoro descendants. All of these statements have analogs for the $U(1)$ charged ϕ we discuss as a warm-up in section 4.2, with $T \to J$.

dictates that CFT correlators must be decomposable as a sum of conformal blocks, bulk scalar fields Φ can be written as a sum of bulk proto-field operators ϕ that are fixed by symmetry.

The proto-fields ϕ also have another interpretation, as sources or sinks for one-particle states in a first-quantized worldline action description. Correlators of protofields $\langle\phi(X_1)\phi(X_2)\cdots\rangle$ with other CFT operators will match to all orders in perturbation theory with the propagation of a particle from X_1 to X_2 in the gravitational background created by these other CFT operators, including the effect of gravitational loops on the propagation. But the proto-field correlators do not include non-gravitational interactions, or mixings with multi-trace operators induced by gravity.

results [8] differ from the proposal [96], which produces a field that does not seem to satisfy the interacting bulk equation of motion in a known gauge when expanded perturbatively in $1/c$ (e.g., compare $\langle\phi\mathcal{O}T\rangle$ correlators to the results of appendix D.4 of [8]).

‘Dressings’ and Correlators with Symmetry Currents

Charged operators in gauge theories and local bulk operators in quantum gravity are not gauge-invariant. This means that their definition is ambiguous, and we need to supply more information to fully specify them. This additional information may be a Wilson line, a specific choice of gauge, or a ‘gravitational dressing’ (by this term we roughly mean ‘gravitational Wilson line’). We discuss the relation between these ideas in section 4.2.1.

The necessity and ambiguity of these dressings has a simple interpretation in the CFT. If we are to write a bulk proto-field $\phi(X)$ as a CFT operator, then the charge and energy in ϕ must be visible to the charge Q and spacetime symmetries $D, P_\mu, K_\nu, M_{\mu\nu}$ in the CFT. These quantities can be computed by integrals of $J_\mu(x)$ or $T_{\mu\nu}(x)$ over Cauchy surfaces on the boundary [55], but the specific spacetime distribution of current and energy-momentum associated with $\phi(X)$ is somewhat arbitrary. This explains the ambiguity in $\phi(X)$, and also suggests how it can be fixed – the gauge and gravitational dressings are specified by the form of $\phi(X)$ correlators with $J_\mu(x)$ or $T_{\mu\nu}(x)$.

To make sense of this logic, it must be possible to distinguish the energy-momentum in $\phi(X)$ from that of other sources in any state or correlator. We accomplish this by assuming that $\phi(X)$ is surrounded by vacuum, so that we can define $\phi(X)$ in a series expansion[2] in the bulk coordinate [126], with local CFT operators as coefficients. In this way we can use radial quantization to define $\phi(X)$.

We will specify general gravitational dressings in two equivalent ways. In section 4.3.2 we use a trick: starting with the proto-field defined through Fefferman-Graham [8], we use a diffeomorphism to bend the gravitational dressing. Whereas in section 4.3.4 we take a more abstract route, and simply construct an operator $\phi(X; x_0)$ at a point X in the bulk, but where $T(x)$ on the boundary detects ϕ's associated stress-energy at a general point x_0.

Summary of Results

Our main result is a simple formula for a proto-field with general dressing

$$\phi\left(u, x, \bar{x}; x_0, \bar{x}_0\right) = \sum_{n=0}^{\infty} \sum_{m,\bar{m}=0}^{\infty} \frac{(-1)^n u^{2h+2n}}{n!(2h)_n} \frac{(x-x_0)^m\,(\bar{x}-\bar{x}_0)^{\bar{m}}}{m!\bar{m}!} \mathcal{L}_{-n-m}\bar{\mathcal{L}}_{-n-\bar{m}}\mathcal{O}(x_0, \bar{x}_0) \tag{4.1.3}$$

The interpretation of this operator is discussed in section 4.3, but roughly speaking, the proto-field is located $(u, x, \bar{x})$ in AdS$_3$, with its associated energy-momentum localized at $(x_0, \bar{x}_0)$ on the boundary. The $\mathcal{L}_{-N}$ are polynomials in the Virasoro generators determined by the bulk primary condition [8], with coefficients that are rational functions in the central

[2] Another common approach [61] defines bulk fields by integrating local CFT operators over a region. This procedure may have equivalent issues when other local CFT operators are present in the region of integration and OPE singularities are encountered.

charge c and holomorphic dimension h of $\mathcal{O}$. We verify that this result has the expected correlators with stress tensors $T(x)$. Our formula can be integrated against a positive, normalized distribution ρ via

$$\phi[\rho](X) \equiv \int d^2x_0\, \rho(x_0, \bar{x}_0)\phi(X; x_0, \bar{x}_0) \tag{4.1.4}$$

to obtain a very general[3] gravitational dressing for the proto-field.

We also show in section 4.4 that correlators of $\phi(X; x_0, \bar{x}_0)$ can be computed by a further adaptation [26, 27] of Zamolodchikov's recursion relations [168, 167]. Then in section 4.5 we analytically calculate the $1/c$ correction to the heavy-light, bulk-boundary propagator on the cylinder using a recent quantization [35] of AdS$_3$ gravity. We demonstrate that our analytic result matches that of the recursion relation. We also observe that as expected [47, 50], the analytic $1/c$ correction to the correlator is not periodic in Euclidean time [27], and so it has a branch cut singularity at the Euclidean horizon. This is surprising from the point of view of perturbation theory in a fixed black hole background.

The outline of the paper is as follows. In section 4.2 we provide a detailed discussion of bulk reconstruction for fields charged under a $U(1)$ Chern-Simons field. This serves as a warm-up where many of the ideas can be more straightforwardly illustrated. Then in section 4.3 we turn to gravity, where many of our results are analogous to the simpler $U(1)$ setting. In section 4.4 we adapt a recursion relation to compute correlators of ϕ with general dressing. In section 4.5 we explain some rather technical calculations, including the recursion relation in a specific configuration and an analytic computation of the one-loop gravitational correction (i.e., order $1/c$) to a $\langle \mathcal{O}_H \mathcal{O}_H \mathcal{O}_L \phi_L \rangle$ correlator. We provide a brief discussion in section 4.6. Many technical results are relegated to the appendices.

4.2 Bulk Proto-Fields with $U(1)$ Chern-Simons Charge

This section will serve as a warm-up in preparation for our eventual discussion of bulk gravity, where most of these ingredients will have a direct analog.

4.2.1 Charged Fields, Wilson Lines, and Gauge Fixing

Consider a bulk field $\varphi(X)$ charged under a $U(1)$ gauge symmetry. It transforms as

$$\varphi(X) \to e^{iq\Lambda(X)}\varphi(X) \tag{4.2.1}$$

[3] It's not entirely clear what operators the full space of gravitational dressings should include, but by letting ρ depend on X we can parameterize a large space of possibilities. An average over x_0 is not equivalent to averaging over different exponents in a Wilson line [40, 41], since the average of an exponential is not the exponential of an average. Wilson lines should be path-ordered, so averaging over complete Wilson lines should be the more generally valid approach.

under the gauge redundancy, so it cannot be regarded as a physical observable. We can remedy this problem in two equivalent ways – by fixing the gauge, or by attaching $\varphi(X)$ to a Wilson line.

The latter approach has the clear advantage that it makes the gauge-invariant nature of our observable manifest. Given a Wilson line

$$W_{\mathcal{C}}(\infty, X) = e^{iq\int_{\mathcal{C}} dx^\mu A_\mu} \tag{4.2.2}$$

running from X to infinity, we can form a non-local operator

$$\phi(X) = W_{\mathcal{C}}(\infty, X)\varphi(X) \tag{4.2.3}$$

Since gauge transformations do not act at infinity, ϕ will be a gauge-invariant observable. However, this means that $\varphi(X)$ itself was highly ambiguous, since ϕ now depends on the path of the Wilson line. Note that once we define a gauge-invariant ϕ in this way, we can compute observables involving it in any convenient gauge, and we will obtain the same results.

The other (fixing the gauge) approach will be easier to discuss when we generalize to quantum gravity. However, it's less flexible and can lead to confusing terminology. In this approach we simply fix a gauge, for example by setting some component of the gauge field $A_y = 0$, and then compute observables involving $\phi(X)$ in this gauge. The results will then be well-defined observables. Note that if $A_y = 0$ then the Wilson line in the $\hat{y}$ direction $W_{\hat{y}} = 1$ identically, so in this case the underlying gauge invariant observable will be $\phi = W_{\hat{y}}\varphi(X)$. But in general it may not be clear how to compute with our observable in other gauges. And it may seem confusing to refer to an observable defined in a specific gauge as gauge-invariant (though this is in fact true).

Let us develop these ideas in the context of a scalar field ϕ in AdS$_3$ charged under a $U(1)$ Chern-Simons theory with level k. The scalar will be dual to a CFT$_2$ primary operator $\mathcal{O}$ with conformal dimension h and charge q, and the gauge field to a holomorphic conserved current $J(z)$. We will work in Euclidean space with a fixed metric

$$ds^2 = \frac{dy^2 + dzd\bar{z}}{y^2} \tag{4.2.4}$$

and in this section we will not include dynamical gravity.

We will be viewing $\phi(y, z, \bar{z})$ through the lens of radial quantization, as discussed previously in [8] and pictured in figure 4.1 (the figure denotes the gravitational case, but with $T \to J$ it also applies to the present discussion). In the CFT ϕ will be a non-local operator, but only because it will be written as an infinite sum of local operators, each a coefficient in the near-boundary or small y expansion of ϕ. If we turn off both gravity and the Chern-Simons interaction, then ϕ is determined by symmetry to be [116]

$$\phi_0(y, z, \bar{z}) = \sum_{n=0}^{\infty}(-1)^n \frac{y^{2h+2n}}{n!(2h)_n} L_{-1}^n \bar{L}_{-1}^n \mathcal{O}(z, \bar{z}) \tag{4.2.5}$$

This follows from the form of the vacuum bulk-boundary propagator

$$\langle \phi_0(y,z,\bar{z})\mathcal{O}^\dagger(w,\bar{w})\rangle = \left(\frac{y}{y^2+(z-w)(\bar{z}-\bar{w})}\right)^{2h} \tag{4.2.6}$$

if we expand in y and identify the coefficients with global descendants of $\mathcal{O}$.

Now let us define a gauge-invariant charged scalar field. As discussed above, we can do this by simply attaching φ to a Wilson line that ends on the boundary at $y=0$. A very simple choice takes the Wilson line to run in the $\hat{y}$ direction, so that

$$\phi(y,z,\bar{z}) = e^{iq\int_0^y dy' A_y(y',z,\bar{z})}\varphi(y,z,\bar{z}) \tag{4.2.7}$$

This operator also has a very simple definition via gauge fixing – it is simply ϕ defined in the gauge $A_y=0$. This makes it clear that the correlator $\langle\phi(y,0,0)\mathcal{O}^\dagger(w,\bar{w})\rangle$ should be equal to the expression (4.2.6).

Correlators involving the boundary current will be non-trivial. Computing in perturbation theory gives

$$\langle J(z_1)\mathcal{O}^\dagger(w,\bar{w})\phi(y,0,0)\rangle = \frac{qw}{z_1(z_1-w)}\left(\frac{y}{y^2+w\bar{w}}\right)^{2h}, \tag{4.2.8}$$

where $\mathcal{O}^\dagger$ has charge $-q$. This reduces to $y^{2h}\langle J\mathcal{O}^\dagger\mathcal{O}\rangle$ in the limit of small y, as expected based on the AdS/CFT dictionary.

If we attach ϕ to the boundary with a more general Wilson line, then we will have

$$\phi(y,z,\bar{z};z_0,\bar{z}) = e^{iq\int_{z_0}^X dY^\mu A_\mu(Y)}\phi(y,z,\bar{z}) \tag{4.2.9}$$

The Wilson line takes some general path from z_0 on the boundary to the location of ϕ in the bulk, yet our notation does not include information about the path. Since Chern-Simons theory is topological, our ϕ will only depend on this path if other charges or Wilson lines entangle with it. However, we are invoking radial quantization to define ϕ, meaning that we will be assuming that there aren't any matter fields or Wilson lines near ϕ, or between it and the boundary. Thus our results for the operator ϕ will be independent of the choice of path, except through the location of z_0.

Correlators of this more general bulk field can still be computed in $A_y=0$ gauge. Since the vacuum equations of motion set $F_{y\mu}=0$, in this gauge A_z is independent of y. Since on the boundary $A_z(0,z,\bar{z})=\frac{1}{k}J(z)$, this identification must hold for all y, so[4]

$$\phi(y,z,\bar{z};z_0,\bar{z}) = e^{i\frac{q}{k}\int_{z_0}^z J(z')dz'}\phi(y,z,\bar{z})\Big|_{A_y=0} \tag{4.2.10}$$

This formula requires some regularization to remain consistent with $\mathcal{O}$ correlators and the dictionary $\mathcal{O}=\lim_{y\to 0}\left[y^{-2h}\phi\right]$. If we expand to first order in q we find

$$\langle J(z_1)\mathcal{O}^\dagger(w,\bar{w})\phi(y,0,0;z_0,0)\rangle \approx \langle J(z_1)\mathcal{O}^\dagger(w,\bar{w})\phi(y,0,0)\rangle + i\frac{q}{k}\int_0^{z_0}\langle J(z_1)\mathcal{O}^\dagger(w,\bar{w})J(z')\phi(y,0,\ 0)\rangle$$

[4] We could also just choose to have the Wilson line run along the boundary from z_0 to z.

$$\approx \frac{q(z_0 - z)}{(z_1 - z_0)(z_1 - z)} \langle \mathcal{O}^\dagger \phi \rangle \tag{4.2.11}$$

This formula has a nice interpretation, as the singularities in z_1 indicate the presence of charge $\pm q$ at z_0 and z. But higher order corrections involving many J will produce divergent integrals. And even the simpler correlator $\langle \mathcal{O}^\dagger \phi \rangle$ also requires regularization.

In the next sections we will see how to avoid regularization by defining ϕ using symmetry when its Wilson line attachments are simple. Then we will extend our results to include general Wilson lines by leveraging the singularity structure of ϕ correlators.

4.2.2 A Bulk Primary Condition from Symmetry

We can take another approach, and constrain the bulk field ϕ using symmetry. If we can determine how to extend CFT symmetries into the bulk, then we can use their action on a charged bulk field to determine how to write it as a sum of CFT operators. This approach will provide an exact definition, without needing to regulate Wilson lines, and it will also generalize more directly to gravity.

The CFT current $J(z)$ can be expanded in modes

$$J\left(z\right) = \sum_{n=-\infty}^{\infty} \frac{J_n}{z^{n+1}}. \tag{4.2.12}$$

The global conformal generators have an algebra with J_n with commutation relations

$$\begin{aligned} \left[L_m, L_n\right] &= \left(m-n\right) L_{m+n}, \\ \left[L_m, J_n\right] &= -nJ_{n+m}, \\ \left[J_m, J_n\right] &= mk\delta_{n+m,0}, \end{aligned} \tag{4.2.13}$$

where the subscripts of the L generators runs from -1 to 1, and the subscript of the J generators run from $-\infty$ to ∞. The current acts on local primary operators via

$$[J_n, \mathcal{O}(z)] = qz^n \mathcal{O}(z) \tag{4.2.14}$$

which can be derived from the $J(x)\mathcal{O}(z)$ OPE. This means that a finite transformation $e^{i\delta J_n}$ will rephase $\mathcal{O}(z) \to e^{iq\delta z^n}\mathcal{O}(z)$.

Now we would like to understand how to extend these symmetries so that they act on bulk fields. This requires either a careful specification of the gauge invariant operators, or a choice of gauge. We will take the latter route and choose $A_y = 0$. We can still transform $A_z \to A_z + \partial_z \lambda(z)$ while preserving this gauge fixing condition. But this is a global (rather than gauge) symmetry transformation, since it acts non-trivially on fields at the boundary $y = 0$.

In the bulk, we expect that a charged field should transform as $\phi \to e^{i\lambda}\phi$. Since λ cannot depend on y, this transforms $\phi(X) \to e^{i\lambda(z)}\phi(X)$. So in $A_y = 0$ gauge, the J_n act on ϕ in the same way that they act on $\mathcal{O}$, giving

$$[J_n, \phi(y,z,\bar{z})] = qz^n \phi(y,z,\bar{z})|_{A_y=0} \tag{4.2.15}$$

where we have indicated explicitly that this only holds in $A_y = 0$ gauge. This further implies a *bulk primary condition*

$$[J_n, \phi(y,0,0)] = 0 \quad \text{for} \quad n \geq 1 \tag{4.2.16}$$

for the bulk field ϕ. This condition is the $U(1)$ Chern-Simons version of the gravitational bulk primary condition originally derived in [8]. Along with the requirement that ϕ has the correct bulk-boundary propagator in vacuum (4.2.6), this bulk primary condition uniquely determines $\phi(y,z,\bar{z})$ as an expansion in y. Furthermore, it is an exact result, and does not require a small coupling expansion.

Notice that a gauge-invariant bulk operator $\phi(y,z,\bar{z})$ attached to the boundary by a Wilson line in the $\hat{y}$-direction must transform in the same way. This simply follows from the fact that $\phi = \varphi$ identically if the latter is defined in the gauge $A_y = 0$.

We can write a formal solution to the bulk primary conditions as

$$\phi(y,z,\bar{z}) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!(2h)_n} y^{2h+2n} \mathcal{J}_{-n} \bar{L}_{-1}^n \mathcal{O}\,(z,\bar{z}) \tag{4.2.17}$$

where $\mathcal{J}_{-n}\mathcal{O}$ is defined as an nth level descendant of $\mathcal{O}$ satisfying the bulk primary condition (4.2.16). In appendix C.2.1, we solve the the bulk primary condition exactly for the first several $\mathcal{J}_{-n}$s. As in section 3.2.2 of [8], it can be shown that $\mathcal{J}_{-n}$ can be written formally in terms of quasi-primaries as

$$\mathcal{J}_{-n}\mathcal{O} = L_{-1}^n \mathcal{O} + n!\,(2h)_n \sum_{j=1}^{n} \sum_i \frac{L_{-1}^{n-j} \mathcal{O}_{h+j}^{(i)}}{\left| L_{-1}^{n-j} \mathcal{O}_{h+j}^{(i)} \right|^2} \tag{4.2.18}$$

where the $\mathcal{O}_{h+j}^{(i)}$ represent the ith quasi-primary at level j (so they satisfy $L_1 \mathcal{O}_{h+j}^{(i)} = 0$) and the denominator is the norm of the corresponding operator. In writing down this equation, we've used the fact that the quasi-primaries can be chosen to be orthogonal to each other. This will be a very useful property for some of the discussions in the following sections. As a concrete example, $\mathcal{J}_{-1}$ is given by

$$\mathcal{J}_{-1}\mathcal{O} = L_{-1}\mathcal{O} + \frac{q^2}{2hk - q^2} \left(L_{-1} - \frac{2h}{q} J_{-1} \right) \mathcal{O}. \tag{4.2.19}$$

where the second term is a quasi-primary satisfying $L_1 \left(L_{-1} - \frac{2h}{q} J_{-1} \right) \mathcal{O} = 0$.

In appendix (C.2.2), we solve the bulk primary condition in the large k limit for the all-order $\frac{1}{k}$ terms in $\mathcal{J}_{-n}$. As shown in both appendix (C.2.1) and (C.2.2), in the $k \to \infty$, we have $\mathcal{J}_{-n} = L_{-1}^n + O(1/k)$, so our expansion in y reduces to that of equation (4.2.5). Note that as in [8], ϕ is a non-local operator in the CFT due to the infinite sum in its definition.

Since the bulk proto-field has been defined as an expansion in descendants of a local CFT primary, we will often informally discuss the 'OPE' of the current J with ϕ. Because of the bulk primary condition above, the singular term in the OPE of $J(z_1)$ and $\phi(y, z, z)$ is very similar to the $J\mathcal{O}$ OPE (which is simply $J(z_1)\mathcal{O}(z,\bar{z}) \sim \frac{q\mathcal{O}(z,\bar{z})}{z_1 - z} + \cdots$):

$$J(z_1)\,\phi(y, z, \bar{z}) \sim \frac{q\phi(y, z, \bar{z})}{z_1 - z} + \cdots \tag{4.2.20}$$

where we have used $[J_0, \phi] = q\phi$, since the descendant operators in ϕ all have the same charge q.

Using these CFT definitions of the bulk charged scalar ϕ, we can compute various correlation functions, such as $\left\langle \phi\mathcal{O}^\dagger J \cdots J \right\rangle$ and $\left\langle \phi^\dagger\phi \right\rangle$. We can first verify that ϕ given in (4.2.17) indeed gives the correct bulk-boundary propagator. Using (4.2.18), one can see $\langle \phi\mathcal{O}^\dagger \rangle$ is given by

$$\left\langle \phi(y, z, z)\,\mathcal{O}^\dagger(w, \bar{w}) \right\rangle = \left(\frac{y}{y^2 + (z - w)(\bar{z} - \bar{w})} \right)^{2h}. \tag{4.2.21}$$

simply because the quasi-primary terms in $\mathcal{J}_{-n}\mathcal{O}$ do not contribute to this two-point function and the calculation reduces to that of $\langle \phi_0\mathcal{O}^\dagger \rangle$ with ϕ_0 given by (4.2.5). The bulk-boundary three-point function $\langle J\mathcal{O}^\dagger\phi \rangle$ can be computed simply using the OPE of $J\mathcal{O}^\dagger$ and $J\phi$ (equation (4.2.20)), and the result is given by

$$\left\langle J(z_1)\,\mathcal{O}^\dagger(w, \bar{w})\,\phi(y, z, z) \right\rangle = q\left(\frac{1}{z_1 - z} - \frac{1}{z_1 - w} \right) \left\langle \phi(y, z, z)\,\mathcal{O}^\dagger(w, \bar{w}) \right\rangle. \tag{4.2.22}$$

Correlation functions of the form $\left\langle \phi\mathcal{O}^\dagger J \cdots J \right\rangle$ can then be computed recursively using OPEs used above and the JJ OPE.

For the bulk two-point function $\left\langle \phi^\dagger\phi \right\rangle$, we compute up to order $1/k$ using the perturbative approximation to ϕ that we derived in appendix (C.2.2), which corresponds to the one photon-loop correction to the bulk propagator. The details of the calculation are given in appendix C.3.1. The result for $\left\langle \phi^\dagger(y_1, z_1, \bar{z}_1)\,\phi(y_2, \bar{z}_2, \bar{z}_2) \right\rangle$ is given by

$$\left\langle \phi^\dagger\phi \right\rangle = \frac{\rho^h}{1 - \rho}\left[1 - \frac{q^2}{k}\left(\frac{\rho\,{}_2F_1(1, 2h+1; 2(h+1); \rho)}{2h+1} + \frac{\rho}{2h} - \log(1 - \rho) \right) \right] + O\left(\frac{1}{k^2} \right) \tag{4.2.23}$$

where $\rho = \left(\frac{\xi}{1 + \sqrt{1 - \xi^2}} \right)^2$ with $\xi = \frac{2y_1y_2}{y_1^2 + y_2^2 + z_{12}\bar{z}_{12}}$.

In appendix C.3 we also use the bulk Witten diagrams to compute $\left\langle J\mathcal{O}^\dagger\phi \right\rangle$ and $\left\langle \phi^\dagger\phi \right\rangle$, and the results exactly match equations (4.2.22) and (4.2.23). This provides a non-trivial check of our definition of a charged bulk scalar field.

4.2.3 Singularities in $J(z)$ and the AdS Equations of Motion

The singularity structure of the $J(z)$ correlators in equation (4.2.22) follow from the bulk equations of motion. So these singularities indicate the placement of Wilson lines attaching

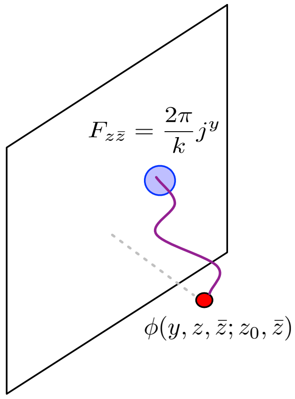
$F_{z\bar{z}} = \frac{2\pi}{k} j^y$
$\phi(y, z, \bar{z}; z_0, \bar{z})$

Figure 4.2: This figure indicates the relationship between singularities in the current $J(z)$, the bulk equations of motion, and Wilson lines. Analogous statements hold for gravity and connect singularities in $T(z)$ to the gravitational dressing.

bulk charges to the boundary, and vice versa. Let us briefly explain these statements, which are illustrated in figure 4.2.

In the $U(1)$ Chern-Simons theory, the equations of motion are

$$\epsilon^{abc} F_{bc}(X) = \frac{2\pi}{k} j^a(X) \tag{4.2.24}$$

for the bulk matter charge $j^a(X)$ and bulk field strength F_{ab}. In the presence of a Wilson line with components in the $\hat{y}$ direction, j^y will receive a delta function contribution localized to the Wilson line. This means that $F_{z\bar{z}}$ must include a delta function. In $A_y = 0$ gauge, we can identify $A_z(y, z, \bar{z}) = \frac{1}{k} J(z)$, so

$$\partial_{\bar{z}} J(z) = 2\pi q \, \delta^2(z, \bar{z}) \tag{4.2.25}$$

or equivalently

$$\oint dz J(z) = 2\pi i q \tag{4.2.26}$$

To satisfy this constraint, the current must have a simple pole $J(z) = \frac{q}{z} + \cdots$, where the ellipsis denotes less singular terms as $z \to 0$. So the singularity structure of correlators with $J(z)$ follows directly from the bulk equations of motion. Conversely, a singularity in $J(z)$ in correlators with other operators indicates the presence of charge.

This means that we can use the singularity structure of correlators with $J(z)$ or $T(z)$ in the case of gravity to help to define a bulk field ϕ with a more general Wilson line attachment or 'gravitational dressing'. Similar observations also hold in higher dimensions, and may be useful for bulk reconstruction more generally.

4.2.4 Charged Bulk Operators and General Wilson Lines

In section 4.2.2 we constructed a charged bulk scalar field by fixing to the gauge $A_y = 0$. In so doing we defined a gauge-invariant bulk field $\phi(y, z, \bar{z})$ connected to the boundary by a Wilson line in the $\hat{y}$ direction. In this section, we are going to construct an exact gauge-invariant bulk field whose associated Wilson line attaches to an arbitrary point z_0 on the boundary.

There are several ways to approach the construction of this general $\phi(y, z, \bar{z}; z_0, \bar{z})$. The most immediate one was already discussed in section 4.2.1, namely including an explicit Wilson line. An issue with this approach is that it requires a regulator for divergences in intermediate calculations, and this makes it difficult to define ϕ non-perturbatively. Another important limitation is that it's challenging to work with Wilson lines for bulk diffeomorphisms, as would be necessary when we turn to gravity.

Instead of inserting an explicit Wilson line, we can define ϕ using the singularity structure of current correlators. Correlators involving $J(z_1)$ and the field $\phi(y, z, \bar{z})$ defined in $A_y = 0$ gauge have singularities as $z_1 \to z$. These singularities represent the charge of the bulk ϕ on the boundary, as emphasized in section 4.2.3. If a Wilson line connects ϕ to the boundary at z_0, then instead we expect that correlators involving $J(z_1)\phi$ will have singularities at z_0.

Thus we need a way to move the singularities in z_1 for all correlators involving $J(z_1)$ and ϕ. In fact, we already have many of the technical tools that we need. The level n descendants $\mathcal{J}_{-n}\mathcal{O}$ defined in section 4.2.2 were constructed so that they would not have any additional singularities in the $J(z_1)\mathcal{J}_{-n}\mathcal{O}$ OPE beyond those already present in $J(z_1)\mathcal{O}$. We can use these $\mathcal{J}_{-n}$ to move an operator without moving the singularities associated with its charge (we develop this idea in more detail in appendix C.1). The point is that since $\mathcal{J}_{-n} = L_{-1}^n + \cdots$ as shown in equation (4.2.18), we have that

$$\sum_{n=0}^{\infty} \frac{x^n}{n!} \mathcal{J}_{-n} = e^{xL_{-1}} + \cdots \tag{4.2.27}$$

where the ellipsis denotes terms proportional to powers of the charge q. So this expression acts like a combination of a translation and a $U(1)$ symmetry transformation. This means that the non-local operator

$$\tilde{\mathcal{O}}\left(z, \bar{z}; z_0, \bar{z}\right) = \sum_{n=0}^{\infty} \frac{\left(z - z_0\right)^n}{n!} \mathcal{J}_{-n}\mathcal{O}\left(z_0, \bar{z}\right). \tag{4.2.28}$$

behaves like a kind of mirage. It is defined so that $J\left(z_1\right)\tilde{\mathcal{O}}\left(z, \bar{z}; z_0, \bar{z}\right)$ only has singularities in $z_1 - z_0$, while correlators of $\tilde{\mathcal{O}}\left(z, \bar{z}; z_0, \bar{z}\right)$ with $\mathcal{O}^\dagger\left(w, \bar{w}\right)$ will instead have singularities when $w - z$ vanishes[5]. Explicitly, by using the OPEs of $\mathcal{J}\mathcal{O}^\dagger$ and $\mathcal{J}\tilde{\mathcal{O}}$ and properties of

[5] Since only the $L_{-1}^n\mathcal{O}$ terms in $\mathcal{J}_{-n}$ contribute in $\left\langle \tilde{\mathcal{O}}\left(z, \bar{z}; z_0, \bar{z}\right) \mathcal{O}^\dagger\left(w, \bar{w}\right) \right\rangle$, the $\mathcal{J}_{-n}$ in (4.2.28) become

$\mathcal{J}_{-n}$, we have

$$\langle J(z_1)\mathcal{O}^\dagger(w,\bar{w})\tilde{\mathcal{O}}(z,\bar{z};z_0,\bar{z})\rangle = \frac{q(z_0-w)}{(z_1-z_0)(z_1-w)}\frac{1}{(z-w)^{2h}(\bar{z}-\bar{w})^{2\bar{h}}} \tag{4.2.29}$$

so $\tilde{\mathcal{O}}(z,\bar{z};z_0,\bar{z})$ behaves as though it is in two places at once.

We can generalize the above idea to obtain a bulk proto-field $\phi\left(y_1,z,\bar{z};z_0,\bar{z}\right)$ at $(y,z,\bar{z})$ with a Wilson line landing at $(z_0,\bar{z})$ on the boundary. This leads to a proposal for a charged bulk field with a more general Wilson line attachment

$$\phi\left(y,z,\bar{z};z_0,\bar{z}\right) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty}(-1)^n\frac{y^{2h+2n}}{n!(2h)_n}\frac{(z-z_0)^m}{m!}\mathcal{J}_{-n-m}\bar{L}_{-1}^n\mathcal{O}\left(z_0,\bar{z}\right), \tag{4.2.30}$$

In fact, this form for ϕ is uniquely fixed by demanding that:

1. $\langle\phi\mathcal{O}^\dagger\rangle$ takes the vacuum form given in equation (4.2.6)
2. Correlators $\langle\mathcal{O}^\dagger(w,\bar{w})J(z_1)\cdots J(z_n)\phi(y,z,\bar{z};z_0,\bar{z})\rangle$ only have simple poles in the z_i, which can only occur when $z_i\to z_0$ or $z_i\to w$.

Note that if $z=z_0$, then only the $m=0$ terms contribute to equation (4.2.30), and ϕ reduces to the bulk field of equation (4.2.17). To obtain $\mathcal{O}(z,\bar{z})$ as we take $y\to 0$, we need to simultaneously send $z_0\to z$; otherwise we obtain the non-local operator $\tilde{\mathcal{O}}(z,\bar{z};z_0,\bar{z})$ of equation (4.2.28).

We can verify that $\phi\left(y,z,\bar{z};z_0,\bar{z}\right)$ is really the desired operator by computing $\left\langle\mathcal{O}^\dagger\phi\right\rangle$ and $\left\langle J\mathcal{O}^\dagger\phi\right\rangle$ using the properties of $\mathcal{J}_{-n}$s. For the bulk-boundary propagator $\left\langle\mathcal{O}^\dagger\left(w,\bar{w}\right)\phi\left(y,z,\bar{z};z_0,\bar{z}\right)\right\rangle$, since the quasi-primaries terms in $\mathcal{J}_{-n}\mathcal{O}$ (equation (4.2.18)) will not contribute, we can simply replace $\mathcal{J}_{-n}\mathcal{O}$ with $L_{-1}^n\mathcal{O}$. But then the sum over m becomes exactly a translation operator, and the ϕ in (4.2.30) becomes the free-field ϕ_0 of (4.2.5), and we have

$$\left\langle\mathcal{O}^\dagger\left(w,\bar{w}\right)\phi\left(y,z,\bar{z};z_0,\bar{z}\right)\right\rangle = \left\langle\mathcal{O}^\dagger\left(w,\bar{w}\right)\phi_0\left(y,z,\bar{z}\right)\right\rangle = \left(\frac{y}{y^2+\left(z-w\right)\left(\bar{z}-\bar{w}\right)}\right)^{2h}, \tag{4.2.31}$$

as expected. For $\left\langle J\mathcal{O}^\dagger\phi\right\rangle$, we can use the OPEs $J\left(z_1\right)\mathcal{O}^\dagger\left(w,\bar{w}\right)\sim\frac{-q\mathcal{O}^\dagger(w,\bar{w})}{z_1-w}+\cdots$ and $J\left(z_1\right)\phi\left(y,z,\bar{z};z_0,\bar{z}\right)\sim\frac{q}{z_1-z_0}\phi\left(y,z,\bar{z};z_0,\bar{z}\right)+\cdots$, where the ellipses denotes non-singular terms. $\left\langle J\mathcal{O}^\dagger\phi\right\rangle$ can then be computed by only including the singular terms in both OPEs, and we get

$$\left\langle J\left(z_1\right)\mathcal{O}^\dagger\left(w,\bar{w}\right)\phi\left(y,z,\bar{z};z_0,\bar{z}\right)\right\rangle = q\frac{z_0-w}{\left(z_1-z_0\right)\left(z_1-w\right)}\left(\frac{y}{y^2+\left(z-w\right)\left(\bar{z}-\bar{w}\right)}\right)^{2h}. \tag{4.2.32}$$

a translation operator, and one can see that we have $\left\langle\tilde{\mathcal{O}}\left(z,\bar{z};z_0,\bar{z}\right)\mathcal{O}^\dagger\left(w,\bar{w}\right)\right\rangle = \left\langle\mathcal{O}\left(z,\bar{z}\right)\mathcal{O}^\dagger\left(w,\bar{w}\right)\right\rangle$.

Note that here the singularities are at $z_1 = w$ and $z_1 = z_0$, while in equation (4.2.22), the singularities were at $z_1 = w$ and $z_1 = z$.

Before concluding this section, we should emphasize that our methods are insensitive to the trajectory that the Wilson line takes from $(y, z, \bar{z})$ in the bulk to the boundary point z_0. This is possible because the bulk theory is topological, and we have assumed that ϕ is surrounded by a region of vacuum, so that it's possible to define ϕ in radial quantization. Our gravitational ϕ will be defined in the same way.

4.2.5 More General Bulk Operators from Sums of Wilson Lines

In prior sections we have developed a formalism for exactly defining and evaluating an operator $\phi(X; z_0)$, where a Wilson line connects $X \equiv (y, z, \bar{z})$ to a point z_0 on the boundary. But there are a host of other choices for gauge-invariant bulk operators. We can explore this space by defining a new bulk operator

$$\phi[\rho](X) \equiv \int d^2 z_0 \, \rho(z_0, \bar{z}_0) \phi(X; z_0, \bar{z}) \tag{4.2.33}$$

where $\rho(z_0, \bar{z}_0) > 0$ and $\int d^2 z_0 \rho(z_0, \bar{z}_0) = 1$. The properties of $\phi[\rho]$ are inherited from $\phi(X; z_0)$, which is simply the special case where $\rho(x) = \delta(x - z_0)$. Note that if we like, we can let ρ depends on X so that ρ varies as we move ϕ to different locations X in the bulk.

Operators like $\phi[\rho]$ make it possible to study bulk fields with far more general 'dressings', which include superpositions of Wilson lines. For example, we might define a bulk field with spherically symmetric dressing. Let the bulk field be located at the point $(y, z, \bar{z})$ in Poincare patch and (t_E, r, θ) in global coordinates[6], and let the point of attachment of the Wilson line to the boundary be z_0. We will average with the measure $\frac{d\theta}{2\pi}$ over the spatial circle on the boundary at fixed t_E. In Poincare patch this means integrating with the measure $\frac{dz_0}{2\pi i z_0}$ along the contour $|z_0| = |z|\sqrt{1 + \frac{y^2}{z\bar{z}}}$. The integration contour ensures that z_0 is on the same time slice as the bulk field. In summary, the integration measure is

$$\rho(z_0, \bar{z}_0) = \frac{1}{2\pi i z_0} \delta\left(|z_0| - |z|\sqrt{1 + \frac{y^2}{z\bar{z}}}\right) \tag{4.2.35}$$

Note that $\int d^2 z_0 \rho(z_0, \bar{z}_0) = 1$. The integration of $\langle J \mathcal{O}^\dagger \phi \rangle$ (equation 4.2.32) then gives

$$\frac{\langle J(z_1) \mathcal{O}^\dagger(w, \bar{w}) \phi[\rho](y, z, \bar{z}) \rangle}{\langle \phi(y, z, \bar{z}) \mathcal{O}(w, \bar{w}) \rangle} = q \left(\frac{\Theta(|z_1| - |z_0|)}{z_1} - \frac{1}{z_1 - w} \right) \tag{4.2.36}$$

[6] The Poincare patch metric $ds^2 = \frac{dy^2 + dz d\bar{z}}{y^2}$ and the global AdS$_3$ metric $ds^2 = \left(r^2 + 1\right) dt_E^2 + \frac{dr^2}{r^2+1} + r^2 d\theta^2$ are related by

$$y = \frac{e^{t_E}}{\sqrt{r^2 + 1}}, \quad z = \frac{r e^{t_E + i\theta}}{\sqrt{r^2 + 1}}, \quad \bar{z} = \frac{r e^{t_E - i\theta}}{\sqrt{r^2 + 1}}. \tag{4.2.34}$$

where $|z_0|$ denotes $|z|\sqrt{1+\frac{y^2}{z\bar{z}}}$ and Θ is the Heaviside step function. Since $J_0 = \frac{1}{2\pi i}\oint dz_1 J(z_1)$, we can further contour integrate the above expression with respect to z_1 to obtain the total charge of the state. The result is given by

$$q\left(\Theta(|z_1|-|z_0|)-\Theta(|z_1|-|w|)\right) \tag{4.2.37}$$

The interpretaton of this is clear: A state created by ϕ or $\mathcal{O}^\dagger$ carries charges q and $-q$ respectively, and the charge content of the state at the time of insertion of J depends upon the insertions that occur before that time. This result should be equivalent to Coulomb gauge [40] to lowest order in perturbation theory, but will not agree more generally, because the exponential of an average is not equal to the average of an exponential.

4.3 Gravitational Proto-Fields with General Dressing

There are no local gauge invariant operators in gravity. To understand this, observe that diffeomorphism gauge redundancies act on local scalar fields via

$$\varphi(x) \to \varphi(x) + \xi^\mu(x)\nabla_\mu\varphi(x) \tag{4.3.1}$$

This is similar to $U(1)$ gauge theory, where a charged field transforms as $\varphi \to \varphi + iq\Lambda(x)\varphi$, insofar as in both cases, local fields by themselves are not gauge-invariant. As we discussed in detail in section 4.2, we can form gauge invariant quasi-local charged operators in a $U(1)$ theory using Wilson line attachments.

Matters are not so simple in the case of gravity, because it is the dependence of φ on the bulk coordinates themselves that renders φ gauge non-invariant. Lacking a simple and general notion of a gravitational Wilson line, we will discuss diffeomorphism gauge-invariant operators as 'gravitationally dressed' local operators. In this section, we consider gravitational dressings[7] that associate local bulk operators with specific points on the boundary. Such dressings are natural analogs of Wilson lines joining a field in the bulk to a point on the boundary.

We will use the notation $\phi(y, z, \bar{z}; z_0, \bar{z}_0)$ to denote a diffeomorphism invariant bulk proto-field at $(y, z, \bar{z})$ in AdS$_3$ that has a gravitational line dressing landing on the boundary at the point $(z_0, \bar{z}_0)$. The simplified notation $\phi(y, z, \bar{z})$ is used when the boundary point of the gravitational dressing is at $(z, \bar{z})$. We sometimes refer to the path that the associated gravitational dressing takes, but strictly speaking our results are not associated with any particular path. In our formalism all paths are equivalent because we assume that the bulk field can be connected to the boundary through the bulk vacuum.

Our typical setup is pictured in figure 4.1. Some of our analysis will be analogous to the simpler and conceptually clearer $U(1)$ Chern-Simons case discussed in section 4.2.

[7] As discussed in section 4.2.1, one way to define dressed, gauge-invariant operators is to define the operators after fixing to a specific gauge. We will use this method and others in this section.

However, we will provide more details about the relationship between diffeomorphisms and gravitational dressing, and explain how some simple correlation functions can be computed.

4.3.1 Review of Protofields in Fefferman-Graham Gauge

In recent work [8, 26, 27], an exact gravitational proto-field was defined with a very specific gravitational dressing. The dressing was determined implicitly, by fixing to a Fefferman-Graham or Banados gauge [12] where the metric takes the form

$$d\hat{s}^2 = \frac{dy^2 + dzd\bar{z}}{y^2} - \frac{6T(z)}{c}dz^2 - \frac{6\bar{T}(\bar{z})}{c}d\bar{z}^2 + y^2\frac{36T(z)\bar{T}(\bar{z})}{c^2}dzd\bar{z}. \tag{4.3.2}$$

Away from sources of bulk energy, $d\hat{s}^2$ may be viewed as an operator whose VEV corresponds to the semiclassical metric. That is, for a CFT state $|\psi\rangle$, the semiclassical metric will be given by $ds^2_{|\psi\rangle} = \langle\psi|d\hat{s}|\psi\rangle$, which is the RHS of the above equation with $T(z) \to \langle\psi|T(z)|\psi\rangle$ and $\bar{T}(z) \to \langle\psi|\bar{T}(z)|\psi\rangle$. In this subsection, we are considering the case where the CFT is living on a flat Euclidean plane with coordinates $(z, \bar{z})$. So for the CFT vacuum $|0\rangle$, we have $\langle 0|T(z)|0\rangle = 0$ and $\langle 0|\bar{T}(z)|0\rangle = 0$, and the bulk metric is the Poincare metric $ds^2_{|0\rangle} = \frac{dy^2+dzd\bar{z}}{y^2}$.

Once we gauge fix, Virasoro symmetry transformations extend to a unique set of vector fields in the bulk. Demanding that $\phi(y, z, \bar{z})$ transforms as a scalar field under the corresponding infinitesimal diffeomorphisms then implies that it must satisfy the bulk primary conditions [8]

$$[L_n, \phi(y, 0, 0)] = 0, \quad [\bar{L}_n, \phi(y, 0, 0)] = 0, \quad n \geq 2, \tag{4.3.3}$$

along with the condition that in the vacuum $ds^2_{|0\rangle} = \frac{dy^2+dzd\bar{z}}{y^2}$, the bulk-boundary propagator is

$$\langle \mathcal{O}(z, \bar{z})\phi(y, 0, 0)\rangle = \frac{y^{2h_L}}{(y^2 + z\bar{z})^{2h_L}}. \tag{4.3.4}$$

These conditions uniquely determine $\phi(y, 0, 0)$ as a CFT operator defined by its series expansion in the radial y coordinate:

$$\phi(y, 0, 0) = y^{2h_L} \sum_{N=0}^{\infty} \frac{(-1)^N y^{2N}}{N!\,(2h_L)_N} \mathcal{L}_{-N}\bar{\mathcal{L}}_{-N}\mathcal{O}(0). \tag{4.3.5}$$

This is a bulk proto-field in the metric of equation (4.3.2). The $\mathcal{L}_{-N}$ are polynomials in the Virasoro generators at level N, with coefficients that are rational functions of the dimension h of the scalar operator $\mathcal{O}$ and of the central charge c. They are obtained by solving the bulk-primary conditions (4.3.3) exactly (no large c expansion is required)[8]. And as shown

[8] As a concrete example

$$\mathcal{L}_{-2} = \frac{(2h+1)(c+8h)}{(2h+1)c + 2h(8h-5)} \left(L_{-1}^2 - \frac{12h}{c+8h}L_{-2}\right). \tag{4.3.6}$$

in section 3.2.2 of [8], they can be written formally in terms of quasi-primaries as

$$\mathcal{L}_{-N}\mathcal{O} = L_{-1}^{N}\mathcal{O} + N!(2h)_N \sum_{n=2}^{N}\sum_{i} \frac{L_{-1}^{N-n}\mathcal{O}_{h+n}^{(i)}}{\left|L_{-1}^{N-n}\mathcal{O}_{h+n}^{(i)}\right|^2} \tag{4.3.7}$$

where $\mathcal{O}_{h+n}^{(i)}$ is the ith quasi-primary[9] at level n and the denominator is the norm of the corresponding operator. In writing this equation, we have used the fact that the quasi-primaries can be chosen so that they are orthogonal to each other. This will be a very useful property of $\mathcal{L}_{-N}\mathcal{O}$ in later discussions.

The correlation function $\langle\phi\mathcal{O}T\rangle$ was computed and it was found that [8]

$$\frac{\langle\phi(y,0,0)\mathcal{O}(z,\bar{z})T(z_1)\rangle}{\langle\phi(y,0,0)\mathcal{O}(z,\bar{z})\rangle} = \frac{hz^2}{z_1^3(z_1-z)^2}\left(z_1 + \frac{2y^2(z_1-z)}{y^2+z\bar{z}}\right). \tag{4.3.8}$$

Notice that this correlator has a cubic singularity when the position z_1 of the energy-momentum tensor T approaches the point $(0,0)$. Roughly speaking, this singularity is the boundary imprint of the energy-momentum of the bulk field. It demonstrates the presence of a gravitational dressing connecting ϕ to the boundary point $(0,0)$. The boundary energy-momentum tensor $T(z_1)$ can detect this dressing via the cubic singularity, just as the current $J(z_1)$ registered the charge of a bulk field by a similar, but lower-order, singularity, as discussed in section 4.2.3. On a more formal level, contour integrals $\oint z_1^{2+n}T(z_1)$ surrounding the singularity pick out the action of conformal generators L_n on the bulk field ϕ.

Using our more general notation $\phi(y,z,\bar{z};z_0,\bar{z}_0)$, the proto-field $\phi(y,0,0)$ given in equation (4.3.5) is the gauge-invariant operator $\phi(y,0,0;0,0)$ evaluated in the Fefferman-Graham gauge. The gravitational dressing is in the $\hat{y}$ direction. In the next subsection, we are going to construct bulk proto-fields that have more general dressings.

4.3.2 Natural Gravitational Dressings from Diffeomorphisms

In many situations it is convenient to place the CFT on a two-dimensional surface other than the plane. For example, in the presence of a BTZ black hole it's more natural to place the CFT on a cylinder, as will be discussed in section 4.5. In this section we will demonstrate how to define scalar proto-fields with a correspondingly natural dressing, greatly generalizing the results of section 4.3.1. The ideas are illustrated in figure 4.3.

General vacuum AdS$_3$ metrics with different boundary geometries can be obtained from the CFT on a plane

$$ds^2 = \frac{du^2 + dxd\bar{x}}{u^2} \tag{4.3.9}$$

[9] In prior work [8, 26, 27] the notation $\mathcal{L}_{-n}^{\mathrm{quasi},i}\mathcal{O}$ has been used, which is $\mathcal{O}_{h+n}^{(i)}$ here.

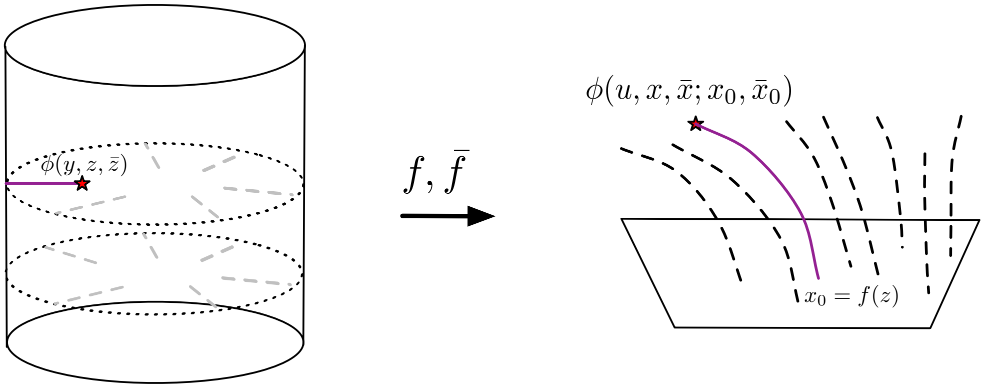

Figure 4.3: This figure suggests the use of a diffeomorphism to generate a natural gravitational dressing for a CFT living on a curved space, such as the cylinder. A curved dressing in the $(u,x,\bar{x})$ coordinates, with the CFT living on a plane, is mapped using equation (4.3.10) to a radial dressing in global AdS and BTZ black hole backgrounds.

by the following coordinate transformation [133]

$$\begin{aligned}
u &= y\frac{4\left(f'(z)\bar{f}'(\bar{z})\right)^{\frac{3}{2}}}{4f'(z)\bar{f}'(\bar{z})+y^2f''(z)\bar{f}''(\bar{z})}\\
x &= f(z)-\frac{2y^2\left(f'(z)\right)^2\bar{f}''(\bar{z})}{4f'(z)\bar{f}'(\bar{z})+y^2f''(z)\bar{f}''(\bar{z})}\\
\bar{x} &= \bar{f}(\bar{z})-\frac{2y^2\left(\bar{f}'(\bar{z})\right)^2 f''(z)}{4f'(z)\bar{f}'(\bar{z})+y^2f''(z)\bar{f}''(\bar{z})}.
\end{aligned} \tag{4.3.10}$$

The resulting vacuum AdS$_3$ metric is then given by

$$ds^2=\frac{dy^2+dzd\bar{z}}{y^2}-\frac{1}{2}S_f(z)dz^2-\frac{1}{2}\bar{S}_{\bar{f}}(\bar{z})d\bar{z}^2+y^2\frac{S_f(z)\bar{S}_{\bar{f}}(\bar{z})}{4}dzd\bar{z}. \tag{4.3.11}$$

where $S_f(z)\equiv\frac{f'''(z)}{f'(z)}-\frac{3}{2}\left(\frac{f''(z)}{f'(z)}\right)^2$ is the Schwarzian derivative.

In this section, we would like to obtain a formula for a bulk proto-field $\phi(y,z,\bar{z})$ with a gravitational dressing in the $\hat{y}$ direction. The difference between this section and section 4.3.1 is that in this section, the CFT is living on a different 2d surface. This may seem confusing, since the boundary here is also described by $(z,\bar{z})$, but here the energy-momentum tensor has a non-vanishing VEV given by $\langle 0|T(z)|0\rangle=\frac{c}{12}S_f$. For example, if $f(z)=e^z$ and $\bar{f}(\bar{z})=e^{\bar{z}}$ (see Section 4.3.3 for more details about this example), then $(z,\bar{z})$ are naturally coordinates on a cylinder and $S_f=-\frac{1}{2}$ (i.e. $\langle 0|T(z)|0\rangle=-\frac{c}{24}$, which is the expectation value of the energy-momentum tensor on the cylinder). In other words, the metric dual to the CFT vacuum $|0\rangle$ here is given by (4.3.11) with a $S_f\neq 0$, whereas in section 4.3.1 the metric dual to the CFT vacuum is given by the Poincare metric $ds^2=\frac{dy^2+dzd\bar{z}}{y^2}$ with planar boundary.

To obtain a bulk proto-field operator $\phi(y,z,\bar{z})$ with a gravitational dressing in the $\hat{y}$ direction, we can consider what this operator corresponds to in the $(u,x,\bar{x})$ coordinates of equation (4.3.9). The position $(y,z,\bar{z})$ of this operator will be mapped to $(u(y,z,\bar{z}),x(y,z,\bar{z}),\bar{x}(y,z,\bar{z}))$, and the boundary point of the gravitational dressing $(y=0,z,\bar{z})$ will be mapped to the boundary point $(u=0,x_0=f(z),\bar{x}_0=\bar{f}(\bar{z}))$. Therefore we can also denote the operator $\phi(y,z,\bar{z})$ as follows

$$\phi(y,z,\bar{z})\equiv\phi(u,x,\bar{x};x_0,\bar{x}_0) \tag{4.3.12}$$

with the understanding that $(u,x,\bar{x})$ are functions of $(y,z,\bar{z})$ as given by (4.3.11) and $x_0=f(z)$ and $\bar{x}_0=\bar{f}(\bar{z})$. Although in the $(y,z,\bar{z})$ coordinates, the gravitational dressing is in the $\hat{y}$ direction, in the $(u,x,\bar{x})$ coordinates the dressing is not in the $\hat{u}$ direction, since typically $x_0\neq x$ and $\bar{x}_0\neq\bar{x}$.

These observations imply that in the $(u,x,\bar{x})$ coordinates, the bulk-boundary propagator should be given by

$$\langle\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)\mathcal{O}\left(x_1,\bar{x}_1\right)\rangle=\left(\frac{u}{u^2+\left(x-x_1\right)\left(\bar{x}-\bar{x}_1\right)}\right)^{2h}. \tag{4.3.13}$$

The new operator must also satisfy the bulk-primary conditions

$$[L_n^{(x_0)},\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)]=0,\quad[\bar{L}_n^{(\bar{x}_0)},\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)]=0,\quad n\geq 2 \tag{4.3.14}$$

The above two conditions uniquely fix $\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)$ to be[10]

$$\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)=\sum_{n=0}^{\infty}\sum_{m,\bar{m}=0}^{\infty}\frac{(-1)^n u^{2h+2n}}{n!(2h)_n}\frac{\left(x-x_0\right)^m\left(\bar{x}-\bar{x}_0\right)^{\bar{m}}}{m!\bar{m}!}\mathcal{L}_{-n-m}\bar{\mathcal{L}}_{-n-\bar{m}}\mathcal{O}\left(x_0,\bar{x}_0\right). \tag{4.3.16}$$

Note that the $\mathcal{L}_{-n-m}$ and $\bar{\mathcal{L}}_{-n-\bar{m}}$ here are defined on the boundary complex plane $(x,\bar{x})$, where $T(x),\bar{T}(\bar{x})$ are quantized by expanding around the point $(x_0,\bar{x}_0)$. It's obvious that the above equation satisfies the bulk-primary conditions (4.3.14), since the $\mathcal{L}_{-N}$s and $\bar{\mathcal{L}}_{-\bar{N}}$s are constructed as solutions to such conditions. Using the property (4.3.7) of the $\mathcal{L}_{-N}$s, it's easy to see why $\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)$ has the correct bulk-boundary propagator (4.3.13) with $\mathcal{O}(x_1,\bar{x}_1)$, since the quasi-primary terms in $\mathcal{L}_{-n-m}\bar{\mathcal{L}}_{-n-\bar{m}}\mathcal{O}$ will not contribute in this two-point function. So when computing (4.3.13), we can simply replace the $\mathcal{L}_{-n-m}\bar{\mathcal{L}}_{-n-\bar{m}}\mathcal{O}$ with $L_{-1}^{n+m}\bar{L}_{-1}^{n+\bar{m}}\mathcal{O}$, and the sums over m and $\bar{m}$ becomes translation operators. We then have $\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)\to\sum_{n=0}^{\infty}\frac{(-1)^n u^{2h+2n}}{n!(2h)_n}L_{-1}^n\bar{L}_{-1}^n\mathcal{O}(x,\bar{x})$, which will give us the desired bulk-boundary propagator (4.3.13).

[10] Performing one of the summations allows us to simplify and write ϕ as

$$\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)=u^{2h}\sum_{N,\bar{N}=0}^{\infty}\frac{(\Delta x)^N(\Delta\bar{x})^{\bar{N}}}{N!\bar{N}!}\,{}_2F_1\left(-N,-\bar{N},2h,-\frac{u^2}{\Delta x\Delta\bar{x}}\right)\mathcal{L}_{-N}\bar{\mathcal{L}}_{-\bar{N}}\mathcal{O}\left(x_0,\bar{x}_0\right). \tag{4.3.15}$$

where we used $\Delta x=x-x_0$ and $\Delta\bar{x}=\bar{x}-\bar{x}_0$ for concision.

In prior work [8] the special case $f(z) = z, \bar{f}(\bar{z}) = z$ was studied, so that $y = u, x = z, \bar{x} = \bar{z}$, and only the terms with $m = \bar{m} = 0$ contribute in the above equation. In that case the result reduces to

$$\phi(y, z, \bar{z}; z, \bar{z}) = \sum_{n=0}^{\infty} \frac{(-1)^n y^{2h+2n}}{n!\,(2h)_n} \mathcal{L}_{-n}\bar{\mathcal{L}}_{-n}\mathcal{O}(z, z) = \phi(y, z, \bar{z}) \tag{4.3.17}$$

where $\phi(y, z, \bar{z})$ here is exactly the bulk proto-field defined in section 4.3.1. And in this case, the gravitational dressing points in the $\hat{u}$ direction, as it coincides with the $\hat{y}$ direction.

If we take the limit that $y \to 0$, then this also implies $u \to 0$, and also $(x - x_0)$, $(\bar{x} - \bar{x}_0) \to 0$, as can be seen from equation (4.3.10), so we find

$$\lim_{y\to 0} y^{-2h}\phi(y, z, \bar{z}) = \left(f'(z)\,\bar{f}'(\bar{z})\right)^h \mathcal{O}\left(f(z), \bar{f}(\bar{z})\right) = \mathcal{O}(z, \bar{z}) \tag{4.3.18}$$

where the Jacobian factor $\left(f'(z)\,\bar{f}'(\bar{z})\right)^h$ comes from $\lim_{y\to 0} u$ using equation (4.3.10). This is exactly what we expect for a bulk operator in the $(y, z, \bar{z})$ coordinates with a gravitational dressing in the $\hat{y}$ direction.

Up to now we have been discussing the operator $\phi(y, z, \bar{z})$ in the CFT vacuum. But our definition also applies to general CFT states $|\psi\rangle$. The semiclassical bulk metric associated with $|\psi\rangle$ is

$$ds^2_{|\psi\rangle} = \frac{dy^2 + dzd\bar{z}}{y^2} - \frac{6T_\psi(z)}{c}dz^2 - \frac{6\bar{T}_\psi(\bar{z})}{c}d\bar{z}^2 + y^2\frac{36T_\psi(z)\bar{T}_\psi(\bar{z})}{c^2}dzd\bar{z} \tag{4.3.19}$$

where $T_\psi(z) \equiv \langle\psi|T(z)|\psi\rangle$ and similarly for $\bar{T}_\psi(\bar{z})$. Note that $T_\psi(z)$ here is the expectation value of the energy-momentum tensor in $|\psi\rangle$, where the CFT lives on a general 2d surface defined via $f, \bar{f}$. $T_\psi(z)$ is related to the $\langle\psi|T(x)|\psi\rangle$ (since we are focusing on the boundary here, we have $x = f(z)$) via the usual transformation rule for the energy-momentum tensor

$$T_\psi(z) = f'(z)^2\langle\psi|T(x)|\psi\rangle + \frac{c}{12}S_f, \tag{4.3.20}$$

where again, S_f is the Schwarzian derivative. Since on the boundary, $x, \bar{x}$ are the coordinates on a complex plane, we have $\langle 0|T(x)|0\rangle = 0$. So in the vacuum, the metric reduces to that of equation (4.3.11).

An interesting example that we will consider later is a map to the cylinder via $f(z) = e^z$ and $\bar{f}(\bar{z}) = e^{\bar{z}}$, where the bulk metric dual to the CFT vacuum is global AdS$_3$. We will study correlators in a heavy state $|\psi\rangle = |\mathcal{O}_H\rangle$, so that the semiclassical bulk geometry is a BTZ black hole. Specifically, we will study the bulk-boundary propagator $\langle\mathcal{O}_H|\phi\mathcal{O}|\mathcal{O}_H\rangle$ in such a heavy state in section 4.5.

Semiclassical Correlators in General Backgrounds

We would like the bulk proto-field to have the property that in *any* semiclassical background arising from heavy sources in a CFT state $|\psi\rangle$, the bulk-boundary propagator $\langle\psi|\phi\mathcal{O}|\psi\rangle$

takes the correct form. Formally, this means that as $c \to \infty$ with heavy source dimensions $h_H \propto c$, but with the dimension of $\mathcal{O}$ fixed, this correlator must obey the free bulk wave equation in the associated metric of equation (4.3.19). Let us explain why this must be the case.

Given any classical fields $T_\psi, \bar{T}_\psi$ forming the metric of equation (4.3.19), we will assume that we can identify new diffeomorphisms $g(z), \bar{g}(\bar{z})$ so that the metric (4.3.19) arises from a diffeomorphism (4.3.10) with $g, \bar{g}$ replacing $f, \bar{f}$. Note that while $f, \bar{f}$ connect the empty Poincaré patch to a corresponding CFT vacuum (4.3.11), the new functions $g, \bar{g}$ relate the empty Poincaré patch to a Fefferman-Graham gauge metric (4.3.19) where there are heavy sources[11].

The bulk primary conditions of equation (4.3.14) were chosen to guarantee that ϕ transforms as a scalar field in Fefferman-Graham gauge, and so it must transform as a scalar under the map from $(u, x, \bar{x}) \leftrightarrow (y, z, \bar{z})$ induced by $g, \bar{g}$. Then the condition (4.3.13) guarantees that the bulk boundary propagator in $(u, x, \bar{x})$ coordinates takes the correct form, and so we can conclude that it must take the correct form in the semiclassical limit in the very non-trivial metric of equation (4.3.19). Thus the conditions we have used to define ϕ ensure that its semiclassical correlators must take the expected form in the background of any sources, as long as ϕ does not intersect with these sources directly.

4.3.3 $\langle \phi \mathcal{O} T \rangle$ on General Vacuum AdS$_3$ Metrics and Examples

We can compute $\langle \phi \mathcal{O} T \rangle$ with ϕ given by (4.3.16) on general vacuum AdS$_3$ metrics given by (4.3.11). Specifically, we study

$$\langle T(x_1)\, \mathcal{O}(x_2, \bar{x}_2)\, \phi(u, x, \bar{x}; x_0, \bar{x}_0) \rangle \tag{4.3.21}$$

with

$$x_0 = f(z), \bar{x}_0 = \bar{f}(\bar{z}), x_1 = f(z_1), x_2 = f(z_2), \text{ and } \bar{x}_2 = \bar{f}(\bar{z}_2) \tag{4.3.22}$$

Although (4.3.21) is written in terms of the $(u, x, \bar{x})$ coordinates, it should really be understood as a correlation function in the coordinates $(y, z, \bar{z})$ with the metric given by (4.3.11) (and S_f given by the Schwarzian derivative of $f(z)$). More precisely, to obtain the correct $\langle \phi \mathcal{O} T \rangle$ in the $(y, z, \bar{z})$ coordinates, we would simply transform $T(x_1)$ and $\mathcal{O}(x_2, \bar{x}_2)$ to $T(z_1)$ and $\mathcal{O}(z_2, \bar{z}_2)$ using the usual transformation rules for the energy-momentum tensor and primary operators, and leave ϕ as it is, since it's a bulk scalar field.

We can use the OPE of $T\mathcal{O}$ and $T\phi$ to evaluate this correlator. Note that when using the OPE of $T\phi$, we need to expand T around x_0 instead of x. Explicitly, the singular terms in the OPEs are

$$T(x_1)\, \mathcal{O}(x_2, \bar{x}_2) \sim \frac{L_{-1}\mathcal{O}(x_2, \bar{x}_2)}{x_1 - x_2} + \frac{h\mathcal{O}(x_2, \bar{x}_2)}{(x_1 - x_2)^2} + \cdots \tag{4.3.23}$$

[11] We'll give an example of this statement in section 4.5.

$$\begin{aligned}T\left(x_1\right)\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)\sim&\frac{L_{-1}\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)}{x_1-x_0}+\frac{L_0\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)}{\left(x_1-x_0\right)^2}\\&+\frac{L_1\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)}{\left(x_1-x_0\right)^3}+\cdots\end{aligned}\tag{4.3.24}$$

where we've used bulk-primary conditions (4.3.14) when writing down the OPE of $T\phi$. So when computing $\langle\phi\mathcal{O}T\rangle$, we simply include all the singular terms in these two OPEs. The L_{-1}s will becomes just the differential operators ∂_{x_2} and ∂_{x_0}, while the terms with $L_0\phi$ and $L_1\phi$ can be computed by commuting L_0 and L_1 with $\mathcal{O}\left(x_2,\bar{x}_2\right)$ or by writing them in terms of contour integrals. Eventually, we get

$$\begin{aligned}&\left\langle T\left(x_1\right)\mathcal{O}\left(x_2,\bar{x}_2\right)\phi\left(u,x,\bar{x};x_0,\bar{x}_0\right)\right\rangle\\=&h\left[\frac{1}{\left(x_1-x_2\right)^2}-\frac{1}{\left(x_1-x_0\right)^2}-2\frac{\left(x_2-x_0\right)}{\left(x_1-x_0\right)^3}\right]\langle\phi\mathcal{O}\rangle\\&-2h\left[-\frac{1}{x_1-x_2}+\frac{x_2-x_0}{\left(x_1-x_0\right)^2}+\frac{1}{x_1-x_0}+\frac{\left(x_2-x_0\right)^2}{\left(x_1-x_0\right)^3}\right]\frac{\left(\bar{x}-\bar{x}_2\right)}{u^2+\left(x-x_2\right)\left(\bar{x}-\bar{x}_2\right)}\langle\phi\mathcal{O}\rangle\,,\end{aligned}\tag{4.3.25}$$

where $\langle\phi\mathcal{O}\rangle=\left(\frac{u}{u^2+(x-x_2)(\bar{x}-\bar{x}_2)}\right)^{2h}$. Correlation functions of the form $\langle\phi\mathcal{O}T\cdots T\rangle$ can then be computed recursively using the above OPEs and also the OPE of TT.

Here, we give two examples of the above result.

Example 1: $\langle\phi\mathcal{O}T\rangle$ on Poincare AdS$_3$

To obtain the $\langle\phi\mathcal{O}T\rangle$ on the Poincare AdS$_3$ metric, we use $f(z)=z,\bar{f}(\bar{z})=\bar{z}$. In this case, we have $u=y,x=z$, and $\bar{x}=\bar{z}$, and the metric (4.3.11) is simply $ds^2=\frac{dy^2+dzd\bar{z}}{y^2}$. After simplification, $\langle\phi\mathcal{O}T\rangle$ is given by

$$\left\langle T\left(z_1\right)\mathcal{O}\left(z_2,\bar{z}_2\right)\phi\left(y,z,\bar{z}\right)\right\rangle=\frac{h\left(z_2-z\right)^2}{\left(z_1-z\right)^3\left(z_1-z_2\right)^2}\left(z_1-z+\frac{2y^2\left(z_1-z_2\right)}{y^2+\left(z-z_2\right)\left(\bar{z}-\bar{z}_2\right)}\right)\langle\phi\mathcal{O}\rangle\tag{4.3.26}$$

This is exactly the result (4.3.8) obtained in [8].

Example 2: $\langle\phi\mathcal{O}T\rangle$ on Global AdS$_3$

To obtain $\langle\phi\mathcal{O}T\rangle$ on global AdS$_3$, whose boundary is a cylinder, we use $f(z)=e^z$ and $\bar{f}\left(\bar{z}\right)=e^{\bar{z}}$. From equation (4.3.10), we have

$$u=\frac{4y\sqrt{\xi\bar{\xi}}}{4+y^2},\quad x=\frac{4-y^2}{4+y^2}\xi,\quad \bar{x}=\frac{4-y^2}{4+y^2}\bar{\xi}.\tag{4.3.27}$$

where we've defined $\xi\equiv e^z$ and $\bar{\xi}\equiv e^{\bar{z}}$ for notational convenience. The resulting metric is

$$ds^2=\frac{dy^2+dzd\bar{z}}{y^2}+\frac{dz^2}{4}+\frac{d\bar{z}^2}{4}+\frac{y^2}{16}dzd\bar{z},\tag{4.3.28}$$

which is related to the usual global AdS$_3$ metric

$$ds^2=\left(r^2+1\right)dt_E^2+\frac{dr^2}{r^2+1}+r^2d\theta^2 \tag{4.3.29}$$

by a simple coordinate transformation

$$z=t_E+i\theta,\quad \bar{z}=t_E-i\theta,\quad y=2\left(\sqrt{r^2+1}-r\right). \tag{4.3.30}$$

The bulk-boundary three-point function $\langle\phi\mathcal{O}T\rangle$ on this metric is given by

$$\begin{aligned}&\left\langle T\left(\xi_1\right)\mathcal{O}\left(\xi_2,\bar{\xi}_2\right)\phi\left(u,x,\bar{x};\xi,\bar{\xi}\right)\right\rangle \\ =&\frac{h\left(\xi-\xi_2\right)^2}{\left(\xi_1-\xi\right)^3\left(\xi_1-\xi_2\right)^2}\left[\xi_1-\xi+\frac{4y^2\xi\left(\xi_1-\xi_2\right)\left(\bar{\xi}+\bar{\xi}_2\right)}{\left(\bar{\xi}\xi+\bar{\xi}_2\xi_2\right)\left(y^2+4\right)+\left(\bar{\xi}\xi_2+\bar{\xi}_2\xi\right)\left(y^2-4\right)}\right] \\ &\times\left\langle\mathcal{O}\left(\xi_2,\bar{\xi}_2\right)\phi\left(u,x,\bar{x};\xi,\bar{\xi}\right)\right\rangle.\end{aligned} \tag{4.3.31}$$

Here, $\left\langle\mathcal{O}\left(\xi_2,\bar{\xi}_2\right)\phi\left(u,x,\bar{x};\xi,\bar{\xi}\right)\right\rangle$ is given by

$$\begin{aligned}\langle\mathcal{O}\phi\rangle&=\left(\frac{u}{u^2+\left(x-\xi_2\right)\left(\bar{x}-\bar{\xi}_2\right)}\right)^{2h} \\ &=\left(\frac{4y\sqrt{\xi\bar{\xi}}}{\left(\bar{\xi}\xi+\bar{\xi}_2\xi_2\right)\left(y^2+4\right)+\left(\bar{\xi}\xi_2+\bar{\xi}_2\xi\right)\left(y^2-4\right)}\right)^{2h}.\end{aligned} \tag{4.3.32}$$

We have also obtained the same result using the effective field theory of gravitons developed in [35] (see appendix C.4 for details of that calculation).

4.3.4 General Gravitational Dressings from Singularity Structure

In this section we will define very general gravitational dressings through a procedure analogous to that of section 4.2.4, where we studied the $U(1)$ Chern-Simons case. To define these bulk proto-fields we will leverage the singularity structure of correlators between $\phi, \mathcal{O}$, and any number of stress tensors. We review how the singularity structure of $T(z)$ is determined by Einstein's equations in appendix C.1.2, generalizing the $U(1)$ case of section 4.2.3. Here the CFT will be living on a flat 2d plane, with the CFT vacuum state corresponding to the pure Poincare metric $ds^2_{|0\rangle}=\frac{du^2+dxd\bar{x}}{u^2}$.

The proto-field operator we will identify takes a similar form to that derived in section 4.3.2, but our interpretation here will be different, and more abstract. Another way of motivating this section would be to ask to what extent equation (4.3.16) can be given a general meaning, independent of the diffeomorphism of equation (4.3.10).

A General Bulk Proto-Field

The energy associated with the bulk operator ϕ must be reflected in the CFT by singularities in $T(x)$ correlators. As in the $U(1)$ Chern-Simons case of section 4.2.4, we can construct a bulk proto-field $\phi(u,x,\bar{x};x_0,\bar{x}_0)$ by demanding:

1. $\langle \phi \mathcal{O} \rangle$ must be given by $\langle \phi\left(u, x, \bar{x}; x_0, \bar{x}_0\right) \mathcal{O}\left(w, \bar{w}\right) \rangle = \left(\frac{u}{u^2+(x-w)(\bar{x}-\bar{w})}\right)^{2h}$ in the vacuum.
2. Correlators $\langle \mathcal{O}(w,\bar{w})T(x_1)\cdots T(x_n)\bar{T}(x_1)\cdots \bar{T}(\bar{x}_{\bar{n}})\phi(u,x,\bar{x};x_0,\bar{x}_0)\rangle$ only have poles of up to third order in the x_i, which can only occur when $x_i \to x_0$ (along with up to second order poles as $x_i \to w$), and equivalently for the antiholomorphic variables.

Note that we allow poles $\propto \frac{1}{(x_i-x_0)^3}$ in $T(x_i)\phi(u,x,\bar{x};x_0,\bar{x}_0)$, whereas only second order poles occur in the OPE of $T(x_i)\mathcal{O}(w)$. This is because ϕ contains descendants of $\mathcal{O}$, including the first descendant $L_{-1}\mathcal{O} \propto \partial\mathcal{O}$, and such operators necessarily induce third-order poles. But higher order singularities are excluded by our assumptions.

The unique operator built from Virasoro descendants of a primary $\mathcal{O}$ and satisfying these conditions is

$$\phi\left(u, x, \bar{x}; x_0, \bar{x}_0\right) = \sum_{n=0}^{\infty} \sum_{m,\bar{m}=0}^{\infty} u^{2h+2n} \frac{(-1)^n}{n!(2h)_n} \frac{(x-x_0)^m(\bar{x}-\bar{x}_0)^{\bar{m}}}{m!\bar{m}!} \mathcal{L}_{-n-m}\bar{\mathcal{L}}_{-n-\bar{m}}\mathcal{O}\left(x_0, \bar{x}_0\right) \tag{4.3.33}$$

As a formal power series expansion in CFT operators, this is identical to equation (4.3.16), except that x_0 and $\bar{x}_0$ are now arbitrary, instead of given by $f(z)$ and $\bar{f}(\bar{z})$. As we will explain below, this object can be interpreted as a bulk proto-field with a dressing that follows the geodesic path from $(u, x, \bar{x})$ to $(x_0, \bar{x}_0)$ in any geometry.

One way to interpret this expression is as a modification of equation (4.3.5) (with the notation $(y, z, \bar{z})$ replaced by $(u, x, \bar{x})$ here), where the boundary imprint of the bulk energy has been translated to $x_0, \bar{x}_0$. In appendix C.1 we develop a theory of non-local 'mirage translation' operators that move the local energy of CFT operators (singularities in $T(x)$ correlators) without moving the apparent location of an operator $\mathcal{O}$ itself (so mirage translations leave OPE singularities between local primaries fixed). Mirage translations also provide an independent motivation for the bulk primary conditions.

To better understand equation (4.3.33), let's consider a few simplifying limits. If $x_0 = x$ and $\bar{x}_0 = \bar{x}$, then only the $m = 0$ terms contribute to equation (4.3.33), and ϕ reduces to the bulk field of equation (4.3.5). To obtain $\mathcal{O}(x_0, \bar{x}_0)$ as we take $u \to 0$, we need to simultaneously send $x \to x_0$; otherwise we obtain a non-local operator $\tilde{\mathcal{O}}(x, \bar{x}; x_0, \bar{x}_0)$, corresponding to $\mathcal{O}(x, \bar{x})$ multiplied with an additional gravitational dressing on the boundary. The non-local operator $\tilde{\mathcal{O}}(x, \bar{x}; x_0, \bar{x}_0)$ can be interpreted is the mirage translation of $\mathcal{O}(x, \bar{x})$, as discussed in appendix C.1.

The non-gravitational limit is also easy to understand. When $c \to \infty$ with h fixed, we have $\mathcal{L}_{-N} \to L_{-1}^N$. In this limit the sum over $m, \bar{m}$ simplifies into a pure translation, and the sums on m simply convert $\mathcal{O}(x_0, \bar{x}_0) \to \mathcal{O}(x, \bar{x})$. Only the sum on n remains, and this reconstructs the non-interacting field defined in equation (4.2.5).

More generally, all of the terms $\mathcal{L}_{-N}\bar{\mathcal{L}}_{-\bar{N}}\mathcal{O}$ were constructed so that they only have OPE singularities of fixed order ≤ 3 with $T(x_1)$, and these singularities only occur when

$x_1 \to x_0$. The field $\phi(u, x, \bar{x}; x_0, \bar{x}_0)$ inherits this property. However, as the $\mathcal{L}_{-N}$ act as a kind of modified translation, other local CFT operators constructed from other primaries will behave as though $\phi(u, x, \bar{x}; x_0, \bar{x}_0)$ is located at $(u, x, \bar{x})$ in the bulk.

The Gravitational Dressing Naturally Follows a Geodesic

We glossed an important issue when defining equation (4.3.33): for this expression to be meaningful, we must have some way of determining where in the bulk this field lives. In the CFT vacuum the operator is at $(u, x, \bar{x})$ in the Poincaré patch metric. This observation is sufficient to determine the location of ϕ in perturbation theory around the Poincaré patch metric. But if we turn on heavy sources, the bulk metric will change by a finite amount. We have not fixed to Fefferman-Graham gauge in $(u, x, \bar{x})$ coordinates, so these coordinates are just labels, which only have unambiguous meaning in perturbation theory or in the limit $u \to 0$.

We can partially resolve this issue by comparing with section 4.3.2. It is easy to see that there exist an infinite family of $f(z), \bar{f}(\bar{z})$ functions so that $x_0, \bar{x}_0 = f(z), \bar{f}(\bar{z})$ and equation (4.3.10) relates $(u, x, \bar{x})$ to $(y, z, \bar{z})$. For each such $f, \bar{f}$ we implicitly define a gauge choice in $(u, x, \bar{x})$ by pulling back the Fefferman-Graham gauge of the $(y, z, \bar{z})$ coordinate system to $(u, x, \bar{x})$ via the diffeomorphism (4.3.10). So the bulk location of the proto-field could be interpreted as a bulk proto-field in any of these gauges.

However, all interpretations of the proto-field share a common feature. The gravitational dressing in $(y, z, \bar{z})$ can be chosen to be a curve with constant $(z, \bar{z})$, so that it points in the $\hat{y}$-direction. In Fefferman-Graham gauge, these curves are all geodesics. This means that the gravitational dressing of $\phi(u, x, \bar{x}; x_0, \bar{x}_0)$ will follow a geodesic from $(u, x, \bar{x})$ in the bulk to $(x_0, \bar{x}_0)$ on the boundary, in any dynamical metric.

Thus we conclude that $\phi(u, x, \bar{x}; x_0, \bar{x}_0)$ will behave like a scalar field in the bulk defined by fixing to any gauge where the gravitational field does not fluctuate along the geodesic connecting $(u, x, \bar{x})$ and $(x_0, \bar{x}_0)$. That is, if $X^\mu(\lambda)$ are coordinates on this geodesic, then $h_{\mu\nu}(X(\lambda))\dot{X}^\nu(\lambda) = 0$, where $h_{\mu\nu}$ is the deviation of the bulk metric from the Poincaré patch form. This leaves the gauge choice elsewhere in spacetime almost entirely undetermined.

More General Dressings

To obtain a more general class of dressed bulk ϕ we can smear $\phi(X; x_0, \bar{x}_0)$ over $(x_0, \bar{x}_0)$ via

$$\phi[\rho](X) \equiv \int dx_0 d\bar{x}_0 \, \rho(x_0, \bar{x}_0)\phi(X; x_0, \bar{x}_0) \tag{4.3.34}$$

with any positive function ρ that integrates to one $\int d^2x_0\rho(x_0, \bar{x}_0) = 1$. If we work in perturbation theory around the vacuum (or any fixed semiclassical metric), then the location of the protofield in $(u, x, \bar{x})$ coordinates will be unambiguous. Then we can obtain results similar to the $U(1)$ case in section 4.2.5.

4.4 Recursion Relation for Bulk-Boundary Vacuum Blocks

Gravitational dynamics can be probed with correlation functions of the bulk proto-field (4.3.16). For example, bulk locality was studied in [26] by computing the bulk two point function $\langle\phi\phi\rangle$ and Euclidean black hole horizons were investigated in [27] by computing the vacuum blocks of heavy-light bulk-boundary correlator $\langle\mathcal{O}_H\phi_L\mathcal{O}_L\mathcal{O}_H\rangle$. In those works, recursions relations were derived for computing correlators involving the proto-field of section 4.3.1 (i.e., the special case of (4.3.16) with $f(z) = z$ and $\bar{f}(\bar{z}) = \bar{z}$). Now that we have the more general bulk proto-field (4.3.16), we can also derive recursion relations for computing[12] its correlators.

In this section, we are going to derive a recursion relation for computing the Virasoro vacuum block contribution to

$$\left\langle\mathcal{O}_H\left(\infty\right)\mathcal{O}_H\left(1\right)\phi_L(u,x,\bar{x};x_0,\bar{x}_0)\mathcal{O}_L\left(0\right)\right\rangle, \tag{4.4.1}$$

where $\phi_L(u, x, \bar{x}; x_0, x_0)$ (with $h = \bar{h} = h_L$) is given by equation (4.3.16) and the coordinates $\infty, 0, 1$ are on the complex plane with $(x, \bar{x})$ coordinates. In order to be more general and include of case of section 4.3.4, we are going to assume that $x_0, \bar{x}_0$ are arbitrary (i.e., we are not assuming that they are given by $f\left(z\right)$ and $\bar{f}\left(\bar{z}\right)$), although this will not affect the discussion in this section. Although we use the subscripts H and L which usually means "heavy" and "light", the conformal dimension h_H and h_L in this section are arbitrary. We will study a special case of this result in section 4.5, and compute the bulk-boundary propagator in a black hole microstate background.

4.4.1 General Structure of the Vacuum Blocks

As usual, the vacuum block $\mathcal{V}_0$ is obtained via the projection

$$\mathcal{V}_0 = \left\langle\mathcal{O}_H\left(\infty\right)\mathcal{O}_H\left(1\right)\mathcal{P}_0\phi_L\left(u,x,\bar{x};x_0,\bar{x}_0\right)\mathcal{O}_L\left(0\right)\right\rangle \tag{4.4.2}$$

where $\mathcal{P}_0$ is the projection operator into the vacuum module[13] (including holomorphic and anti-holomorphic parts). $\phi_L\left(u, x, \bar{x}; x_0, \bar{x}_0\right)$ of equation (4.3.16) can be simplified to the

[12] Although we don't develop it in this paper, the recursion relation for computing the bulk two-point function $\langle\phi\phi\rangle$ for ϕ given by (4.3.16) should be very similar to the one derived in [26].

[13] To be clear, the projection operator for a representation of the Virasoro algebra with lowest weight state $|\mathcal{O}_{h_1}\rangle$ factorizes, that is, $\mathcal{P}_{h_1} = \mathcal{P}_{h_1}^{\text{holo}}\mathcal{P}_{h_1}^{\text{anti-holo}}$ and the holomorphic part is given symbolically by

$$\mathcal{P}_{h_1}^{\text{holo}} = \sum_{\{m_i\},\{n_j\}}\frac{L_{-m_1}\cdots L_{-m_i}|\mathcal{O}_{h_1}\rangle\langle\mathcal{O}_{h_1}|L_{n_j}\cdots L_{n_1}}{\mathcal{N}_{\{m_i\},\{n_j\}}}, \tag{4.4.3}$$

following form:

$$\phi_L\left(u,x,\bar{x};x_0,\bar{x}_0\right)=u^{2h}\sum_{N,\bar{N}=0}^{\infty}\mathcal{A}_{N,\bar{N}}\left(u,\Delta x,\Delta\bar{x}\right)\frac{\mathcal{L}_{-N}\bar{\mathcal{L}}_{-\bar{N}}\mathcal{O}\left(x_0,\bar{x}_0\right)}{\left(2h_L\right)_N\left(2h_L\right)_{\bar{N}}N!\bar{N}!} \tag{4.4.4}$$

where $\Delta x=x-x_0$ and $\Delta\bar{x}=\bar{x}-x_0$, and

$$\mathcal{A}_{N,\bar{N}}\left(u,\Delta x,\Delta\bar{x}\right)\equiv\left(2h_L\right)_N\left(2h_L\right)_{\bar{N}}\left(\Delta x\right)^N\left(\Delta\bar{x}\right)^{\bar{N}}\ {}_2F_1\left(-N,-\bar{N},2h_L,-\frac{u^2}{\Delta x\Delta\bar{x}}\right). \tag{4.4.5}$$

The extra factors of $\left(2h_L\right)_N\left(2h_L\right)_{\bar{N}}$ in the expressions above are inserted for later convenience.

Although the Virasoro vacuum block of (4.4.2) doesn't factorize into holomorphic and anti-holomorphic parts, we can make use of the fact that it does factorizes for a specific N and $\bar{N}$, since the projection operator $\mathcal{P}_0$ factorizes. Similar to the case in [27], we can define the holomorphic part of ϕ_h to be

$$\tilde{\phi}_h^{\text{holo}}\left(u,x;x_0\right)\equiv\sum_{N=0}^{\infty}\frac{\mathcal{L}_{-N}\mathcal{O}_h\left(x_0,\bar{x}_0\right)}{\left(2h_L\right)_N N!}, \tag{4.4.6}$$

and we'll obtain a recursion relation for computing the more general holomorphic block:

$$\mathcal{V}_{\text{holo}}\left(h_1,h_2,c\right)\equiv\left\langle\mathcal{O}_H(\infty)|\mathcal{O}_H(1)\mathcal{P}_{h_1}^{\text{holo}}\tilde{\phi}_{h_2}^{\text{holo}}\left(u,x;x_0\right)|\mathcal{O}_L(0)\right\rangle. \tag{4.4.7}$$

Here the holomorphic projection operator $\mathcal{P}_{h_1}^{\text{holo}}$ only includes the holomorphic descendants of $|\mathcal{O}_{h_1}\rangle$. We are considering this more general block for the convenience of discussing the recursion relation in the next subsection. Eventually, we are interested in the vacuum block $\mathcal{V}_{\text{holo}}\left(0,h_L,c\right)$, and it will be given in the following form

$$\mathcal{V}_{\text{holo}}\left(0,h_L,c\right)=\frac{1}{x_0^{2h_L}}\sum_{N=0}^{\infty}\frac{1}{x_0^N}F_N\left(x_0\right) \tag{4.4.8}$$

$F_N\left(x_0\right)$ is an infinite series of x_0, starting with $F_N\left(x_0\right)=1+\cdots$ (we'll explain how to obtain $F_N(x_0)$ in next subsection). The full vacuum block $\mathcal{V}_0$ is then obtained via the following equation

$$\mathcal{V}_0=\left(\frac{u}{x_0\bar{x}_0}\right)^{2h_L}\sum_{N,\bar{N}=0}^{\infty}\frac{\mathcal{A}_{N,\bar{N}}\left(u,\Delta x,\Delta\bar{x}\right)}{(\Delta x_0)^N(\Delta\bar{x}_0)^{\bar{N}}}F_N\left(x_0\right)F_{\bar{N}}\left(\bar{x}_0\right), \tag{4.4.9}$$

where $F_{\bar{N}}\left(\bar{x}_0\right)$ is simply $F_{\bar{N}}(x_0)$ with x_0 replaced by $\bar{x}_0$.

where $\frac{1}{\mathcal{N}_{\{m_i\},\{n_j\}}}$ is the inverse of the inner-product matrix between the states.

4.4.2 Recursion Relation

Our task now is to obtain the recursion relation for computing $\mathcal{V}_{\text{holo}}\left(h_1,h_2,c\right)$ based on the singularity structure of $\mathcal{V}_{\text{holo}}\left(h_1,h_2,c\right)$ as a function of the central charge c. Actually, the recursion relation for computing $\mathcal{V}_{\text{holo}}$ here is almost the same as the recursion in [27] for computing the $\mathcal{V}_{\text{holo}}$ in that case[14], except that the seed of the recursion is different. We reproduce the recursion relation here for convenience

$$\begin{aligned}\mathcal{V}_{\text{ holo }}\left(h_1,h_2,c\right)=&\mathcal{V}_{\text{ holo }}\left(h_1,h_2,c\to\infty\right)\\&+\sum_{m\geq2,n\geq1}\frac{R_{m,n}\left(h_1,h_2\right)}{c-c_{m,n}\left(h_1\right)}\mathcal{V}_{\text{ holo }}\left(h_1\to h_1+mn,h_2,c\to c_{mn}\left(h_1\right)\right)\\&+\sum_{m\geq2,n\geq1}\frac{S_{m,n}\left(h_1,h_2\right)}{c-c_{m,n}\left(h_2\right)}\mathcal{V}_{\text{ holo }}\left(h_1,h_2\to h_2+mn,c\to c_{mn}\left(h_2\right)\right).\end{aligned}\tag{4.4.10}$$

For more details about the meaning of the symbols $R_{m,n}$, $S_{m,n}$, $c_{m,n}$ and how to solve this recursion relation, please see section 4 and appendix C of [27].

As mentioned above, the seed of the recursion $\mathcal{V}_{\text{ holo }}\left(h_1,h_2,c\to\infty\right)$ is different from that of [27]. In the $c\to\infty$ limit, only global descendants of the intermediate state $\left|\mathcal{O}_{h_1}\right\rangle$ and global descendants of $\mathcal{O}_{h_2}$ contribute, so $\mathcal{V}_{\text{ holo }}\left(h_1,h_2,c\to\infty\right)$ is actually the global block, i.e.

$$G\left(h_1,h_2\right)\equiv\mathcal{V}_{\text{ holo }}\left(h_1,h_2,c\to\infty\right).\tag{4.4.11}$$

$G\left(h_1,h_2\right)$ can be obtain by direct computation, as follows

$$\begin{aligned}G\left(h_1,h_2\right)=&\sum_{m_1,m_2=0}^{\infty}\frac{\left\langle\mathcal{O}_H\middle|\mathcal{O}_H\middle|L_{-1}^{m_1}\mathcal{O}_1\right\rangle\left\langle L_{-1}^{m_1}\mathcal{O}_1\left|L_{-1}^{m_2}\mathcal{O}_{h_2}\left(x_0\right)\right|\mathcal{O}_L\right\rangle}{\left|L_{-1}^{m_1}\right|\mathcal{O}_1\left.\right\rangle|^2\left|L_{-1}^{m_2}\right|\mathcal{O}_2\left.\right\rangle|^2}\\=&\frac{1}{x_0^{2h_L}}\sum_{m_1,m_2=0}^{\infty}\frac{\left(h_1\right)_{m_1}\rho_{m_1,m_2,0}\left(h_1,h_2,h_L\right)}{\left(2h_1\right)_{m_1}m_1!\left(2h_2\right)_{m_2}m_2!}x_0^{h_1+m_1}\tilde{x}_0^{m_2+h_2-h_L}\end{aligned}\tag{4.4.12}$$

where $\tilde{x}_0\equiv\frac{1}{x_0}$. We are using $\tilde{x}_0$ here for keeping tracking the origin of each term (so that we can obtain the F_N in (4.4.8) after we compute $\mathcal{V}_{\text{holo}}$) and for the convenience of implementation in Mathematica. Here, $\rho_{i,j,k}\left(h_1,h_2,h_3\right)$ is the three point functions of global descendants, and it's given by [7]

$$\begin{aligned}\rho_{i,j,k}\left(h_1,h_2,h_3\right)\equiv&\left\langle L_{-1}^{i}\mathcal{O}_{h_1}\middle|L_{-1}^{j}\mathcal{O}_{h_2}\left(1\right)\middle|L_{-1}^{k}\mathcal{O}_{h_3}\right\rangle\\=&\left(h_1+i-h_2-j+1-h_3-k\right)_j s_{ik}\left(h_1,h_2,h_3\right)\end{aligned}\tag{4.4.13}$$

[14] The recursion relation in [27] is an special case of the recursion here, with $f\left(z\right)=z$ and $\bar{f}\left(\bar{z}\right)=\bar{z}$, but the general structures are almost the same.

with

$$s_{ik}\left(h_1,h_2,h_3\right)=\sum_{p=0}^{\min(i,k)}\frac{i!}{p!(i-p)!}\left(2h_3+k-p\right)_p\left(i-p+1\right)_p \\ \times\left(h_3+h_2-h_1\right)_{k-p}\left(h_1+h_2-h_3+p-k\right)_{i-p}. \tag{4.4.14}$$

And we only need the $\rho_{i,j,k}$ with $k=0$.

As in [27], solving the recursion produces $\mathcal{V}_{\text{holo}}\left(h_1,h_2,c\right)$ as the following sum[15]

$$\mathcal{V}_{\text{holo}}\left(h_1,h_2,c\right)=\sum_{m,n=0}^{\infty}C_{m,n}G\left(h_1+m,h_2+n\right). \tag{4.4.16}$$

Here, $G\left(h_1+m,h_2+n\right)$ is the global block (4.4.12) with $h_1\to h_1+m$ and $h_2\to h_2+n$. The summand $C_{m,n}G\left(h_1+m,h_2+n\right)$ is the contribution to $\mathcal{V}_{\text{holo}}$ from all the level m quasi-primaries $\mathcal{O}^{(i)}_{h_1+m}$ and level n quasi-primaries $\mathcal{O}^{(j)}_{h_2+n}$ (denoted as $\mathcal{L}^{\text{quasi},i}_{-m}\mathcal{O}_{h_1}$ and $\mathcal{L}^{\text{quasi},j}_{-n}\mathcal{O}_{h_2}$ in previous papers [26, 27]) , and their global descendants. The coefficients $C_{m,n}$ are exactly the same as the coefficients in [27]. Basically, they encode the three point functions of the quasi-primaries with primaries as follows

$$C_{m,n}=\sum_{i,j}\frac{\left\langle\mathcal{O}_H|\mathcal{O}_H(1)|\mathcal{O}^{(i)}_{h_1+m}\right\rangle\left\langle\mathcal{O}^{(i)}_{h_1+m}\left|\mathcal{O}^{(j)}_{h_2+n}\left(1\right)\right|\mathcal{O}_L\right\rangle}{\left|\mathcal{O}^{(i)}_{h_1+m}\right|^2\left|\mathcal{O}^{(j)}_{h_2+n}\right|^2}, \tag{4.4.17}$$

where we've assumed that the quasi-primaries are orthogonalized and the sum $\sum_{i,j}$ is over all level m and level n quasi-primaries. In section 4 and appendix C of [27], we discussed in detail how to obtain the above result and how to compute them using the recursion.

After obtaining $\mathcal{V}_{\text{holo}}(0,h_L,c)$ as a polynomial of $\tilde{x}_0$, the coefficient of $\tilde{x}_0^N$ is the $F_N(x_0)$ in (4.4.8). This is why we keep $\tilde{x}_0$ explicitly in the global blocks (4.4.12), instead of using the fact that $\tilde{x}_0\equiv\frac{1}{x_0}$ to simplify the calculation of the global blocks, because that will mix $\frac{1}{x_0}$ with the x_0 in $F_N(x_0)$, and we will not be able to extract $F_N(x_0)$. After obtaining $F_N(x_0)$, we can simply use equation (4.4.9) to compute the full vacuum block $\mathcal{V}_0$. The Mathematica code for implementing this recursion relation is attached with this paper.

Generally, the recursion relations for computing the boundary Virasoro blocks [168, 167, 29] and the bulk-boundary Virasoro blocks consist of two parts. One is the computation

[15] As shown in section 3.2.2 of [8], solving the bulk primary conditions will give us $\mathcal{L}_{-n}\mathcal{O}$ in terms of quasi-primaries and their global descendants (see the paragraph below (4.3.7) for explanations of the notations here)

$$\mathcal{L}_{-N}\mathcal{O}=N!\left(2h\right)_N\sum_{n=0}^{N}\sum_{i}\frac{L_{-1}^{N-n}\mathcal{O}^{(i)}_{h+n}}{\left|L_{-1}^{N-n}\mathcal{O}^{(i)}_{h+n}\right|^2} \tag{4.4.15}$$

Similarly, $\mathcal{P}^{\text{holo}}_{h_1}$ can be written in terms of quasi-primaries and their global descendants. So $\mathcal{V}_{\text{holo}}\left(h_1,h_2,c\right)$ defined in (4.4.7) can also be written as a sum over quasi-primaries and their global descendants. In this way, it's easier to see why $\mathcal{V}_{\text{holo}}\left(h_1,h_2,c\right)$ can be decomposed into a sum over global blocks as in (4.4.16).

of the coefficients $C_{m,n}$. And the other part is the computation of the global blocks. The computation of $C_{m,n}$ is the most complicated part of the recursion relations, but luckily, for most observables of interest, it's universal. The difference between observables is in the global blocks, which are the seed for the recursion relations.

4.5 Heavy-Light Bulk-Boundary Correlator on the Cylinder

Our study of bulk reconstruction was motivated by the desire to understand near horizon dynamics and the black hole information paradox. This program may be advanced by computing correlation functions of bulk proto-fields in a black hole microstate background. One object of interest is the heavy-light bulk-boundary vacuum block $\mathcal{V}_0$ of $\langle \mathcal{O}_H | \mathcal{O}_L \phi_L | \mathcal{O}_H \rangle$ for ϕ_L defined in global AdS. When $|\mathcal{O}_H\rangle$ is dual to a BTZ black hole microstate ($h_H > \frac{c}{24}$), $\mathcal{V}_0$ will be dual to the bulk-boundary propagator of the light operators in such a background. In this section, we'll compute this vacuum block using two different methods.

Our first method utilizes the recursion relations introduced in the last section. This method will give us an exact result for the vacuum block as an expansion in the kinematic variables, with coefficients evaluated exactly at finite c, including all the gravitational interactions between the light probe operator and the heavy state. Our second method is based on the idea of bulk-boundary OPE blocks [8] (or bulk-boundary bi-local operators) and effective theory for boundary gravitons in AdS$_3$/CFT$_2$ [35]. This second method will give us the vacuum block in a large c expansion with $\frac{h_H}{c}$ fixed. We'll carry out the calculation up to order $\frac{1}{c}$, which corresponds to the gravitational one-loop correction to the bulk-boundary propagator in a microstate BTZ black hole background. We have verified that the results from these methods agree.

We will also show analytically that the one-loop corrections are singular at the Euclidean horizon. This effect only arises because the $1/c$ corrections to the correlators are not periodic in Euclidean time [47, 50]. When they are interpreted as correlators in the BTZ geometry, they must have a branch cut at the horizon.

Throughout this section, we'll assume the following limit

$$c \to \infty, \quad h_H \sim O(c), \quad h_L \sim O(1), \tag{4.5.1}$$

although our computation using the recursion relation is valid at finite c. We'll study the bulk protofield

$$\phi_L(y, z, \bar{z}) \equiv \phi_L(u, x, \bar{x}; f(z), f(z))$$

(with $h = \bar{h} = h_L$) to be given by equation (4.3.16) with $f(z) = e^z$ and $\bar{f}(\bar{z}) = e^{\bar{z}}$. In this case, we have

$$u = \frac{4y\sqrt{\xi\bar{\xi}}}{4+y^2}, \quad x = \frac{4-y^2}{4+y^2}\xi, \quad \bar{x} = \frac{4-y^2}{4+y^2}\bar{\xi} \tag{4.5.2}$$

with $\xi \equiv e^z$ and $\bar{\xi} \equiv e^{\bar{z}}$. As discussed in section (4.3.2), in this case, the CFT is living on the boundary cylinder with coordinates $(z, \bar{z})$ and the bulk metric (4.3.11) that's dual the

CFT vacuum $|0\rangle$ is given by

$$ds^2_{|0\rangle} = \frac{dy^2 + dzd\bar{z}}{y^2} + \frac{dz^2}{4} + \frac{d\bar{z}^2}{4} + \frac{y^2}{16}dzd\bar{z}, \tag{4.5.3}$$

which becomes the usual global AdS$_3$ metric[16] (4.3.29) via the coordinate transformation (4.3.30).

In this section, we are interested in studying bulk-boundary propagator in a heavy state background $|\mathcal{O}_H\rangle$. The semiclassical bulk metric that's dual to this heavy state is given by equation (4.3.19), i.e.

$$ds^2_{|\mathcal{O}_H\rangle} = \frac{dy^2 + dzd\bar{z}}{y^2} + \frac{1}{4}\alpha^2 dz^2 + \frac{1}{4}\bar{\alpha}^2 d\bar{z}^2 + \frac{\alpha^2\bar{\alpha}^2 y^2}{16}dzd\bar{z}, \tag{4.5.4}$$

with $\alpha = \sqrt{1 - \frac{24h_H}{c}}$ and $\bar{\alpha}$ its complex conjugate[17]. This metric is related the usual BTZ black hole metric

$$ds^2 = \left(r^2 - r_+^2\right) dt_E^2 + \frac{dr^2}{r^2 - r_+^2} + r^2 d\theta^2 \tag{4.5.5}$$

via a simple coordinate transformation,

$$z = t_E + i\theta, \quad \bar{z} = t_E - i\theta, \quad r^2 = \frac{\left(\alpha^2 y^2 - 4\right)\left(\bar{\alpha}^2 y^2 - 4\right)}{16y^2}, \tag{4.5.6}$$

where $\alpha = ir_+$ and $\alpha = -ir_+$ with $r_+ = \sqrt{\frac{24h_H}{c} - 1}$. This is why the vacuum block $\mathcal{V}_0$ of $\langle\mathcal{O}_H|\mathcal{O}_L\phi_L|\mathcal{O}_H\rangle$ has the interpretation of the bulk-boundary propagator in a BTZ black hole microstate background.

4.5.1 Recursion Relation on the Cylinder

In last section, we obtained the recursion relation for computing the bulk-boundary vacuum block in the configuration

$$\left\langle\mathcal{O}_H(\infty)\mathcal{O}_H(1)\phi_L\left(u, x, \bar{x}; f\left(z\right), \bar{f}\left(\bar{z}\right)\right)\mathcal{O}_L(0)\right\rangle. \tag{4.5.7}$$

We emphasize again that the coordinates $(0, 1, \infty)$ here are on the boundary $(x, \bar{x})$ complex plane (rather than the $(z, \bar{z})$ coordinates). In order to study the bulk-boundary propagator in a heavy background in this section, we'll consider the following configuration

$$\left\langle\mathcal{O}_H(\infty)\mathcal{O}_H(1)\phi_L\left(u, 1 - x, 1 - \bar{x}; 1 - f\left(z\right), 1 - \bar{f}\left(\bar{z}\right)\right)\mathcal{O}_L(0)\right\rangle \tag{4.5.8}$$

[16] The reason that we consider this specific ϕ_L is because it's easier to relate the global AdS$_3$ metric to the BTZ black hole metric (since their boundaries are both cylindrical), and it enables us to circumvent some technical (numerical) issues that we encounter in [27].

[17] In this section, we mostly consider the non-rotating BTZ black holes, but most of our formulas (especially those of section 4.5.2, since we've kept α and $\bar{\alpha}$ independent) are easily generalized to the rotating case. In the rotating black hole case, the relations between $\alpha, \bar{\alpha}$, $h_H, \bar{h}_H$ and r_+, r_- are a little bit more complicated.

which is equivalent to $\left\langle \mathcal{O}_H|\phi_L\left(u,x,\bar{x};f\left(z\right),\bar{f}\left(\bar{z}\right)\right)\mathcal{O}_L(1)|\mathcal{O}_H\right\rangle$ due to translational symmetry on the complex plane[18].

To obtain the bulk-boundary vacuum block $\mathcal{V}_0$ of (4.5.8), we just need to adopt the result of last section to the special case considered here. For the configuration in (4.5.8), we have

$$u=\frac{4y\sqrt{\xi\bar{\xi}}}{4+y^2},\quad \Delta x=\frac{2y^2}{y^2+4}\xi,\quad \Delta\bar{x}=\frac{2y^2}{y^2+4}\xi,\quad x_0=1-\xi,\quad \bar{x}_0=1-\bar{\xi}. \tag{4.5.9}$$

So we just need to plug the above expressions for $u,\Delta x,\Delta\bar{x},x_0,\bar{x}_0$ into (4.4.8), (4.4.9) and (4.4.12) to obtain $\mathcal{V}_0$. The Mathematica code for computing $\mathcal{V}_0$ using the recursion relation is attached with this paper. The first several terms of $\mathcal{V}_0$ from the recursion relation are given by

$$\begin{aligned}\frac{\mathcal{V}_0}{\left(\frac{u}{x_0\bar{x}_0}\right)^{2h_L}}&=1+\frac{2\left(\bar{z}^2+z^2\right)h_Hh_L}{c}-\frac{h_L}{72}\left(z^2\left(\bar{z}^2+12\right)-36z\bar{z}+12\left(\bar{z}^2+12\right)\right)s \quad (4.5.10)\\&+\frac{h_Hh_L\left(3z^3\bar{z}h_L-2z^2\left(\bar{z}^2+6\right)\left(h_L-1\right)+3z\bar{z}^3h_L-12\bar{z}^2\left(h_L-1\right)\right)}{3c}s+\cdots\end{aligned}$$

where we've defined $s\equiv\frac{y^2}{z\bar{z}}$ and expanded the LHS in terms of small s,z and $\bar{z}$ to get the RHS. We've checked that this result is consistent with the semiclassical result and $1/c$ corrections to be computed in next subsection. Since the recursion relation is valid at finite c, we can use it to study non-perturbative physics near the black hole horizon. Due to numerical difficulties of obtaining convergent and reliable result near the horizon, we postpone it to future work.

4.5.2 Quantum Corrections to $\langle\mathcal{O}_H|\phi_L\mathcal{O}_L|\mathcal{O}_H\rangle$ on the Cylinder

We can use bulk-boundary bi-local operators (as a generalization of the boundary bi-local operators in [35]) and the effective theory for boundary gravitons developed in [35] to compute the semiclassical limit of $\mathcal{V}_0$ and its $1/c$ corrections. We will briefly discuss the physical interpretation of these results at the end of this section.

Semiclassical Result

First, we notice that the semiclassical metric (4.5.4) can be obtained from the Poincare patch

$$ds^2=\frac{du^2+dxd\bar{x}}{u^2} \tag{4.5.11}$$

[18] Another way of obtaining ϕ_L in (4.5.8) is to substitute $f\left(z\right)$ and $\bar{f}\left(\bar{z}\right)$ with $1-f\left(z\right)$ and $1-\bar{f}\left(\bar{z}\right)$ in equation (4.3.16), which will not change the metrics (4.5.3).

via the coordinate transformation (4.3.10) with

$$f(z)=e^{\alpha z},\qquad \bar{f}(\bar{z})=e^{\bar{\alpha}\bar{z}},\tag{4.5.12}$$

where $\alpha=\sqrt{1-\frac{24h_H}{c}}$ and $\bar{\alpha}$ is the complex conjugate of α. This means that semiclassically, the effect of the heavy operators is trivialized via this map back to the $(u,x,\bar{x})$ coordinates[19]. So the semiclassical result of $\mathcal{V}_0$ of $\langle\mathcal{O}_H|\phi_L(y,z_1,\bar{z}_1)\,\mathcal{O}_L(z_2,\bar{z}_2)\,|\mathcal{O}_H\rangle$ must be given by the bulk-boundary propagator $\langle\phi_L(u_1,x_1,\bar{x}_1)\,\mathcal{O}_L(x_2,\bar{x}_2)\rangle$ in the $(u,x,\bar{x})$ coordinates, that is

$$\mathcal{V}_0=\left(f'(z_2)\,\bar{f}'(z_2)\right)^{h_L}\left(\frac{u_1}{u_1^2+(x_1-x_2)(\bar{x}_1-\bar{x}_2)}\right)^{2h_L}+O\left(\frac{1}{c}\right)\tag{4.5.13}$$

where the factor $\left(f'(z_1)\,\bar{f}'(z_1)\right)^{h_L}$ comes from the transformation rule for the primary operator $\mathcal{O}_L$ and

$$u_1=\frac{4y\sqrt{\alpha\bar{\alpha}e^{\alpha z_1+\bar{\alpha}\bar{z}_1}}}{\alpha\bar{\alpha}y^2+4},\quad x_1=\frac{e^{\alpha z_1}\left(4-\alpha\bar{\alpha}y^2\right)}{\alpha\bar{\alpha}y^2+4},\quad \bar{x}_1=\frac{e^{\bar{\alpha}\bar{z}_1}\left(4-\alpha\bar{\alpha}y^2\right)}{\alpha\bar{\alpha}y^2+4}$$

and $x_2=e^{\alpha z_2}$, $\bar{x}_2=e^{\bar{\alpha}\bar{z}_2}$. For later convenience, we'll denote the semi-classical result (the first term in (4.5.13)) as $\mathcal{V}_0^{\text{s-c}}$, and in terms of $\left(y,\xi,\bar{\xi}\right)$ with $\xi\equiv e^{z_1-z_2}$ and $\bar{\xi}\equiv e^{\bar{z}_1-\bar{z}_2}$, it's given by [82, 27]

$$\mathcal{V}_0^{\text{s-c}}=\left(\frac{16y^2\alpha^2\bar{\alpha}^2\xi^{\alpha}\bar{\xi}^{\bar{\alpha}}}{\left(4\left(1-\xi^{\alpha}\right)\left(1-\bar{\xi}^{\bar{\alpha}}\right)+y^2\alpha\bar{\alpha}\left(1+\xi^{\alpha}\right)\left(1+\bar{\xi}^{\bar{\alpha}}\right)\right)^2}\right)^{h_L}.\tag{4.5.14}$$

The configuration of last subsection corresponds to $z_1=z$ and $z_2=0$, so the ξ and $\bar{\xi}$ defined here are the same as those of last subsection.

$1/c$ Corrections

In order to compute the $1/c$ corrections (gravitational one-loop corrections) to the semiclassical result of $\mathcal{V}_0$, we must include perturbations in $f(z)$ and $\bar{f}(\bar{z})$, and be more precise about the central charge c. It turns out that, as in [35], we should use the following $f(z)$ and $\bar{f}(\bar{z})$

$$f(z)=e^{\alpha_0 z+\frac{i\epsilon(z)}{\sqrt{C}}}\qquad\text{and } \bar{f}(\bar{z})=e^{\bar{\alpha}_0\bar{z}+\frac{i\bar{\epsilon}(\bar{z})}{\sqrt{C}}}.\tag{4.5.15}$$

where $C=c-1$, $\alpha_0=\sqrt{1-24\frac{h_H}{C}}$ and $\bar{\alpha}_0$ is the complex conjugate of α_0. Here ϵ and $\bar{\epsilon}$ are to be understood as operators. We then obtain the large c expansion (with $\frac{h_H}{c}$ fixed) of $\mathcal{V}_0$ via

$$\mathcal{V}_0=\left\langle\left(f'(z_2)\,\bar{f}'(z_2)\right)^{h_L}\left(\frac{u_1}{u_1^2+(x_1-x_2)(\bar{x}_1-\bar{x}_2)}\right)^{2h_L}\right\rangle\tag{4.5.16}$$

[19] Note that the $(u,x,\bar{x})$ in this subsection are different from those of last subsection since $f(z),\bar{f}(z)$ are different now.

upon plugging in the expressions of various terms using (4.3.10) with f and $\bar{f}$ of (4.5.15), and expanding in large c or small $\epsilon, \bar{\epsilon}$. The idea here is roughly the same as the idea of the bulk-boundary OPE blocks used in [8] to compute $\langle \phi \mathcal{O} T \rangle$ and in [27] to compute the large c expansion of vacuum block of $\langle \mathcal{O}_H \mathcal{O}_H \phi_L \mathcal{O}_L \rangle$ (with $h_H \sim O(1)$) in Poincare AdS$_3$. It's also a generalization of the boundary bi-local operators in [35] to the bulk-boundary case.

At leading order of the large C limit (with $h_H \sim \mathcal{O}(C)$), we have

$$\mathcal{V}_0 = \left(\frac{16 y^2 \alpha_0^2 \bar{\alpha}_0^2 \xi^{\alpha_0} \bar{\xi}^{\bar{\alpha}_0}}{\left(4\left(1-\xi^{\alpha_0}\right)\left(1-\bar{\xi}^{\bar{\alpha}_0}\right) + y^2 \alpha_0 \bar{\alpha}_0 \left(1+\xi^{\alpha_0}\right)\left(1+\bar{\xi}^{\bar{\alpha}_0}\right)\right)^2} \right)^{h_L} \left(1 + \mathcal{O}\left(\frac{1}{C}\right)\right) \tag{4.5.17}$$

where $\xi \equiv e^{z_1 - z_2}$ and $\bar{\xi} \equiv e^{\bar{z}_1 - \bar{z}_2}$. At order $1/c$, we'll get two different kinds of contributions, one from the ϵ and $\bar{\epsilon}$ terms, and another from the large c expansion of the leading order result (recalling that $C = c - 1$). Since the terms linear in ϵ or $\bar{\epsilon}$ will have zero expectation value, we'll only keep the terms quadratic in ϵ or $\bar{\epsilon}$ in the large c expansion. We can then package the order $1/c$ contributions to $\mathcal{V}_0$ in the following form:

$$\mathcal{V}_0 = \mathcal{V}_0^{\text{s-c}} \left(1 + \frac{h_L}{c} \mathcal{V}_{h_L/c} + \frac{h_L^2}{c} \mathcal{V}_{h_L^2/c} + \mathcal{O}\left(\frac{1}{c^2}\right)\right) \tag{4.5.18}$$

where $\mathcal{V}_0^{\text{s-c}}$ is the semiclassical result given in (4.5.14). The order $1/c$ terms of the large c expansion of (4.5.17) will be linear in h_L and will be included in $\mathcal{V}_{h_L/c}$.

The order h_L/c term $\mathcal{V}_{h_L/c}$ of (4.5.18) is given by

$$\mathcal{V}_{h_L/c} = \frac{96\left(1-\bar{\xi}^{\bar{\alpha}}\right)^2 \left\langle \mathcal{V}_{h_L/c}^{(0)} \right\rangle + 4\alpha\bar{\alpha}\left(1-\bar{\xi}^{2\bar{\alpha}}\right) \left\langle \mathcal{V}_{h_L/c}^{(2)} \right\rangle y^2 + \bar{\alpha}^2 \left(1+\bar{\xi}^{\bar{\alpha}}\right)^2 \left\langle \mathcal{V}_{h_L/c}^{(4)} \right\rangle y^4}{D^2} + \frac{\mathcal{V}_{h_L/c}^{\text{s-c}}}{D} + c.c. \tag{4.5.19}$$

where we've defined

$$D \equiv 4\left(1-\xi^{\alpha}\right)\left(1-\bar{\xi}^{\bar{\alpha}}\right) + y^2 \alpha\bar{\alpha}\left(1+\xi^{\alpha}\right)\left(1+\bar{\xi}^{\bar{\alpha}}\right) \tag{4.5.20}$$

for notational convenience and the various terms in the numerator are given by

$$\begin{aligned}
\mathcal{V}_{h_L/c}^{(0)} &= \frac{1}{12\alpha^2}\left(1-\xi^{\alpha}\right)^2 \left(\left(\epsilon_1'\right)^2 + \left(\epsilon_2'\right)^2\right) - \frac{1}{6}\xi^{\alpha}\left(\epsilon_1 - \epsilon_2\right)^2 \qquad (4.5.21)\\
\mathcal{V}_{h_L/c}^{(2)} &= \tfrac{1}{\alpha^3}\left(4\alpha^2 \xi^{\alpha}\left(\epsilon_1 - \epsilon_2\right)\epsilon_1' + \left(1-\xi^{\alpha}\right)\left(2\left(\xi^{\alpha}-1\right)\epsilon_1'\epsilon_1'' + \alpha\left(\xi^{\alpha}+1\right)\left(\left(\epsilon_1'\right)^2 + \left(\epsilon_2'\right)^2\right)\right)\right)\\
\mathcal{V}_{h_L/c}^{(4)} &= \tfrac{1}{2}\left(\xi^{\alpha}+1\right)^2\left(\left(\epsilon_2'\right)^2 - \left(\epsilon_1'\right)^2\right) + \xi^{\alpha}\left(\alpha^2\left(\epsilon_1-\epsilon_2\right) - 4\epsilon_1''\right)\left(\epsilon_1-\epsilon_2\right) + \tfrac{\left(4\alpha\left(\xi^{2\alpha}-1\right)\epsilon_1' - \left(1-\xi^{\alpha}\right)^2\epsilon_1''\right)\epsilon_1''}{\alpha^2}
\end{aligned}$$

with $\epsilon_1 \equiv \epsilon(z_1)$ and $\epsilon_2 \equiv \epsilon(z_2)$ and the primes means the derivatives with respect to their arguments, respectively. The complex conjugate *c.c.* in Equation (4.5.19) means replacing $\xi, \bar{\xi}, \alpha, \bar{\alpha}$ with $\bar{\xi}, \xi, \bar{\alpha}, \alpha$, respectively, and also changing ϵ to $\bar{\epsilon}$. The $\mathcal{V}_{h_L/c}^{\text{s-c}}$ term plus it's complex conjugate is from the $1/c$ expansion of (4.5.17), and it's given by

$$\mathcal{V}_{h_L/c}^{\text{s-c}} = \left(1-\alpha^2\right)\left(\frac{2\left(1-\bar{\xi}^{\bar{\alpha}}\right)\left(2\xi^{\alpha} - \alpha\left(1+\xi^{\alpha}\right)\log(\xi) - 2\right)}{\alpha^2} - \frac{y^2}{2}\bar{\alpha}\left(1-\xi^{\alpha}\right)\left(1+\bar{\xi}^{\bar{\alpha}}\right)\log(\xi)\right) \tag{4.5.22}$$

The order h_L^2/c term $\mathcal{V}_{h_L^2/c}$ of (4.5.18) is given by

$$\mathcal{V}_{h_L^2/c} = \frac{\left\langle \mathcal{V}^{(0)}_{h_L^2/c} \right\rangle + \left\langle \mathcal{V}^{(2)}_{h_L^2/c} \right\rangle y^2 + \left\langle \mathcal{V}^{(4)}_{h_L^2/c} \right\rangle y^4}{D^2} + c.c., \tag{4.5.23}$$

where D is given by (4.5.20). The numerator is given by

$$\left\langle \mathcal{V}^{(0)}_{h_L^2/c} \right\rangle + \left\langle \mathcal{V}^{(2)}_{h_L^2/c} \right\rangle y^2 + \left\langle \mathcal{V}^{(4)}_{h_L^2/c} \right\rangle y^4 = \frac{1}{-2\alpha^2} \left\langle \left(V_1 + y^2 V_2\right)^2 \right\rangle \tag{4.5.24}$$

with

$$\begin{aligned} V_1 &= 4\left(\bar{\xi}^{\bar{\alpha}} - 1\right)\left(\alpha\left(\epsilon_1 - \epsilon_2\right)\left(\xi^\alpha + 1\right) + \left(1 - \xi^\alpha\right)\left(\epsilon_1' + \epsilon_2'\right)\right), \\ V_2 &= \bar{\alpha}\left(\bar{\xi}^{\bar{\alpha}} + 1\right)\left(\alpha^2\left(\xi^\alpha - 1\right)\left(\epsilon_1 - \epsilon_2\right) + \alpha\left(\xi^\alpha + 1\right)\left(\epsilon_1' - \epsilon_2'\right) + 2\left(1 - \xi^\alpha\right)\epsilon_1''\right), \end{aligned} \tag{4.5.25}$$

Now to compute $\mathcal{V}_{h_L/c}$ and $\mathcal{V}_{h_L^2/c}$, we need the ϵ propagator. This is worked out in [35] using the effective theory for boundary gravitons developed in that paper, and it's given by[20]

$$\langle \epsilon(z_1)\epsilon(z_2) \rangle = \frac{6}{C}\left(2\ln(1-\xi) + \Phi(\xi, 1, \alpha) + \Phi(\xi, 1, -\alpha)\right), \quad \xi = e^{z_1 - z_2} \tag{4.5.26}$$

where $\Phi(z, s, a)$ is the Lerch transcendant [21]

$$\Phi(z, s, a) = \sum_{n=0}^{\infty} \frac{z^n}{(n+a)^s}, \tag{4.5.28}$$

Using the ϵ propagator (4.5.26), we can evaluate $\mathcal{V}_{h_L/c}$ and $\mathcal{V}_{h_L^2/c}$, but they are logarithmically divergent and need to be renormalized. We follow the procedure in [35], and define the renormalized expectation value of the $\left\langle \mathcal{V}^{(i)}_{h_L/c} \right\rangle$ as follows

$$\left\langle \mathcal{V}^{(i)}_{h_L/c} \right\rangle_{\mathrm{R}} = \left\langle \mathcal{V}^{(i)}_{h_L/c} \right\rangle - \left(\left\langle \mathcal{V}^{(i)}_{h_L/c} \right\rangle_{\alpha \to 1, \bar{\alpha} \to 1} \right)_{w \to \alpha w, \bar{w} \to \bar{\alpha} \bar{w}, y \to y\sqrt{\alpha\bar{\alpha}}} \tag{4.5.29}$$

and similarly for $\left\langle \mathcal{V}^{(i)}_{h_L^2/c} \right\rangle_{\mathrm{R}}$.

[20] There is a subtlety about the ordering of the ϵ operators in the ϵ two-point function. When computing $\mathcal{V}_0$, we substitute $\langle \epsilon(z_1)\epsilon(z_2) \rangle$ with the symmetric average of the two different ordering $\frac{1}{2}\left(\langle \epsilon(z_1)\epsilon(z_2) \rangle + \langle \epsilon(z_2)\epsilon(z_1) \rangle\right)$. This procedure gives a result matching the recursion relation.

[21] For $s = 1$ it is related to a certain incomplete Beta function as

$$B(z, a, 0) = z^a \Phi(z, 1, a). \tag{4.5.27}$$

Since the result of $\mathcal{V}_0$ only depends on $z_1 - z_2$, we'll set $z_1 = z$ and $z_2 = 0$. Eventually, the $\left\langle \mathcal{V}^{(i)}_{h_L/c} \right\rangle_{\mathrm{R}}$ terms are given by

$$\begin{aligned}
\left\langle \mathcal{V}^{(0)}_{h_L/c} \right\rangle_{\mathrm{R}} &= e^{\alpha z}\left(\mathcal{F}_1 + 4\log\alpha + 4\log\left(2\sinh\left(\frac{z}{2}\right)\right) + 2\right) + \left(1 - e^{2\alpha z}\right)\alpha z \\
&\quad + \left(1 + e^{2\alpha z}\right)\left(H_{-\alpha} + H_{\alpha} - 1 + i\pi - 2\log\alpha\right) + 2\left(1 - e^{\alpha z}\right)^2 \log\left(2\sinh\left(\frac{\alpha z}{2}\right)\right), \qquad (4.5.30)\\
\left\langle \mathcal{V}^{(2)}_{h_L/c} \right\rangle_{\mathrm{R}} &= \left(1 - e^{2\alpha z}\right)\left(H_{-\alpha} + H_{\alpha} + i\pi - 1 - 2\log\alpha + 2\log\left(2\sinh\left(\frac{\alpha z}{2}\right)\right)\right) \\
&\quad + \left(1 + e^{2\alpha z}\right)\alpha z + e^{\alpha z}\mathcal{F}_2, \qquad (4.5.31)\\
\left\langle \mathcal{V}^{(4)}_{h_L/c} \right\rangle_{\mathrm{R}} &= \left(1 - \xi^{\alpha}\right)^2 + 6\alpha^3\left(1 - e^{2\alpha z}\right) z \qquad (4.5.32)\\
&\quad + 6\alpha^2\left(1 + e^{2\alpha z}\right)\left(H_{-\alpha} + H_{\alpha} + 2\log\left(2\sinh\left(\frac{\alpha z}{2}\right)\right) - 2\log\alpha - \frac{7}{6} + i\pi\right) \\
&\quad + 6\alpha^2 e^{\alpha z}\left(\mathcal{F}_1 - 4\log\alpha - 4\log\left(2\sinh\left(\frac{z}{2}\right)\right) + 4\log\left(2\sinh\left(\frac{\alpha z}{2}\right)\right) + \frac{7}{3}\right),
\end{aligned}$$

and the $\left\langle \mathcal{V}^{(i)}_{h_L^2/c} \right\rangle_{\mathrm{R}}$ terms are given by

$$\begin{aligned}
\left\langle \mathcal{V}^{(0)}_{h_L^2/c} \right\rangle_{\mathrm{R}} &= 192\left(1 - e^{\bar{\alpha}\bar{z}}\right)^2\left(\left(\log\left(\sinh\left(\frac{z}{2}\right)\right) - \log\left(\sinh\left(\frac{\alpha z}{2}\right)\right) + \log\alpha + 1\right)\left(1 - e^{\alpha z}\right)^2\right. \\
&\quad \left. + \mathcal{F}_3 + 2e^{\alpha z}\left(H_{-\alpha} + H_{\alpha} + 2\log\left(2\sinh\left(\frac{z}{2}\right)\right) + i\pi\right)\right), \qquad (4.5.33)\\
\left\langle \mathcal{V}^{(2)}_{h_L^2/c} \right\rangle_{\mathrm{R}} &= 48\bar{\alpha}\left(1 - e^{\alpha z}\right)\left(1 - e^{2\bar{\alpha}\bar{z}}\right)\left(\coth\left(\frac{z}{2}\right)\left(1 - e^{\alpha z}\right)\right. \qquad (4.5.34)\\
&\quad \left. + 2\alpha\left(e^{\alpha z} + 1\right)\left(\log\alpha - \log\left(\sinh\left(\frac{\alpha z}{2}\right)\right) + \log\left(\sinh\left(\frac{z}{2}\right)\right) + \frac{1}{2}\right)\right), \\
\left\langle \mathcal{V}^{(4)}_{h_L^2/c} \right\rangle_{\mathrm{R}} &= 2\alpha^2\bar{\alpha}^2\left(1 + e^{\bar{\alpha}\bar{z}}\right)^2\left(\frac{6}{\alpha}\coth\left(\frac{z}{2}\right)\left(1 - e^{2\alpha z}\right) + 6\log\alpha\left(1 + e^{\alpha z}\right)^2 - e^{2\alpha z}\right. \\
&\quad + 6\left(1 - e^{\alpha z}\right)^2\left(\log\left(\sinh\left(\frac{z}{2}\right)\right) - \log\left(\sinh\left(\frac{\alpha z}{2}\right)\right) + \frac{1}{6\alpha^2}\right) - 6\mathcal{F}_3 \\
&\quad \left. - 1 - 12e^{\alpha z}\left(H_{-\alpha} + H_{\alpha} + 2\log\left(2\sinh\left(\frac{\alpha z}{2}\right)\right) + i\pi - \frac{13}{6}\right)\right). \qquad (4.5.35)
\end{aligned}$$

where we've define

$$\begin{aligned}
\mathcal{F}_1 &\equiv \Phi\left(e^{z}, 1, \alpha\right) + \Phi\left(e^{z}, 1, -\alpha\right) + \Phi\left(e^{-z}, 1, \alpha\right) + \Phi\left(e^{-z}, 1, -\alpha\right) \\
\mathcal{F}_2 &\equiv \Phi\left(e^{z}, 1, \alpha\right) - \Phi\left(e^{z}, 1, -\alpha\right) - \Phi\left(e^{-z}, 1, \alpha\right) + \Phi\left(e^{-z}, 1, -\alpha\right) \\
\mathcal{F}_3 &\equiv e^{2\alpha z}\Phi\left(e^{z}, 1, \alpha\right) + \Phi\left(e^{z}, 1, -\alpha\right) + \Phi\left(e^{-z}, 1, \alpha\right) + e^{2\alpha z}\Phi\left(e^{-z}, 1, -\alpha\right)
\end{aligned}$$

for notational convenience.

We checked that these results agree with the recursion relation of last subsection by expanding in small[22] $s \equiv \frac{y^2}{z\bar{z}}$, z and $\bar{z}$. We have also verified that when we take ϕ_L to the boundary, we recover the $1/c$ correction [47] to the heavy-light vacuum block [48].

[22] The branch cuts in the various functions in the equations (4.5.30)-(4.5.35) were chosen such that when they are expanded in small z assuming $z > 0$ in Mathematica, the result matches that of the recursion relation. One should be careful when evaluating (4.5.30)-(4.5.35) for $z < 0$.

One-Loop Corrections Near the Black Hole Horizon

We would like to see what the semiclassical result and the $1/c$ corrections computed here tell us about physics near the Euclidean black hole horizon. For the non-rotating case considered here, the horizon is at $r = r_+ = \sqrt{\frac{24h_H}{c} - 1}$ (which corresponds to $y = \frac{2}{r_+}$). In this case, the semi-classical result $\mathcal{V}_0^{\text{s-c}}$ of equation (4.5.14) written in terms of the (r, t_E, θ) (using the coordinate relations (4.5.6)) is given by

$$\mathcal{V}_0^{\text{s-c}} = \left(\frac{r_+}{2}\right)^{2h_L} \frac{1}{\left(\frac{r}{r_+}\cosh\left(r_+\theta\right) - \frac{\sqrt{r^2-r_+^2}}{r_+}\cos\left(r_+t_E\right)\right)^{2h_L}}. \tag{4.5.36}$$

which is the $n = 0$ terms in the full semiclassical bulk-boundary correltator for a free field in a BTZ black hole given by the image sum in [82]. So one can see that the semiclassical result is periodic in t_E, and it's smooth at the horizon $r = r_+$ (and its dependence on t_E drops out there).

In terms of $(y, z, \bar{z})$, the horizon is at $y = \frac{2}{r_+}$, and at this value of y, the $\frac{1}{c}$ corrections $\mathcal{V}_{h_L/c}$ and $\mathcal{V}_{h_L^2/c}$ to the vacuum block of $\langle \mathcal{O}_H | \phi_L \mathcal{O}_L | \mathcal{O}_H \rangle$ are finite, since their numerators truncate at order y^4, and their denominators are just the same as the denominator of the semiclassical result $\mathcal{V}_0^{\text{s-c}}$, which is also non-singular at this value of y. However, unlike the semiclassical vacuum block, the functions $\mathcal{V}_{h_L/c}$ and $\mathcal{V}_{h_L^2/c}$ are not periodic in Euclidean time [47, 50]. This means that the $1/c$ correction to the bulk-boundary heavy-light correlator will have a branch point at the Euclidean horizon. So the singularity of these correlators at the Euclidean horizon arises already in perturbation theory, and does not require non-perturbative effects [27].

4.6 Discussion

Our primary goal has been to develop an exact bulk reconstruction procedure with very general gravitational dressings. The motivation was to enable future investigations into the dressing-dependence of bulk observables, as these ambiguities present a major caveat when drawing physical conclusions. For example, using our results it should be possible to determine if the breakdown of bulk locality at short-distances in AdS$_3$ [26] persists with a general class of gravitational dressings. We can also investigate BTZ black hole horizons [27], though the necessary numerics may be rather formidable. We took the first steps in this direction in section 4.5. By exploiting the connection between the singularity structure of CFT stress-tensor correlators and gravitational dressings, it may be possible to generalize some of our results to higher dimensions.

Our reconstruction procedure only incorporates effects arising as a mandatory consequence of Virasoro symmetry. With hard work one could add other perturbative interactions, but such methods would likely just reproduce bulk perturbation theory, without providing a deeper understanding of quantum gravity. So our methods are limited, as they are only able to address certain universal features of quantum gravity. Unfortunately, in the case of quantum gravity it would seem that we must either solve toy models completely, and then try to argue that they are representative, or solve a universal sector of a general class of models, and try to argue that the effects from this sector determine the relevant physics. Given the universal nature of the gravitational force, we believe that the latter route is a more compelling way to address locality and near-horizon dynamics.

Most work on bulk reconstruction suffers from a nagging conceptual problem. As physical observers, we do not setup experiments by making reference to the boundary of spacetime. And defining the bulk by reference to the boundary seems even more perverse in a cosmological setting. Furthermore, it has been shown that using the boundary as a reference point leads to fundamental problems, such as bulk fields that do not commute outside the lightcone [41], even at low-orders in gravitational perturbation theory. Perhaps a more sensible approach defines observables relative to other objects in spacetime, just as we define our local reference frame with respect to the earth, solar system, galaxy, and galactic neighborhood. This also seems more in keeping with interpretations of the Wheeler-DeWitt equation [39]. We hope to formalize such a definition of local observables in future work.

Acknowledgments

We would like to thank Ibou Bah, Liam Fitzpatrick, and David E. Kaplan for discussions. HC, JK and US have been supported in part by NSF grant PHY-1454083. JK was also supported in part by the Simons Collaboration Grant on the Non-Perturbative Bootstrap.

Appendix A

Appendix to Chapter 2

A.1 Technical Details and Minor Results

A.1.1 Fitting

To extract the scaling exponent α we need to fit power-laws to the empirical $L(N)$ for trained models with N parameters. For this purpose we simply fit straight lines to $\log L$ vs $\log N$, assuming that the error in $\log L$ was independent of N (ie we assumed Gaussian errors in $\log L$). We fit from the smallest value of N tested until the power-law behavior breaks down. This point is quite clear visually in most cases, as seen in figures 2.5, A.2, and 2.9. For the case where we had networks with both different widths and different depths 2.5 we only used the networks that performed among the best at each model size (ie we used points on the 'convex hull' in the L vs N plane).

However, to avoid bias we determined the last point to include in the fit in the following way. We fit a circle (parameterized by its center and radius) to the first $n \geq 3$ points in the $\log L$ vs $\log N$ plane (starting at $N = N_{\text{min}}$), and evaluated $r(n)$, the radius of the best-fit circle for each n. We then chose the value of n that achieved the maximal radius r, as this is the 'most linear' set of points. Finally, we fit a straight line $\log L = -\alpha \log N + b$ to this collection of points to determine α.

Note that this provides an alternative way to determine N_{max}, the largest network in the power-law scaling region. This was the input for figure A.1, where we show N_{max} as a function of d for teacher/student experiments.

The power-law scaling breaks down in CIFAR10 and other small image datasets due to overfitting. We do not have a complete understanding of why it breaks down for the teacher/student experiments, but it seems to be due to a failure of optimization, perhaps related to numerical precision. We note that the power-law behavior persists to larger model size and smaller loss with the deeper networks in figure 2.5.

A.1.2 Teacher/Student Experiments

Network Architectures

Our teacher networks had shape $[20, 600, 600, 2]$ (i.e. 20 dimensional input, two hidden layers of output dimension 600, and final layer ouput of dimension 2) for experiments with cross entropy loss (figures 2.5, 2.7 and 2.8), $[20, 600, 600, 1]$ for MSE loss (figure A.2) and $[9, 240, 240, 2]$ for cross entropy loss with vetted teacher (figure A.3). The teachers are randomly initialized, with biases set to zero, and weights picked from a gaussian distribution of mean zero and standard deviation $1/\sqrt{N}$, where N is the input size of the layer. We experimented with including random non-zero biases, but did not find that they significantly alter the behavior of teachers.

For experiments with mean-squared error loss, the teacher and student networks each outputted a single real value. For experiments using a cross-entropy loss, networks output two logits, and we computed the cross entropy directly from these teacher outputs (ie we did not sample discrete values from the teacher, but used its exact output distribution). For cross-entropy experiments we used students with 2, 3, and 4 hidden layers, and let the best performing models define the $L(N)$ fits, while for MSE loss we simply used students with 2 hidden layers.

We ran 10 trials each for cross-entropy and MSE losses, and in each case selected the ones with the 9 lowest losses. Intrinsic dimension calculations were done using the same 9 networks. For vetted teacher experiments, we took 90 trials and computed the mean of the loss excluding the 10 worst performing students.

Optimization and LR Schedule

We use the ADAM optimizer [84] with default settings except for the learning rate. In order to optimize effectively, we scanned over a grid of learning rates, and experimented with cosine, linear, and step-function learning rate schedules. We ended up using step function schedules for teacher/student experiments, and a constant learning rate for CIFAR10 and other image datasets, as these performed roughly as well or better than other choices. We did not find it necessary to vary the overall learning rate among different network sizes, but the schedules themselves were important for optimization. Our learning rate schedules for the various teacher/student experiments in the paper (labeled by associated figures) are summarized in table A.1.

A.1.3 Vetting Teachers to Increase Intrinsic Dimension

In figure 2.6, the ID is typically smaller than the number of features, especially when the latter is large. One might worry that this indicates ID measurements are inaccurate. In

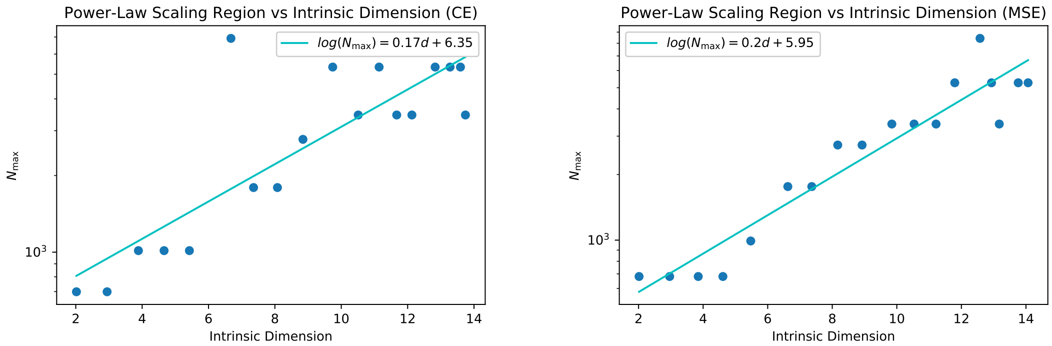

Figure A.1: This figure shows the maximum number of parameters N_{max} at which we observe power-law scaling of $L(N)$, as a function of the intrinsic dimension, for teacher/student experiments. This N_{max} is determined as described in appendix A.1.1. The left plot uses cross-entropy loss, while the right uses MSE loss. This plot should be viewed as a more empirical (but less well understood) alternative to figure 2.2.

Experiment (T/S)	student architecture	training steps	batch size	learning rate (ADAM)
(random)	MSE: [20,n,n,1]	0-200k	200	0.01
figures 2.6, 2.7, 2.8	CE: [20,n,n,2]	200-220k	1000	0.01
		220-240k	4000	0.001
(vetted)		0-100k	200	0.01
figure A.3	[9,n,n,2]	100-150k	200	0.001
		150-170k	200	0.0001

Table A.1:: Architectures and training schedules for Teacher/Student experiments in the paper, referenced by the figures in which the results are described.

fact, we believe that this occurs partly because randomly initialized teacher networks do not typically produce fully generic functions of their inputs.

We can partially remedy this problem by generating a large number of teachers and vetting them, keeping only those that produce the most complicated and non-linear functions of their inputs. The result is pictured in figure A.3, where we repeat the experiment of section 2.3.1 with up to 9 features. We see that sufficiently vetted teachers have ID nearly equal to their feature count, and that the relationship $\alpha \approx \frac{4}{d}$ continues to hold.

Presumably many vetting procedures could be successfully applied to filter the teacher networks. To increase the complexity and non-linearity of teachers so that ID would better match the number of input features, we followed this ad-hoc approach:

1. For a given teacher, we took a random slice along each input coordinate axis (i.e. the values of the other coordinates are chosen uniformly at random from $[-1/2, 1/2)$). We performed linear regression on this slice and computed the score(R^2, the coefficient of determination), and took the mean of the scores across coordinate axes. A low score implies more non-linearity.

2. We repeated this procedure 200 times and computed the mean score of all the trials. This is the score for the teacher.

3. We iterated over 5000 randomly generated teachers and selected the one with the minimum score.

A.1.4 CNNs on CIFAR10, MNIST, FMNIST, and SVHN

For CIFAR10 we used the architecture from the tensorflow CNN tutorial [107], and modified the channel width. The architecture is recorded in table A.2.

The networks were trained for 50 epochs with the ADAM optimizer with default hyperparameters. We use 40 iterations of each network and average the loss (on log scale) over the iterations. Note that we record the test and training loss at the early stopping point where the test loss reaches its minimum value. These are the results in figure 2.9.

For MNIST [91], fashion MNIST [162], and SVHN [120], we use a slightly smaller network (3 instead of 4 hidden layers) with architecture shown in table A.3. We used a smaller network in the hopes of identifying a power-law scaling region without significant overfitting.

For MNIST and fashion MNIST, we ran each network for 20 trials and took the mean loss (on log scale). The networks were trained for 50 epochs with the ADAM optimizer with default hyperparameters. As with CIFAR10, we take the minimum test loss during training (i.e. early stopping), and also report training loss at this point.

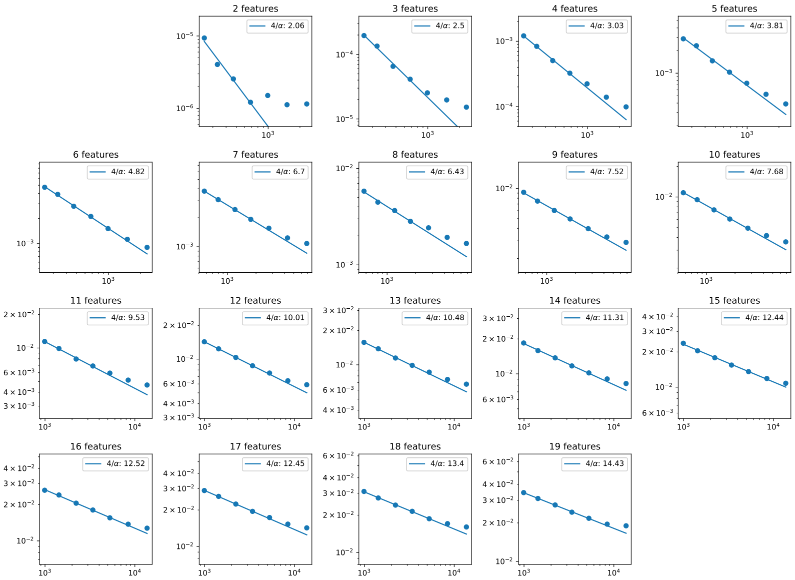

Figure A.2: This figure shows $L(N)$ with a MSE loss for students (all with 2 hidden layers) learning from a randomly initialized teacher with 2-19 features. Figure 2.5 shows the results for cross-entropy loss.

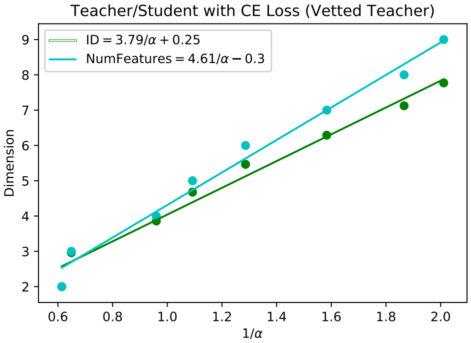

Figure A.3: This figure shows the number of features and ID vs $1/\alpha$ for vetted teachers. ID is still smaller than the number of input features, but vetting partially closes the gap. Compare the slope of 4.61 for number of features vs $1/\alpha$ here to the left of figure 2.6, where the slope was 5.48. Slopes for ID vs $1/\alpha$ are very similar with or without vetting.

For SVHN, the networks were trained for 5 epochs with both training and additional datasets used for training (total 604k images), and test dataset (26k images) for testing. We used default hyperparameters.

A.1.5 Scaling of KL Divergence with Piecewise Linear Logits

We assume the logits $c_i(x)$ are linear in a small region of volume s^d we take to surround the origin, and that the underlying probability distribution $f_i(x)$ over k discrete choices is smooth. The loss in this region is

$$L = \sum_{i=1}^{k} \int d^d x f_i(x) \log \frac{f_i(x)}{q_i(x)} \tag{A.1.1}$$

where $\log q_i(x) = c_i(x) + \log\left(\sum_{j=1}^{k} e^{c_j(x)}\right)$. If we write $q_i(x) = f_i(x) + \delta_i(x)$ then as is well known

$$\begin{aligned} L &= \sum_{i=1}^{k} \int d^d x f_i(x) \log \frac{f_i(x)}{f_i(x) + \delta_i(x)} \\ &= \int d^d x \sum_{i=1}^{k} f_i(x) \left(0 - \frac{\delta_i(x)}{f_i(x)} + \frac{1}{2}\left(\frac{\delta_i(x)}{f_i(x)}\right)^2 + \cdots\right) \\ &\approx \int d^d x \sum_{i=1}^{k} \frac{1}{2} \frac{\delta_i(x)^2}{f_i(x)} \end{aligned} \tag{A.1.2}$$

After optimization the linear $c_i(x)$ will determine a $\delta_i(x)$ that is quadratic in x, and so the loss per unit volume will scale as s^4, as claimed.

A.2 Review of Intrinsic Dimension Estimation Methods

In this section we review the two nearest neighbor method [10] and explain that it can be extended to k-nearest neighbors. Then we note that the same analysis derives the maximum likelihood method [95].

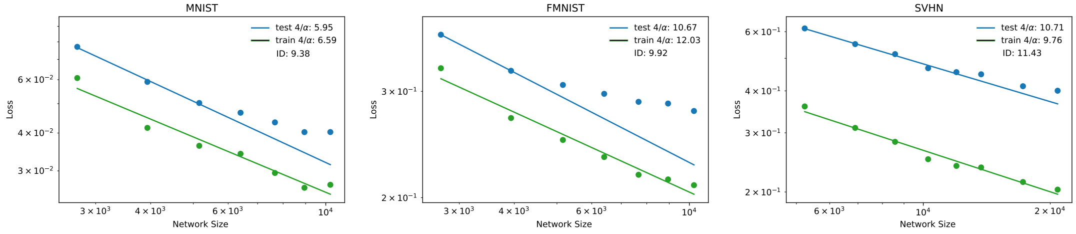

Figure A.4: This shows train and test loss on MNIST, Fashion MNIST, and test loss on SVHN, along with the exponents and ID measurement.

Layer	Output shape
Conv2D	$(32, 32, n)$
MaxPooling2D	$(16, 16, n)$
Conv2D	$(16, 16, 2n)$
MaxPooling2D	$(8, 8, 2n)$
Conv2D	$(6, 6, 2n)$
Dense	(64)
Output	(10)

Table A.2:: Architecture of the CNN network used for CIFAR10. We chose n in the range $1 \leq n \leq 13$ to minimize overfitting. All convolutions were 3×3 with unit stride, and the images have 3 colors, so the network has a total of $N = 714 + 4640n + 54n^2$ parameters.

Layer	Output shape
Conv2D	$(28, 28, n)$
MaxPooling2D	$(14, 14, n)$
Conv2D	$(12, 12, n)$
MaxPooling2D	$(6, 6, n)$
Dense	(32)
Output	(10)

Layer	Output shape
Conv2D	$(32, 32, n)$
MaxPooling2D	$(16, 16, n)$
Conv2D	$(14, 14, n)$
MaxPooling2D	$(7, 7, n)$
Dense	(32)
Output	(10)

Table A.3:: Architecture of the CNN network used for MNIST and fashion MNIST (left) and SVHN (right). All convolutions were 3×3 with unit stride.

A.2.1 The Two Nearest Neighbor Method

Assume that points are drawn from a distribution with density $\rho(x)$ with support on a d-dimensional manifold in a potentially much higher dimensional ambient space. We will see that $\rho(x)$ drops out of our results, assuming that it is constant across the first few nearest neighbors, so we will drop its explicit x-dependence in what follows.

The probability of finding n points from the dataset in a region with d-dimensional volume V is Poisson:

$$P_n(V) = \frac{(\rho V)^n}{n!} e^{-\rho V} \tag{A.2.1}$$

To see this, note that in an infinitesimal volume δV, $P_0 = 1 - \rho\delta V$ and $P_1 = \rho\delta V$, with all $P_{n>1} = 0$. Thus the generating function for P_n in a finite volume V can be found by taking the product of binomial distributions over all δV in V, giving

$$G(x;V) = \lim_{\delta V\to 0} ((1 - \rho\delta V) + x\rho\delta V)^{\frac{V}{\delta V}} = \sum_{n=0}^{\infty} \frac{(x\rho V)^n}{n!} e^{-\rho V} \tag{A.2.2}$$

The coefficients of x^n are the P_n above.

With this result in hand, we can consider the distribution of nearest-neighbor distances. Consider some point in the dataset. The probability for its nearest neighbor to be in $[r_1, r_1 + dr]$ is given by the product of the probability that there are no points in $r < r_1$ times the probability of finding a point in the shell $r_1 < r < r_1 + dr$, which is

$$P(r_1)dr_1 = \left(d\rho\omega_d r_1^{d-1} dr_1\right) e^{-\rho\omega_d r_1^d} \tag{A.2.3}$$

where ω_d is the volume of a unit d-ball. This result easily generalizes to the case where there are many r_i corresponding to the first k nearest neighbors. For example for two nearest neighbors we find

$$P(r_1, r_2)dr_1 dr_2 = (\rho\omega_d d)^2 e^{-\rho\omega_d r_2^d} r_1^{d-1} r_2^{d-1} dr_1 dr_2 \tag{A.2.4}$$

since we are demanding that there are two points on two infinitesimal shells at radii r_1, r_2 and no points otherwise.

Now we can compute the distribution over nearest neighbor distances, and their ratios. The TwoNN method [10] is based on the distribution of the ratio $\mu_2 = r_2/r_1$, which we can compute by integrating over r_1, r_2 while fixing their ratio:

$$\begin{aligned} P(\mu_2) &= \int dr_1 dr_2 \delta\left(\mu_2 - \frac{r_2}{r_1}\right)(\rho\omega_d d)^2 e^{-\rho\omega_d r_2^d} r_1^{d-1} r_2^{d-1} \\ &= \int dr_1 (\rho\omega_d d)^2 e^{-\rho\omega_d \mu^d r_1^d} r_1^{2d-1} \mu_2^{d-1} \\ &= \frac{d}{\mu_2^{d+1}} \end{aligned} \tag{A.2.5}$$

This means that the cumulative distribution for μ_2 is

$$C(\mu) = \int_1^{\mu} \frac{d}{\mu_2^{d+1}} d\mu = 1 - \frac{1}{\mu_2^d} \tag{A.2.6}$$

This means that *we can identify the dimension d by measuring the slope of a linear fit of* $\log \mu_2$ *vs* $\log(1 - C(\mu_2))$. That's the TwoNN method, as seen in figure A.5.

A.2.2 Extension to k-Neighbors and MLE

The beauty of the TwoNN method [10] is that it uses very short-distance information, and so it's plausible that the density $\rho(x)$ can be well-approximated as a constant. A downside of this method is that it primarily measures the dimension on short scales. This can be mitigated by applying the method while sampling different numbers of points from the data distribution, but it's also easy to validate the TwoNN method by simply using more neighbors.

Let's see what happens with three neighbors, and then we will generalize. We can compute the distribution of $\mu_2 = r_2/r_1, \mu_3 = r_3/r_1$, and use it for validation. We have

$$\begin{aligned} P(\mu_2, \mu_3) &= \int dr_i \delta\left(\mu_2 - \frac{r_2}{r_1}\right) \delta\left(\mu_3 - \frac{r_3}{r_1}\right) (\rho\omega_d d)^3 e^{-\rho\omega_d r_3^d} (r_1 r_2 r_3)^{d-1} \\ &= \int dr_1 (\rho\omega_d d)^3 e^{-\rho\omega_d \mu_3 r_3^d} r_1^{3d-1} \mu_2^{d-1} \mu_3^{d-1} \\ &= \frac{2d^2 \mu_2^{d-1}}{\mu_3^{2d+1}} \end{aligned} \tag{A.2.7}$$

Intuitively, large μ_3 becomes unlikely because it implies that there are few points inside a large radius, but with fixed μ_3, a larger value of μ_2 is more probable due to the larger volume at large radius.

We find a nice simplification when we study $P(\mu_3)$ and its cumulative distribution after

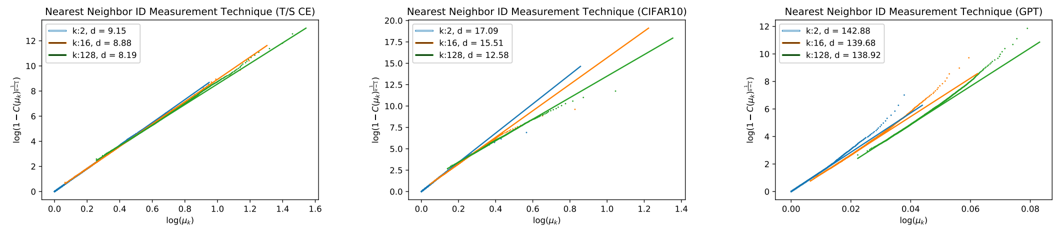

Figure A.5: This figure shows the relationship in equation A.2.16, which we use to determine the ID using the nearest neighbor method. We display examples using teacher/student data, CIFAR10, and GPT.

marginalizing over μ_2. The probability distribution is

$$P(\mu_3) = \int_1^{\mu_3} d\mu_2 \frac{2d^2\mu_2^{d-1}}{\mu_3^{2d+1}} = \frac{2d}{\mu_3^{2d+1}}\left(\mu_3^d - 1\right) \tag{A.2.8}$$

The cumulative distribution is then

$$C(\mu_3) = \left(1 - \frac{1}{\mu_3^d}\right)^2 \tag{A.2.9}$$

Thus we also find a simple method for identifying d based on μ_3 alone, namely

$$d = \frac{\log\left(1 - \sqrt{C(\mu_3)}\right)}{\log \mu_3} \tag{A.2.10}$$

This directly generalizes the TwoNN; in practice we measure d via a linear fit to the numerator as a function of the denominator in this expression.

Generalizing to k neighbors, the probability distribution for $\mu_2, \cdots, \mu_k$ is

$$P(\mu_i) = d^{k-1}(k-1)! \frac{\prod_{i=2}^{k-1} \mu_i^{d-1}}{\mu_k^{1+d(k-1)}} \tag{A.2.11}$$

for $\mu_i = r_i/r_1$. This can be used directly for maximum likelihood estimation [95]. If we maximize $\log P$ with respect to d we find

$$d = \frac{k-1}{(k-1)\log \mu_k - \sum_{j=2}^{k-1} \log \mu_j} \tag{A.2.12}$$

In fact, this MLE estimator is biased; the unbiased estimator is [95]

$$d = \mathbb{E}\left[\frac{k-2}{(k-1)\log \mu_k - \sum_{j=2}^{k-1} \log \mu_j}\right] \tag{A.2.13}$$

In practice, we can compute the RHS for all points in the manifold (after fixing some value for the number of neighbors k) and compute the mean. We display a histogram of the MLE estimates over many points in the data manifold for two examples in figure A.6. The variance provides some measure of the errors. Alternatively, we could directly measure $\log P$ and evaluate the likelihood as a function of d. The variance of this estimator was studied in [95]. They also found numerically that it can be useful to tune of the value of k, as very small k overestimates ID while large k underestimates ID.

We can use these results to extend the TwoNN method in a simple way to general k. Marginalizing over all but μ_k, we find that

$$P(\mu_k) = \frac{(k-1)d}{\mu_k^{(k-1)d+1}}\left(\mu_k^d - 1\right)^{k-1} \tag{A.2.14}$$

which leads to the cumulative distribution

$$C(\mu_k) = \left(1 - \frac{1}{\mu_k^d}\right)^{k-1} \tag{A.2.15}$$

and the formula

$$d = \frac{\log\left(1 - C(\mu_k)^{\frac{1}{k-1}}\right)}{\log \mu_k} \tag{A.2.16}$$

for the kth nearest neighbor. This can be used as a cross-check for TwoNN. For examples of the relationship between the numerator and denominator with various k, and the relevant fits, see figure A.5. Just as with MLE, we find empirically that larger k leads to smaller estimates of ID (see figure A.10).

A.3 Examples and Tests of Intrinsic Dimension Estimation

The MLE and TwoNN methods have been tested and demonstrated by their authors [95, 10]. We conduct a few tests with synthetic data. Then we provide some other examples of the ID measurement process, including errors, using our student/teacher, CIFAR10, and language data.

A.3.1 Tests on Synthetic Data

As a baseline test, we evaluate the TwoNN and MLE methods on synthetic datasets with dimensions ranging from 2 to 128, with results in figure A.7. We display synthetic data on the hypercube $[0, 1]^d$ as well as a d-torus $S^1 \times S^1 \times \cdots \times S^1$ embedded in $2d$ dimensions (in the simplest way, by embedding each circle factor in 2 Euclidean dimensions).

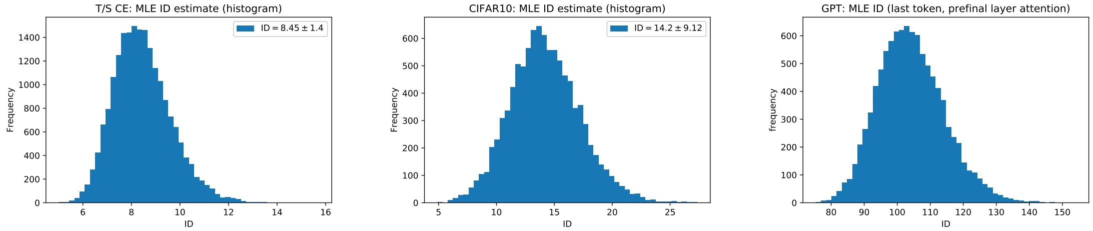

Figure A.6: These figures show a histogram of the results for d from MLE (with 100 neighbors) among all of the points used for measurement. On the left we have a teacher with 10 features, in the middle we have the $n = 5$ CNN trained on CIFAR10, while on the right we have the GPT model's prefinal attention output for the last token in the text sequence. Smaller numbers of neighbors typically give larger IDs.

We notice that 1) results are more accurate for smaller d, with quite reliable results for the TwoNN method for $d \lesssim 20$, 2) at large d all methods tend to underestimate the true ID, but 3) its certainly possible to both under and over-estimate the true ID, and measurements are not necessarily even monotonic with the number of points used for the measurement. We also see that for the torus the ID estimates are reasonably accurate even for dimensions ~ 100, though there's certainly no guarantee that this will hold for unknown data manifolds.

As other authors have noted [22], the ID is under-estimated on the hypercube, likely because cubes have sharp boundaries and corners which reduce the number of neighbors. Similarly, we believe that the ID is often over-estimated for the torus because (due to the curvature of the circles in the embedding space) points are often closer together than they would be in flat Euclidean space. We have also seen as shown in [95] that for small k the MLE method typically overestimates ID. The NN method seems a bit less sensitive to k as compared to MLE.

A.3.2 Tests on Neural Network Activations

In all cases we measure ID from fully trained networks, and we always use students (not teachers) in that context. There are a large variety of potential statistical and systematic errors associated with these measurements:

- Variation among IDs measured from students of the same size and trained with the same teacher network (or dataset), but with different initialization (see figure A.9).
- Variation of ID measurements among random groups of points sampled from the same data manifold
- Dependence of ID on the number of points used (and so the overall density) from the data manifold. More points samples shorter distance scales on the manifold. See figure A.8.
- Dependence of ID on how many nearest neighbor points are used, either for NN (see figure A.10) or MLE type estimation.
- Variation of ID from among points in different locations on the data data manifold (we show a histogram of results from MLE in figure A.6)
- Dataset specific distinctions, eg from the same or different classes in an image classifier, or from the same or different text sequences in a language model (discussed in section 2.3.4)
- Dependence of ID measurements on the layer studied (see figures 2.10 and A.8)

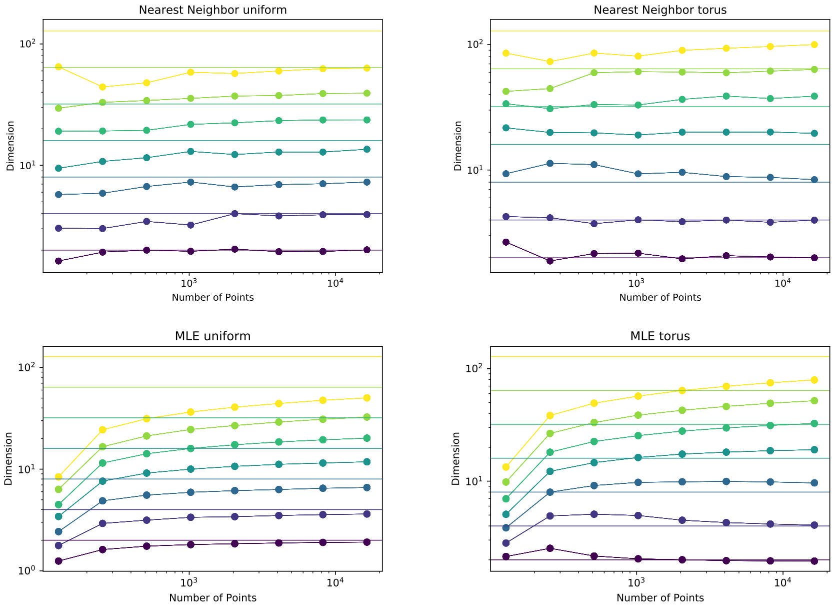

Figure A.7: Here we show measured ID as a function of the number of points in the dataset used for the measurement, for both the TwoNN (top) and MLE (bottom) methods (with $k = 100$). The left plots show a uniform distribution in the hypercube $[0, 1]^d$, while the plot on the right show a d-torus embedded in $2d$ dimensions.

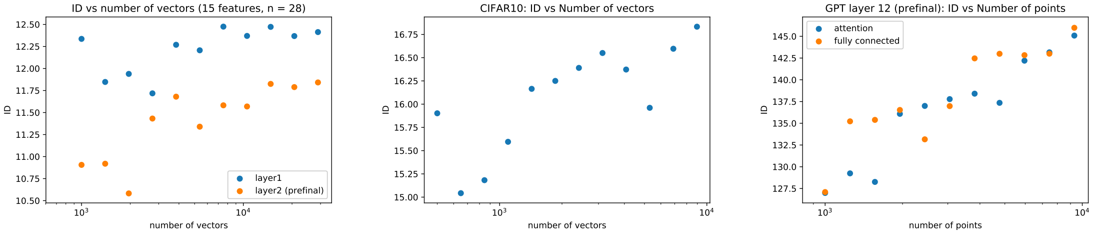

Figure A.8: Variation of Intrinsic Dimension(ID) with number of vectors for a single student network (left), for the last layer of an $n = 5$ CNN trained on CIFAR10 (middle), and also for the last layer and last token of GPT (right). The student is of size $[15, 28, 28, 2]$ and was trained on teacher with 15 features.

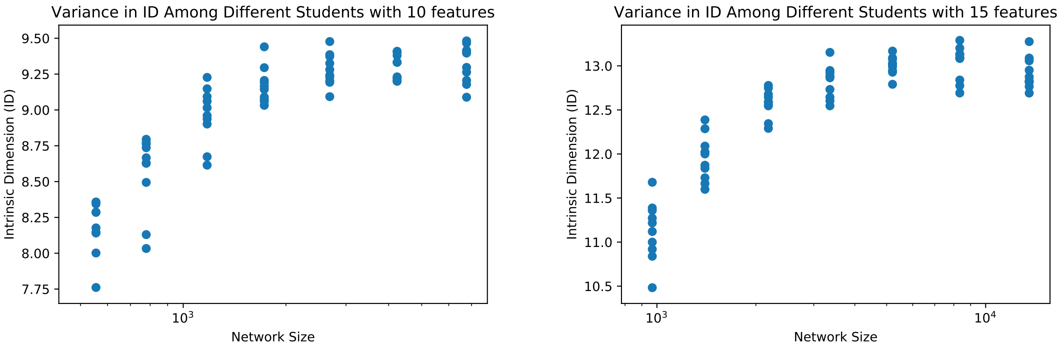

Figure A.9: Variation of Intrinsic Dimension (ID) across network sizes for a single teacher. The figure on the left shows number of inputs features = 10 and the one on the right has 15. Each point on either figure is one student. All students on each figure are trained on the same teacher, but the teacher for the left and right figures are different.

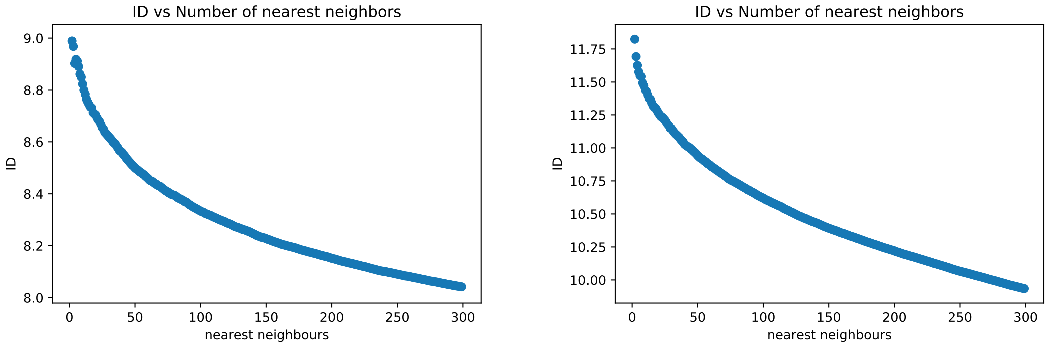

Figure A.10: Variation of Intrinsic Dimension (ID) with number of neighbors used in the algorithm. The figure on the left shows a student of size $[20, 25, 25, 2]$ trained on a teacher with 10 features, while the one on the right has student shape $[15, 28, 28, 2]$ trained on teacher with 15 features.

We provide some brief information about many of these sources of variation in the referenced plots. In most cases we find that the variation of the ID is small as long as it is measured with sufficiently many vectors. It would be interesting obtain a more precise theoretical and experimental characterization of these methods in the future.

But as evidenced by the synthetic examples in figure A.7, this does not lead us to believe that the IDs are fully trustworthy, especially when they are measured to be large. Though the apparent statistical errors in ID measurements may seem small, there may be systematic errors that are more difficult to observe.

It's conceivable that deficiencies in ID measurement actually work to the advantage of the theory relating d and $4/\alpha$. For example, d tends to be underestimated when the data manifold has a boundary (or simply less support in some region), but this may also correlate with regions of the manifold where there really is less data, and these regions do not need to be modeled as precisely to achieve a good test loss. But we leave a more thorough investigation of such subtleties to future work.

Appendix B

Appendix to Chapter 3

B.1 Experimental setup

Figure 3.1 (top-left) Experiments are done using Neural Tangents [122] based on JAX [18]. All experiment except denoted as (CNN), use 3-layer, width-8 fully-connected networks. CNN architecture used is Myrtle-5 network [140] with 8 channels. Relu activation function with critical initialization [138, 92, 163] was used. Unless specified softmax-cross-entropy loss was used. We performed full-batch gradient descent update for all dataset sizes without L2 regularization. 20 different training data sampling seed was averaged for each point. For fully-connected network input pooling of size 4 was performed for CIFAR-10/100 dataset and pooling of size 2 was performed for MNIST and Fashion-MNIST dataset. This was to reduce number of parameters in the input layer (# of pixels × width) which can be quite large even for small width networks.

Figure 3.1 (top-right) All experiments were performed using a Flax [66] implementation of Wide ResNet 28-10 [166], and performed using the Caliban experiment manager [132]. Models were trained for 78125 total steps with a cosine learning rate decay [100] and an augmentation policy consisting of random flips and crops. We report final loss, though we found no qualitative difference between using final loss, best loss, final accuracy or best accuracy (see Figure B.1).

Figure 3.1 (bottom-left) The setup was identical to Figure 1 (top-right) except that the model considered was a depth 10 residual network with varying width.

Figure 3.1 (bottom-right) Experiments are done using Neural Tangents. All experiments use 100 training samples and two-hidden layer fully-connected networks of varying width (ranging from $w = 64$ to $W = 11,585$) with Relu nonlinearities unless specified as Erf. Full-batch gradient descent and cross-entropy loss were used unless specified as MSE, and the figure shows curves from a random assortment of training times ranging from 100 to 500 steps (equivalently, epochs). Training was done with learning rates small enough so as to avoid catapult dynamics [97] and no $L2$ regularization; in such a setting, the infinite-width learning dynamics is known to be equivalent to that of linearized models [94]. Consequently, for each random initialization of the parameters, the test loss of the finite-width linearized model was additionally computed in the identical training setting. This value approximates the limiting behavior $L(\infty)$ known theoretically and is subtracted off from the final test loss of the (nonlinear) neural network before averaging over 50 random initializations to yield each of the individual data points in the figure.

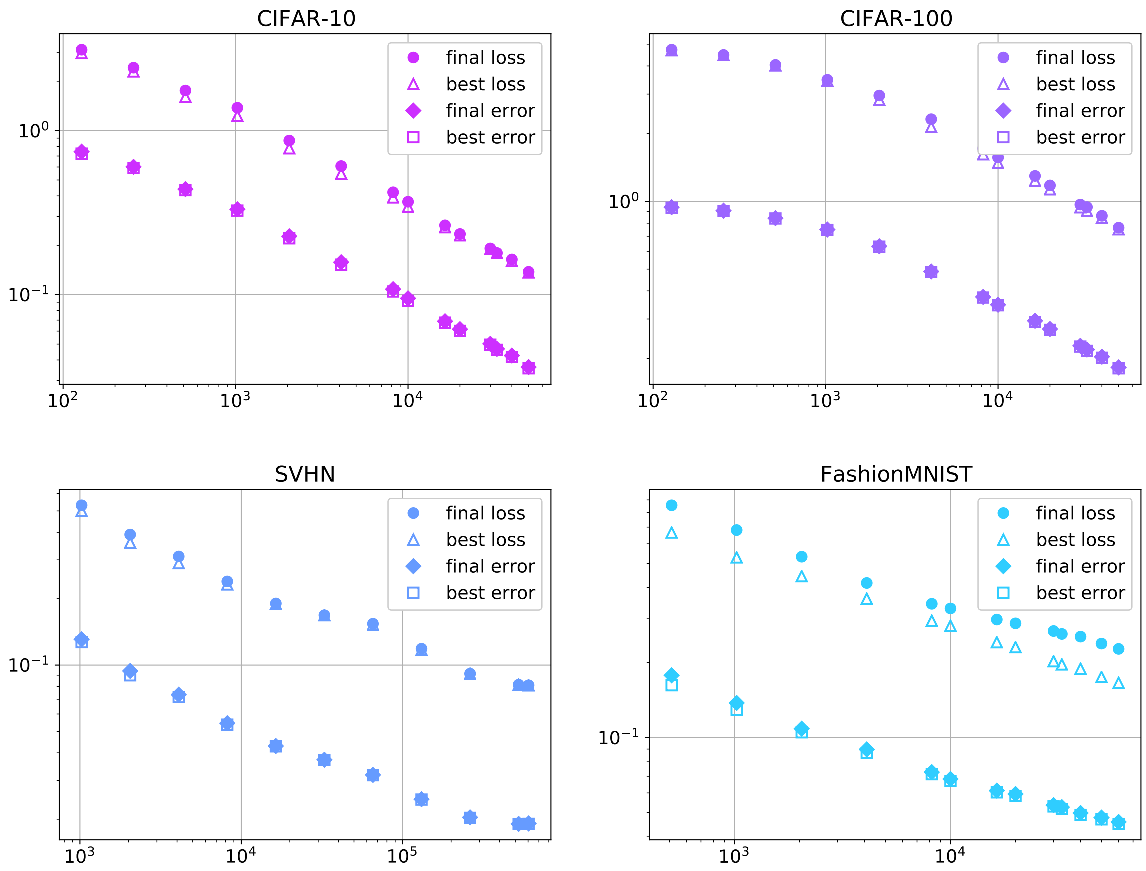

Figure B.1: **Alternate metrics and stopping conditions** We find similar scaling behavior for both the loss and error, and for final and best (early stopped) metrics.

B.1.1 Deep teacher-student models

The teacher-student scaling with dataset size (figure B.2) was performed with fully-connected teacher and student networks with two hidden layers and widths 96 and 192, respectively, using PyTorch [125]. The inputs were random vectors sampled uniformly from a hypercube of dimension $d = 2, 3, \cdots, 9$. To mitigate noise, we ran the experiment on eight different

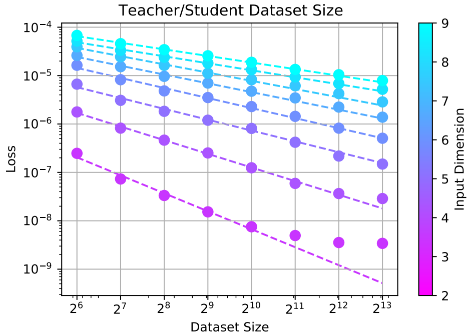

Figure B.2: This figure shows scaling trends of MSE loss with dataset size for teacher/student models. The exponents extracted from these fits and their associated input-space dimensionalities are shown in figure 3.2.

random seeds, fixing the random seed for the teacher and student as we scanned over dataset sizes. We also used a fixed test dataset, and a fixed training set, which was sub-sampled for the experiments with smaller D. The student networks were trained using MSE loss and Adam optimizer with a maximum learning rate of 3×10^{-3}, a cosine learning rate decay, and a batch size of 64, and $40,000$ steps of training. The test losses were measured with early stopping. We combine test losses from different random seeds by averaging the logarithm of the loss from each seed.

In our experiments, we always use inputs that are uniformly sampled from a d-dimensional hypercube, following the setup of [141]. They also utilized several intrisic dimension (ID) estimation methods and found the estimates were close to the input dimension, so we simply use the latter for comparisons. For the dataset size scans we used randomly initialized teachers with width 96, and students with width 192. We found similar results with other network sizes.

The final scaling exponents and input dimensions are show in the bottom of figure 3.2. We used the same experiments for the top of that figure, interpolating the behavior of both teacher and a set of students between two fixed training points. The students only differed by the size of their training sets, but had the same random seeds and were trained in the same way. In that figure the input space dimension was four.

Finally, we also used a similar setup to study variance-limited exponents and scaling. In that case we used much smaller models, with 16-dimensional hidden layers, and a correspondingly larger learning rate. We then studied scaling with D again, with results pictured in figure 3.1.

B.1.2 CNN architecture for resolution-limited scaling

Figure 3.2 includes data from CNN architectures trained on image datasets. The architectures are summarized in Table B.1. We used Adam optimizer for training, with cross-entropy loss. Each network was trained for long enough to achieve either a clear minimum or a plateau in test loss. Specifically, CIFAR10, MNIST and fashion MNIST were trained for 50 epochs, CIFAR100 was trained for 100 epochs and SVHN was trained for 10 epochs. The default keras training parameters were used. In case of SVHN we included the additional images as training data. We averaged (in log space) over 20 runs for CIFAR100 and CIFAR10, 16 runs for MNIST, 12 runs for fashion MNIST, and 5 runs for SVHN. The results of these experiments are shown in figure B.3.

The measurement of input-space dimensionality for these experiemnts was done using the nearest-neighbour algorithm, described in detail in appendix B and C in [141]. We used 2, 3 and 4 nearest neighbors and averaged over the three.

B.1.3 Teacher-student experiment for scaling of loss with model size

We replicated the teacher-student setup in [141] to demonstrate the scaling of loss with model size. The resulting variation of $-4/\alpha_P$ with input-space dimensionality is shown in figure B.4. In our implementation we averaged (in log space) over 15 iterations, with a fixed, randomly generated teacher.

B.2 Effect of aspect ratio on scaling exponents

We trained Wide ResNet architectures of various widths and depths on CIFAR-10 accross dataset sizes. We found that the effect of depth on dataset scaling was mild for the range studied, while the effect of width impacted the scaling behavior up until a saturating width, after which the scaling behavior fixed. See Figure B.5.

B.3 Proof of Theorems 1 and 2

In this section we detail the proof of Theorems 1 and 2. The key observation is to make use of the fact that nearest neighbor distances for D points sampled i.i.d. from a d-dimensional manifold have mean $\mathbb{E}_{D,x}\left[|x-\hat{x}|\right] = \mathcal{O}\left(D^{-1/d}\right)$, where $\hat{x}$ is the nearest neighbor of x and the expectation is the mean over data-points and draws of the dataset see e.g. [95].

The theorem statements are copied for convenience. In the main, in an abuse of notation, we used $L(f)$ to indicate the value of the test loss as a function of the network f, and $L(D)$ to indicate the test loss averaged over the population, draws of the dataset, model

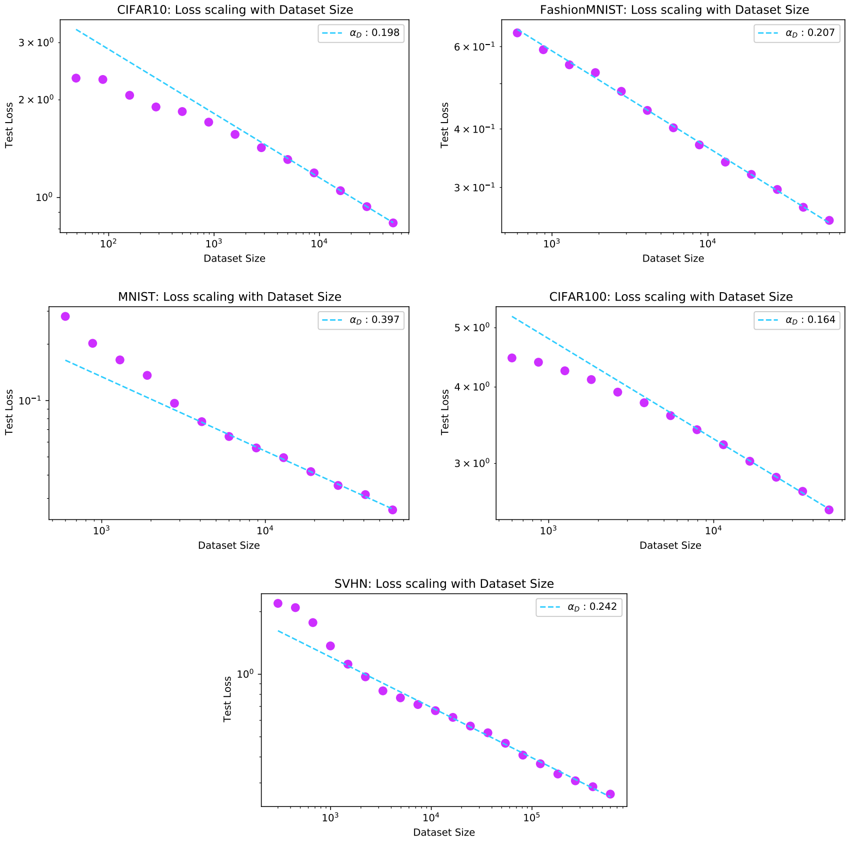

Figure B.3: This figure shows scaling trends of CE loss with dataset size for various image datasets. The exponents extracted from these fits and their associated input-space dimensionalities are shown in figure 3.2.

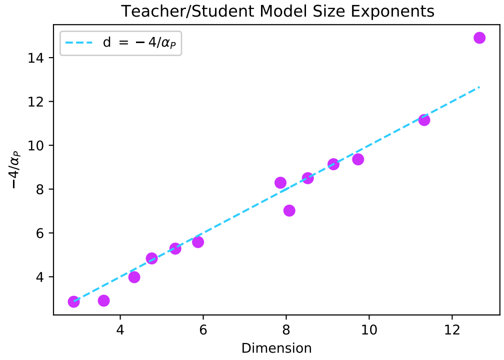

Figure B.4: This figure shows the variation of α_P with the input-space dimension. The exponent α_P is the scaling exponent of loss with model size for Teacher-student setup.

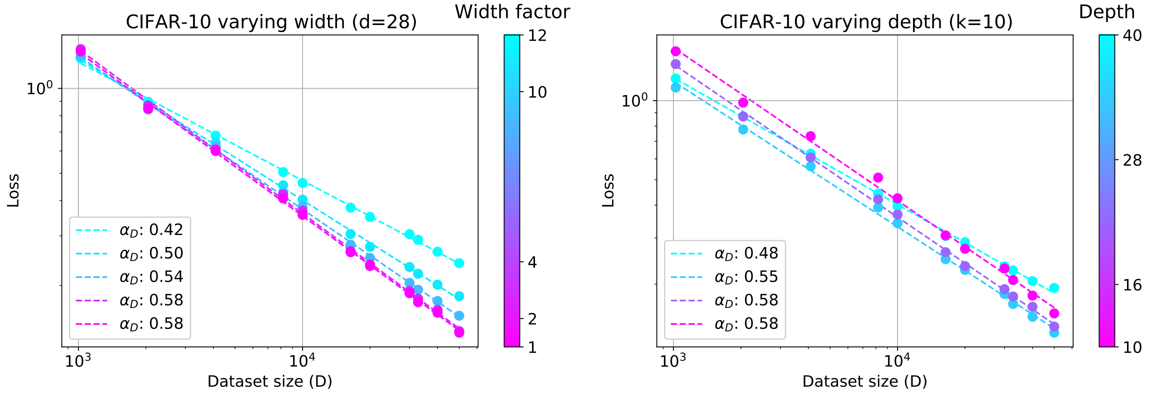

Figure B.5: **Effect of aspect ratio on dataset scaling** We find that for WRN-d-k trained on CIFAR-10, varying depth from 10 to 40 has a relatively mild effect on scaling behavior, while varying the width multiplier, k, from 1 to 12 has a more noticeable effect, up until a saturating width.

initializations and training. To be more explicit below, we will use the notation $\ell(f(x))$ to indicate the test loss for a single network evaluated at single test point.

Theorem 1. *Let $\ell(f)$, f and $\mathcal{F}$ be Lipschitz with constants K_L, K_f, and $K_{\mathcal{F}}$ and $\ell(\mathcal{F}) = 0$. Further let $\mathcal{D}$ be a training dataset of size D sampled i.i.d from $\mathcal{M}_d$ and let $f(x) = \mathcal{F}(x)$, $\forall x \in \mathcal{D}$ then $L(D) = \mathcal{O}\left(K_L max(K_f, K_{\mathcal{F}}) D^{-1/d}\right)$.*

Proof. Consider a network trained on a particular draw of the training data. For each training point, x, let $\hat{x}$ denote the neighboring training data point. Then by the above Lipschitz assumptions and the vanishing of the loss on the true target, we have $\ell(f(x)) \leq K_L \left|f(x) - \mathcal{F}(x)\right| \leq K_L \left(K_f + K_{\mathcal{F}}\right) \left|x - \hat{x}\right|$. With this, the average test loss is bounded as

$$L(D) \leq K_L \left(K_f + K_{\mathcal{F}}\right) \mathbb{E}_{D,x}\left[\left|x - \hat{x}\right|\right] = \mathcal{O}\left(K_L \max(K_f, K_{\mathcal{F}}) D^{-1/d}\right) . \tag{B.3.1}$$

In the last equality, we used the above mentioned scaling of nearest neighbor distances. □

Theorem 2. *Let $\ell(f)$, f and $\mathcal{F}$ be Lipschitz with constants K_L, K_f, and $K_{\mathcal{F}}$. Further let $f(x) = \mathcal{F}(x)$ for P points sampled i.i.d from $\mathcal{M}_d$ then $L(P) = \mathcal{O}\left(K_L max(K_f, K_{\mathcal{F}}) P^{-1/d}\right)$.*

Proof. Denote by $\mathcal{P}$ the P points, z, for which $f(z) = \mathcal{F}(z)$. For each test point x let $\hat{x}$ denote the closest point in $\mathcal{P}$, $\hat{x} = \text{argmin}_{\mathcal{P}}\left(\left|x - z\right|\right)$. Adopting this notation, the result follows by the same argument as Theorem 1. □

B.4 Random feature models

Here we present random feature models in more detail. We begin by reviewing exact expressions for the loss. We then go onto derive its asymptotic properties. We again consider training a model $f(x) = \sum_{\mu=1}^{P} \theta_\mu f_\mu(x)$, where f_μ are drawn from some larger pool of features, $\{F_M\}$, $f_\mu(x) = \sum_{M=1}^{S} \mathcal{P}_{\mu M} F_M(x)$.

Note, if $\{F_M(x)\}$ form a complete set of functions over the data distribution, than any target function, $y(x)$, can be expressed as $y = \sum_{M=1}^{S} \omega_M F_M(x)$. The extra constraint in a teacher-student model is specifying the distribution of the ω_M. The variance-limited scaling goes through with or without the teacher-student assumption, however it is crucial for analysing the variance-limited behavior.

As in Section 3.2.3 we consider models with weights initialized to zero and trained to convergence with mean squared error loss.

$$L_{\text{train}} = \frac{1}{2D} \sum_{a=1}^{D} \left(f(x_a) - y_a\right)^2 \,. \tag{B.4.1}$$

The data and feature second moments play a central role in our analysis. We introduce the notation,

$$\begin{gathered}
\mathcal{C} = \mathbb{E}_x\left[F(x)F^T(x)\right] \,, \quad \bar{\mathcal{C}} = \frac{1}{D}\sum_{a=1}^{D} F(x_a)F^T(x_a) \,, \quad C = \mathcal{P}\mathcal{C}\mathcal{P}^T \,, \quad \bar{C} = \mathcal{P}\bar{\mathcal{C}}\mathcal{P}^T \,. \\
\mathcal{K}(x,x') = \frac{1}{S}F^T(x)F(x') \,, \quad \bar{\mathcal{K}} = \mathcal{K}\Big|_{\mathcal{D}_{\text{train}}} \,, \quad K(x,x') = \frac{1}{P}f^T(x)f(x') \,, \quad \bar{K} = K\Big|_{\mathcal{D}_{\text{train}}} \,.
\end{gathered} \tag{B.4.2}$$

Here the script notation indicates the full feature space while the block letters are restricted to the student features. The bar represents restriction to the training dataset. We will also indicate kernels with one index in the training set as $\vec{\mathcal{K}}(x) := \mathcal{K}(x, x_{a=1\ldots D})$ and $\vec{K}(x) := K(x, x_{a=1\ldots D})$. After this notation spree, the test loss can be written for under-parameterized models, $P \leq D$ as

$$L(D,P) = \frac{1}{2S}\mathbb{E}_D\left[\operatorname{Tr}\left(\mathcal{C} + \bar{\mathcal{C}}\mathcal{P}^T\bar{C}^{-1}C\bar{C}^{-1}\mathcal{P}\bar{\mathcal{C}} - 2\bar{\mathcal{C}}\mathcal{P}^T\bar{C}^{-1}\mathcal{P}\mathcal{C}\right)\right] \,. \tag{B.4.3}$$

and for over-parameterized models (at the unique minimum found by GD, SGD, or projected Newton's method),

$$L(D,P) = \frac{1}{2}\mathbb{E}_{x,D}\left[\mathcal{K}(x,x) + \vec{K}(x)^T\bar{K}^{-1}\bar{\mathcal{K}}\bar{K}^{-1}\vec{K}(x) - 2\vec{K}(x)^T\bar{K}^{-1}\vec{\mathcal{K}}(x)\right] \,. \tag{B.4.4}$$

Here the expectation $\mathbb{E}_D\left[\bullet\right]$ is an expectation with respect to iid draws of a dataset of size D from the input distribution, while $\mathbb{E}_x\left[\bullet\right]$ is an ordinary expectation over the input distribution. Note, expression (B.4.3) is also valid for over-parameterized models and (B.4.4) is valid for under-parameterized models if the inverses are replaces with the Moore-Penrose pseudo-inverse. Also note, the two expressions can be related by echanging the projections onto finite features with the projection onto the training dataset and the sums of teacher features with the expectation over the data manifold. This realizes the duality between dataset and features discussed above.

B.4.1 Asymptotic expressions

We are interested in (B.4.3) and (B.4.4) in the limits of large P and D.

Variance-limited scaling We begin with the under-parameterized case. In the limit of lots of data the sample estimate of the feature feature second moment matrix, $\bar{\mathcal{C}}$, approaches the true second moment matrix, $\mathcal{C}$. Explicitly, if we define the difference, $\delta\mathcal{C}$ by $\bar{\mathcal{C}} = \mathcal{C} + \delta\mathcal{C}$. We have

$$\begin{aligned} \mathbb{E}_D\left[\delta\mathcal{C}\right] &= 0 \\ \mathbb{E}_D\left[\delta\mathcal{C}_{M_1N_1}\delta\mathcal{C}_{M_2N_2}\right] &= \frac{1}{D}\left(\mathbb{E}_x\left[F_{M_1}(x)F_{N_1}(x)F_{M_2}(x)F_{N_2}(x)\right] - \mathcal{C}_{M_1N_1}\mathcal{C}_{M_2N_2}\right) \\ \mathbb{E}_D\left[\delta\mathcal{C}_{M_1N_1}\cdots\delta\mathcal{C}_{M_nN_n}\right] &= \mathcal{O}\left(D^{-2}\right) \quad \forall n > 2\,. \end{aligned} \tag{B.4.5}$$

The key takeaway from (B.4.5) is that the dependence on D is manifest.

Using these expressions in (B.4.3) yields.

$$\begin{aligned} L(D,P) = &\frac{1}{2S}\operatorname{Tr}\left(\mathcal{C} - \mathcal{C}\mathcal{P}^T C^{-1}\mathcal{P}\mathcal{C}\right) \\ &+ \frac{1}{2DS}\sum_{M_{1,2}N_{1,2}=1}^{P} T_{M_1N_1M_2N_2}\Big[\delta_{M_1M_2}\left(\mathcal{P}^T C^{-1}\mathcal{P}\right)_{N_1N_2} + (C^{-1}\mathcal{P}\mathcal{C}^2\mathcal{P}^T C^{-1})_{M_1M_2}C^{-1}_{N_1N_2} \\ &\qquad -2\left(\mathcal{C}\mathcal{P}^T C^{-1}\mathcal{P}\right)_{M_1M_2}\left(\mathcal{P}^T C^{-1}\mathcal{P}\right)_{N_1N_2}\Big] + \mathcal{O}\left(D^{-2}\right)\,. \end{aligned} \tag{B.4.6}$$

Here we have introduced the notation, $T_{M_1N_1M_2N_2} = \mathbb{E}_x\left[F_{M_1}(x)F_{N_1}(x)F_{M_2}(x)F_{N_2}(x)\right]$.

As above, defining

$$L(P) := \lim_{D\to\infty} L(D,P) = \frac{1}{2S}\operatorname{Tr}\left(\mathcal{C} - \mathcal{C}\mathcal{P}^T C^{-1}\mathcal{P}\mathcal{C}\right)\,. \tag{B.4.7}$$

we see that though $L(D,P) - L(P)$ is a somewhat cumbersome quantity to compute, involving the average of a quartic tensor over the data distribution, its dependence on D is simple.

For the over-parameterized case, we can similarly expand (B.4.4) using $K = \mathcal{K} + \delta\mathcal{K}$. With fluctuations satisfying,

$$\begin{aligned} \mathbb{E}_P\left[\delta\mathcal{K}\right] &= 0 \\ \mathbb{E}_P\left[\delta\mathcal{K}_{a_1b_1}\delta\mathcal{K}_{a_2b_2}\right] &= \frac{1}{P}\left(\mathbb{E}_P\left[f_\mu(x_{a_1})f_\mu(x_{b_1})f_\mu(x_{a_2})f_\mu(x_{b_2})\right] - \mathcal{K}_{a_1b_1}\mathcal{K}_{a_2b_2}\right) \\ \mathbb{E}_P\left[\delta\mathcal{K}_{a_1a_1}\cdots\delta\mathcal{K}_{a_na_n}\right] &= \mathcal{O}\left(P^{-2}\right) \quad \forall n > 2\,. \end{aligned} \tag{B.4.8}$$

This gives the expansion

$$L(D,P) = \frac{1}{2}\mathbb{E}_{x,D}\left[\mathcal{K}(x,x) - \vec{\mathcal{K}}(x)^T\bar{\mathcal{K}}^{-1}\vec{\mathcal{K}}(x)\right] + \mathcal{O}(P^{-1})\,, \tag{B.4.9}$$

and

$$L(D) = \frac{1}{2}\mathbb{E}_{x,D}\left[\mathcal{K}(x,x) - \vec{\mathcal{K}}(x)^T \bar{\mathcal{K}}^{-1}\vec{\mathcal{K}}(x)\right] . \tag{B.4.10}$$

Resolution-limited scaling We now move onto studying the parameter scaling of $L(P)$ and dataset scaling of $L(D)$. We explicitly analyse the dataset scaling of $L(D)$, with the parameter scaling following via the dataset parameter duality.

Much work has been devoted to evaluating the expression, (B.4.10) [158, 104, 143]. One approach is to use the *replica trick* – a tool originating in the study of disordered systems which computes the expectation of a logarithm of a random variable via simpler moment contributions and analyticity assumption [124]. The replica trick has a long history as a technique to study the generalization properties of kernel methods [142, 106, 105, 151, 32, 54, 17]. We will most closely follow the work of [23] who use the replica method to derive an expression for the test loss of linear feature models in terms of the eigenvalues of the kernel $\mathcal{C}$ and $\bar{\omega}$, the coefficient vector of the target labels in terms of the model features.

$$\begin{aligned} L(D) &= \frac{\kappa^2}{1-\gamma}\sum_i \frac{\lambda_i \bar{\omega}_i^2}{\left(\kappa + D\lambda_i\right)^2} , \\ \kappa &= \sum_i \frac{\kappa \lambda_i}{\kappa + D\lambda_i} , \quad \gamma = \sum_i \frac{D\lambda_i^2}{\left(\kappa + D\lambda_i\right)^2} . \end{aligned} \tag{B.4.11}$$

This is the ridge-less, noise-free limit of equation (4) of [23]. Here we analyze the asymptotic behavior of these expressions for eigenvalues satisfying a power-law decay, $\lambda_i = i^{-(1+\alpha_K)}$ and for targets coming from a teacher-student setup, $w \sim \mathcal{N}(0, 1/S)$.

To begin, we note that for teacher-student models in the limit of many features, the overlap coefficients $\bar{\omega}$ are equal to the teacher weights, up to a rotation $\bar{\omega}_i = O_{iM} w_M$. As we are choosing an isotropic Gaussian initialization, we are insensitive to this rotation and, in particular, $\mathbb{E}_w\left[\bar{\omega}_i^2\right] = 1/S$. See Figure B.8 for empirical support of the average constancy of $\bar{\omega}_i$ for the teacher-student setting and contrast with realistic labels.

With this simplification, we now compute the asymptotic scaling of (B.4.11) by approximating the sums with integrals and expanding the resulting expressions in large D. We use the identities:

$$\begin{aligned} \int_1^\infty dx \frac{x^{-n(1+\alpha)}}{\left(\kappa + Dx^{-(1+\alpha)}\right)^m} &= \kappa^{-m}\frac{\Gamma\left(n - \frac{1}{1+\alpha}\right)}{(1+\alpha)\Gamma\left(n + \frac{\alpha}{1+\alpha}\right)} {}_2F_1\left(m, n - \frac{1}{1+\alpha}, n + \frac{\alpha}{1+\alpha}, \frac{-D}{\kappa}\right) \\ {}_2F_1\left(a, b, c, -y\right) &\propto y^{-a} + \mathcal{B}y^{-b} + \dots , \end{aligned} \tag{B.4.12}$$

Here ${}_2F_1$ is the hypergeometric function and the second line gives its asymptotic form at large y. $\mathcal{B}$ is a constant which does not effect the asymptotic scaling.

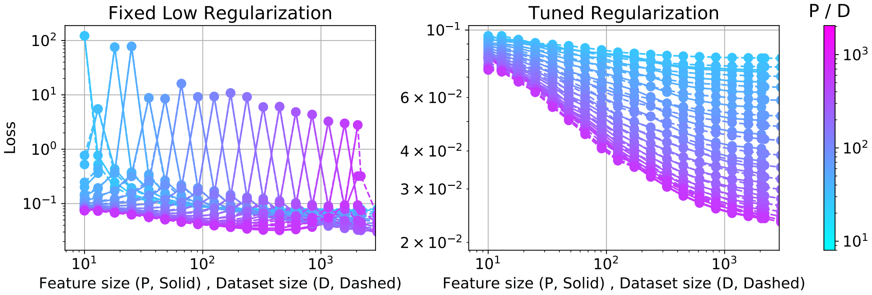

Figure B.6: **Duality between dataset size vs feature number in pretrained features** Using pretrained embedding features of EfficientNet-B5 [148] for different levels of regularization, we see that loss as function of dataset size or loss as a function of the feature dimension track each other both for small regularization (**left**) and for tuned regularization (**right**). Note that regularization strength with trained-feature kernels can be mapped to inverse training time [6, 93]. Thus (**left**) corresponds to long training time and exhibits double descent behavior, while (**right**) corresponds to optimal early stopping.

Using these relations yields

$$\kappa \propto D^{-\alpha_K}, \quad \gamma \propto D^0, \quad \text{and} \quad L(D) \propto D^{-\alpha_K}\,, \tag{B.4.13}$$

as promised. Here we have dropped sub-leading terms at large D. Scaling behavior for parameter scaling $L(P)$ follow via the dataset parameter duality.

B.4.2 Duality beyond asymptotics

Expressions (B.4.3) and (B.4.4) are related by changing projections onto finite feature set, and finite dataset even without taking any asymptotic limits. We thus expect the dependence of test loss on parameter count and dataset size to be related quite generally in linear feature models. See Section B.5 for further details.

B.5 Learned Features

In this section, we consider linear models with features coming from pretrained neural networks. Such features are useful for transfer learning applications (e.g. [86, 85]). In Figures B.6 and B.7, we take pretrained embedding features from an EfficientNet-B5 model [148] using TF hub[1]. The EfficientNet model is pretrained using the ImageNet dataset with input image size of $(456, 456)$. To extract features for the $(32, 32)$ CIFAR-10 images, we use *bilinear* resizing. We then train a linear classifier on top of the penultimate pretrained features.

[1] https://www.tensorflow.org/hub

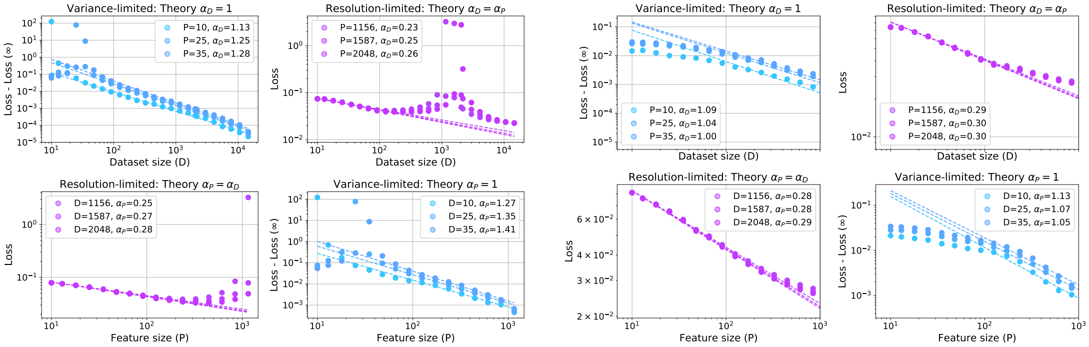

Figure B.7: **Four scaling regimes exhibited by pretrained embedding features** Using pretrained embedding features of EfficientNet-B5 [148] for fixed low regularization (**left**) and tuned regularization (**right**), we can identify four regimes of scaling using real CIFAR-10 labels.

To explore the effect feature size, P, and dataset size D, we randomly subset the feature dimension and training dataset size and average over 5 random seeds. Prediction on test points are obtained as a kernel ridge regression problem with linear kernel. We note that the regularization ridge parameter can be mapped to an inverse early-stopping time [6, 93] of a corresponding ridgeless model trained via gradient descent. Inference with low regularization parameter denotes training for long time while tuned regularization parameter is equivalent to optimal early stopping.

In Figure B.7 we see evidence of all four scaling regimes for low regularization (left four) and optimal regularization (right four). We speculate that the deviation from the predicted variance-limited exponent $\alpha_P = \alpha_D = 1$ for the case of fixed low regularization (late time) is possibly due to the double descent resonance at $D = P$ which interferes with the power law fit.

In Figure B.6, we observe the duality between dataset size D (solid) and feature size P (dashed) – the loss as a function of the number of features is identical to the loss as function of dataset size for both the optimal loss (tuned regularization) or late time loss (low regularization).

In Figure B.8, we also compare properties of random features (using the infinite-width limit) and learned features from trained WRN 28-10 models. We note that teacher-student models, where the feature class matches the target function and ordinary, fully trained models on real data (Figure 3.1), have significantly larger exponents than models with fixed features and realistic targets.

The measured $\bar{\omega}_i$ – the coefficient of the task labels under the i-th feature (B.4.11) are approximately constant as function of index i for all teacher-student settings. However for real targets, $\bar{\omega}_i$ are only constant for the well-performing Myrtle-10 and WRN trained features (last two columns).

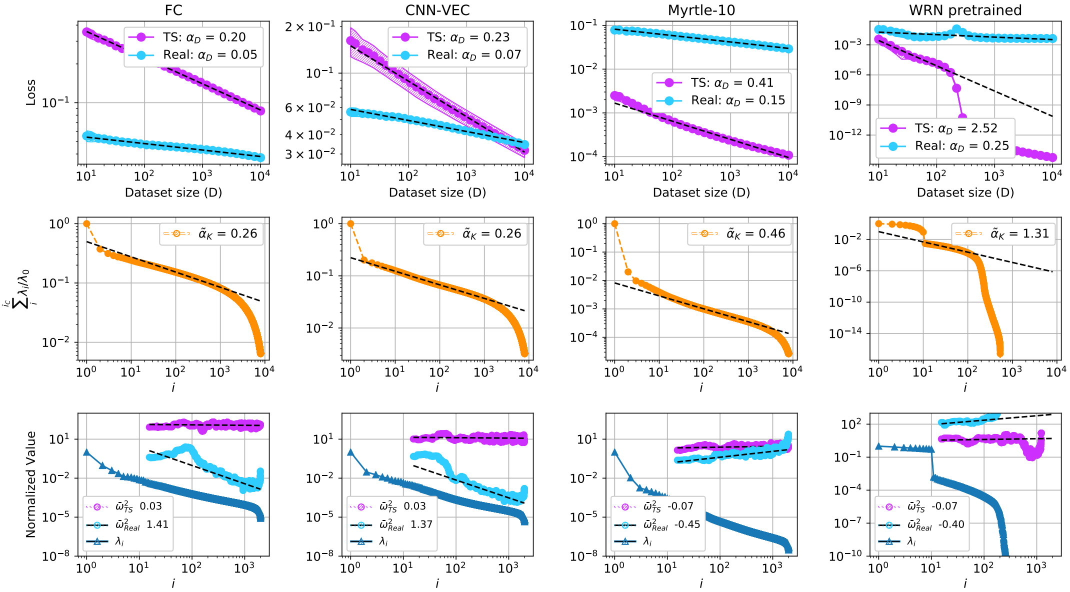

Figure B.8: **Loss on the teacher targets scale better than real targets for both untrained and trained features** The first three columns are infinite width kernels while the last column is a kernel built out of features from the penultimate layer of pretrained WRN 28-10 models on CIFAR-10. The first row is the loss as a function of dataset size D for teacher-student targets vs real targets. The observed dataset scaling exponent is denoted in the legend. The second row is the normalized partial sum of kernel eigenvalues. The partial sum's scaling exponent is measured to capture the effect of the finite dataset size when empirical α_K is close to zero. The third row shows $\bar{\omega}_i$ for teacher-student and real target compared against the kernel eigenvalue decay. We see the teacher-student $\bar{\omega}_i$ are approximately constant.

Layer	**Width**
CNN window (3, 3)	50
2D Max Pooling (2, 2)	
CNN window (3, 3)	100
2D Max Pooling (2, 2)	
CNN window (3, 3)	100
Dense	64
Dense	10

Layer	**Width**
CNN window (3, 3)	50
2D Max Pooling (2, 2)	
CNN window (3,3)	100
2D Max Pooling (2, 2)	
CNN window (3, 3)	200
Dense	256
Dense	100

Layer	**Width**
CNN window (3, 3)	64
2D Max Pooling (2, 2)	
CNN window (3, 3)	64
2D Max Pooling (2, 2)	
Dense	128
Dense	10

Table B.1:: CNN architectures for CIFAR10, MNIST, Fashion MNIST (left), CIFAR100 (center) and SVHN (right)

Appendix C

Appendix to Chapter 4

C.1 Bulk Primary Conditions as Mirage Translations

Bulk operators must leave an imprint in boundary correlators representing their conserved charges and energies. This imprint manifests as singularities in correlators involving conserved currents $J(z)$ (see section 4.2.3) or the stress tensor $T(z)$. In this section we develop some formalism for displacing the singularities associated with local charge or energy in a CFT$_2$. This will allow us to alter the 'gravitational dressing' of bulk operators. We will also identify an alternative explanation for the bulk primary condition [8]. In appendix C.1.2 we provide a review of the singularity structure of $T(z)$ correlators with CFT primary operators as derived from the bulk.

C.1.1 Mirage Translations

In a translation-invariant theory such as a CFT, we use the momentum generator P_μ to move local operators around. In a CFT$_2$ this means that

$$\mathcal{O}(z,\bar{z}) = e^{zL_{-1}+\bar{z}\bar{L}_{-1}}\mathcal{O}(0)e^{-zL_{-1}-\bar{z}\bar{L}_{-1}} \tag{C.1.1}$$

since $L_{-1}, \bar{L}_{-1}$ are the holomorphic and anti-holomorphic momentum generators.

Now let us assume that the CFT has a holomorphic $U(1)$ current $J(z)$. Correlators with the current such as

$$\langle J(z_1)\mathcal{O}^\dagger(z,\bar{z})\mathcal{O}(0)\rangle = q\frac{z}{(z-z_1)z_1}\frac{1}{z^{2h}\bar{z}^{2\bar{h}}} \tag{C.1.2}$$

have singularities in z_1 when J collides with charged operators, which indicate the presence of charge localized at 0 and z. We will pose the following question: *can we find an operator that moves local charge without moving the associated primary operators?* Or equivalently, can we move the primary operators while leaving its charge in place?

We can sharpen these questions into precise criteria for correlators. We would like to find a modified translation operator $G_{h,q}(z_f)$ that can appear in correlators as[1]

$$\langle \mathcal{O}^\dagger(z) J(z_k) \cdots J(z_1) \left[G_{h,q}(z_f) \mathcal{O}(0) \right] \rangle \tag{C.1.3}$$

We wish to choose $G_{h,q}$ so that $\mathcal{O}^\dagger(z, \bar{z})$ only has an OPE singularity with $[G_{h,q}(z_f)\mathcal{O}(0)]$ when $z - z_f$ vanish, but the currents $J(z_j)$ have OPE singularities with $[G_{h,q}(z_f)\mathcal{O}(0)]$ when $z_j \to 0$. So the non-local object $[G_{h,q}\mathcal{O}]$ behaves like a mirage, present at both 0 and z_f.

Conventional translation operators automatically satisfy the first condition. They also satisfy the second condition when the charge of $\mathcal{O}$ is $q = 0$, suggesting that $G_{h,0}(z_f, 0) = e^{z_f L_{-1}}$. So let us modify the translation generator and define

$$G_{h,q}(z_f) = \sum_{n=0}^{\infty} \frac{z_f^n}{n!} \mathcal{J}_{-n} \tag{C.1.4}$$

where we have $\mathcal{J}_{-n} = L_{-1}^n + O(q)$, so that $\mathcal{J}_{-n}$ implicitly depends on h and q. Our criterion require an OPE

$$J(z_1) \left[G_{h,q}(z_f) \mathcal{O}(0) \right] = \frac{q}{z_1} \left[G_{h,q}(z_f, 0) \mathcal{O}(0) \right] + \cdots \tag{C.1.5}$$

where the ellipsis denotes finite terms, so that we are demanding that the only singularity is a simple pole at $z_1 = 0$. This condition will automatically be satisfied if

$$[J_m, \mathcal{J}_{-n} \mathcal{O}(0)] = 0 \quad \text{for all} \quad m \geq 1 \tag{C.1.6}$$

and for any n. These conditions uniquely determine $\mathcal{J}_{-n}$ up to an overall factor. These overall factors will be fixed as in equation (C.1.4) by the requirement that $G_{h,q}(z_f)$ acts on $\mathcal{O}$ as a conventional translation in its two-point function with $\mathcal{O}^\dagger$.

We can repeat this exercise and replace charge with energy-momentum, and $J(z)$ with the CFT$_2$ energy-momentum tensor $T(z)$. In that case, we could write

$$G_h(z_f) = \sum_{n=0}^{\infty} \frac{z_f^n}{n!} \mathcal{L}_{-n} \tag{C.1.7}$$

where $\mathcal{L}_{-n}$ implicitly depends on the dimension h of the primary $\mathcal{O}$ to which we are applying G_h.

We must have $\mathcal{L}_{-1} = L_{-1}$ simply because there are no other level-one combinations of Virasoro generators. This means that the OPE $T(z_1) [G_h(z_f)\mathcal{O}(0)]$ necessarily contains a third-order pole $\frac{1}{z_1^3}$, but it needn't have any higher order singularities. The absence of any further singularities at $z_1 \to 0$ or anywhere else in the complex plane implies that the $\mathcal{L}_{-n}$ must satisfy the bulk primary conditions

$$[L_m, \mathcal{L}_{-n} \mathcal{O}(0)] = 0 \quad \text{for all} \quad m \geq 2, \tag{C.1.8}$$

[1] We are implicitly assuming we can separate z_i and w_j from 0 and z_f and perform radial quantization about the non-local object $[G_{h,q}\mathcal{O}]$.

which were previously discovered in the context of bulk reconstruction. Here we see them appearing in the answer to a question concerning the CFT alone.

C.1.2 Singularity Structure of $\langle T(z)\rangle$ from Einstein's Equations

In this section, we generalize the discussion of section 4.2.3 to gravity, showing how Einstein's equations in the presence of a massive source on the boundary dictate the singularity structure of correlators with the CFT stress tensor. This is elementary, as in essence it amounts to Gauss's law for AdS$_3$ gravity [11]. But we review the argument to emphasize the connection between gravitational dressing and singularities.

We wish to establish that the OPE of a scalar primary with the CFT stress tensor has $\frac{1}{z^2}$ singularity by using the bulk equations of motion. A scalar primary inserted at the origin will create a scalar particle propagating in the bulk. We assume that the particle is sufficiently heavy to model its wavefunction with a worldline.

In global AdS$_3$ with metric $ds^2 = \left(r^2+1\right) dt_E^2 + \frac{1}{r^2+1} dr^2 + r^2 d\theta^2$, the only non-vanishing component of the bulk energy-momentum tensor of this particle is

$$T_{\rm B}^{tt} = \frac{m}{2\pi r}\delta(r), \tag{C.1.9}$$

where we denote the bulk energy-momentum tensor with a subscript "B" to avoid confusion ($T, \bar{T}$ will still be the boundary bulk energy-momentum tensor). We are interested in describing the above particle in Poincare patch $ds^2 = \frac{dy^2 + dz d\bar{z}}{y^2}$. The coordinate maps that connect these two metrics are

$$y = \frac{e^{t_E}}{\sqrt{r^2+1}}, \quad z = \frac{r e^{t_E + i\theta}}{\sqrt{r^2+1}}, \quad \bar{z} = \frac{r e^{t_E - i\theta}}{\sqrt{r^2+1}}. \tag{C.1.10}$$

The trajectory of the particle, which is simply $r = 0$ in the global coordinates, is corresponding to $(y, z, \bar{z}) = (e^{t_E}, 0, 0)$. Since we want to study the singularity of the boundary stress tensor $T(z)$ as it approaches the source, we need to localize to a small neighborhood around $r = 0$, that is, we'll take the limit $z, \bar{z} \to 0$ in the following calculations. The full backreacted metric will take the form of equation (4.3.2), where here we will interpret T and $\bar{T}$ as components of a classical gravitational field. The delta function in the bulk energy-momentum tensor (C.1.9) can be transformed to the Poincare patch via

$$\frac{1}{2\pi r}\delta(r) \to \frac{1}{y\sqrt{|g|}}\delta^2(z, \bar{z}) \tag{C.1.11}$$

where the Jacobian accounts for $y = e^t$ at $r = 0$. Thus, the covariant bulk energy-momentum tensor in Poincare patch will be given by

$$T_{\rm B}^{\mu\nu} = m\frac{v^\mu v^\nu}{v^\alpha v^\beta g_{\alpha\beta}}\frac{1}{y\sqrt{|g|}}\delta^2(z, \bar{z}) \tag{C.1.12}$$

where the velocity of the particle following a geodesic is $v^\mu = (\dot{y}, \dot{z}, \dot{\bar{z}}) = (y, z, \bar{z})$ with constant r and θ. We'll assume $T(z)$ to be more singular than $\frac{1}{z}$ and similarly for $\bar{T}(\bar{z})$. In the small $z, \bar{z}$ limit, we have $\sqrt{|g|} \approx \frac{18yT\bar{T}}{c^2}$ and $v^\alpha v^\beta g_{\alpha\beta} \approx \frac{36y^2 z\bar{z}T\bar{T}}{c^2}$. The resulting simplified form of the stress tensor is

$$T_{\mathrm{B}}^{\mu\nu} \approx \frac{mc^4 v^\mu v^\nu}{648y^4 z\bar{z}T^2\bar{T}^2}\delta^2(z,\bar{z}) \tag{C.1.13}$$

We apply the same limit to the LHS of Einstein equation to find the $y\bar{z}$ component to be

$$G^{y\bar{z}} - g^{y\bar{z}} \approx \frac{c^3 \partial_{\bar{z}} T}{54y^3 T^2 \bar{T}^2}. \tag{C.1.14}$$

So the $y\bar{z}$ component of the Einstein's equation $G^{\mu\nu} - g^{\mu\nu} = 8\pi G_N T_{\mathrm{B}}^{\mu\nu}$ is

$$\partial_{\bar{z}} T = \frac{\pi m}{z}\delta^2(z,\bar{z}), \tag{C.1.15}$$

where we've used $G_N = \frac{3}{2c}$. So we find

$$T(z) = \frac{m}{2z^2}, \tag{C.1.16}$$

where we've used $\partial_{\bar{z}}\frac{1}{z} = \pi\delta^2(z,\bar{z})$. Similarly, from the yz component of the Einstein's equation, we can get $\bar{T}(\bar{z}) = \frac{m}{2\bar{z}^2}$. Other components of the Einstein's equation are trivially satisfied once we substitute these solutions for $T(z)$ and $\bar{T}(\bar{z})$.

So in the large mass approximation $m \approx 2h$ we can conclude that $T = \frac{h}{z^2}$ in the presence of a source localized at the origin. Bulk fields must be leave a similar imprint on the boundary T correlators.

C.2 Solving the Charged Bulk Primary Conditions

In this appendix we solve the charged bulk primary conditions for the operators $\mathcal{J}_{-n}$, first exactly for the first few n in appendix C.2.1, and then in appendix C.2.2, we study the large k limit and obtain all the terms at order $1/k$ in $\mathcal{J}_{-n}$ for all n.

C.2.1 Exact Solutions

We'll expand the bulk charged field as

$$\phi(y,z,\bar{z}) = \sum_{n=0}^{\infty} y^{2h+2n}\lambda_n \mathcal{J}_{-n}\bar{L}_{-1}^n \mathcal{O}(z,\bar{z}), \qquad \text{with } \lambda_n \equiv \frac{(-1)^n}{n!\,(2h)_n} \tag{C.2.1}$$

where we've factored out λ_n in ϕ for later convenience. Now our task is to solve for $\mathcal{J}_{-n}$s, which satisfies the following two conditions

$$J_m \mathcal{J}_{-n}\mathcal{O} = 0, \quad \text{for } m \geq 1,$$

$$L_1^n \mathcal{J}_{-n} \mathcal{O} = n!\, (2h)_n\, \mathcal{O}, \tag{C.2.2}$$

where the first one is simply the bulk-primary condition (4.2.16), and the second one just is to ensure that ϕ has the correct bulk-boundary propagator with $\mathcal{O}(w,\bar{w})$, i.e. $\langle \phi(y,z,\bar{z})\mathcal{O}(w,\bar{w})\rangle = \left(\frac{y}{y^2+(z-w)(\bar{z}-\bar{w})}\right)^{2h}$. One can also understand the second one as giving a normalization condition for $\mathcal{J}_{-n}$. It can be shown that the above two conditions uniquely fix $\mathcal{J}_{-n}$.

At each level n, we simply write $\mathcal{J}_{-n}\mathcal{O}$ as a sum over all possible level n descendant operators with unknown coefficients, and use the above equations to fix the coefficients. There will be equal number of unknown coefficients and independent equations at each level n. The solutions for n up to 4 are given by

$$\begin{aligned}
\mathcal{J}_{-1} &= \frac{1}{1-\frac{q^2}{2hk}}\left(L_{-1} - \frac{q}{k}J_{-1}\right), \\
\mathcal{J}_{-2} &= \frac{1}{1-\frac{4h+1}{2h(2h+1)}\frac{q^2}{k}+\frac{1}{2h(2h+1)}\frac{q^4}{k^2}}\left(L_{-1}^2 - \frac{q}{k}\left(J_{-2}+2J_{-1}L_{-1}\right)+\frac{q^2}{k^2}J_{-1}^2\right), \\
\mathcal{J}_{3} &= \frac{1}{1-\frac{6h^2+6h+1}{2h(h+1)(2h+1)}\frac{q^2}{k}+\frac{3}{4h(h+1)}\frac{q^4}{k^2}-\frac{1}{4h(h+1)(2h+1)}\frac{q^6}{k^3}} \\
&\quad \times \left(L_{-1}^3 - \frac{q}{k}\left(2J_{-3}+3J_{-2}L_{-1}+3J_{-1}L_{-1}^2\right)+\frac{3q^2}{k^2}\left(J_{-2}J_{-1}+J_{-1}J_{-1}L_{-1}\right)-\frac{q^3}{k^3}J_{-1}^3\right),
\end{aligned} \tag{C.2.3}$$

and

$$\begin{aligned}
\mathcal{J}_{-4} &= \frac{1}{1-\frac{(16h^3+36h^2+22h+3)q^2}{2h(2h+3)(2h^2+3h+1)k}+\frac{(24h^2+36h+11)q^4}{4h(2h+3)(2h^2+3h+1)k^2}-\frac{(4h+3)q^6}{2h(2h+3)(2h^2+3h+1)k^3}+\frac{q^8}{4h(2h+3)(2h^2+3h+1)k^4}} \\
&\quad \times \Big[L_{-1}^4 - \frac{q}{k}\left(6J_{-4}+8J_{-3}L_{-1}+6J_{-2}L_{-1}^2+4J_{-1}L_{-1}^3\right) \\
&\qquad + \frac{q^2}{k^2}\left(8J_{-3}J_{-1}+3J_{-2}J_{-2}+12J_{-2}J_{-1}L_{-1}+6J_{-1}^2L_{-1}^2\right) \\
&\qquad - \frac{6q^3}{k^3}\left(6J_{-2}J_{-1}^2+4J_{-1}^3L_{-1}\right)+\frac{q^4}{k^4}J_{-1}^4\Big].
\end{aligned}$$

In principle, one can continue this calculation up to arbitrarily large n. However, as n increase, the number of descendant operators at level n will increase very fast and the calculation becomes very complicated.

One useful way of organizing the terms in $\mathcal{J}_{-n}\mathcal{O}$ is to separate the contribution of the global descendants of $\mathcal{O}$ from that of quasi-primary operators and their global descendants, i.e.

$$\mathcal{J}_{-n}\mathcal{O} = L_{-1}^n\mathcal{O} + \text{quasi-primaries plus their global descendants.} \tag{C.2.4}$$

For example, we can rewrite $\mathcal{J}_{-1}\mathcal{O}$ as

$$\mathcal{J}_{-1}\mathcal{O} = L_{-1}\mathcal{O} + \frac{q^2}{2hk-q^2}\left(L_{-1} - \frac{2h}{q}J_{-1}\right)\mathcal{O} \tag{C.2.5}$$

where $\left(L_{-1}-\frac{2h}{q}J_{-1}\right)\mathcal{O}$ is a quasi-primary operator since it satisfies $L_1\left(L_{-1}-\frac{2h}{q}J_{-1}\right)\mathcal{O}=0$.

Actually, we can be more precise about the statement (C.2.4). Similar to the case of gravity considered in section 3.2.2 of [8], one can show that $\mathcal{J}_{-n}\mathcal{O}$ can be written as

$$\mathcal{J}_{-n}\mathcal{O}=L_{-1}^{n}\mathcal{O}+n!\,(2h)_n\sum_{j=0}^{n}\sum_{i}\frac{L_{-1}^{n-j}\mathcal{O}_{h+j}^{(i)}}{\left|L_{-1}^{n-j}\mathcal{O}_{h+j}^{(i)}\right|^2}\tag{C.2.6}$$

where the $\mathcal{O}_{h+j}^{(i)}$ represent the ith quasi-primary at level j (so they satisfy $L_1\mathcal{O}_{h+j}^{(i)}=0$) and we've assumed that the quasi-primaries are orthogonal to each other. The denominator in each term is the norm of the corresponding operator. The simplest example of the above expression is given by $\mathcal{J}_{-1}\mathcal{O}$ in (C.2.5), which can be written as

$$\mathcal{J}_{-1}\mathcal{O}=L_{-1}+2h\frac{\left(L_{-1}-\frac{2h}{q}J_{-1}\right)\mathcal{O}}{\left|\left(L_{-1}-\frac{2h}{q}J_{-1}\right)\mathcal{O}\right|^2}\tag{C.2.7}$$

where we've used the fact that $\left|\left(L_{-1}-\frac{2h}{q}J_{-1}\right)\mathcal{O}\right|^2=\frac{2h\left(2hk-q^2\right)}{q^2}$.

Note that the quasi-primaries $\mathcal{O}_{h+j}^{(i)}$ must include at least one J generator in them (like the J_{-1} in $\left(L_{-1}-\frac{2h}{q}J_{-1}\right)\mathcal{O}$), so their norms must be at least order k in the large k limit. This means that in the large k limit, we have

$$\lim_{k\to\infty}\mathcal{J}_{-n}=L_{-1}^{n},$$

as expected.

The decomposition of $\mathcal{J}_{-n}\mathcal{O}$ in (C.2.4) is useful is because we can make use of this fact when computing some correlation function. For example, since the two-point functions of $\mathcal{O}^\dagger$ with quasi-primaries vanish, the terms that will contribute to $\left\langle\phi\mathcal{O}^\dagger\right\rangle$ are those global descendants terms $L_{-1}^{n}\mathcal{O}$. Therefore, we have

$$\begin{aligned}\left\langle\phi\left(y,z,\bar z\right)\mathcal{O}\left(z_1,\bar z_1\right)\right\rangle&=\sum_{n=0}^{\infty}y^{2h+2n}\lambda_n\left\langle L_{-1}^{n}\bar L_{-1}^{n}\mathcal{O}\left(z,\bar z\right)\mathcal{O}^\dagger\left(z_1,\bar z_1\right)\right\rangle\\&=\left(\frac{y}{y^2+\left(z-z_1\right)\left(\bar z-\bar z_1\right)}\right)^{2h}\end{aligned}\tag{C.2.8}$$

C.2.2 Large k Expansion

In this section, we are going to solve the charged bulk primary conditions to next to leading order of the large k limit. We can assume the ansatz for $\mathcal{J}_{-n}\mathcal{O}$ up to this order to be

$$\mathcal{J}_{-n}=\left(1+\frac{a_0^{(n)}}{k}\right)L_{-1}^{n}+\sum_{i=1}^{n}\frac{a_i^{(n)}}{k}J_{-i}L_{-1}^{n-i}+O\left(\frac{1}{k^2}\right)\tag{C.2.9}$$

where $O\left(\frac{1}{k^2}\right)$ includes all the higher oder terms. For the exact $\mathcal{J}_{-n}\mathcal{O}$, we have $J_m\mathcal{J}_{-n}\mathcal{O}=0$ for $m\geq 1$. This means that we must have

$$J_m\left[\left(1+\frac{a_0^{(n)}}{k}\right)L_{-1}^n+\sum_{i=1}^n\frac{a_i^{(n)}}{k}J_{-i}L_{-1}^{n-i}\right]\mathcal{O}=O\left(\frac{1}{k}\right),\tag{C.2.10}$$

since J_m acting on some of the order $\frac{1}{k^2}$ terms may give order $\frac{1}{k}$ result. So the leading order contribution to the LHS of the above equation should vanish,

$$J_m\left(\left(1+\frac{a_0^{(n)}}{k}\right)L_{-1}^n\mathcal{O}+\sum_{i=1}^n\frac{a_i^{(n)}}{k}J_{-i}L_{-1}^{n-i}\right)\mathcal{O}\approx\left(\begin{array}{c}n\\m\end{array}\right)m!qL_{-1}^{n-m}+ma_m^{(n)}L_{-1}^{n-m}\mathcal{O}=0\tag{C.2.11}$$

where we discarded the order $\frac{1}{k}$ terms on the RHS of the approximation symbol and we've assumed $1\leq m\leq n$ in the above equation. Solving the equation on the RHS for $a_m^{(n)}$, we get

$$a_m^{(n)}=-\frac{n!}{(n-m)!m}q\tag{C.2.12}$$

Now using the condition that $L_1^n\mathcal{J}_{-n}\mathcal{O}=n!\,(2h)_n\,\mathcal{O}$, we can solve for $a_0^{(n)}$. We have

$$L_1^n\mathcal{J}_{-n}\mathcal{O}=\left[\left(1+\frac{a_0^{(n)}}{k}\right)n!\,(2h)_n+\sum_{i=1}^n\frac{a_i^{(n)}}{k}i!\left(\begin{array}{c}n\\i\end{array}\right)(n-i)!\,(2h)_{n-i}\right]\mathcal{O}+O\left(\frac{1}{k^2}\right).\tag{C.2.13}$$

Requiring the RHS to be equal to $n!\,(2h)_n\,\mathcal{O}$ up to order $\frac{1}{k^2}$, we get

$$a_0^{(n)}=-\frac{1}{(2h)_n}\sum_{i=1}^n a_i^{(n)}\,(2h)_{n-i}=-q(\psi^{(0)}(1-2h)-\psi^{(0)}(-2h-n+1))\tag{C.2.14}$$

where $\psi^{(0)}$ is the digamma function.

In summary, the order $\frac{1}{k}$ terms in $\mathcal{J}_{-n}$ are given by (C.2.9) with $a_i^{(n)}$ and $a_0^{(n)}$ given by (C.2.12) and (C.2.14), respectively. In appendix C.3.1, we are going to use this result to compute the $\frac{1}{k}$ correction to the bulk propagator $\langle\phi^\dagger\phi\rangle$.

C.3 Bulk Correlation Functions in $U(1)$ Chern-Simons Theory

In this appendix we first compute the $1/k$ corrections to the bulk propagator $\langle\phi^\dagger\phi\rangle$ using CFT techniques (i.e., using the result we obtained in last section for the $1/k$ corrections in ϕ). Then in appendix C.3.2, we compute $\langle\phi\mathcal{O}^\dagger J\rangle$ and $\langle\phi^\dagger\phi\rangle$ using Witten diagrams, and get exactly the same results as the CFT calculations. These calculations provide a non-trivial check of our definition of a bulk charged scalar field using the bulk primary condition (4.2.16).

C.3.1 CFT Calculation of $\langle \phi^\dagger \phi \rangle$

We can compute $\left\langle \phi^\dagger \left(y_1, z_1, \bar{z}_1\right) \phi \left(y_2, \bar{z}_2, \bar{z}_2\right) \right\rangle$, where $\phi^\dagger$ is ϕ with $q \to -q$, using the result we obtained in last section for the $1/k$ correction terms in ϕ. Up to order $\frac{1}{k}$, ϕ is given by

$$\phi \left(y, z, \bar{z}\right) = y^{2h} \sum_{n=0}^{\infty} y^{2n} \lambda_n \mathcal{J}_{-n} \bar{L}_{-1}^{n} \mathcal{O} \left(z, \bar{z}\right), \qquad \lambda_n = \frac{\left(-1\right)^n}{n! \left(2h\right)_n} \tag{C.3.1}$$

with

$$\mathcal{J}_{-n} = \left(1 + \frac{a_0^{(n)}}{k}\right) L_{-1}^{n} + \sum_{i=1}^{n} \frac{a_i^{(n)}}{k} J_{-i} L_{-1}^{n-i} + O\left(\frac{1}{k^2}\right). \tag{C.3.2}$$

and $a_i^{(n)} = -q \frac{n!}{(n-i)! i}$ and $a_0^{(n)} = -q(\psi^{(0)}(1-2h) - \psi^{(0)}(-2h-n+1))$, as computed in appendix C.2.2. We can compute $\left\langle \phi \left(y_1, z_1, \bar{z}_1\right) \phi \left(y_2, \bar{z}_2, \bar{z}_2\right) \right\rangle$ by directly inserting the above expansion of ϕ into $\left\langle \phi^\dagger \phi \right\rangle$ and summing over all the contributions. So up to order $\frac{1}{k}$, we have

$$\left\langle \phi^\dagger \left(y_1, z_1, \bar{z}_1\right) \phi \left(y_2, \bar{z}_2, \bar{z}_2\right) \right\rangle = \left(y_1 y_2\right)^{2h} \sum_{n_1, n_2 = 0}^{\infty} y_1^{2n_1} y_2^{2n_2} \lambda_{n_1} \lambda_{n_2} \left(\mathcal{A} + \mathcal{B} + \mathcal{C} + \mathcal{D}\right) \tag{C.3.3}$$

with

$$\mathcal{A} \equiv \left(1 + \frac{a_0^{(n_1)} + a_0^{(n_2)}}{k}\right) \left\langle L_{-1}^{n_1} \bar{L}_{-1}^{n_1} \mathcal{O}^\dagger \left(z_1, \bar{z}_1\right) L_{-1}^{n_2} \bar{L}_{-1}^{n_2} \mathcal{O} \left(z_2, \bar{z}_2\right) \right\rangle, \tag{C.3.4}$$

$$\mathcal{B} \equiv \sum_{i_2=1}^{n_2} \frac{a_{i_2}^{(n_2)}}{k} \left\langle L_{-1}^{n_1} \bar{L}_{-1}^{n_1} \mathcal{O}^\dagger \left(z_1, \bar{z}_1\right) J_{-i_2} L_{-1}^{n_2 - i_2} \bar{L}_{-1}^{n_2} \mathcal{O} \left(z_2, \bar{z}_2\right) \right\rangle, \tag{C.3.5}$$

$$\mathcal{C} \equiv \sum_{i_1=1}^{n_1} \frac{a_{i_1}^{(n_1)}}{k} \left\langle J_{-i_1} L_{-1}^{n_1 - i_1} \bar{L}_{-1}^{n_1} \mathcal{O}^\dagger \left(z_1, \bar{z}_1\right) L_{-1}^{n_2} \bar{L}_{-1}^{n_2} \mathcal{O} \left(z_2, \bar{z}_2\right) \right\rangle, \tag{C.3.6}$$

$$\mathcal{D} \equiv \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \frac{a_{i_1}^{(n_1)} a_{i_2}^{(n_1)}}{k^2} \left\langle J_{-i_1} L_{-1}^{n_1 - i_1} \bar{L}_{-1}^{n_1} \mathcal{O}^\dagger \left(z_1, \bar{z}_1\right) J_{-i_2} L_{-1}^{n_2 - i_2} \bar{L}_{-1}^{n_2} \mathcal{O} \left(z_2, \bar{z}_2\right) \right\rangle. \tag{C.3.7}$$

where we've only kept the terms that may contribute to order $\frac{1}{k}$.

The $\frac{1}{k}$ term in $\mathcal{A}$ will cancel the $\frac{1}{k}$ terms in $\mathcal{B}$ and $\mathcal{C}$[2], so the sum of $\mathcal{A} + \mathcal{B} + \mathcal{C}$ will be

$$\mathcal{A} + \mathcal{B} + \mathcal{C} = \left\langle L_{-1}^{n_1} \bar{L}_{-1}^{n_1} \mathcal{O}^\dagger \left(z_1, \bar{z}_1\right) L_{-1}^{n_2} \bar{L}_{-1}^{n_2} \mathcal{O} \left(z_2, \bar{z}_2\right) \right\rangle. \tag{C.3.9}$$

[2] This can be checked by direct calculation. On the other hand, we can also see that this must be true by separating ϕ into two parts: $\phi = \phi_0 + \phi_{\text{q-p}}$, where ϕ_0 is the free-field

$$\phi_0 = y^{2h} \sum_{n=0}^{\infty} y^{2n} \lambda_n L_{-1}^{n} \bar{L}_{-1}^{n} \mathcal{O} \tag{C.3.8}$$

and $\phi_{\text{q-p}}$ includes the contributions from quasi-primaries and their global descendants. We can do this separation because of the property of $\mathcal{J}_{-n}$ in (4.2.18). The point here is that $\langle \phi_0 \phi_{\text{q-p}} \rangle$ must be zero, since the two-point function $\mathcal{O}$ with a quasi-primary is zero and this is also true for the two-points of their global descendants. So the $1/k$ corrections can only come from $\langle \phi_{\text{q-p}}^\dagger \phi_{\text{q-p}} \rangle$, that is, the terms in $\mathcal{D}$ of (C.3.7).

Since L_{-1} and $\bar{L}_{-1}$ are simply derivatives ∂_z and $\partial_{\bar{z}}$, this is simply computed to be

$$\mathcal{A}+\mathcal{B}+\mathcal{C}=\frac{\left[(2h)_{n_1+n_2}\right]^2}{z_{12}^{2h+n_1+n_2}\,\bar{z}_{12}^{2h+n_1+n_2}}. \tag{C.3.10}$$

The sum of this term in equation (C.3.3) will give the free field limit of $\langle\phi\phi\rangle$, which is

$$\langle\phi_0\phi_0\rangle=\frac{\rho^h}{1-\rho},\qquad \text{with } \rho=\left(\frac{\xi}{1+\sqrt{1-\xi^2}}\right)^2,\quad \xi=\frac{2y_1y_2}{y_1^2+y_2^2+z_{12}\bar{z}_{12}} \tag{C.3.11}$$

where ϕ_0 is given in (C.3.8).

Now the only term left to compute is $\mathcal{D}$. The anti-holomorphic part can be computed simply, and we have

$$\begin{aligned}&\left\langle J_{-i_1}L_{-1}^{n_1-i_1}\bar{L}_{-1}^{n_1}\mathcal{O}^\dagger\left(z_1,\bar{z}_1\right)J_{-i_2}L_{-1}^{n_2-i_2}\bar{L}_{-1}^{n_2}\mathcal{O}\left(z_2,\bar{z}_2\right)\right\rangle \\ =&\frac{(2h)_{n_1+n_2}(-1)^{n_1}}{\bar{z}_{12}^{2h+n_1+n_2}}\left\langle J_{-i_1}L_{-1}^{n_1-i_1}\mathcal{O}\left(z_1\right)J_{-i_2}L_{-1}^{n_2-i_2}\mathcal{O}\left(z_2\right)\right\rangle\end{aligned} \tag{C.3.12}$$

Since we only care about the order $\frac{1}{k}$ terms in $\mathcal{D}$, we only need to compute the order k term in the two-point function on the RHS of the above equation, which is

$$\begin{aligned}&\left\langle J_{-i_1}L_{-1}^{n_1-i_1}\mathcal{O}\left(z_1\right)J_{-i_2}L_{-1}^{n_2-i_2}\mathcal{O}\left(z_2\right)\right\rangle \\ =&\frac{1}{2\pi i}\oint_{z_1}dz\left(z-z_1\right)^{-i_1}\frac{1}{\left(z-z_2\right)^{i_2+1}}\left\langle L_{-1}^{n_1-i_1}\mathcal{O}\left(z_1\right)J_{i_2}J_{-i_2}L_{-1}^{n_2-i_2}\mathcal{O}\left(z_2\right)\right\rangle+... \\ =&\frac{1}{2\pi i}\oint_{z_1}dz\frac{i_2k}{\left(z-z_1\right)^{i_1}\left(z-z_2\right)^{i_2+1}}\left\langle L_{-1}^{n_1-i_1}\mathcal{O}\left(z_1\right)L_{-1}^{n_2-i_2}\mathcal{O}\left(z_2\right)\right\rangle+... \\ =&k\frac{(-1)^{n_1-1}(2h)_{n_1-i_1+n_2-i_2}}{z_{12}^{2h+n_1+n_2}}\frac{(i_2)_{i_1}}{(i_1-1)!}+...\end{aligned} \tag{C.3.13}$$

where we've only kept the leading order terms.

Putting everything together, we have

$$\begin{aligned}\left\langle\phi^\dagger\phi\right\rangle=&\langle\phi_0\phi_0\rangle-\frac{q^2}{k}\left(X_1X_2\right)^h \\ &\times\sum_{n_1,n_2=0}^{\infty}\frac{(-1)^{n_1+n_2}(2h)_{n_1+n_2}}{(2h)_{n_1}(2h)_{n_2}}\left(\sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\frac{(2h)_{n_1+n_2-i_1-i_2}(i_2)_{i_1}}{(n_1-i_1)!\,(n_2-i_2)!\,(i_1-1)!i_1i_2}\right)X_1^{n_1}X_2^{n_2}+O\left(\frac{1}{k^2}\right)\end{aligned} \tag{C.3.14}$$

where we've defined $X_1\equiv\frac{y_1^2}{z_{12}\bar{z}_{12}}, X_2\equiv\frac{y_2^2}{z_{12}\bar{z}_{12}}$. Mathematica is not able to perform the above sums, but it can be checked that the following expression

$$\left\langle\phi^\dagger\phi\right\rangle=\frac{\rho^h}{1-\rho}\left[1-\frac{q^2}{k}\left(\frac{\rho^2\,{}_2F_1(1,2h+1;2(h+1);\rho)}{2h+1}+\frac{\rho}{2h}-\log(1-\rho)\right)\right]+O\left(\frac{1}{k^2}\right) \tag{C.3.15}$$

gives the same expansion when expanded in small X_1 and X_2 . In appendix C.3.2, we'll show that the above expression is the result of the bulk Witten diagram calculation. Thus this provides a non-trivial check of our definition of a bulk charged proto-field in $U(1)$ Chern-Simons theory.

C.3.2 Bulk Witten Diagram Calculation for $\left\langle J\mathcal{O}^{\dagger}\phi\right\rangle$ and $\langle\phi^{\dagger}\phi\rangle$

In this section, we are going to compute $\left\langle J\mathcal{O}^{\dagger}\phi\right\rangle$ and $\langle\phi^{\dagger}\phi\rangle$ using Witten diagrams. The calculations here will be very similar to that of $\langle T\mathcal{O}\phi\rangle$ in [8] and that of $\langle\phi\phi\rangle$ in [27], except that the $\langle A_z A_z\rangle$ two-point function and the bulk three-point vertex are different, but actually simpler.

The action of the $U(1)$ Chern-Simons theory in Poincare AdS$_3$ is given by

$$I=\int d^3x\frac{k}{4\pi}\epsilon^{\mu\nu\lambda}A_\mu\partial_\nu A_\lambda+\sqrt{|g|}\left(\nabla_\mu\phi\left(\nabla^\mu\phi\right)^\dagger+m^2\phi\phi^\dagger\right)\tag{C.3.16}$$

with $\nabla_\mu\phi=\left(\partial_\mu+iqA_\mu\right)\phi$ and $\left(\nabla_\mu\phi\right)^\dagger=\left(\partial_\mu-iqA_\mu\right)\phi^\dagger$. The Poincare AdS$_3$ metric and its inverse are

$$g_{\mu\nu}=\left(\begin{array}{ccc}\frac{1}{y^2}&0&0\\0&0&\frac{1}{2y^2}\\0&\frac{1}{2y^2}&0\end{array}\right),\quad g^{\mu\nu}=\left(\begin{array}{ccc}y^2&0&0\\0&0&2y^2\\0&2y^2&0\end{array}\right)\tag{C.3.17}$$

in the coordinate system $(y,z,\bar{z})$. From the Chern-Simons action (C.3.16), we can see that the photon two point function is given by

$$H\left(Y_1,Y_2\right)\equiv\left\langle A_z\left(y_1,z_1,\bar{z}_1\right)A_z\left(y_2,z_2,\bar{z}_2\right)\right\rangle=\frac{1}{k\left(z_1-z_2\right)^2}.\tag{C.3.18}$$

Here and in some other parts of this paper, we'll use X or Y to denote $(y,z,\bar{z})$ for convenience. The scalar two point function is just the usual bulk-bulk propagator as in (C.3.11)

$$G\left(Y_1,Y_2\right)\equiv\left\langle\phi^\dagger\left(y_1,z_1,\bar{z}_1\right)\phi\left(y_2,z_2,\bar{z}_2\right)\right\rangle=\frac{\rho^h}{1-\rho},\tag{C.3.19}$$

with $m^2=4h\left(h-1\right)$. The photon-scalar-scalar three point vertex is given by

$$iqg^{\mu\nu}A_\mu\left(\phi\partial_\mu\phi^\dagger-\phi^\dagger\partial_\mu\phi\right).\tag{C.3.20}$$

And since we'll only be interested in the z component of A_z, the above vertex becomes

$$i2qy^2A_z\left(\phi\partial_{\bar{z}}\phi^\dagger-\phi^\dagger\partial_{\bar{z}}\phi\right)\tag{C.3.21}$$

where we've used $g^{\bar{z}z}=2y^2$. We'll assume[3] that we are free to perform an integration by parts, so that the above vertex can be written as $-4iqy^2A_z\phi^\dagger\partial_{\bar{z}}\phi$.

We'll also need to bulk-boundary propagator for the charged scalar field and the photon field. These are given by

$$K\left(Y_1,\left(z_2,\bar{z}_2\right)\right)\equiv\left\langle\phi\left(y_1,z_1,\bar{z}_1\right)\mathcal{O}^\dagger\left(z_2,\bar{z}_2\right)\right\rangle=\left(\frac{y_1}{y_1^2+\left(z_1-z_2\right)\left(\bar{z}_1-\bar{z}_2\right)^2}\right)^{2h}\tag{C.3.22}$$

[3] There may be some subtleties about this, which we explain in the calculation of $\langle\phi\mathcal{O}J\rangle$ in next subsection.

and

$$\langle A_z(Y_1)\, J(z_2)\rangle = \frac{1}{(z_1-z_2)^2}, \tag{C.3.23}$$

where we've used the fact that $J(z) = \frac{A_z}{k}$.

Witten Diagram Calculation for $\left\langle J\mathcal{O}^\dagger\phi\right\rangle$

The Witten diagram for $\left\langle J\mathcal{O}^\dagger\phi\right\rangle$ is given by figure C.1. We'll use the saddle point approximation to evaluate this diagram. The idea is that instead of integrating the bulk vertex point over AdS$_3$, we only integrate along the geodesic connecting ϕ and $\mathcal{O}^\dagger$. In appendix D.4 of [8], we showed that the saddle point approximation for $\langle\phi\mathcal{O}T\rangle$ actually gives the exact result. We expect the same thing to happen here.

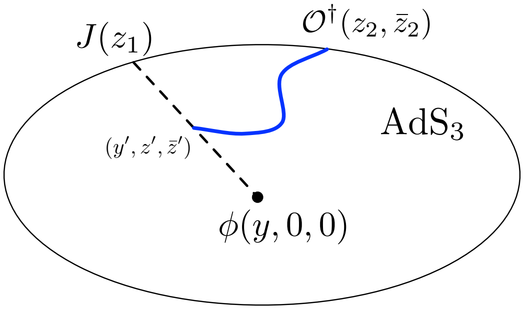

Figure C.1: Witten diagram for $\langle J\mathcal{O}^\dagger\phi\rangle$

The bulk-boundary three-point function $\left\langle J(z_1)\,\mathcal{O}^\dagger(z_2,\bar{z}_2)\,\phi(X_1)\right\rangle$ (with $X_1 = (y,0,0)$) is computed as usual by

$$\left\langle J\mathcal{O}^\dagger\phi\right\rangle = \int_{\text{AdS}_3}\sqrt{g}d^3X'\frac{1}{(z'-z_1)^2}2iqy'^2\left[G\partial_{\bar{z}'}K - G\partial_{\bar{z}'}K\right] \tag{C.3.24}$$

where where we've denoted $X' = (y',z',\bar{z}')$, and also used the bulk-boundary propagator for the photons (C.3.23). Explicitly, the coordinate dependence of G and K are $G = G(X_1, X')$ and $K = K(X',(z_2,\bar{z}_2))$. Now as mentioned above, we integrate by parts[4] to find

$$\left\langle J\mathcal{O}^\dagger\phi\right\rangle = -\int_{\text{AdS}_3}\sqrt{g}d^3X'\frac{4iqy'^2}{(z'-z_1)^2}G\partial_{\bar{z}'}K \tag{C.3.25}$$

Now we can do a saddle point approximation as in appendix D.4 of [8]. The idea is as follows. We can write

$$G(X_1,X') = \frac{e^{-2h\sigma(X_1,X')}}{1-e^{-2\sigma(X_1,X')}} \tag{C.3.26}$$

[4] In fact, when performing an integration by parts, there is a delta function term coming from $\partial_{\bar{z}'}\frac{1}{z'-z_1} = \pi\delta^2(z'-z_1,\bar{z}'-\bar{z}_1)$. Such terms would contaminate pure CFT correlators (as well as bulk correlators) and violate Ward identities, and so we have dropped them. This can be viewed as a choice of regulator. It would be very interesting to better understand regulation, and the role of these terms, in future work.

where $\rho \equiv e^{-2\sigma}$ and $\sigma_{(X_1,X')}$ is the geodesic length between $X_1 = (y,0,0)$ and $X' = (y', z', \bar{z}')$, whose expression can be obtained from (C.3.11). Similarly, the bulk boundary propagator can be written as

$$K\left(X', (z_2, \bar{z}_2)\right) = e^{-2h\sigma_{(X',(z_2,\bar{z}_2))}} \tag{C.3.27}$$

where $\sigma_{(X',(z_2,\bar{z}_2))} = \log \frac{y'^2 + (z'-z_2)(\bar{z}'-\bar{z}_2)}{y'}$ is the (regularized) bulk-boundary geodesic length between $X' = (y', z', \bar{z}')$ and $(z_2, \bar{z}_2)$. So the integral (C.3.25) after simplification can be written in the following form

$$\left\langle J\mathcal{O}^\dagger\phi\right\rangle = 8ihq \int \frac{dz'd\bar{z}'dy'}{y'^2} e^{-2hL(y',z',\bar{z}')} \frac{e^{-\sigma_{(y',z',\bar{z}'),(z_2,\bar{z}_2)}}}{1-e^{-2\sigma_{(y,0,0),(y',z',\bar{z}')}}} \frac{(z'-z_2)}{(z'-z_1)^2} \tag{C.3.28}$$

where $L(y', z', \bar{z}') \equiv \sigma_{(X_1,X')} + \sigma_{(X',(z_2,\bar{z}_2))}$. Now we can take the large h limit and the integration will be dominated by the line integral along the geodesic from $X_1 = (y,0,0)$ to $(z_2, \bar{z}_2)$. This geodesic parameterized by z' is given by

$$\bar{z}' = \frac{\bar{z}}{z} z', \quad y'^2 = \left(1 - \frac{z'}{z}\right)\left(y^2 + z'\bar{z}\right), \tag{C.3.29}$$

so that the saddle point approximation to equation (C.3.25) is

$$\langle J\mathcal{O}^\dagger\phi\rangle = 8\pi iqe^{-2hL(y,0,0)} \int_0^{z_2} dz' \frac{1}{\sqrt{|\det \partial^2 L|}} \frac{e^{-2\sigma_{(X',(z_2,\bar{z}_2))}}}{1-e^{-2\sigma_{(X_1,X')}}} \frac{1}{y'^2} \frac{(z'-z_2)}{(z'-z_1)^2}, \tag{C.3.30}$$

where the determinant is given by

$$\det \partial^2 L = \det \begin{pmatrix} \partial^2_{\bar{z}'} L & \partial_{\bar{z}'}\partial_{y'} L \\ \partial_{y'}\partial_{\bar{z}'} L & \partial^2_{y'} L \end{pmatrix} = \frac{4z_2^5\left(z'\bar{z}_2 + y^2\right)}{z'^2\left(z'-z_2\right)\left(z_2\bar{z}_2 + y^2\right)^4}. \tag{C.3.31}$$

Plugging this in (C.3.30), we have

$$\left\langle J(z_1)\mathcal{O}^\dagger(z_2,\bar{z}_2)\phi(y,0,0)\right\rangle = q\left\langle \phi\mathcal{O}^\dagger\right\rangle \int_0^{z_2} dz' \frac{1}{(z_1-z')^2} = \frac{qz_2}{(z_2-z_1)\,z_1}\left\langle\phi\mathcal{O}^\dagger\right\rangle \tag{C.3.32}$$

where we've used the fact that $\langle\phi(y,0,0)\mathcal{O}^\dagger(z_2,\bar{z}_2)\rangle = e^{-2hL(y,0,0)} = \left(\frac{y}{y^2+z_2\bar{z}_2}\right)^{2h}$ and we neglected a numerical constant in obtaining the above result. The above result is exactly the same as the result (4.2.22) obtained via assuming ϕ to be a charged bulk field defined by the bulk primary condition (4.2.16).

One-loop Correction to $\langle\phi^\dagger\phi\rangle$

In this subsection, we are going to compute one loop correction to $\langle\phi^\dagger\phi\rangle$. The Witten diagram is given in figure C.2, and it's computed as follows

$$\left\langle \phi^\dagger\left(X_1\right)\phi\left(X_2\right)\right\rangle_{1\text{ loop}} = -16q^2 \int \sqrt{|g|}d^3X' \int \sqrt{|g|}d^3X'' y'^2 y''^2 \tag{C.3.33}$$

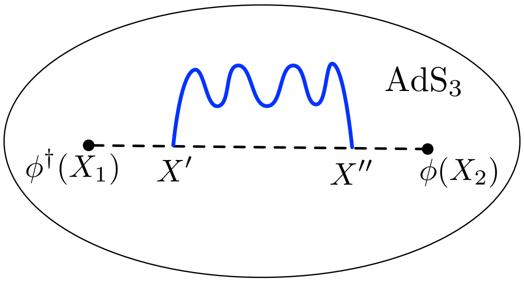

Figure C.2: Witten diagram for $\langle \phi^\dagger \phi \rangle$

$$\times G\left(X_1, X'\right) \partial_{\bar{z}'} G\left(X', X''\right) H\left(X', X''\right) \partial_{\bar{z}''} G\left(X'', X_2\right)$$

Acting on this two-point function with the Klein-Gordon operator twice, we have

$$\left(\nabla_1^2 + m^2\right)\left(\nabla_2^2 + m^2\right)\left\langle \phi^\dagger\left(X_1\right)\phi\left(X_2\right)\right\rangle_{1\text{ loop}} = -16q^2 H\left(X_1, X_2\right) y_1^2 y_2^2 \partial_{\bar{z}_1}\partial_{\bar{z}_2} G\left(X_1, X_2\right) \tag{C.3.34}$$

By using the photon propagator (C.3.18) and the bulk-bulk propagator (C.3.19), one can show that the RHS of the above equation is

$$-\frac{32q^2\rho^{h+1}\left(2h^2(1-\rho)^2 + h\left(-5\rho^2+4\rho+1\right)+3\rho(\rho+1)\right)}{k(1-\rho)^5}. \tag{C.3.35}$$

Assuming $P(\rho) \equiv \left\langle \phi^\dagger\left(X_1\right)\phi\left(X_2\right)\right\rangle_{1\text{ loop}}$ only depends on the ρ, equation (C.3.34) becomes

$$\begin{aligned} &16\left(h-1\right)^2 h^2 P\left(\rho\right) + \frac{64}{\rho-1}\left(-h^2+h+1\right)\rho^2 P'\left(\rho\right) \qquad \text{(C.3.36)} \\ &-\frac{32\rho^2\left(\left(h-1\right)h\left(\rho-1\right)-7\rho+1\right)P''\left(\rho\right)}{\rho-1} + \frac{64\rho^3\left(2\rho-1\right)P^{(3)}\left(\rho\right)}{\rho-1} + 16\rho^4 P^{(4)}\left(\rho\right) \\ =&-\frac{32q^2\rho^{h+1}\left(2h^2(1-\rho)^2+h\left(-5\rho^2+4\rho+1\right)+3\rho(\rho+1)\right)}{k(1-\rho)^5} \end{aligned}$$

Luckily, Mathematica is able to solve the above fourth order differential equation. Fixing the integration constants, we get

$$P\left(\rho\right) = -\frac{q^2}{k}\frac{\rho^h}{1-\rho}\left[\frac{\rho^2\,{}_2F_1(1,2h+1;2(h+1);\rho)}{2h+1} + \frac{\rho}{2h} - \log(1-\rho)\right], \tag{C.3.37}$$

which is exactly the same as the $\frac{1}{k}$ correction to the two-point function of the bulk charged proto-field computed in appendix C.3.1 using pure CFT technics.

C.4 Computations of $\langle \phi \mathcal{O} T \rangle$ in Global AdS$_3$

In this section, we use two methods to calculate $\langle \phi \mathcal{O} T \rangle$ in global AdS$_3$. These two methods are essentially the same. Both of them are based on the idea of bulk-boundary OPE blocks (as we called them in [8] and [27]) or the bulk-boundary bi-local operator $\phi\mathcal{O}$ (C.4.2) (a generalization of the boundary bi-local operator as defined in [35]). The first method is to expand the vacuum channel of the bulk-boundary bi-local operator in terms of the ϵ operator defined in section 6 of [35], and then use the ϵ propagator, which was derived from the Alekseev-Shatashvilli theory of boundary gravitons in that paper, to compute $\langle\phi\mathcal{O}T\rangle$. The second method is to expand the vacuum channel of the bi-local in terms of the energy-momentum tensor T and use the two-point function of T to compute $\langle\phi\mathcal{O}T\rangle$. Both of them give the same result as (4.3.31) obtained using the properties of the bulk proto-field.

C.4.1 $\langle\phi\mathcal{O}T\rangle$ in Global AdS$_3$ from Alekseev-Shatashvili

The idea here very similar to that of section 4.5.2, where we computed the $1/c$ corrections to the bulk-boundary propagator in a heavy background. The difference is that here we are considering bulk-boundary bi-local operator $\phi\mathcal{O}$ in the vacuum.

As in section 4.3.3, we use $f(z) = e^{z}$ and $\bar f(\bar z) = e^{\bar z}$ to obtain the global AdS$_3$ metric (4.3.28). But now we are going to include perturbation as follows

$$f(z) = e^{z+i\frac{\epsilon(z)}{c}}, \text{ and } \bar f(\bar z) = e^{\bar z - i\frac{\bar\epsilon(\bar z)}{c}} \tag{C.4.1}$$

(with $\epsilon(z)$ and $\bar\epsilon(z)$ promoted to operators). Specifically, the vacuum contribution to the bulk-boundary bi-local operator is given by

$$\phi(y_1, z_1, \bar z_1)\,\mathcal{O}(z_2, z_2)\,|_{\text{vac}} = \left(f'(z_2)\,\bar f'(\bar z_2)\right)^h \left(\frac{u_1}{u_1^2 + (x_1 - f(z_2))(\bar x_1 - f(\bar z_2))}\right)^{2h}. \tag{C.4.2}$$

The $(u_1, x_1, \bar x_1)$ here are functions of $(y_1, z_1, \bar z_1)$ given by (4.3.10) with $f(z)$ and $\bar f(\bar z)$ given by (C.4.1). One can see that the above expression becomes the boundary bi-local defined in [35] when we send ϕ to the boundary by taking $y_1 \to 0$. Written in terms of the coordinates $(y, z, \bar z)$ and expanded in large c, the above vacuum block becomes

$$\frac{\phi(y_1, z_1, \bar z_1)\,\mathcal{O}(z_2, \bar z_2)\,|_{\text{vac}}}{\langle\phi(y_1, z_1, \bar z_1)\,\mathcal{O}(z_2, \bar z_2)\rangle_{\text{global AdS}_3}} = 1 + \frac{1}{c}\frac{B\epsilon_1 + C\epsilon_1' + D\epsilon_1'' + E\epsilon_2 + F\epsilon_2'}{A} + \mathcal{O}\left(\frac{1}{c^2}\right) \tag{C.4.3}$$

where $\epsilon_i \equiv \epsilon(z_i)$ and the derivatives are with respect to z_i. The bulk-boundary propagator $\langle\phi(y_1, z_1, \bar z_1)\,\mathcal{O}(z_2, \bar z_2)\rangle_{\text{global AdS}_3}$ in global AdS3 is given in (4.3.32) (up to a factor of $(\xi_2\bar\xi_2)^h$ coming from the $\left(f'(z_2)\,\bar f'(\bar z_2)\right)^h$ in (C.4.2)), i.e.

$$\langle\phi\mathcal{O}\rangle_{\text{global AdS}_3} = \left(\frac{4y_1\sqrt{\xi_1\bar\xi_1\xi_2\bar\xi_2}}{\left(\bar\xi_1\xi_1 + \bar\xi_2\xi_2\right)\left(y_1^2+4\right) + \left(\bar\xi_1\xi_2 + \bar\xi_2\xi_1\right)\left(y_1^2-4\right)}\right)^{2h}, \tag{C.4.4}$$

where we've defined $\xi_i \equiv e^{z_i}$ and $\bar\xi_i = e^{\bar z_i}$. The denominator and the coefficients in the numerator of (C.4.3) are given by

$$A = i\left(\left(y_1^2-4\right)\left(\xi_2\bar\xi_1 + \xi_1\bar\xi_2\right) + \left(y_1^2+4\right)\left(\xi_1\bar\xi_1 + \xi_2\bar\xi_2\right)\right),$$

$$\begin{aligned}
B &= \left(y_1^2-4\right)\left(\xi_1\bar{\xi}_2-\xi_2\bar{\xi}_1\right)+\left(y_1^2+4\right)\left(\xi_1\bar{\xi}_1-\xi_2\bar{\xi}_2\right), \\
C &= \left(y_1^2+4\right)\left(\xi_2\bar{\xi}_1+\xi_1\bar{\xi}_2\right)+\left(y_1^2-4\right)\left(\xi_1\bar{\xi}_1+\xi_2\bar{\xi}_2\right), \\
D &= -2y_1^2\left(\xi_1-\xi_2\right)\left(\bar{\xi}_1+\bar{\xi}_2\right), \\
E &= \left(y_1^2+4\right)\left(\xi_2\bar{\xi}_2-\xi_1\bar{\xi}_1\right)+\left(y_1^2-4\right)\left(\xi_2\bar{\xi}_1-\xi_1\bar{\xi}_2\right), \\
F &= -\left(y_1^2-4\right)\left(\xi_2\bar{\xi}_1+\xi_1\bar{\xi}_2\right)-\left(y_1^2+4\right)\left(\xi_1\bar{\xi}_1+\xi_2\bar{\xi}_2\right).
\end{aligned} \tag{C.4.5}$$

We've only kept the holomorphic ϵ terms in (C.4.3), since the anti-holomorphic $\bar{\epsilon}$ will not contribute to $\langle\phi\mathcal{O}T\rangle$.

To compute $\langle\phi\mathcal{O}T\rangle$, we also need to write the energy-momentum tensor T in terms of the ϵ operators. As explained in section 2 of [8], T is simply given by the Schwarzian derivative as follows:

$$T\left(z\right)=\frac{c}{12}\left[\frac{f'''(z)f'(z)-\frac{3}{2}\left(f''(z)\right)^2}{\left(f'(z)\right)^2}\right]=-\frac{c}{24}-\frac{i}{12}\left(\epsilon'(z)-\epsilon^{(3)}(z)\right)+\mathcal{O}\left(\frac{1}{c}\right). \tag{C.4.6}$$

The last piece of information we need to compute $\langle\phi\mathcal{O}T\rangle$ is the ϵ propagator $\langle\epsilon\epsilon\rangle$. The action obeyed by ϵ and $\bar{\epsilon}$ is the saddle point quadratic action derived from the Alekseev-Shatashvilli action. And the ϵ propagator is given in section 6 of [36],

$$\langle\epsilon\left(z\right)\epsilon\left(0\right)\rangle=6c\left(-1+\frac{3\xi}{2}-\frac{\left(1-\xi\right)^2}{\xi}\ln\left(1-\xi\right)\right),\quad \xi\equiv e^z \tag{C.4.7}$$

Note that we have a different normalization for the ϵs.

Eventually, $\langle\phi\left(y_1,z_1,\bar{z}_1\right)\mathcal{O}\left(z_2,\bar{z}_2\right)T\left(z\right)\rangle$ is given by[5]

$$\frac{\langle\phi\mathcal{O}T\rangle}{\langle\phi\mathcal{O}\rangle_{\text{global AdS}_3}}=\frac{h\left(\xi_1-\xi_2\right)^2\xi^2}{\left(\xi-\xi_1\right)^3\left(\xi-\xi_2\right)^2}\left[\xi-\xi_1+\frac{4y_1^2\xi_1\left(\xi-\xi_2\right)\left(\bar{\xi}_1+\bar{\xi}_2\right)}{\left(\bar{\xi}_1\xi_1+\bar{\xi}_2\xi_2\right)\left(y_1^2+4\right)+\left(\bar{\xi}_1\xi_2+\bar{\xi}_2\xi_1\right)\left(y_1^2-4\right)}\right]-\frac{c}{24} \tag{C.4.8}$$

which agrees with (4.3.31) up to a trivial transformation for $\mathcal{O}$ and T from the complex plane $\left(\xi,\bar{\xi}\right)$ to the cylinder $(z,\bar{z})$.

C.4.2 $\langle\phi\mathcal{O}T\rangle$ in Global AdS$_3$ from $\langle TT\rangle$

We can also compute $\langle\phi\mathcal{O}T\rangle$ in global AdS$_3$ without using the ϵ propagator as we did in last subsection. Instead, we'll use the two-point function of the energy-momentum tensor T. This was the method that we used in [8] to compute $\langle\phi\mathcal{O}T\rangle$ (equation (4.3.26)) in Poincare AdS$_3$, and also in appendix B of [27] to compute the $1/c$ corrections to the vacuum block of all-light bulk-boundary correlator $\langle\mathcal{O}_2\mathcal{O}_2\phi_1\mathcal{O}_1\rangle$ (with ϕ_1 the proto-field of section 4.3.1) up to order $1/c^2$.

[5] There could be $1/c$ corrections to $\langle\phi\mathcal{O}T\rangle$ coming from the higher order terms in (C.4.3) and (C.4.6). But since the result for $\langle\phi\mathcal{O}T\rangle$ is exact without any $1/c$ correction (as computed using the properties of the bulk proto-field in 4.3.3), such higher order have been dropped by the regulator, as in [49, 8, 27].

The idea is actually very similar to that of last section. Instead of denoting the perturbation via ϵ, we expand $f\left(z\right)$ in the large c limit as

$$f(z) = e^{z} + \frac{1}{c}f_1(z) + \dots , \tag{C.4.9}$$

and similarly for $\bar{f}\left(\bar{z}\right)$. Then similar to last subsection, the energy-momentum tensor $T\left(z\right)$ is given by

$$T(z) = \frac{c}{12}\left[\frac{f'''(z)f'(z) - \frac{3}{2}\left(f''(z)\right)^2}{\left(f'(z)\right)^2}\right] = -\frac{c}{24} + \frac{1}{12}e^{-z}\left(2f_1' - 3f_1'' + f_1^{(3)}\right) + O\left(\frac{1}{c}\right). \tag{C.4.10}$$

The energy-momentum tensor $T\left(\xi\right)$ on the complex plane with coordinates $\left(\xi = e^{z}, \bar{\xi} = e^{\bar{z}}\right)$ is related to $T\left(z\right)$ as usual by

$$T\left(z\right) = -\frac{c}{24} + \xi^2 T\left(\xi\right), \tag{C.4.11}$$

so from (C.4.10), we have

$$T\left(e^{z}\right) = \frac{1}{12}e^{-3z}\left(2f_1' - 3f_1'' + f_1^{(3)}\right) \tag{C.4.12}$$

We can now solve for $f_1\left(z\right)$ in terms of $T\left(e^{z}\right)$, and the solution is given by

$$f_1(z) = 12\int_0^{z}\left(\int_0^{z'} e^{z'+z''}(e^{z'} - e^{z''})T\left(e^{z''}\right)\, dz''\right)dz'. \tag{C.4.13}$$

The bulk-boundary OPE block or the bulk boundary bi-local operator (C.4.2) can then be expanded in large c,

$$\frac{\phi\left(y,0,0\right)\mathcal{O}\left(z_2,\bar{z}_2\right)|_{\text{vac}}}{\left\langle\phi\left(y,0,0\right)\mathcal{O}\left(z_2,\bar{z}_2\right)\right\rangle_{\text{global AdS}_3}} = 1 + \frac{1}{c}\left(\frac{hf_1'\left(z_2\right)}{\xi_2} - \frac{2h\left(\left(y^2+4\right)\bar{\xi}_2 + y^2 - 4\right)f_1\left(z_2\right)}{\left(y^2+4\right)\left(1+\bar{\xi}_2\xi_2\right) + \left(y^2-4\right)\left(\xi_2+\bar{\xi}_2\right)}\right) + O\left(\frac{1}{c^2}\right) \tag{C.4.14}$$

Using $\left\langle T\left(e^{z}\right)T\left(e^{z_2}\right)\right\rangle = \frac{c}{\left(e^{z}-e^{z_2}\right)^4}$ on the complex plane, we find

$$\left\langle f_1\left(z_2\right)T\left(e^{z}\right)\right\rangle = \frac{c\left(\xi_2-1\right)^3}{\left(\xi-1\right)^3\left(\xi-\xi_2\right)} \tag{C.4.15}$$

$$\left\langle f_1'\left(z_2\right)T\left(e^{z}\right)\right\rangle = -\frac{c\left(\xi_2-1\right)^2\xi_2\left(1-3\xi+2\xi_2\right)}{\left(\xi-1\right)^3\left(\xi-\xi_2\right)^2}$$

and $\left\langle\phi\left(y,0,0\right)\mathcal{O}\left(z_2,\bar{z}_2\right)T\left(z\right)\right\rangle$ (after transforming T to the cylinder) is given by

$$\frac{\langle\phi\mathcal{O}T\rangle}{\langle\phi\mathcal{O}\rangle_{\text{global AdS}_3}} = \frac{h\left(1-\xi_2\right)^2\xi^2}{\left(\xi-1\right)^3\left(\xi-\xi_2\right)^2}\left[\xi - 1 + \frac{4y_1^2\left(\xi-\xi_2\right)\left(1+\bar{\xi}_2\right)}{\left(1+\bar{\xi}_2\xi_2\right)\left(y^2+4\right) + \left(\xi_2+\bar{\xi}_2\right)\left(y^2-4\right)}\right] - \frac{c}{24} \tag{C.4.16}$$

which is exactly the result obtained in last subsection (C.4.8) with $y_1 = y, z_1 = \bar{z}_1 = 0$.

References

[1] Ben Adlam and Jeffrey Pennington. “The Neural Tangent Kernel in high dimensions: Triple descent and a multi-scale theory of generalization”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 74–84.

[2] Ben Adlam and Jeffrey Pennington. “Understanding Double Descent Requires A Fine-Grained Bias-Variance Decomposition”. In: *Advances in Neural Information Processing Systems* 33 (2020).

[3] Madhu S Advani and Andrew M Saxe. “High-dimensional dynamics of generalization error in neural networks”. In: *arXiv preprint arXiv:1710.03667* (2017).

[4] Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. “High-dimensional dynamics of generalization error in neural networks”. In: *Neural Networks* 132 (2020), pp. 428–446.

[5] Subutai Ahmad and Gerald Tesauro. “Scaling and generalization in neural networks: a case study”. In: *Advances in neural information processing systems*. 1989, pp. 160–168.

[6] Alnur Ali, J Zico Kolter, and Ryan J Tibshirani. “A continuous-time view of early stopping for least squares regression”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 1370–1378.

[7] K. B. Alkalaev and V. A. Belavin. “From global to heavy-light: 5-point conformal blocks”. In: *JHEP* 03 (2016), p. 184. DOI: 10.1007/JHEP03(2016)184.

[8] Nikhil Anand et al. “An Exact Operator That Knows Its Location”. In: (2017).

[9] Anders Andreassen and Ethan Dyer. “Asymptotics of Wide Convolutional Neural Networks”. In: *arxiv preprint arXiv:2008.08675* (2020).

[10] Alessio Ansuini et al. *Intrinsic dimension of data representations in deep neural networks*. 2019.

[11] Vijay Balasubramanian and Per Kraus. “A Stress tensor for Anti-de Sitter gravity”. In: *Commun. Math. Phys.* 208 (1999), pp. 413–428. DOI: 10.1007/s002200050764.

[12] Maximo Banados. “Three-dimensional quantum geometry and black holes”. In: (1998). [AIP Conf. Proc.484,147(1999)], pp. 147–169. DOI: 10.1063/1.59661.

[13] Ronen Basri and David Jacobs. *Efficient Representation of Low-Dimensional Manifolds using Deep Networks.* 2016.

[14] Luís M. A. Bettencourt et al. "Growth, innovation, scaling, and the pace of life in cities". In: *Proc Natl Acad Sci U S A.* 104(17) (2007). DOI: `10.1073/pnas.0610172104`.

[15] GÃŠrard Biau. "Analysis of a random forests model". In: *Journal of Machine Learning Research* 13.Apr (2012), pp. 1063–1095.

[16] Peter J Bickel, Bo Li, et al. "Local polynomial regression on unknown manifolds". In: *Complex datasets and inverse problems.* Institute of Mathematical Statistics, 2007, pp. 177–186.

[17] Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. "Spectrum dependent learning curves in kernel regression and wide neural networks". In: *International Conference on Machine Learning.* PMLR. 2020, pp. 1024–1034.

[18] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs.* Version 0.2.5. 2018. URL: `http://github.com/google/jax`.

[19] The Editors of Encyclopaedia Britannica. *Curie point.* 2016. URL: `https://www.britannica.com/science/Curie-point`.

[20] Tom B Brown et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[21] Lars Buitinck et al. "API design for machine learning software: experiences from the scikit-learn project". In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning.* 2013, pp. 108–122.

[22] Francesco Camastra and Antonino Staiano. "Intrinsic dimension estimation: Advances and open problems". In: *Information Sciences* 328 (2016), pp. 26–41.

[23] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. "Statistical Mechanics of Generalization in Kernel Regression". In: *arXiv preprint arXiv:2006.13198* (2020).

[24] Bruno Carneiro da Cunha and Monica Guica. "Exploring the BTZ bulk with boundary conformal blocks". In: (2016).

[25] Alejandra Castro, Nabil Iqbal, and Eva Llabrés. "Wilson lines and Ishibashi states in AdS_3/CFT_2". In: *JHEP* 09 (2018), p. 066. DOI: `10.1007/JHEP09(2018)066`.

[26] Hongbin Chen et al. "The AdS_3 propagator and the fate of locality". In: *JHEP* 04 (2018), p. 075. DOI: `10.1007/JHEP04(2018)075`.

[27] Hongbin Chen et al. "The Bulk-to-Boundary Propagator in Black Hole Microstate Backgrounds". In: (2018).

[28] Lenaic Chizat, Edouard Oyallon, and Francis Bach. "On lazy training in differentiable programming". In: *Advances in Neural Information Processing Systems.* 2019, pp. 2937–2947.

[29] Minjae Cho, Scott Collier, and Xi Yin. “Recursive Representations of Arbitrary Virasoro Conformal Blocks”. In: (2017).

[30] M. Chui et al. *Notes from the AI frontier: Applications and value of deep learning.* 2018. URL: `https://www.mckinsey.com/featured-insights/artificial-intelligence/notes-from-the-ai-frontier-applications-and-value-of-deep-learning#`.

[31] Karl Cobbe et al. *Leveraging Procedural Generation to Benchmark Reinforcement Learning.* 2019.

[32] Omry Cohen, Or Malka, and Zohar Ringel. “Learning curves for deep neural networks: a gaussian field theory perspective”. In: *arXiv preprint arXiv:1906.05301* (2019).

[33] David Cohn and Gerald Tesauro. “Can neural networks do better than the Vapnik-Chervonenkis bounds?” In: *Advances in Neural Information Processing Systems.* 1991, pp. 911–917.

[34] Wikimedia Commons. URL: `https://commons.wikimedia.org/wiki/File:Colored_neural_network.svg`.

[35] Jordan Cotler and Kristan Jensen. “A theory of reparameterizations for AdS_3 gravity”. In: (2018).

[36] Jordan S. Cotler et al. “Black Holes and Random Matrices”. In: (2016).

[37] Stéphane d’Ascoli et al. “Double trouble in double descent: Bias and variance (s) in the lazy regime”. In: *International Conference on Machine Learning.* PMLR. 2020, pp. 2280–2290.

[38] Suchetan Das and Bobby Ezhuthachan. “Modular Hamiltonians and large diffeomorphisms in AdS_3”. In: *JHEP* 12 (2018), p. 096. DOI: `10.1007/JHEP12(2018)096`.

[39] Bryce S. DeWitt. “Quantum Theory of Gravity. 1. The Canonical Theory”. In: *Phys. Rev.* 160 (1967), pp. 1113–1148. DOI: `10.1103/PhysRev.160.1113`.

[40] Paul A. M. Dirac. “Gauge invariant formulation of quantum electrodynamics”. In: *Can. J. Phys.* 33 (1955), p. 650. DOI: `10.1139/p55-081`.

[41] William Donnelly and Steven B. Giddings. “Diffeomorphism-invariant observables and their nonlocal algebra”. In: *Phys. Rev.* D93.2 (2016). [Erratum: Phys. Rev.D94,no.2,029903(2016)], p. 024030. DOI: `10.1103/PhysRevD.94.029903,10.1103/PhysRevD.93.024030`.

[42] Ethan Dyer and Guy Gur-Ari. “Asymptotics of Wide Networks from Feynman Diagrams”. In: *International Conference on Learning Representations.* 2020. URL: `https://openreview.net/forum?id=S1gFvANKDS`.

[43] B. Enquist and K. Niklas. “Invariant scaling relations across tree-dominated communities”. In: *Nature* 410 (2001), pp. 655–660. DOI: `10.1038/35070500`.

[44] Elena Facco et al. “Estimating the intrinsic dimension of datasets by a minimal neighborhood information”. In: *Scientific Reports* 7 (Dec. 2017). DOI: `10.1038/s41598-017-11873-y`.

[45] William Fedus, Barret Zoph, and Noam Shazeer. *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity*. 2021.

[46] JC Ferreira and VA Menegatto. “Eigenvalues of integral operators defined by smooth positive definite kernels”. In: *Integral Equations and Operator Theory* 64.1 (2009), pp. 61–81.

[47] A. Liam Fitzpatrick and Jared Kaplan. “Conformal Blocks Beyond the Semi-Classical Limit”. In: (2015).

[48] A. Liam Fitzpatrick, Jared Kaplan, and Matthew T. Walters. “Universality of Long-Distance AdS Physics from the CFT Bootstrap”. In: *JHEP* 1408 (2014), p. 145. DOI: `10.1007/JHEP08(2014)145`.

[49] A. Liam Fitzpatrick et al. “Exact Virasoro Blocks from Wilson Lines and Background-Independent Operators”. In: (2016).

[50] A. Liam Fitzpatrick et al. “On Information Loss in $\text{AdS}_3/\text{CFT}_2$”. In: (2016).

[51] International Transport Forum. *Managing the Transition to Driverless Road Freight Transport*. 2017. URL: `https://www.itf-oecd.org/managing-transition-driverless-road-freight-transport`.

[52] Adrià Garriga-Alonso, Laurence Aitchison, and Carl Edward Rasmussen. “Deep convolutional networks as shallow Gaussian processes”. In: *International Conference on Learning Representations*. 2019.

[53] Mario Geiger et al. “Scaling description of generalization with number of parameters in deep learning”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2020.2 (2020), p. 023401.

[54] Federica Gerace et al. “Generalisation error in learning with random features and the hidden manifold model”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3452–3462.

[55] Paul H. Ginsparg. “Applied Conformal Field Theory”. In: (1988).

[56] Gabriel Goh. “Why Momentum Really Works”. In: *Distill* (2017). DOI: `10.23915/distill.00006`. URL: `http://distill.pub/2017/momentum`.

[57] Will Grathwohl et al. “Your classifier is secretly an energy based model and you should treat it like one”. In: *International Conference on Learning Representations*. 2020. URL: `https://openreview.net/forum?id=Hkxzx0NtDB`.

[58] Roger Grosse. *University of Toronto CSC2541 Winter 2021 Neural Net Training Dynamics, Lecture Notes*. 2021. URL: `https://www.cs.toronto.edu/~rgrosse/courses/csc2541_2021`.

[59] Monica Guica. “Bulk fields from the boundary OPE”. In: (2016).

[60] Monica Guica and Daniel L. Jafferis. "On the construction of charged operators inside an eternal black hole". In: (2015).

[61] Alex Hamilton et al. "Holographic representation of local bulk operators". In: *Phys. Rev.* D74 (2006), p. 066009. DOI: `10.1103/PhysRevD.74.066009`.

[62] Boris Hanin and Mihai Nica. "Finite Depth and Width Corrections to the Neural Tangent Kernel". In: *International Conference on Learning Representations*. 2020. URL: `https://openreview.net/forum?id=SJgndT4KwB`.

[63] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[64] Trevor Hastie et al. "Surprises in high-dimensional ridgeless least squares interpolation". In: *arXiv preprint arXiv:1903.08560* (2019).

[65] S.W. Hawking. "Particle creation by black holes." In: *Commun.Math. Phys.* 43 (1975). DOI: `10.1007/BF02345020`. URL: `https://link.springer.com/article/10.1007/BF02345020`.

[66] Jonathan Heek et al. *Flax: A neural network library and ecosystem for JAX*. Version 0.3.0. 2020. URL: `http://github.com/google/flax`.

[67] Tom Henighan et al. "Scaling Laws for Autoregressive Generative Modeling". In: *arXiv preprint arXiv:2010.14701* (2020).

[68] Joel Hestness, Newsha Ardalani, and Gregory Diamos. "Beyond human-level accuracy: computational challenges in deep learning". In: *Proceedings of the 24th Symposium on Principles and Practice of Parallel Programming*. 2019, pp. 1–14.

[69] Joel Hestness et al. *Deep Learning Scaling is Predictable, Empirically*. 2017.

[70] Joel Hestness et al. "Deep learning scaling is predictable, empirically". In: *arXiv preprint arXiv:1712.00409* (2017).

[71] Markus Heusler. *Black Hole Uniqueness Theorems*. Cambridge Lecture Notes in Physics. Cambridge University Press, 1996. DOI: `10.1017/CBO9780511661396`.

[72] Wei Huang et al. "Implicit bias of deep linear networks in the large learning rate phase". In: *arXiv preprint arXiv:2011.12547* (2020).

[73] Marcus Hutter. "Learning Curve Theory". In: *arXiv preprint arXiv:2102.04074* (2021).

[74] Andrew Ilyas et al. "Adversarial Examples Are Not Bugs, They Are Features". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 125–136. URL: `http://papers.nips.cc/paper/8307-adversarial-examples-are-%20not-bugs-they-are-features.pdf`.

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

[76] Daniel Kabat and Gilad Lifschytz. "CFT representation of interacting bulk gauge fields in AdS". In: *Phys. Rev.* D87.8 (2013), p. 086004. DOI: 10.1103/PhysRevD.87.086004.

[77] Daniel Kabat and Gilad Lifschytz. "Decoding the hologram: Scalar fields interacting with gravity". In: *Phys. Rev.* D89.6 (2014), p. 066010. DOI: 10.1103/PhysRevD.89.066010.

[78] Daniel Kabat and Gilad Lifschytz. "Locality, bulk equations of motion and the conformal bootstrap". In: *JHEP* 10 (2016), p. 091. DOI: 10.1007/JHEP10(2016)091.

[79] Daniel Kabat, Gilad Lifschytz, and David A. Lowe. "Constructing local bulk observables in interacting AdS/CFT". In: *Phys. Rev.* D83 (2011), p. 106009. DOI: 10.1103/PhysRevD.83.106009.

[80] Jared Kaplan. *Lectures on AdS/CFT from the Bottom Up.* URL: https://sites.krieger.jhu.edu/jared-kaplan/writing/.

[81] Jared Kaplan et al. *Scaling Laws for Neural Language Models.* 2020.

[82] Esko Keski-Vakkuri. "Bulk and boundary dynamics in BTZ black holes". In: *Phys. Rev.* D59 (1999), p. 104001. DOI: 10.1103/PhysRevD.59.104001.

[83] Young-Bum Kim. *The Scalable Neural Architecture behind Alexa's Ability to Select Skills.* 2018. URL: https://www.amazon.science/blog/the-scalable-neural-architecture-behind-alexas-ability-to-select-skills.

[84] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization.* 2014.

[85] Alexander Kolesnikov et al. "Big transfer (bit): General visual representation learning". In: *arXiv preprint arXiv:1912.11370* 6.2 (2019), p. 8.

[86] Simon Kornblith, Jonathon Shlens, and Quoc V Le. "Do better imagenet models transfer better?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2019, pp. 2661–2671.

[87] Filip Kos et al. "Precision islands in the Ising and $O(N)$ models". In: *Journal of High Energy Physics volume 2016* 2016 (2016). DOI: 10.1007/JHEP08(2016)036. URL: https://link.springer.com/article/10.1007/JHEP08(2016)036#citeas.

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

[89] Thomas Kühn. "Eigenvalues of integral operators with smooth positive definite kernels". In: *Archiv der Mathematik* 49.6 (1987), pp. 525–534.

[90] David de Laat. "Approximating manifolds by meshes: asymptotic bounds in higher codimension". In: *Master's Thesis, University of Groningen, Groningen* (2011).

[91] Yann LeCun and Corinna Cortes. "MNIST handwritten digit database". In: (2010). URL: http://yann.lecun.com/exdb/mnist/.

[92] Jaehoon Lee et al. “Deep Neural Networks as Gaussian Processes”. In: *International Conference on Learning Representations*. 2018.

[93] Jaehoon Lee et al. “Finite Versus Infinite Neural Networks: an Empirical Study”. In: *Advances in Neural Information Processing Systems* 33 (2020).

[94] Jaehoon Lee et al. “Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent”. In: *Advances in Neural Information Processing Systems*. 2019.

[95] Elizaveta Levina and Peter J Bickel. “Maximum likelihood estimation of intrinsic dimension”. In: *Advances in neural information processing systems*. 2005, pp. 777–784.

[96] Aitor Lewkowycz, Gustavo J. Turiaci, and Herman Verlinde. “A CFT Perspective on Gravitational Dressing and Bulk Locality”. In: (2016).

[97] Aitor Lewkowycz et al. “The large learning rate phase of deep learning: the catapult mechanism”. In: *arXiv preprint arXiv:2003.02218* (2020).

[98] Chunyuan Li et al. *Measuring the Intrinsic Dimension of Objective Landscapes*. 2018.

[99] Peter J. Liu et al. “Generating Wikipedia by Summarizing Long Sequences”. en. In: *arXiv:1801.10198 [cs]* (2018). URL: `http://arxiv.org/abs/1801.10198`.

[100] Ilya Loshchilov and Frank Hutter. “Sgdr: Stochastic gradient descent with warm restarts”. In: *arXiv preprint arXiv:1608.03983* (2016).

[101] Andreas Loukas. “How close are the eigenvectors of the sample and actual covariance matrices?” In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2228–2237.

[102] Xingjun Ma et al. *Dimensionality-Driven Learning with Noisy Labels*. 2018.

[103] Juan Maldacena. “The Large-N Limit of Superconformal Field Theories and Supergravity”. In: *International Journal of Theoretical Physics* 38 (1999), pp. 1113–1133. DOI: `10.1023/A:1026654312961`.

[104] Dörthe Malzahn and Manfred Opper. “A Variational Approach to Learning Curves”. In: *Advances in Neural Information Processing Systems*. Ed. by T. Dietterich, S. Becker, and Z. Ghahramani. Vol. 14. MIT Press, 2002, pp. 463–469. URL: `https://proceedings.neurips.cc/paper/2001/file/26f5bd4aa64fdadf96152ca6e6408068-Paper.pdf`.

[105] Dörthe Malzahn and Manfred Opper. “Learning curves and bootstrap estimates for inference with Gaussian processes: A statistical mechanics study”. In: *Complexity* 8.4 (2003), pp. 57–63.

[106] Dörthe Malzahn and Manfred Opper. “Learning curves for Gaussian processes regression: A framework for good approximations”. In: *Advances in neural information processing systems* (2001), pp. 273–279.

[107] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: `https://www.tensorflow.org/`.

[108] Alexander G. de G. Matthews et al. "Gaussian Process Behaviour in Wide Deep Neural Networks". In: *International Conference on Learning Representations*. 2018.

[109] Henry Maxfield. "A view of the bulk from the worldline". In: (2017).

[110] P McCullagh and John A Nelder. *Generalized Linear Models*. Vol. 37. CRC Press, 1989.

[111] Song Mei and Andrea Montanari. "The generalization error of random features regression: Precise asymptotics and double descent curve". In: *arXiv preprint arXiv:1908.05355* (2019).

[112] Charles W. Misner, Kip S. Thorne, and John Archibald Wheeler. *Gravitation*. New York, NY: Princeton University Press, 2017. ISBN: 9780691177793.

[113] Guido F Montufar et al. "On the number of linear regions of deep neural networks". In: *Advances in neural information processing systems*. 2014, pp. 2924–2932.

[114] Yu Nakayama. "Scale invariance vs conformal invariance". In: *Physics Reports* 569 (2015). Scale invariance vs conformal invariance, pp. 1–93. ISSN: 0370-1573. DOI: `https://doi.org/10.1016/j.physrep.2014.12.003`. URL: `https://www.sciencedirect.com/science/article/pii/S0370157314004499`.

[115] Yu Nakayama and Hirosi Ooguri. "Bulk Local States and Crosscaps in Holographic CFT". In: *JHEP* 10 (2016), p. 085. DOI: `10.1007/JHEP10(2016)085`.

[116] Yu Nakayama and Hirosi Ooguri. "Bulk Locality and Boundary Creating Operators". In: *JHEP* 10 (2015), p. 114. DOI: `10.1007/JHEP10(2015)114`.

[117] Preetum Nakkiran. "More data can hurt for linear regression: Sample-wise double descent". In: *arXiv preprint arXiv:1912.07242* (2019).

[118] Preetum Nakkiran and Yamini Bansal. "Distributional Generalization: A New Kind of Generalization". In: *arXiv preprint arXiv:2009.08092* (2020).

[119] Radford M. Neal. "Bayesian Learning for Neural Networks". PhD thesis. University of Toronto, Dept. of Computer Science, 1994.

[120] Yuval Netzer et al. "Reading digits in natural images with unsupervised feature learning". In: (2011).

[121] Roman Novak et al. "Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes". In: *International Conference on Learning Representations*. 2019.

[122] Roman Novak et al. "Neural Tangents: Fast and Easy Infinite Neural Networks in Python". In: *International Conference on Learning Representations*. 2020. URL: `https://github.com/google/neural-tangents`.

[123] Chris Olah et al. “Zoom In: An Introduction to Circuits”. In: *Distill* (2020). https://distill.pub/2020/circu in. DOI: `10.23915/distill.00024.001`.

[124] Giorgio Parisi. “A sequence of approximated solutions to the SK model for spin glasses”. In: *Journal of Physics A: Mathematical and General* 13.4 (1980), p. L115.

[125] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 8026–8037.

[126] Miguel F. Paulos et al. “The S-matrix Bootstrap I: QFT in AdS”. In: (2016).

[127] Alec Radford et al. “Improving language understanding by generative pre-training”. In: *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/lan understanding paper. pdf* (2018).

[128] Alec Radford et al. “Language Models are Unsupervised Multitask Learners”. In: *openai.com* (2019).

[129] Ali Rahimi and Benjamin Recht. “Weighted sums of random kitchen sinks: replacing minimization with randomization in learning.” In: *Nips*. Citeseer. 2008, pp. 1313–1320.

[130] JB Reade. “Eigenvalues of positive definite kernels”. In: *SIAM Journal on Mathematical Analysis* 14.1 (1983), pp. 152–157.

[131] Ryan M Rifkin and Ross A Lippert. *Notes on regularized least squares*. 2007.

[132] Sam Ritchie, Ambrose Slone, and Vinay Ramasesh. “Caliban: Docker-based job manager for reproducible workflows”. In: *Journal of Open Source Software* 5.53 (2020), p. 2403. DOI: `10.21105/joss.02403`. URL: `https://doi.org/10.21105/joss.02403`.

[133] Matthew M. Roberts. “Time evolution of entanglement entropy from a pulse”. In: *JHEP* 12 (2012), p. 027. DOI: `10.1007/JHEP12(2012)027`.

[134] David Roodman. *Modeling the Human Trajectory*. 2020. URL: `https://www.openphilanthropy.org/blog/modeling-human-trajectory`.

[135] Jonathan S. Rosenfeld et al. *A Constructive Prediction of the Generalization Error Across Scales*. 2019.

[136] Jonathan S. Rosenfeld et al. “A Constructive Prediction of the Generalization Error Across Scales”. In: *International Conference on Learning Representations*. 2020.

[137] Jonathan S. Rosenfeld et al. “On the Predictability of Pruning Across Scales”. In: *arXiv preprint arXiv:2006.10621* (2020).

[138] Samuel S Schoenholz et al. “Deep Information Propagation”. In: *International Conference on Learning Representations* (2017).

[139] Klaus Schwab. *The Fourth Industrial Revolution: what it means, how to respond*. 2016. URL: `https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/`.

[140] Vaishaal Shankar et al. "Neural Kernels Without Tangents". In: *International Conference on Machine Learning*. 2020.

[141] Utkarsh Sharma and Jared Kaplan. "A Neural Scaling Law from the Dimension of the Data Manifold". In: *arXiv preprint arXiv:2004.10802* (2020).

[142] Peter Sollich. "Learning curves for Gaussian processes". In: *Proceedings of the 11th International Conference on Neural Information Processing Systems*. 1998, pp. 344–350.

[143] Peter Sollich and Anason Halees. "Learning curves for Gaussian process regression: Approximations and bounds". In: *Neural computation* 14.6 (2002), pp. 1393–1428.

[144] Stefano Spigler, Mario Geiger, and Matthieu Wyart. *Asymptotic learning curves of kernel methods: empirical data v.s. Teacher-Student paradigm*. 2019.

[145] Stefano Spigler, Mario Geiger, and Matthieu Wyart. "Asymptotic learning curves of kernel methods: empirical data versus teacher–student paradigm". In: *Journal of Statistical Mechanics: Theory and Experiment* 2020.12 (2020), p. 124001.

[146] Michael L Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media, 1999.

[147] Yaniv Taigman et al. "DeepFace: Closing the Gap to Human-Level Performance in Face Verification". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2014).

[148] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 6105–6114.

[149] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *CoRR* abs/1905.11946 (2019). URL: `http://arxiv.org/abs/1905.11946`.

[150] David Tong. *Lectures on Statistical Field Theory*. URL: `https://www.damtp.cam.ac.uk/user/tong/sft.html`.

[151] Matthew J Urry and Peter Sollich. "Replica theory for learning curves for Gaussian processes on random graphs". In: *Journal of Physics A: Mathematical and Theoretical* 45.42 (2012), p. 425005.

[152] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 5998–6008. URL: `http://papers.nips.cc/paper/7181-attention-is-all-you-%20need.pdf`.

[153] W. Wagner and A. Pruß. "The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use". In: *Journal of Physical and Chemical Reference Data* 31 (2002), p. 387. DOI: `10.1063/1.1461829`.

[154] Larry Wasserman. *All of nonparametric statistics.* Springer Science & Business Media, 2006.

[155] Geoffrey West and Jim Brown. "Life's Universal Scaling Laws". In: *Physics Today* 57 (2004). DOI: 10.1063/1.1809090.

[156] Geoffrey B. West, James H. Brown, and Brian J. Enquist. "A General Model for the Origin of Allometric Scaling Laws in Biology". In: *Science* 276.5309 (1997), pp. 122–126. ISSN: 0036-8075. DOI: 10.1126/science.276.5309.122. URL: https://science.sciencemag.org/content/276/5309/122.

[157] Hermann Weyl. "Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung)". In: *Mathematische Annalen* 71.4 (1912), pp. 441–479.

[158] Christopher KI Williams and Francesco Vivarelli. "Upper and lower bounds on the learning curve for Gaussian processes". In: *Machine Learning* 40.1 (2000), pp. 77–102.

[159] Kenneth G. Wilson. "Renormalization Group and Critical Phenomena. I. Renormalization Group and the Kadanoff Scaling Picture". In: *Phys. Rev. B* 4 (9 Nov. 1971), pp. 3174–3183. DOI: 10.1103/PhysRevB.4.3174. URL: https://link.aps.org/doi/10.1103/PhysRevB.4.3174.

[160] Kenneth G. Wilson. "Renormalization Group and Critical Phenomena. II. Phase-Space Cell Analysis of Critical Behavior". In: *Phys. Rev. B* 4 (9 Nov. 1971), pp. 3184–3205. DOI: 10.1103/PhysRevB.4.3184. URL: https://link.aps.org/doi/10.1103/PhysRevB.4.3184.

[161] Peter Woit. *Not Even Wrong: The Failure of String Theory and the Search for Unity in Physical Law for Unity in Physical Law.* Basic Books. ISBN: 9780465003631.

[162] Han Xiao, Kashif Rasul, and Roland Vollgraf. "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms". In: (Aug. 2017).

[163] Lechao Xiao et al. "Dynamical Isometry and a Mean Field Theory of CNNs: How to Train 10,000-Layer Vanilla Convolutional Neural Networks". In: *International Conference on Machine Learning.* 2018.

[164] Sho Yaida. "Non-Gaussian processes and neural networks at finite widths". In: *Mathematical and Scientific Machine Learning Conference.* 2020.

[165] Greg Yang. "Scaling Limits of Wide Neural Networks with Weight Sharing: Gaussian Process Behavior, Gradient Independence, and Neural Tangent Kernel Derivation". In: *arXiv preprint arXiv:1902.04760* (2019).

[166] Sergey Zagoruyko and Nikos Komodakis. "Wide residual networks". In: *British Machine Vision Conference.* 2016.

[167] A. Zamolodchikov. "Conformal Symmetry in Two-dimensional Space: Recursion Representation of the Conformal Block". In: *Teoreticheskaya i Matematicheskaya Fizika* 73 (1987), pp. 103–110.

[168] A.B. Zamolodchikov. “Conformal Symmetry in Two-Dimensions: An Explicit Recurrence Formula for the Conformal Partial Wave Amplitude”. In: *Commun.Math.Phys.* 96 (1984), pp. 419–422. DOI: `10.1007/BF01214585`.