

# Neural Scaling Laws Rooted in the Data Distribution

Ari Brill<sup>1</sup>

## Abstract

Deep neural networks exhibit empirical neural scaling laws, with error decreasing as a power law with increasing model or data size, across a wide variety of architectures, tasks, and datasets. This universality suggests that scaling laws may result from general properties of natural learning tasks. We develop a mathematical model intended to describe natural datasets using percolation theory. Two distinct criticality regimes emerge, each yielding optimal power-law neural scaling laws. These regimes, corresponding to power-law-distributed discrete subtasks and a dominant data manifold, can be associated with previously proposed theories of neural scaling, thereby grounding and unifying prior works. We test the theory by training regression models on toy datasets derived from percolation theory simulations. We suggest directions for quantitatively predicting language model scaling.

## 1. Introduction

Deep neural networks leverage massive amounts of computation to achieve superb performance on real-world tasks. In particular, neural scaling laws are the empirical finding that deep neural networks, including large language models (LLMs), perform predictably better as the model or dataset size increases, with the test error dropping as a power law over many orders of magnitude (Hestness et al., 2017; Kaplan et al., 2020; Hoffmann et al., 2022; Bachmann et al., 2024). Neural scaling appears to be a universal phenomenon, with power-law exponents often of order  $\sim 0.1$  observed across a wide variety of architectures, tasks, and datasets (Henighan et al., 2020; Ghorbani et al., 2021; Jones, 2021; Alabdulmohsin et al., 2022; 2024).

It is unclear why neural scaling laws exist. One influential theoretical approach treats neural networks as approximating a continuous function defined on a manifold with low intrinsic dimension relative to the input space (Bahri et al.,

2024; Sharma & Kaplan, 2022). Increasing the model or dataset size allows for increasingly fine piecewise function approximation, producing power-law scaling with an exponent inversely proportional to the data manifold dimension.

On the other hand, deep neural networks do more than approximate functions. They learn representations, constructing a hierarchy of abstract explanatory features that reflect the data’s underlying structure (Bengio et al., 2013). If a neural network achieves good performance by learning good features, it stands to reason that a scaled-up network that achieves even better performance may do so by learning more or better features. Indeed, works in mechanistic interpretability have empirically studied how scaling laws may connect to specific model behaviors and circuits (Hernandez et al., 2022; Olsson et al., 2022; Michaud et al., 2024).

Several models have been proposed connecting neural scaling laws to power-law feature distributions. Hutter (2021) studied a toy model of learning, showing that a dataset of Zipf-distributed discrete features can yield power-law data scaling with a range of exponents. In kernel regression, power-law kernel spectra can produce power-law scaling laws (Spigler et al., 2020; Bordelon et al., 2020; Bahri et al., 2024). Maloney et al. (2022) considered a generative data model based on a set of latent features with power-law spectral structure, and computed the resulting scaling laws in the context of a random feature model. Notably, Michaud et al. (2024) conjectured that natural prediction problems involve discrete subtasks or quanta, which can be ordered into a *Q Sequence* by frequency of usefulness. By assuming that the use frequencies are power-law-distributed and models learn quanta in the order of the *Q Sequence*, power-law neural scaling laws can be derived.

These prior works show that power-law dataset structure can produce power-law neural scaling laws, but they do not explain why this structure would emerge across disparate domains. Unifying manifold-approximation and feature-learning theories of power-law scaling, and understanding how data distributions bound scaling exponents, have been identified as key research questions to resolve foundational challenges for assuring the alignment and safety of LLMs (Anwar et al., 2024).

In this work, we propose a model of emergent power-law dataset structure that yields power-law neural scaling laws.

<sup>1</sup>Independent. Correspondence to: Ari Brill <aryeh.brill@gmail.com>.

We make two key assumptions meant to describe natural datasets, *context-dependent target function* and *general-purpose learning*, discussed in Sec. 2. In Sec. 3, we introduce percolation theory and use it to translate these assumptions into a mathematical model of dataset structure.

Two scaling regimes emerge that integrate and recontextualize previously proposed theoretical models of neural scaling laws. Below a critical threshold, the data distribution consists of a power-law distribution of subtasks, which we identify with the quanta proposed by Michaud et al. (2024). Each quantum corresponds to a low-dimensional structure in data space. Above the critical threshold, a single structure dominates the data distribution, and we identify this regime with the manifold-approximation model proposed by Sharma & Kaplan (2022).

In Sec. 4, we derive theoretical scaling laws in model and data size, and perform experimental tests in Sec. 5. We discuss implications of the theory in Sec. 6, highlighting progress toward quantitatively predicting LLM scaling laws.

## 2. Definitions and Assumptions

### 2.1. Data Space

We consider supervised regression with inputs  $X \subseteq \mathbb{R}^{d'}$ , target labels  $Y \subseteq \mathbb{R}$ , and mean squared error loss  $\mathcal{L}$ . The inputs and labels are sampled uniformly at random.

Real-world data often have physical invariances reflecting underlying symmetries of the physical generating process (Lin et al., 2017; Roberts, 2021). For example, image data are subject to locality and translation invariance. Encoding such invariances into the model can improve sample complexity and reduce the data’s effective dimension (Tahmasebi & Jegelka, 2023). We accordingly define the *data space*  $\mathcal{X}$  with dimension  $d \leq d'$  such that each axis represents one of the  $d$  contingent, empirical degrees of freedom remaining after accounting for any symmetries or invariances. The domain of each dimension of  $\mathcal{X}$  is the domain of its corresponding input degree of freedom.

In practice, continuous inputs can be distinguished only above a precision cutoff. This might be set by sensor resolution, human perception, or floating-point numerical precision. The number of distinct values along each data space dimension determines a characteristic length scale  $L$ . We approximate  $\mathcal{X}$  as a hypercubic lattice with  $L^d$  discrete elements, and discretize the labels  $Y$  similarly. We denote discrete data-space elements as  $x \in \mathcal{X}$ , with corresponding labels  $y \in Y$ . For convenience, we’ll write  $f(x)$  for a function evaluated at any point corresponding to  $x$  in the  $(d' - d)$ -dimensional manifold in the input space  $X$ .

### 2.2. Context-Dependent Target Function

Modeling real-world data often requires different behaviors in different contexts. A language model might compose a sonnet when prompted with Shakespeare, write Python code given a software specification, and generate moves for a grandmaster-level chess game. In general, a useful target function could as well correspond to an accumulation of disparate behaviors as to a unified concept (Minsky, 1988).

We model a context-dependent target function as a higher-order target function  $HOF$  that generates first-order functions, which then return the target labels. That is,  $HOF$  generates first-order functions  $f_x : \mathcal{X} \rightarrow Y$  such that  $HOF(x) = f_x$  and  $f_x(x) = y$ . Each of the functions  $f_x$  is assumed to be Lipschitz continuous on its domain, following Sharma & Kaplan (2022).

Factorizing the target function this way splits the learning task into two parts. The functions  $f_x$  determine the modeling difficulty of subtasks, while  $HOF$  controls the difficulty of generalizing among subtasks. For example, a natural language dataset comprising multiple languages might be expected to have an  $HOF$  with a greater rate of change than one composed of a single language.

### 2.3. General-Purpose Learning

A target function corresponding to a real-world task has latent structure rooted in the real, outside world. Reflecting reality, this structure is arbitrarily complex, contingent, and *a priori* unknown. Reality’s apparently unbounded complexity is one reason why AI systems applying general, scalable learning algorithms have outperformed methods reliant on built-in human knowledge (Sutton, 2019). The input representation used by a general-purpose learning system without built-in task-specific knowledge is essentially arbitrary, since by definition it’s unrelated to the target function’s extrinsic structure.

We assume a general-purpose learning task. Because the input representation is arbitrary, we’ll model the data distribution statistically by assuming that the input representation is random. We elaborate on the conceptual underpinnings of the general-purpose learning assumption and the kinds of datasets to which it’s meant to apply in Appendix A.

## 3. Percolation Model

### 3.1. Setup

We draw a *bond* between each pair of adjacent data-space elements  $x_i, x_j \in \mathcal{X}$  if they require interchangeable functional behavior, that is, if  $f_i(x_i) = f_j(x_i)$  and  $f_i(x_j) = f_j(x_j)$ . Let  $p$  be the fraction of adjacent pairs connected by a bond. A group of connected elements forms a cluster. An element with no bonds is *out of distribution*, because

its function can't be learned via generalization, only memorized. Elements with bonds are *in distribution*.

Due to general-purpose learning, the bond pattern's projection in data space can be modeled as random. We analyze the dataset's overall statistical properties by considering a hypercubic lattice with elements occupied at random with probability  $p$ . We'll do so using percolation theory, which has been applied in fields including physics, materials science, network theory, biology, hydrology, and geochemistry (Sahimi, 1994). Sec. 3.2 reviews foundational results in percolation theory, which can be found in an introductory textbook, such as Stauffer & Aharony (1994).

### 3.2. Percolation Theory

Percolation theory concerns the statistical and structural properties of clusters of randomly occupied units on a lattice or network. These systems exhibit phase transitions about a critical occupation probability  $p_c$ , called the percolation threshold. When  $p < p_c$ , clusters are finite and disconnected, while for  $p \geq p_c$ , the system *percolates*: a so-called infinite cluster emerges that scales with the size of the system, its size going to infinity in an infinite lattice. The numerical value of  $p_c$  depends on details of the system.

The randomly occupied units of percolation can be either lattice sites (site percolation) or bonds between neighboring sites (bond percolation). In either case, a cluster consists of a group of occupied units connected by a chain of nearest-neighbor relations, and a cluster's size  $s$  is its total number of units. Fig. 1 illustrates site percolation on a 3D lattice. Bond and site percolation are closely related, and all results here are valid in either model<sup>1</sup>. Following Stauffer & Aharony (1994), we'll use the language of site percolation.

On a large  $d$ -dimensional hypercubic lattice of size  $L$ , the number of finite clusters of size  $s$  per lattice site,  $n_s$ , can be approximated for large  $s$  as

$$n_s \propto s^{-\tau} e^{-as}, \quad (1)$$

where  $a \propto |p - p_c|^{1/\sigma}$  and  $\tau$  and  $\sigma$  are universal critical exponents that depend only on  $d$ . At the percolation threshold  $p = p_c$ , the cluster size distribution is a power law,  $n_s \propto s^{-\tau}$ . Away from  $p_c$ , the size distribution is cut off exponentially, with most occupied sites instead belonging to the infinite cluster when  $p > p_c$ .

Clusters have nontrivial geometric structure. Finite clusters are fractal objects characterized by an intrinsic dimension  $D < d$ . Specifically, if an  $s$ -cluster's linear extent is de-

scribed by its radius of gyration  $R_s$ , then  $R_s \propto s^{1/D}$ . At the percolation threshold, the incipient infinite cluster is also a fractal object with  $R_s \propto s^{1/D}$ , transitioning to Euclidean geometry with  $R_s \propto s^{1/d}$  for  $p \gg p_c$ .

A realistic data space is likely high-dimensional. Percolation on a Bethe lattice, an infinite tree in which every site has the same number of edges, provides an accurate model of percolation on a lattice with any  $d \geq 6$ . In this case,  $\tau = 5/2$ ,  $\sigma = 1/2$ , and  $D = 4$ .

## 4. Neural Scaling Laws

### 4.1. Emergent Quanta at the Percolation Threshold

At the percolation threshold  $p = p_c$ , the probability that a site belongs to a cluster of size  $s$  is given by

$$p(s) = \frac{n_s s}{\int_1^\infty n_s s ds} = \frac{s^{-\tau} s}{\int_1^\infty s^{-\tau} s ds} = (\tau - 2) s^{-(\tau-1)}. \quad (2)$$

The probability that a site belongs to a cluster of size  $s$  or larger is then given by

$$P(s) = \int_s^\infty (\tau - 2) s'^{-(\tau-1)} ds' = s^{-(\tau-2)}. \quad (3)$$

We assign each cluster a rank  $k$  based on its size, with  $k = 1$  indicating the largest cluster,  $k = 2$  the next largest, and so on. To do so, we use the fact that the rank distribution is directly proportional to the cumulative distribution function,  $k(s) \propto P(s)$  (Newman, 2005). By inverting Eq. 3, we obtain a Zipf power-law distribution

$$s \propto k^{-\frac{1}{\tau-2}}. \quad (4)$$

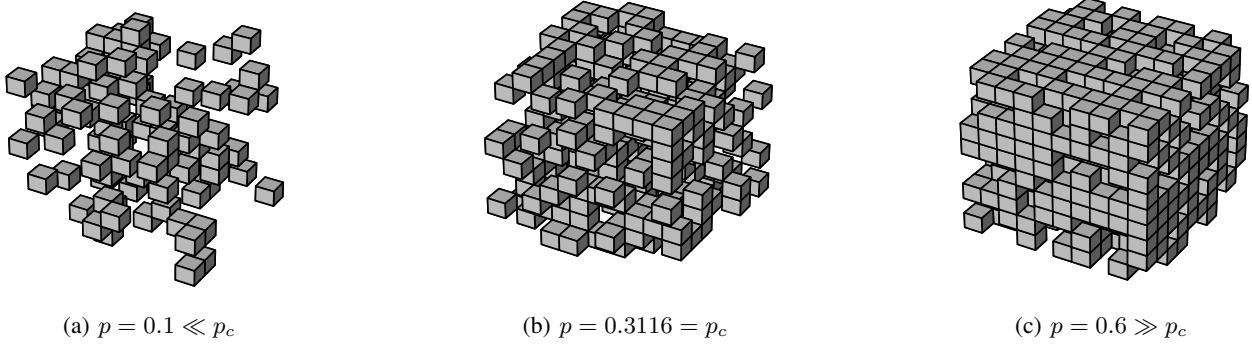
We show in Appendix B that each cluster contributes to the baseline loss approximately in proportion to that cluster's size,  $\Delta \mathcal{L}_s \propto s$ . We have therefore derived a *Q Sequence*, as proposed by Michaud et al. (2024). Each cluster's function defines a "quantum" of model behavior to learn. The functions (quanta) form an ordered sequence in which fully learning the  $k^{\text{th}}$  quantum reduces the loss such that

$$-\Delta \mathcal{L}_k \propto k^{-(\alpha+1)}, \quad (5)$$

with exponent  $\alpha > 0$ . Equating Eqs. 4 and 5 gives  $\alpha = (3 - \tau)/(\tau - 2)$ . For  $d \geq 6$ ,  $\tau = 5/2$ , yielding  $\alpha = 1$ .

Following Michaud et al. (2024), throughout this work we will sometimes use the terms "quantum" and "Q Sequence" to refer to a function defined on a discrete cluster and to the sequence of quanta ordered by cluster size, respectively.

<sup>1</sup>Sec 3.1 described bonds between lattice sites. In the equivalent site percolation description, data-space elements are in-distribution with probability  $p$ , and groups of adjacent in-distribution elements form same-function clusters.


 Figure 1. Visualization of site percolation on a  $10 \times 10 \times 10$  lattice.

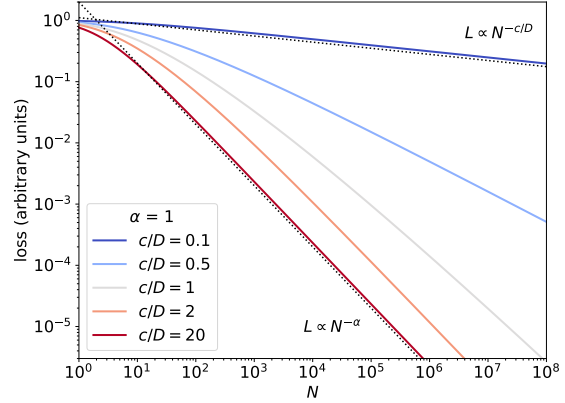
#### 4.2. Model Scaling

We now study scaling in model size. We consider resolution-limited scaling (Bahri et al., 2024), taking the dataset size and computational resources to be effectively infinite. We assume that the model is nonparametric and characterized by  $N$  independent effective degrees of freedom (DOF). Empirical scaling laws are typically reported in terms of parameter count  $P$ . The correct way to convert between DOF and parameters may be architecture-dependent. We discuss this issue in Appendix C. We report scaling laws in terms of  $N$  and leave open the conversion to  $P$  for future investigation.

In nonparametric regression and density estimation, the error with respect to DOF (or data samples) scales exponentially in the input dimension  $d$ ,  $\epsilon \propto N^{-1/\mathcal{O}(d)}$  (Wasserman, 2006). This results from the curse of dimensionality:  $\mathcal{O}((1/l)^d)$  hypercubes are required to piecewise approximate a function by slicing space into hypercubes with side length  $l$ . Furthermore, if the data lie on a low-dimensional manifold of intrinsic dimension  $D < d$ , deep neural networks can achieve much more efficient  $c/D$  scaling, for a constant  $c$  (Chen et al., 2019; Sharma & Kaplan, 2022; Bahri et al., 2024). For a piecewise constant model,  $c \geq 2$ , and for a piecewise linear model such as a ReLU neural network,  $c \geq 4$ . Equality holds for generic Lipschitz continuous functions, with faster scaling possible for non-generically simple functions (Sharma & Kaplan, 2022).

The data distribution at  $p_c$  consists of power-law-distributed clusters each with intrinsic dimension  $D$ . From Eq. 5, each cluster contributes to the loss in proportion to its size. If  $n_k$  of the model’s  $N$  effective DOF are used to approximate the function of the rank- $k$  cluster, then that cluster’s loss contribution further scales as  $n_k^{-c/D}$ . The sum of cluster losses with DOF allocated optimally, accounting for both factors, gives the overall scaling law.

The predicted model scaling law is derived in Appendix D. Fig. 2 shows how it changes as  $c/D$  varies relative to  $\alpha$ .


 Figure 2. Theoretical scaling law as a function of model DOF  $N$ , for  $\alpha = 1$  and various values of  $c/D$ .

Intuitively, we can think about the loss-minimizing model as allocating DOF to model those cluster functions that on the margin most reduce the loss. At first, this is the largest cluster only. As that cluster becomes better modeled, DOF are also assigned to the next-largest cluster, and so on, following the Q Sequence. The optimal DOF allocation is

$$n_k = \begin{cases} ak^{b-1} & k < k_{\text{br}} \\ 0 & k \geq k_{\text{br}}. \end{cases} \quad (6)$$

where  $a$  is a normalization factor, the allocation exponent is

$$b = 1 - \frac{\alpha + 1}{c/D + 1} = \frac{c/D - \alpha}{c/D + 1},$$

and the cluster rank where a break occurs is



$$k_{\text{br}} \approx \begin{cases} N & c/D \gg \alpha, c/D \gtrsim 1 \\ (\ln \frac{N}{\ln N})^{-1} N & c/D \approx \alpha \text{ or } c/D \ll 1, \alpha \ll 1 \\ (|b|N)^{1/(1+|b|)} & c/D \ll \alpha, \alpha \gtrsim 1. \end{cases}$$

The predicted loss is

$$\mathcal{L} \propto \left( \frac{N}{k_{\text{br}}} + \frac{1}{\alpha} \right) k_{\text{br}}^{-\alpha}. \quad (7)$$

In the limiting case where  $c/D \gg \alpha$  and  $c/D \gtrsim 1$ , each cluster function is quickly modeled well. The loss is determined by the number of clusters learned, scaling as

$$\mathcal{L} \sim \left( 1 + \frac{1}{\alpha} \right) N^{-\alpha}. \quad (8)$$

When  $c/D \ll \alpha$  and  $\alpha \gtrsim 1$ , clusters are modeled slowly, and only the functions for a handful of the largest clusters can be learned. For large  $N$ , the loss scales as

$$\mathcal{L} \sim \left( \frac{\alpha}{c/D + 1} \right)^{-(c/D+1)} \left( 1 + \frac{1}{\alpha} N^{-\frac{\alpha}{1+\alpha}} \right) N^{-c/D}. \quad (9)$$

In the critical regime,  $\alpha = 1$  (Sec 4.1). For large clusters when  $d \geq 6$ ,  $D = 4$ . For ReLU activations,  $c = 4$ , so  $c/D = 1$ . A ReLU neural network or equivalent is thus the minimally powerful function approximator for which model scaling begins to transition from the manifold approximation to the Q Sequence regime for a data distribution at criticality.

### 4.3. Data Scaling

We now consider resolution-limited scaling in dataset size  $D$  (i.e. number of training examples). We idealize the model as an optimal approximator, with the finite dataset size bottlenecking performance. While before DOF were distributed to optimize performance, each cluster's expected number of data points is now fixed by its size.

A cluster's size affects its loss in three (opposing) ways. First, larger size means more test data points to model, increasing the loss. Second, larger size means more training data points for modeling its function, decreasing the loss in accordance with manifold approximation scaling. Third, larger size increases the probability that the cluster is represented in the randomly sampled training data, decreasing the loss due to the chance it might not be learned at all. The total loss is the sum of each cluster's loss.

Intuitively, if the cluster functions are relatively easy to model, then the first and third considerations dominate, and

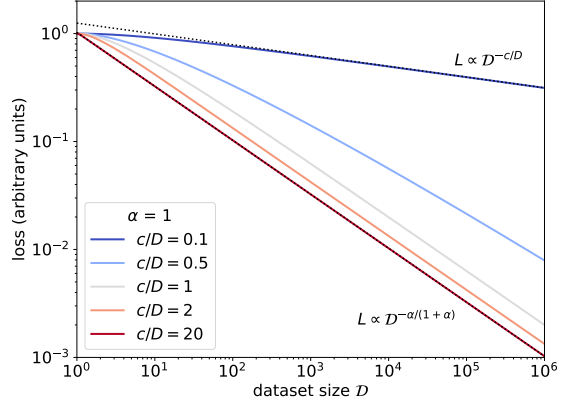


Figure 3. Theoretical scaling law as a function of dataset size  $D$ , for  $\alpha = 1$  and various values of  $c/D$ .

the data scaling law will reflect the cluster size distribution. Conversely, if the functions are relatively hard to model, then the second consideration is most important, and the scaling law will correspond to manifold approximation.

The predicted data scaling law is derived in Appendix E. Fig. 3 shows how it changes as  $c/D$  varies relative to  $\alpha$ . We obtain, up to an overall constant factor,

$$\mathcal{L} \approx \frac{\alpha^{-\alpha/(1+\alpha)-1}}{1 - \frac{\alpha/(1+\alpha)}{c/D}} D^{-\alpha/(1+\alpha)} + \frac{\alpha^{-c/D-1}}{1 - \frac{c/D}{\alpha/(1+\alpha)}} D^{-c/D}, \quad (10)$$

assuming that  $c/D \neq \alpha/(1+\alpha)$ . In the limiting case where  $c/D \gg \alpha/(1+\alpha)$ , the first term dominates and we obtain  $\mathcal{L} \propto D^{-\alpha/(1+\alpha)}$ . In the limiting case where  $\alpha/(1+\alpha) \gg c/D$ , the second term dominates and we obtain  $\mathcal{L} \propto D^{-c/D}$ . If  $c/D = \alpha/(1+\alpha)$ , we obtain

$$\mathcal{L} \approx \alpha^{-c/D-1} \left( 1 + \frac{c}{D} \log \alpha + \frac{c}{D} \log D \right) D^{-c/D}. \quad (11)$$

### 4.4. Subcritical and Supercritical Scaling

In the subcritical regime  $p < p_c$ , the cluster size distribution has an exponential cutoff for large  $s$  (Eq. 1), but the power-law exponent is the same. Scaling should thus be similar to the critical regime  $p = p_c$ . A detailed analysis is left for future work. The supercritical regime  $p > p_c$  is qualitatively different. The infinite cluster dominates the data distribution, and a single function rather than a Q Sequence best describes the learning task. This regime appears to correspond to the manifold approximation model proposed by Sharma & Kaplan (2022). The intrinsic manifold dimension  $D$  can be identified with the infinite cluster's fractal dimension, which is the same as for finite clusters when  $p = p_c$  and becomes Euclidean,  $D = d$ , when  $p \gg p_c$ .

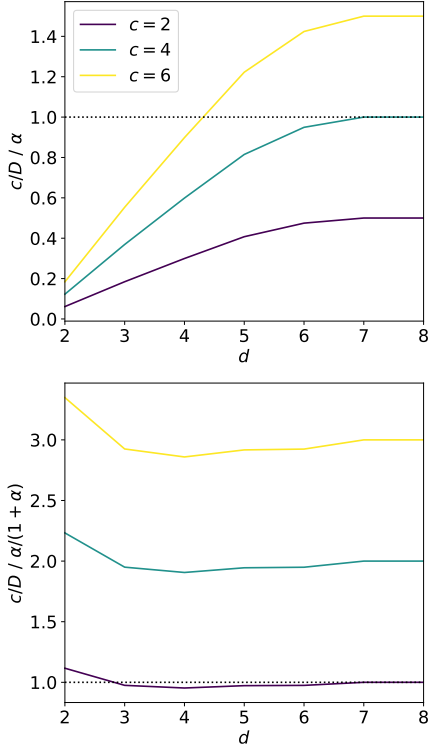


Figure 4. Exponent ratios controlling model scaling (top) and data scaling (bottom), meaningful when  $p \lesssim p_c$ . The overall scaling law regime transitions about a ratio of 1 for both.

#### 4.5. Scaling Regimes

The derived scaling laws have two main free parameters:  $p$ , which relates to the dataset’s degree of context-dependence, and the manifold approximation constant  $c$ , which represents the model’s ability to fit complex functions. In the subcritical regime, for high-dimensional data,  $\alpha$  and  $D$  are fixed by percolation theory, varying only for  $d < 6$ . In the supercritical regime,  $D$  is determined by the dimension of the infinite cluster, which depends on  $p$ .

Fig. 4 shows the ratio of scaling exponents for several values of  $c$ , assuming that  $p \lesssim p_c$ . Model scaling begins to transition from manifold approximation to quanta scaling when  $c = 4$ . For data scaling, the transition begins at  $c = 2$ , suggesting that manifold approximation data scaling should commonly hold for natural datasets only when  $p \gtrsim p_c$ . In Appendix F, we discuss in more detail how the proposed model’s criticality regimes and limiting cases relate to previously proposed theories of neural scaling laws.

### 5. Experiments

We tested the derived model and data scaling laws in a minimal toy setting that allows efficient dataset generation and

model fitting, with more general investigations left for future work. We trained a model based on nearest-neighbor regression on datasets generated from percolation simulations. Because such simulations are inherently random, we generated and fit multiple datasets<sup>2</sup>.

#### 5.1. Dataset

Each dataset was made by simulating percolation clusters at criticality on a Bethe lattice. A Bethe lattice provides a suitable approximation for percolation on a high-dimensional lattice, and yields exact tree structure enabling efficient cluster generation and downstream computation. To emulate percolation on a  $d$ -dimensional lattice, we simulated a Bethe lattice with degree  $z = 2d$  at the critical threshold,  $p = p_c = 1/(z - 1) = 1/(2d - 1)$ . We set  $d = 100$ .

Clusters were generated by breadth-first iteration starting from a root site. Each active site’s number of neighbors was sampled from a binomial distribution  $B(z - 1, p)$ <sup>3</sup>. Active sites were added until all sites had neighbors or a maximum size threshold was reached. To ensure that all clusters had enough sites without using too much memory, clusters with fewer than 300 or more than  $3 \times 10^7$  sites were discarded. To avoid unduly distorting the distribution, datasets were limited to 316 clusters, chosen to yield one cluster above the maximum size threshold in expectation. For simplicity and efficiency, clusters were represented as graphs, without embedding them into a high-dimensional Euclidean space.

While simulating a cluster, real-valued target values were generated for each active site. This was done following a branching random walk, with each site’s value randomly drawn from a normal distribution centered at its parent’s value. Afterward, each completed cluster’s values were standardized to have zero mean and unit standard deviation. Examples of simulated clusters with superimposed target functions are shown in Appendix G.

This generation procedure is intended to reflect our motivating assumptions in Sec. 2. However, we do not formalize or instantiate them in an explicit target function in this work. Implicitly, a random walk represents a continuous target function, and the exclusive dependence of target values on cluster topology corresponds to general-purpose learning. Standardizing the target values eliminates noise from varying cluster loss scales, reducing experimental variance.

#### 5.2. Machine Learning Model

We studied scaling laws using a model based on nearest-neighbor regression. This is a piecewise-constant nonparametric function approximator with an expected manifold

<sup>2</sup>For all plots, unless noted otherwise, the random seed was 0.

<sup>3</sup>Along with a site’s parent, this gives a maximum of  $z$  neighbors. For the root,  $B(z, p)$  was used instead.

approximation constant of  $c = 2$ . For all experiments, non-overlapping data subsets were randomly selected for training and testing. The number of training data points corresponds to both DOF (model scaling) and dataset size (data scaling). The loss function was mean squared error.

A key challenge was efficiently emulating an optimal model. We began with the simple and fast core of nearest-neighbor regression. The distance metric used was shortest-path graph distance, with every cluster’s full graph representation provided and every data point’s cluster identified. Because caching the distances between all site pairs consumed too much memory, nearest-neighbor values were efficiently computed by caching each site’s distance to the root and an index to its parent, and exploiting the cluster’s tree structure to reconstruct distances.

This model was augmented in two ways. First, a normal prior distribution was incorporated to provide less noisy predictions than naive nearest-neighbor regression. This was done using the Bayesian estimator, derived in Appendix H,

$$y_{\text{pred}} = \frac{y_{\text{nn}}}{1 + \frac{d_{\text{nn}}}{\sqrt{s}}}, \quad (12)$$

where  $y_{\text{pred}}$  is the prediction,  $y_{\text{nn}}$  the nearest neighbor value,  $d_{\text{nn}}$  the distance to the nearest neighbor, and  $s$  the cluster size. When  $d_{\text{nn}}$  is large,  $y_{\text{pred}}$  approaches 0, the prior mean.

With this improved estimator, the scaling curves for individual clusters were found to be consistent with manifold approximation,  $\mathcal{L} \sim N^{-c/D}$ . Empirically, however, each cluster’s scaling law had a different constant prefactor. We modeled this using cluster-dependent scale factors  $m \geq 1$  to convert training data points  $P$  to DOF  $N$ , i.e.,  $P = mN$ . Because learning requires at least one DOF, this manifests in the scaling curve as an initial “small data” region or plateau of random-guess loss (Hestness et al., 2017).

Each cluster’s value of  $m$  was then fit using the scaling law,

$$\mathcal{L} = \begin{cases} 1, & P < m \\ (P/m)^{-c/D}, & \text{otherwise.} \end{cases} \quad (13)$$

Appendix I shows representative examples of fitted cluster scaling curves. Overall,  $m$  varied between 1 and roughly 20, with an approximate central value of  $m \sim 5$ . For all following experiments, each cluster’s assigned number of DOF was scaled up by its fitted  $m$  to get the number of training points. Because  $\alpha = 1$ ,  $c = 2$ , and  $D = 4$  are fixed, the predicted scaling laws have no further free parameters.

### 5.3. Results

Eq. 6 gives the predicted optimal allocation of DOF among clusters for model scaling. We tested this using a parameter-

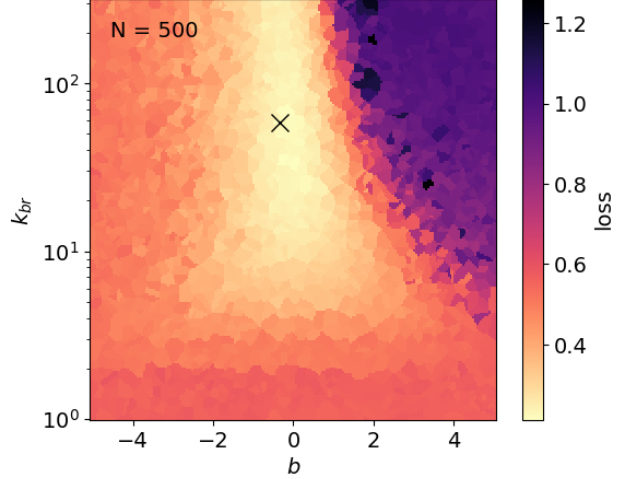


Figure 5. Loss achieved by a function approximator with parameterized DOF distribution for  $N = 500$ . A black cross indicates the parameters predicted to be optimal for model scaling.

ized DOF distribution with the functional form of Eq. 6, but allowing  $b$  and  $k_{\text{br}}$  to vary freely. The loss was computed for 2000 random combinations of  $b$  and  $k_{\text{br}}$ , with the results for  $N = 500$  shown in Fig. 5. Appendix J provides similar plots for other  $N$ . The best parameters appear consistent with the theoretically optimal ones.

Next, we investigated model and data scaling, with the DOF allocation for model scaling fixed to Eq. 6. Scaling curves were computed using 50 random seeds to account for the randomly generated datasets’ inherent variability. Fig. 6 shows the results. Each blue point with 1 standard deviation error bars shows the median with 16th and 84th percentiles. The results match theoretical predictions well.

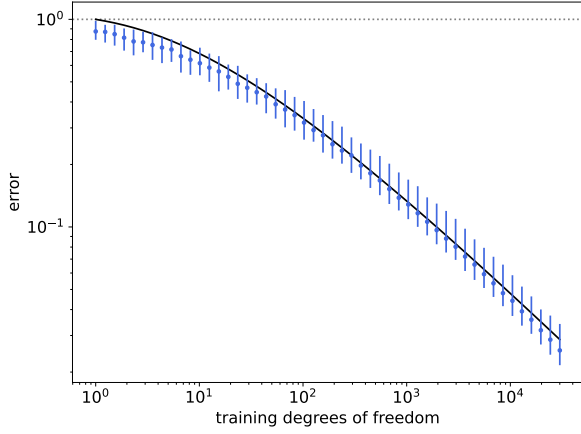
## 6. Discussion

### 6.1. Scaling Large Language Models

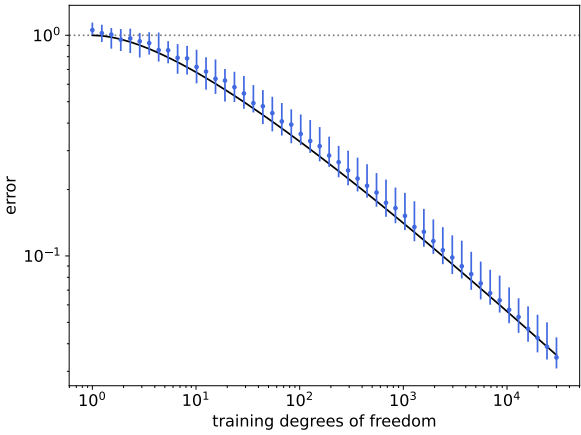
LLMs have achieved human-level performance across a bewildering array of tasks and disciplines (e.g. Bubeck et al., 2023). This diversity suggests that subcritical percolation may model natural language well. Indeed, Youn et al. (2016) used semantic network analysis across diverse human languages to reveal universal structure consisting of almost disconnected clusters of closely related concepts.

Understanding how close scaling laws achieved by LLMs<sup>4</sup> are to theoretically optimal is essential for forecasting their future performance and transformative potential (e.g. Bran-

<sup>4</sup>For LLM training, compute scaling can be derived from model and data scaling (Hoffmann et al., 2022), and this can be generalized to account for inference (Sardana et al., 2023).



(a) Model scaling



(b) Data scaling

Figure 6. Scaling laws computed using 50 random seeds. Results are shown as blue points with 1 standard deviation error bars. Theoretical predictions are overplotted in black (not a fit).

wen, 2020; Cotra, 2020; Barnett & Besiroglu, 2023). We discuss the complex relation between scale and capabilities in Appendix K. Possibly the most influential measurement of LLM scaling laws comes from Chinchilla (Hoffmann et al., 2022), which fit the loss using the parametric form<sup>5</sup>,

$$\mathcal{L}(P, \mathcal{D}) = E + \frac{A}{P^{\alpha_{m, \text{obs}}}} + \frac{B}{\mathcal{D}^{\alpha_{d, \text{obs}}}}, \quad (14)$$

with constants  $A$ ,  $B$ , and  $E$ , and model and data scaling exponents  $\alpha_{m, \text{obs}}$  and  $\alpha_{d, \text{obs}}$ . Hoffmann et al. (2022) reported exponents of  $\alpha_{m, \text{obs}} = 0.34$  and  $\alpha_{d, \text{obs}} = 0.28$ . Besiroglu et al. (2024) conducted a statistical reanalysis to get exponents of  $\alpha_{m, \text{obs}} = 0.35 \pm 0.02$  and  $\alpha_{d, \text{obs}} = 0.37 \pm 0.02$ .

As discussed in Appendix C, a comparison to empirical model scaling requires converting DOF  $N$  to parameters  $P$ .

<sup>5</sup>The notation has been changed for compatibility with ours.

We look at two scenarios. First, the infinite-width relation  $N = P$  yields  $\alpha_{m, \text{th}} = \alpha = 1$ . Another conjecture could be  $N \propto d_{\text{model}}$ . For example, associative memories in transformers (Geva et al., 2021; Meng et al., 2022) may yield scaling laws with respect to feedforward layer size  $d_{\text{ff}} \propto d_{\text{model}}$  (Cabannes et al., 2023), and generally  $d_{\text{ff}} \propto d_{\text{model}}$ . For Chinchilla, we estimate in Appendix L that  $d_{\text{model}} \propto P^{0.387}$ , yielding  $\alpha_{m, \text{th}} = 0.387 \cdot \alpha = 0.387$ . Thus, more work is needed to tell if LLMs have approached our theory’s limit to model scaling, or if more efficient scaling is possible.

For data scaling, we predict that  $\alpha_{d, \text{th}} = \alpha / (1 + \alpha) = 0.5$ . Our theory therefore suggests that somewhat more efficient data scaling remains possible for LLMs.

## 6.2. Data Distributions Near Criticality

In this work,  $p$  is a free parameter, and no inherent mechanism exists to push the data distribution to criticality (cf. self-organized criticality, Bak et al., 1987). However, real-world datasets may in practice have near-critical  $p$ , because they’re selected for feasible learning. A dataset with  $p \ll p_c$  resembles random data and allows minimal generalization. Memorizing it is inefficient and useless. On the other hand,  $p \gg p_c$  describes a dataset with simple functional structure but irreducible, strongly-coupled dependence on many input dimensions. It’s very inefficient to learn, as the scaling exponent is proportional to  $1/d$ . Only near criticality with  $p \approx p_c$  is learning both useful and efficient. In this regime, the data distribution contains one or more large but low-dimensional clusters in data space, corresponding to the well-known manifold hypothesis (Bengio et al., 2013).

## 6.3. Future Directions

This work made theoretical predictions of neural scaling laws. It could be built on by training neural networks on toy datasets to test scaling regimes with  $c > 2$ ; searching for percolation cluster structure in natural data distributions; and mechanistically interpreting neural networks to see whether they learn features that recapitulate dataset structure.

To derive power-law scaling, we assumed that data sampling was uniform and random. Non-uniform methods such as data pruning to undersample large clusters or oversample small ones (e.g. Sorscher et al., 2022), or active learning to query for desired data (Ren et al., 2021), could possibly outpace power-law data scaling. In addition, same-context samples can be correlated, so realistic sampling from a clustered data distribution may be nonergodic. A capacity tradeoff might then exist between modeling the data distribution and learning from context. Indeed, clustered data with many rare classes can drive emergent in-context learning in transformers (Chan et al., 2022). By linking neural networks’ internal mechanisms to data distributional properties, we can better understand how these models learn and scale.



## Acknowledgements

Thanks to Philipp Kreer, Eduardo López, Eric Michaud, Adam Shai, and Alok Singh for feedback and discussions on drafts of this work. This research was supported by a grant from the Long-Term Future Fund (EA Funds). Research was sponsored by the National Aeronautics and Space Administration (NASA) through a contract with ORAU. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Aeronautics and Space Administration (NASA) or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

## Impact Statement

This work theoretically investigates the origin of neural scaling laws. This study may advance the field of machine learning, which can have many potential societal consequences. Improved theoretical understanding of neural scaling laws may help in crafting policy aimed at ensuring that advanced machine learning systems are beneficial and safe.

## References

- Alabdulmohsin, I. M., Neyshabur, B., and Zhai, X. Revisiting neural scaling laws in language and vision. *Advances in Neural Information Processing Systems*, 35:22300–22312, 2022.
- Alabdulmohsin, I. M., Zhai, X., Kolesnikov, A., and Beyer, L. Getting vit in shape: Scaling laws for compute-optimal model design. *Advances in Neural Information Processing Systems*, 36, 2024.
- Anwar, U., Saparov, A., Rando, J., Paleka, D., Turpin, M., Hase, P., Lubana, E. S., Jenner, E., Casper, S., Sourbut, O., et al. Foundational challenges in assuring alignment and safety of large language models. *arXiv preprint arXiv:2404.09932*, 2024.
- Arora, S. and Goyal, A. A theory for emergence of complex skills in language models. *arXiv preprint arXiv:2307.15936*, 2023.
- Bachmann, G., Anagnostidis, S., and Hofmann, T. Scaling mpls: A tale of inductive bias. *Advances in Neural Information Processing Systems*, 36, 2024.
- Bahri, Y., Dyer, E., Kaplan, J., Lee, J., and Sharma, U. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024.
- Bak, P., Tang, C., and Wiesenfeld, K. Self-organized criticality: An explanation of the  $1/f$  noise. *Physical review letters*, 59(4):381, 1987.
- Barnett, M. and Besiroglu, T. The direct approach, 2023. URL <https://epochai.org/blog/the-direct-approach>. Accessed: 2024-02-06.
- Bender, E. M. and Koller, A. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5185–5198, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.463. URL <https://aclanthology.org/2020.acl-main.463>.
- Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623, 2021.
- Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.
- Besiroglu, T., Erdil, E., Barnett, M., and You, J. Chinchilla scaling: A replication attempt. *arXiv preprint arXiv:2404.10102*, 2024.
- Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.
- Bordelon, B., Canatar, A., and Pehlevan, C. Spectrum dependent learning curves in kernel regression and wide neural networks. In *International Conference on Machine Learning*, pp. 1024–1034. PMLR, 2020.
- Branwen, G. <https://gwern.net/scaling-hypothesis>, 2020. Accessed: 2024-02-06.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf).

- Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.
- Caballero, E., Gupta, K., Rish, I., and Krueger, D. Broken neural scaling laws. *arXiv preprint arXiv:2210.14891*, 2022.
- Cabannes, V., Dohmatob, E., and Bietti, A. Scaling laws for associative memories. *arXiv preprint arXiv:2310.02984*, 2023.
- Chan, S., Santoro, A., Lampinen, A., Wang, J., Singh, A., Richemond, P., McClelland, J., and Hill, F. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 35:18878–18891, 2022.
- Chen, M., Jiang, H., Liao, W., and Zhao, T. Efficient approximation of deep relu networks for functions on low dimensional manifolds. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/fd95ec8df5dbeea25aa8e6c808bad583-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/fd95ec8df5dbeea25aa8e6c808bad583-Paper.pdf).
- Chughtai, B., Chan, L., and Nanda, N. A toy model of universality: Reverse engineering how networks learn group operations. In *International Conference on Machine Learning*, pp. 6243–6267. PMLR, 2023.
- Cotra, A. <https://www.lesswrong.com/posts/KrJfoZzpSDpnrV9va/draft-report-on-ai-timelines>, 2020. Accessed: 2024-02-06.
- Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021.
- Ghorbani, B., Firat, O., Freitag, M., Bapna, A., Krikun, M., Garcia, X., Chelba, C., and Cherry, C. Scaling laws for neural machine translation. *arXiv preprint arXiv:2109.07740*, 2021.
- Henighan, T., Kaplan, J., Katz, M., Chen, M., Hesse, C., Jackson, J., Jun, H., Brown, T. B., Dhariwal, P., Gray, S., et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.
- Hernandez, D., Brown, T., Conerly, T., DasSarma, N., Drain, D., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Henighan, T., Hume, T., et al. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022.
- Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.
- Huh, M., Cheung, B., Wang, T., and Isola, P. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.
- Hutter, M. Learning curve theory. *arXiv preprint arXiv:2102.04074*, 2021.
- Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.
- Jones, A. L. Scaling scaling laws with board games. *arXiv preprint arXiv:2104.03113*, 2021.
- Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.
- Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.
- Lin, H. W., Tegmark, M., and Rolnick, D. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168:1223–1247, 2017.
- Liu, B., Ash, J. T., Goel, S., Krishnamurthy, A., and Zhang, C. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.
- Lubana, E. S., Kawaguchi, K., Dick, R. P., and Tanaka, H. A percolation model of emergence: Analyzing transformers trained on a formal language. *arXiv preprint arXiv:2408.12578*, 2024.

- Maloney, A., Roberts, D. A., and Sully, J. A solvable model of neural scaling laws. *arXiv preprint arXiv:2210.16859*, 2022.
- Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.
- Michaud, E., Liu, Z., Girit, U., and Tegmark, M. The quantization model of neural scaling. *Advances in Neural Information Processing Systems*, 36, 2024.
- Minsky, M. *Society of mind*. Simon and Schuster, 1988.
- Nanda, N., Chan, L., Liberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.
- Neal, R. M. *Bayesian Learning for Neural Networks*. Springer, 1996.
- Newman, M. E. Power laws, pareto distributions and zipf’s law. *Contemporary physics*, 46(5):323–351, 2005.
- Okawa, M., Lubana, E. S., Dick, R., and Tanaka, H. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 50173–50195. Curran Associates, Inc., 2023. URL [https://proceedings.neurips.cc/paper\\_files/paper/2023/file/9d0f188c7947eac0c07f709576824f6-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/9d0f188c7947eac0c07f709576824f6-Paper-Conference.pdf).
- Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. <https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html>.
- Perniss, P. and Vigliocco, G. The bridge of iconicity: from a world of experience to the experience of language. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1651):20130300, 2014.
- Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X., and Wang, X. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.
- Roberts, D. A. Why is ai hard and physics simple? *arXiv preprint arXiv:2104.00008*, 2021.
- Roberts, D. A., Yaida, S., and Hanin, B. *The principles of deep learning theory*. Cambridge University Press Cambridge, MA, USA, 2022.
- Rosenfeld, J. S., Rosenfeld, A., Belinkov, Y., and Shavit, N. A constructive prediction of the generalization error across scales. *arXiv preprint arXiv:1909.12673*, 2019.
- Sahimi, M. *Applications of percolation theory*. CRC Press, 1994.
- Sardana, N., Portes, J., Doubov, S., and Frankle, J. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. *arXiv preprint arXiv:2401.00448*, 2023.
- Schaeffer, R., Miranda, B., and Koyejo, S. Are emergent abilities of large language models a mirage? In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 55565–55581. Curran Associates, Inc., 2023. URL [https://proceedings.neurips.cc/paper\\_files/paper/2023/file/adc98a266f45005c403b8311ca7e8bd7-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/adc98a266f45005c403b8311ca7e8bd7-Paper-Conference.pdf).
- Sharma, U. and Kaplan, J. Scaling laws from the data manifold dimension. *Journal of Machine Learning Research*, 23(9):1–34, 2022. URL <http://jmlr.org/papers/v23/20-1111.html>.
- Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., and Morcos, A. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.
- Spigler, S., Geiger, M., and Wyart, M. Asymptotic learning curves of kernel methods: empirical data versus teacher-student paradigm. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124001, 2020.
- Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*, 2023.
- Stauffer, D. and Aharony, A. *Introduction To Percolation Theory*. CRC Press, 1994.
- Sutton, R. The bitter lesson. *Incomplete Ideas (blog)*, 13(1), 2019.

- Tahmasebi, B. and Jegelka, S. The exact sample complexity gain from invariances for kernel regression. *Advances in Neural Information Processing Systems*, 36, 2023.
- Wasserman, L. *All of nonparametric statistics*. Springer Science & Business Media, 2006.
- Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.
- Yang, G., Hu, E. J., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., and Gao, J. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.
- Youn, H., Sutton, L., Smith, E., Moore, C., Wilkins, J. F., Maddieson, I., Croft, W., and Bhattacharya, T. On the universal structure of human lexical semantics. *Proceedings of the National Academy of Sciences*, 113(7):1766–1771, 2016.



## A. General-Purpose Learning

Our assumption of general-purpose learning describes a learning task that has latent data distributional structure independent of the format used to represent it. The latent structure comes from the world, while the input format is determined by the learning system. When this relationship exists, we equivalently say that the data’s latent structure is *extrinsic*, because no information about the external reality it derives from is built into the learning system. Indeed, deep neural networks trained on disparate datasets and modalities appear to learn convergent internal representations, consistent with independent general-purpose learning systems each modeling reality’s shared extrinsic structure (Huh et al., 2024).

To illustrate, there are thousands of natural languages. These languages can have many observable realizations or forms, including speech, written words, visual gestures and signs, and token embedding vectors. Language in any form also has meaning: humans use it to communicate, which means it must in some way be structured by a correspondence with shared reality (Bender & Koller, 2020). Language’s form is essentially arbitrary, but its latent structure or meaning derives from external reality. A similar division between form and meaning exists in computer vision. A picture of a dog (say) is a meaningful image because it represents an actual or imagined dog, regardless of the values of its pixel-space representation.

Not all datasets have extrinsic structure. For example, a tabular dataset of handcrafted features doesn’t, because its features define meaningful axes of variation by construction. The same is true of an algorithmic dataset representing a group-theoretic transformation or procedural task in which the symbolic forms are isomorphic with the mathematical operations represented (e.g. Nanda et al., 2023; Liu et al., 2022; Chughtai et al., 2023; Michaud et al., 2024). In these cases, the input format provides built-in information relevant to the task at hand.

General-purpose learning doesn’t matter if the model fails to recognize or exploit a dataset’s extrinsic structure. This might occur if adequate performance can be achieved by exclusively modeling surface statistics. Notably, Bender et al. (2021) claimed that language models are “stochastic parrots” that generate text based solely on statistical modeling of linguistic forms, without reference to meaning. By contrast, we assume that the model under consideration is not a stochastic parrot and can be usefully modeled as learning form-independent latent extrinsic structure.

A dataset’s structure might be partly extrinsic and partly form-dependent, so that general-purpose learning is only partially realized. A language model may combine a meaningful world model with heuristics based on surface statistics. In addition, language can possess iconicity, or non-arbitrary resemblances between linguistic form and meaning (Perniss & Vigliocco, 2014). We expect our theory to apply in proportion to the extent to which the dataset’s structure is extrinsic.

## B. Expected Loss Reduction from Learning a Cluster Function

In supervised learning, each in-distribution element of the data space constitutes a possible prediction task. There are then  $pL^d$  possible tasks. Learning the function (or quantum) defined on a cluster of size  $s$  reduces the error on  $s$  of these tasks, so the expected loss reduction from learning an  $s$ -cluster’s function is  $\Delta\mathcal{L}_s \propto -s/(pL^d) \propto s$ .

Neural networks are also commonly (pre-)trained on self-supervised generative objectives, including masked language modeling and next token prediction for language models, and masked denoising approaches in computer vision. In this setting, the model is trained to predict a portion of the input that has been masked or corrupted, by exploiting the surrounding context. The masked input may be one or more tokens or pixel patches, depending on the method and modality. If  $m$  dimensions are masked, there are  $\binom{d}{m}pL^d$  possible prediction tasks. As each prediction task is done independently in practice, we consider the case  $m = 1$ .

In this setting, the expected loss reduction from learning an  $s$ -cluster’s function is given by the expected number of relevant prediction tasks multiplied by the expected loss reduction per relevant task,  $\Delta\mathcal{L}_s = N_{\text{task}} \times \Delta\mathcal{L}_{\text{task}}$ . If one dimension is masked, then the intersection of an  $s$ -cluster with the unmasked hypersurface has fractal dimension  $D + (d - 1) - d = D - 1$ , so that  $N_{\text{task}} \propto R_s^{D-1}$ .

The cluster’s projection onto the masked dimension has expected length  $R_s$ . However,  $\Delta\mathcal{L}_{\text{task}}$  corresponds to how well learning a cluster function concentrates probability mass onto the correct output, which depends nonlinearly on  $R_s$ . If  $R_s$  is small, the zero-information prior may overwhelm the cluster’s contribution, leading to small  $\Delta\mathcal{L}_{\text{task}}$ . If  $R_s$  is large, the model can’t pinpoint which of the cluster’s possible output values is correct, and  $\Delta\mathcal{L}_{\text{task}}$  is again small. For moderately sized  $R_s$ , however, we will see below that  $\Delta\mathcal{L}_{\text{task}} \propto R_s$ .

To do so, we consider the expected loss reduction from learning an  $s$ -cluster compared to a zero-information baseline for a

prediction task with a cross-entropy loss function and vocabulary size  $L$ . We assume that the prediction task has exactly one correct output  $i$ , giving true labels  $q_i = 1, q_{l \neq i} = 0$  for all  $l \in (1, L)$ . The zero-information baseline assigns an equal probability  $p_l = p / \sum_{l=1}^L p = 1/L$  to each possible output along the masked dimension, yielding loss  $\mathcal{L} = -\log \frac{1}{L} = \log L$ .

Now, suppose the model fully learns a relevant quantum corresponding to a cluster of size  $s$  and radius  $R_s$ . We assume the cluster is small compared to the full data distribution,  $s \ll pL^d$ , so that learning it doesn't appreciably reduce the probability mass assigned to out-of-cluster elements. All data-space elements consistent with the unmasked input are possible outputs, with in-cluster elements being equally probable with unit weight and out-of-cluster elements having lower zero-information weight  $p$ . The probability that the correct output is in the cluster is then

$$p_{\text{in}} = \frac{R_s}{R_s + p(L - R_s)}. \quad (15)$$

The probabilities that the model should predict for outputs in and out of the cluster, respectively, are

$$p_{l,\text{in}} = \frac{1}{R_s + p(L - R_s)}, \quad (16)$$

and

$$p_{l,\text{out}} = \frac{p}{R_s + p(L - R_s)}. \quad (17)$$

The total expected loss reduction is, after some algebra,

$$\begin{aligned} \Delta \mathcal{L}_{\text{task}} &= \mathcal{L}_{\text{in}} + \mathcal{L}_{\text{out}} - \mathcal{L}_{\text{baseline}} \\ &= p_{\text{in}} (-\log p_{l,\text{in}}) + (1 - p_{\text{in}}) (-\log p_{l,\text{out}}) - \log L \\ &= \log(1 + \delta) + \frac{\delta}{1 + \delta} \frac{\log p}{1 - p}, \end{aligned} \quad (18)$$

where

$$\delta = \frac{(1 - p)R_s}{pL}.$$

Note that Eq. 18 vanishes either as  $R_s \rightarrow 0$  or  $R_s \rightarrow L$ . We can think of  $\delta$  as the ratio between the probability mass gained by the in-cluster elements when the cluster is learned and the total expected in-distribution probability mass. If we make the reasonable assumption that the probability mass attributable to learning a cluster is small compared to the expected total, then to first order,

$$\Delta \mathcal{L}_{\text{task}} \approx \delta + \delta \frac{\log p}{1 - p} = \frac{1 - p + \log p}{pL} R_s \propto R_s. \quad (19)$$

The approximate expected loss reduction from learning an  $s$ -cluster is then  $\Delta \mathcal{L}_s = N_{\text{task}} \times \Delta \mathcal{L}_{\text{task}} \propto R_s^{D-1} R_s = s$ .

### C. Neural Network Model Capacity

Comparing theoretical predictions of model scaling with empirical measurements requires a way to consistently determine model capacity. Empirical scaling laws are typically reported in terms of the number of weight and bias parameters  $P$ , but relating parameter count to a notion of model capacity such as nonparameteric DOF  $N$  isn't necessarily straightforward.

Many prior works theoretically investigating neural scaling laws appear to assume that model capacity and parameter count are directly proportional (e.g. [Sharma & Kaplan, 2022](#); [Bahri et al., 2024](#); [Maloney et al., 2022](#); [Michaud et al., 2024](#)). But

this assumption doesn't seem obviously true in general. For example, the capacity of a network with a narrow bottleneck layer (e.g. an autoencoder) is intuitively constrained by its bottleneck dimension, regardless of the other layers' sizes. The relation between parameter count and model capacity seems to be architecture-dependent. From an empirical standpoint, counting only non-embedding parameters provides significantly cleaner scaling laws (Kaplan et al., 2020). This suggests that model capacity may depend on how parameters are employed in a network, not just their total count. Also, regularization such as weight decay or dropout can reduce model capacity but isn't reflected in the raw parameter count.

Effective field theories of deep neural networks (Roberts et al., 2022) may provide a useful theoretical perspective. By analogy to statistical mechanics, one can describe a neural network either in terms of "microscopic" weight and bias parameters or "macroscopic" effective DOF. The macroscopic representation depends on architecture and scale. In the infinite-width limit, a deep neural network effectively interpolates the training data and is equivalent to a Gaussian processes (Neal, 1996; Lee et al., 2018). In this limit, a deep neural network's output can be described as depending on a number of fixed random macroscopic features equal to the number of parameters, giving  $N = P$  (Jacot et al., 2018; Lee et al., 2019).

However, in practice, width and depth are typically scaled up in tandem (e.g. Kaplan et al., 2020; Hoffmann et al., 2022). To analyze this regime, Roberts et al. (2022) computed a perturbative expansion to the infinite-width limit, finding that representation learning arises when the network's depth-to-width ratio is small but finite<sup>6</sup>. In this limit, the model output now depends on  $P$  macroscopic features that are constrained by an implicit inductive bias to learn a nontrivial representation from the data. Since such a constraint implies that the learned features aren't independent, we conjecture that when representation learning occurs,  $N < P$ . Further theoretical and empirical investigation of model capacity in neural networks may be fruitful for shedding light on these issues.

## D. Model Scaling

We assume an idealized training procedure with unlimited training data and computation, and perfect optimization. We consider the pure scaling regime with irreducible error assumed to be negligible. Complexities resulting from relaxing these idealized assumptions could be empirically modeled using a flexible fitting procedure (e.g. Hestness et al., 2017; Rosenfeld et al., 2019; Alabdulmohsin et al., 2022; Caballero et al., 2022).

We consider a nonparametric model characterized by  $N$  effective DOF. At convergence, the cluster of rank  $k$  is allocated  $n_k \geq 0$  DOF to model its function (quantum) such that

$$N = \sum_{k=1}^{\infty} n_k. \quad (20)$$

We assume that operations besides modeling cluster functions, such as identifying an input's corresponding cluster or converting the model's internal representations to the output format, consume negligible model capacity and can be neglected.

The total loss  $\mathcal{L}$  is the sum of the loss from each cluster  $k$ ,

$$\mathcal{L} = \sum_{k=1}^{\infty} \mathcal{L}_k, \quad (21)$$

where each cluster's loss is given by manifold approximation (Sharma & Kaplan, 2022),

$$\mathcal{L}_k = A_k k^{-(\alpha+1)} \begin{cases} n_k^{-c_k/D_k} & n_k \geq 1 \\ 1 & \text{otherwise.} \end{cases} \quad (22)$$

In Eq. 22,  $A_k$  is a constant that determines the loss scale for cluster  $k$ . This constant combines numerical factors coming from sources including the efficiency of the model class and training procedure; the target label scale and any other factors in the loss function; the conversion between cluster rank  $k$  and size  $s$ ; and the function's complexity or rate of change. We assume that  $A_k = A$  for all  $k$ , that is, all cluster functions have the same complexity. Without loss of generality we let

<sup>6</sup>Representation learning can occur in the infinite-width limit if maximal-update parameterization is used (Yang et al., 2022).

$A = 1$ . We further assume that all cluster functions are generic, giving  $c_k = c$  for all  $k$ , where  $c$  is a constant representing the complexity of the nonparametric function approximator. As discussed in Sec 3.2, all clusters can be approximated as having the same intrinsic dimension  $D$ , so we let  $D_k = D$  for all  $k$ .

Each marginal DOF goes toward modeling the cluster function giving the greatest marginal loss reduction. Each cluster can be allocated one or more DOF, or none. In equilibrium, among clusters allocated DOF, the marginal loss reduction from adding DOF must be the same and equal to that of the first cluster allocated none, up to discretization. We call the rank of that cluster where a break occurs  $k_{\text{br}}$ . Assigning DOF to any cluster after  $k_{\text{br}}$  would reduce the loss suboptimally.

In equilibrium, after training has converged, for any  $k_1, k_2 < k_{\text{br}}$ , we have

$$\frac{\partial}{\partial n_{k_1}} \left( n_{k_1}^{-c/D} k_1^{-(\alpha+1)} \right) = \frac{\partial}{\partial n_{k_2}} \left( n_{k_2}^{-c/D} k_2^{-(\alpha+1)} \right), \quad (23)$$

where

$$\frac{\partial}{\partial n_k} \left( n_k^{-c/D} k^{-(\alpha+1)} \right) = -\frac{c}{D} n_k^{-(c/D+1)} k^{-(\alpha+1)}, \quad (24)$$

yielding

$$\frac{n_{k_1}}{n_{k_2}} = \left( \frac{k_1}{k_2} \right)^{-\frac{\alpha+1}{c/D+1}}. \quad (25)$$

To solve for  $n_k$ , we adopt the ansatz that, for  $a = a(c, D, \alpha, N)$  and  $b = b(c, D, \alpha, N) < 1$ ,

$$n_k = \begin{cases} a k^{b-1} & k < k_{\text{br}} \\ 0 & k \geq k_{\text{br}}. \end{cases} \quad (26)$$

Substituting, we obtain

$$b = 1 - \frac{\alpha + 1}{c/D + 1} = \frac{c/D - \alpha}{c/D + 1}. \quad (27)$$

In addition, we have from Eq. 20,

$$N = \sum_{k=1}^{k_{\text{br}}} a k^{b-1} \approx \int_1^{k_{\text{br}}} a k^{b-1} dk = \frac{a}{b} (k_{\text{br}}^b - 1),$$

giving

$$a = \frac{bN}{k_{\text{br}}^b - 1}. \quad (28)$$

To solve for  $k_{\text{br}}$ , we use the fact that the marginal loss reduction from allocating a marginal DOF for modeling any cluster with rank  $k_i < k_{\text{br}}$  is approximately equal to the marginal loss reduction from allocating the first DOF at  $k_{\text{br}}$ ,

$$-\frac{c}{D} (a k_i^{b-1})^{-(c/D+1)} k_i^{-(\alpha+1)} \approx -\frac{c}{D} (1)^{-(c/D+1)} k_{\text{br}}^{-(\alpha+1)}.$$

Using the expressions for  $b$  and  $a$  from Eqs. 27 and 28 and simplifying, we obtain

$$bN = k_{\text{br}} (1 - k_{\text{br}}^{-b}). \quad (29)$$



If  $c/D \gg \alpha$  and  $c/D \gtrsim 1$ ,  $b \approx 1$  and  $k_{\text{br}} \approx N + 1 \approx N$ . If  $c/D \approx \alpha$  or both  $c/D \ll 1$  and  $\alpha \ll 1$ ,  $|b| \ll 1$  and Eq. 29 approaches  $N \approx k_{\text{br}} \ln k_{\text{br}}$ . The solution is  $k_{\text{br}} = e^{W(N)} = N/W(N) \approx N/\ln(N/\ln N)$ , where  $W(x)$  is the Lambert  $W$  function. Finally, if  $c/D \ll \alpha$  and  $\alpha \gtrsim 1$ , then  $b < 0$  and  $|b| > 1$ , so  $k_{\text{br}} \approx (|b|N)^{1/(1+|b|)}$ . Summarizing,

$$k_{\text{br}} \approx \begin{cases} N & c/D \gg \alpha, c/D \gtrsim 1 \\ \left(\ln \frac{N}{\ln N}\right)^{-1} N & c/D \approx \alpha \text{ or } c/D \ll 1, \alpha \ll 1 \\ (|b|N)^{1/(1+|b|)} & c/D \ll \alpha, \alpha \gtrsim 1. \end{cases} \quad (30)$$

Putting everything together, the expected loss is given by

$$\begin{aligned} \mathcal{L} &= \sum_{k=1}^{k_{\text{br}}} (ak^{b-1})^{-c/D} k^{-(\alpha+1)} + \sum_{k=k_{\text{br}}}^{\infty} k^{-(\alpha+1)} \\ &= \sum_{k=1}^{k_{\text{br}}} a^{-c/D} k^{b-1} + \sum_{k=k_{\text{br}}}^{\infty} k^{-(\alpha+1)} \\ &\approx \int_1^{k_{\text{br}}} a^{-c/D} k^{b-1} dk + \int_{k_{\text{br}}}^{\infty} k^{-(\alpha+1)} dk \\ &= \left(\frac{k_{\text{br}}^b - 1}{b}\right) a^{-c/D} + \alpha^{-1} k_{\text{br}}^{-\alpha} \\ &= \left(\frac{k_{\text{br}}^b - 1}{b}\right)^{1+c/D} N^{-c/D} + \alpha^{-1} k_{\text{br}}^{-\alpha} \\ &= k_{\text{br}}^{-(1-b)(1+c/D)} \left(\frac{k_{\text{br}}(1 - k_{\text{br}}^{-b})}{b}\right)^{1+c/D} N^{-c/D} + \alpha^{-1} k_{\text{br}}^{-\alpha} \\ &= \left(\frac{N}{k_{\text{br}}} + \frac{1}{\alpha}\right) k_{\text{br}}^{-\alpha}. \end{aligned} \quad (31)$$

In the regime  $c/D \gg \alpha$ ,  $c/D \gtrsim 1$ ,  $k_{\text{br}} \approx N$ , and we obtain

$$\begin{aligned} \mathcal{L} &\approx \left(1 + \frac{1}{\alpha}\right) N^{-\alpha} \\ &\propto N^{-\alpha}. \end{aligned} \quad (32)$$

In the regime  $c/D \ll \alpha$ ,  $\alpha \gtrsim 1$ , we obtain

$$\begin{aligned} \mathcal{L} &\approx \left(\frac{N}{(|b|N)^{1/(1+|b|)}} + \frac{1}{\alpha}\right) (|b|N)^{-\alpha/(1+|b|)} \\ &= |b|^{-\frac{\alpha+1}{1+|b|}} \left(1 + \frac{1}{\alpha} |b|^{\frac{1}{1+|b|}} N^{-\frac{|b|}{1+|b|}}\right) N^{-\frac{\alpha-|b|}{1+|b|}} \\ &= |b|^{-(c/D+1)} \left(1 + \frac{1}{\alpha} |b|^{\frac{1}{1+|b|}} N^{-\frac{|b|}{1+|b|}}\right) N^{-c/D} \\ &\approx \left(\frac{\alpha}{c/D+1}\right)^{-(c/D+1)} \left(1 + \frac{1}{\alpha} N^{-\frac{\alpha}{1+\alpha}}\right) N^{-c/D} \\ &\propto N^{-c/D} \text{ for large } N. \end{aligned} \quad (33)$$

## E. Data Scaling

We assume that training examples are randomly sampled from the data distribution, which has  $M \equiv pL^d$  elements. Suppose that the training set contains  $\mathcal{D} \equiv \epsilon M$  examples, such that  $\epsilon \ll 1$  and  $\mathcal{D} \gg 1$ . Then the cluster with rank  $k$  will be sampled  $l$  times, with  $l$  having a binomial distribution,

$$l \sim \binom{\mathcal{D}}{l} \left(1 - \alpha k^{-(\alpha+1)}\right)^{\mathcal{D}-l} \left(\alpha k^{-(\alpha+1)}\right)^l, \quad (34)$$

where  $\alpha k^{-(\alpha+1)}$  is the probability that any given training example belongs to that cluster.

A cluster must be sampled at least once,  $l \gtrsim 1$ , to contribute to reducing the loss<sup>7</sup>. If a cluster isn't sampled ( $l = 0$ ), it can't be learned. For the cluster of rank  $k$ , the characteristic transition between these regimes occurs when  $\mathcal{D}\alpha k^{-(\alpha+1)} \approx 1$ , yielding  $k_{\text{br}} \approx (\mathcal{D}\alpha)^{1/(1+\alpha)}$ .

The total loss is then, up to an overall constant,

$$\mathcal{L} \approx \sum_{k=1}^{k_{\text{br}}} \sum_{l=0}^{\infty} p(l|k) l(k)^{-c/D} k^{-(\alpha+1)} + \sum_{k=k_{\text{br}}}^{\infty} k^{-(\alpha+1)}. \quad (35)$$

To simplify this expression, we'll replace the sums with integrals and make the further approximation that  $p(l|k) \approx \delta(l(k) - \mathbb{E}(l|k))$ , thereby substituting  $l(k)$  with its mean value for each  $k$ . This yields

$$\begin{aligned} \mathcal{L} &\approx \int_1^{k_{\text{br}}} \left(\mathcal{D}\alpha k^{-(\alpha+1)}\right)^{-c/D} k^{-(\alpha+1)} dk + \int_{k_{\text{br}}}^{\infty} k^{-(\alpha+1)} dk \\ &= \frac{(\mathcal{D}\alpha)^{-c/D}}{\frac{c}{D}(1+\alpha) - \alpha} \left(k_{\text{br}}^{-\alpha + \frac{c}{D}(1+\alpha)} - 1\right) + \alpha^{-1} k_{\text{br}}^{-\alpha} \\ &= \frac{1}{\frac{c}{D}(1+\alpha) - \alpha} \left((\mathcal{D}\alpha)^{-\alpha/(1+\alpha)} - (\mathcal{D}\alpha)^{-c/D}\right) + \alpha^{-1} (\mathcal{D}\alpha)^{-\alpha/(1+\alpha)} \\ &= \frac{\alpha^{-\alpha/(1+\alpha)-1}}{1 - \frac{\alpha/(1+\alpha)}{c/D}} \mathcal{D}^{-\alpha/(1+\alpha)} + \frac{\alpha^{-c/D-1}}{1 - \frac{c/D}{\alpha/(1+\alpha)}} \mathcal{D}^{-c/D}, \end{aligned} \quad (36)$$

assuming that  $c/D \neq \alpha/(1+\alpha)$ . In the limiting case where  $c/D \gg \alpha/(1+\alpha)$ , the first term dominates and we obtain  $\mathcal{L} \propto \mathcal{D}^{-\alpha/(1+\alpha)}$ . In the limiting case where  $\alpha/(1+\alpha) \gg c/D$ , the second term dominates and we obtain  $\mathcal{L} \propto \mathcal{D}^{-c/D}$ .

If  $c/D = \alpha/(1+\alpha)$ , we have instead of Eq. 36,

$$\begin{aligned} \mathcal{L} &\approx (\mathcal{D}\alpha)^{-c/D} \log k_{\text{br}} + \alpha^{-1} k_{\text{br}}^{-\alpha} \\ &= \alpha^{-\alpha/(1+\alpha)-1} \mathcal{D}^{-\alpha/(1+\alpha)} + \alpha^{-c/D-1} \frac{c}{D} \log(\alpha \mathcal{D}) \mathcal{D}^{-c/D} \\ &= \alpha^{-c/D-1} \left(1 + \frac{c}{D} \log \alpha + \frac{c}{D} \log \mathcal{D}\right) \mathcal{D}^{-c/D}. \end{aligned} \quad (37)$$

<sup>7</sup>A cluster's size  $s$  can be estimated if it's sampled multiple times, as the average distance between random cluster sites is proportional to  $R_s$ . If a cluster is only sampled once ( $l = 1$ ), its overall size and shape can't be effectively estimated, but generalization is still possible in that sample's immediate neighborhood. One could model this using an estimated cluster size  $s_{\text{est}}(l)$ , with  $s_{\text{est}}(l) \approx 1$  for  $l = 1$  and  $s_{\text{est}}(l) \approx s$  for  $l \gg 1$ . We elide this complication as its effect on the total estimated loss is small.

## F. Theories of Neural Scaling Laws

The proposed neural scaling model encompasses several distinct criticality regimes and limiting cases that can be identified with previously proposed models of neural scaling laws. Table 1 highlights several of these connections. In the subcritical regime, the scaling behavior depends on the relative magnitudes of  $c/D$  and either  $\alpha$  (for model scaling) or  $\alpha/(1 + \alpha)$  (for data scaling). When  $c/D$  is large, the achievable loss is bounded by the exponent of the cluster size distribution. This limit appears to correspond to the quantization model of neural scaling proposed by [Michaud et al. \(2024\)](#). In both that work and this one, the predicted exponents for model and data scaling are  $\alpha$  and  $\alpha/(1 + \alpha)$ , respectively.

Criticality	Limit	Scaling	Related Works
$p \leq p_c$	$c/D \gg \alpha$	Model	<a href="#">Michaud et al. (2024)</a>
	$c/D \gg \alpha/(1 + \alpha)$	Data	<a href="#">Hutter (2021)</a> ; <a href="#">Michaud et al. (2024)</a>
	$c/D \ll \alpha$	Model	<a href="#">Sharma &amp; Kaplan (2022)</a> ; <a href="#">Bahri et al. (2024)</a>
	$c/D \ll \alpha/(1 + \alpha)$	Data	
$p > p_c$	N/A	Model Data	

Table 1. Connections between the proposed model and related works.

Our model extends the work of [Michaud et al. \(2024\)](#) in several ways. We derive a power-law distribution of quantized subtasks from first principles, thus supporting the quantization hypothesis conjectured in that work. Furthermore, we quantitatively predict that  $\alpha = 1$  (for  $d \geq 6$ ), so  $\alpha$  is not a free parameter. Qualitatively, our model suggests that neural networks learn quanta gradually in parallel, rather than fully one by one in sequence. This description complicates the connection between quantized subtasks and emergent capabilities, which we discuss further in Appendix K. Unlike [Michaud et al. \(2024\)](#), we don’t consider training dynamics or make predictions for single-epoch scaling.

A data scaling exponent of  $\alpha/(1 + \alpha)$  was also predicted by [Hutter \(2021\)](#) for a toy model of data scaling with Zipf-distributed data. [Hutter \(2021\)](#) obtained this exponent by considering a dataset of discrete labeled features and an algorithm that correctly predicts a feature if and only if it’s been previously seen. The quantitative similarity between that model and ours can be understood by considering the discrete features to represent clusters and the learning algorithm to approximately correspond to the limit  $c/D \rightarrow \infty$ . [Cabannes et al. \(2023\)](#) obtained the same scaling behavior by analyzing the storage and retrieval of Zipf-distributed discrete associative memories using a simple model of a transformer layer.

In the supercritical regime, or in the subcritical regime when  $c/D$  is small compared to either  $\alpha$  (for model scaling) or  $\alpha/(1 + \alpha)$  (for data scaling), the achievable loss is bounded by the model’s ability to nonparametrically approximate the data distribution. This regime appears to correspond to the manifold approximation model proposed by [Sharma & Kaplan \(2022\)](#) and extended by [Bahri et al. \(2024\)](#). Our theory extends these prior works by grounding the predicted scaling laws in a percolation model of the data distribution. A connected data distribution with well-defined intrinsic dimension emerges when  $p > p_c$ . When  $p \leq p_c$ ,  $c/D$  scaling can still occur, with the data distribution consisting of disconnected clusters.

[Maloney et al. \(2022\)](#) proposed a solvable model of neural scaling laws in which the dataset is generated from a set of latent features with power-law spectral structure. The dataset’s properties are controlled by two independent hyperparameters: the latent space dimension  $M$ , which must be larger than any other scale, and the spectral index  $\alpha$ , which determines the intrinsic dataset dimension such that  $d_{\text{in}} \propto 1/\alpha$ . The dimensions  $M$  and  $d_{\text{in}}$  represent different properties jointly characterizing the latent data manifold. In our model, the dataset is instead essentially characterized by a single hyperparameter,  $p$  (since  $D$  and  $\alpha$  are fixed for  $d \geq 6$ ). No explicit latent feature space is required. Instead, a power-law distribution of latent features — clusters — naturally emerges from a context-dependent target function and general-purpose learning.

## G. Visualizations of Simulated Clusters

Fig. 7 shows visualizations of functions defined on simulated percolation clusters. The clusters shown are restricted to less than 2000 sites to avoid incurring excessive computational cost for generating the visualizations.

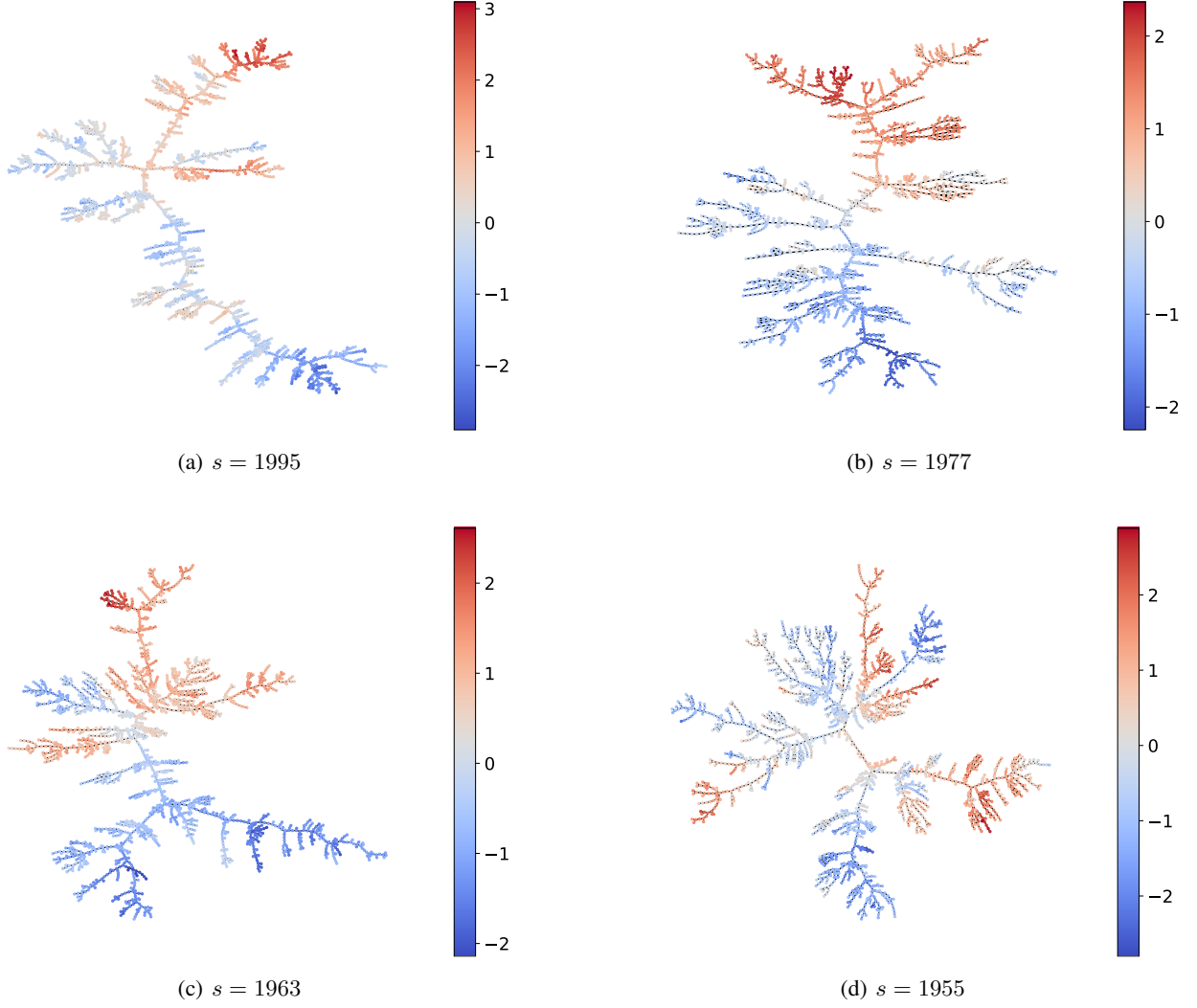


Figure 7. Visualizations of percolation clusters simulated on a Bethe lattice. Site color indicates each cluster's superimposed function.



## H. Bayesian Nearest Neighbor Estimator

The value predicted by a conventional nearest-neighbor estimator is the nearest neighbor's value,  $y_{\text{pred}} = y_{\text{nn}}$ . This estimator can be improved by using Bayesian inference to include prior information. In particular, adjusting the predicted value toward the prior mean reduces the expected error when the nearest neighbor is distant.

For our setup, the marginal distribution of sites' target values in a cluster can be modeled as standard normal,  $y \sim \mathcal{N}(0, 1)$ . We are given an observation of the nearest-neighbor training site's value  $y_{\text{nn}}$ , and need to predict the value at a test site at a topological distance  $d_{\text{nn}}$ , given by the number of sites along the minimum path length. The cluster size  $s$  is known.

Each cluster's  $y$  values were generated following a random walk with random terms  $\epsilon$  initially distributed as standard normal. After standardization, the random terms have a distribution we model as  $\epsilon \sim \mathcal{N}(0, \sigma_0)$ , with a standard deviation  $\sigma_0$  estimated below. The difference between any two sites' values is normally distributed, being a sum of normal terms. Since the random terms are independent, the standard deviation between sites at distance  $d_{\text{rw}}$  follows the relation,

$$\sigma \sim \sigma_0 \sqrt{d_{\text{rw}}}. \quad (38)$$

The expression for Bayesian inference when the prior and likelihood function are both normal is well known (see e.g. [Bishop, 2006](#), Sec. 2.3). In our case, it is given by

$$y_{\text{pred}} = \frac{y_{\text{nn}}}{1 + \sigma^2}. \quad (39)$$

To estimate  $\sigma$ , we use fundamental properties of percolation on a Bethe lattice. First, the minimum path length  $l$  between two sites on a percolation cluster is related to the geometric distance  $R$  via an exponent  $D_{\text{min}}$  such that  $l \propto R^{D_{\text{min}}}$  ([Stauffer & Aharony, 1994](#)). For a Bethe lattice,  $D_{\text{min}} = 2$ . We recall that the size of a percolation cluster scales with the geometric length as  $s \propto R^D$ , where for a Bethe lattice  $D = 4$ . We then have the relation

$$l \propto \sqrt{s}. \quad (40)$$

Combining Eq. 38 and Eq. 40 to obtain a fixed marginal standard deviation with  $d_{\text{rw}} = l$  yields  $\sigma_0 \propto s^{-1/4}$ . Applying Eq. 38 again with  $d_{\text{rw}} = d_{\text{nn}}$ , we obtain for the variance,

$$\sigma^2 = \text{const} \times \frac{d_{\text{nn}}}{\sqrt{s}}. \quad (41)$$

Empirically, a constant of 1 works well. Combining Eq. 39 and Eq. 41 yields the estimator,

$$y_{\text{pred}} = \frac{y_{\text{nn}}}{1 + \frac{d_{\text{nn}}}{\sqrt{s}}}. \quad (42)$$

## I. Scaling Laws for Individual Clusters

Fig. 8 shows example scaling curves for individual clusters. After an initial break, each curve appears consistent with predicted manifold-approximation scaling, with power-law exponent  $c/D = 0.5$ .

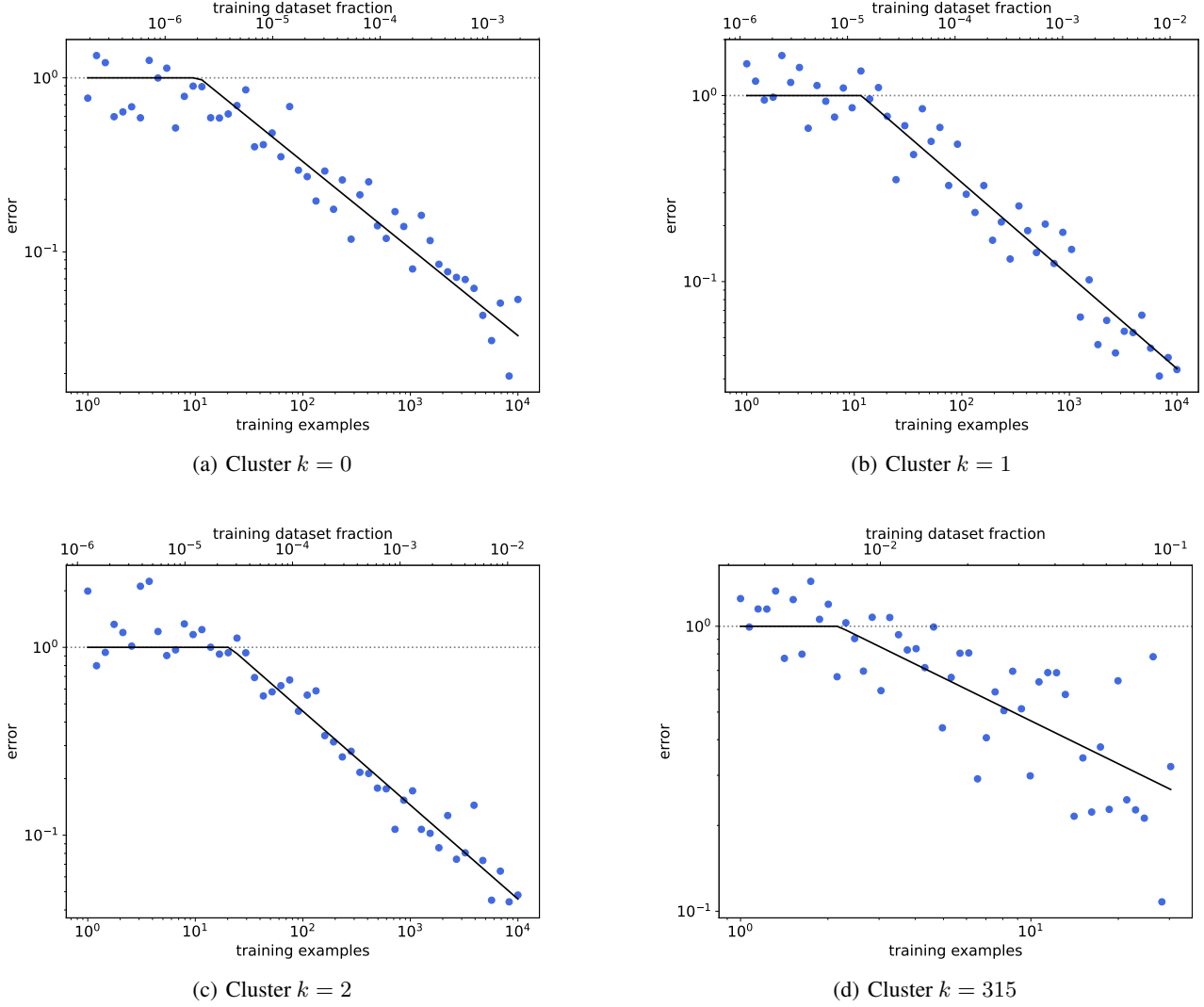


Figure 8. Scaling curves for training and test sets restricted to individual clusters. Fits are shown in black, with the only free parameter for each being the break location. The first three plots show a dataset’s largest three clusters, while the last shows its smallest cluster.

## J. Parameterized DOF Allocation

Fig. 9 shows the loss achieved with parameterized DOF allocations given different total values of DOF  $N$ . The case  $N = 500$  is shown in the main text in Fig. 5.

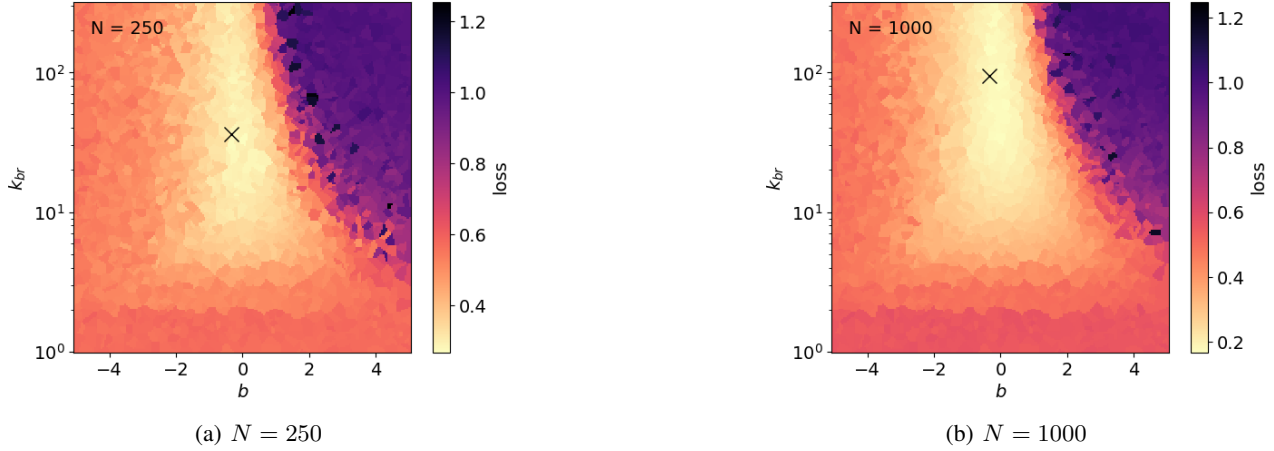


Figure 9. Loss achieved by function approximators with parameterized DOF distributions, where  $N$  is the total number of DOF. Black crosses indicate the parameters predicted to be optimal for model scaling.

## K. Scaling Laws and Emergent Capabilities

In this work, we examined scaling laws relating a machine learning system’s scale to its overall loss, but capabilities on given tasks are what determine utility and safety. Loss and capabilities can have a complex, nonlinear relationship. In particular, LLMs appear to display emergent capabilities<sup>8</sup>, or abilities arising abruptly and unpredictably at particular model scales (Brown et al., 2020; Wei et al., 2022; Srivastava et al., 2023). It remains unclear how best to reconcile discontinuous capability emergence with smooth power-law scaling laws. Indeed, Schaeffer et al. (2023) suggested that apparent emergent capabilities may be artifacts of discontinuous metrics rather than fundamental model behavior changes.

Several works have proposed phenomenological models to explain emergent capabilities. One approach models emergence through the composition of subtasks, since good task performance would emerge abruptly only once every subtask is learned (Okawa et al., 2023; Arora & Goyal, 2023). Notably, Lubana et al. (2024) used percolation theory to model emergence in transformers trained on a formal language. Lubana et al. (2024) followed a very different approach than the one we employed, as they used percolation theory to model a network’s learning dynamics rather than a static data distribution.

In addition, the quantization model of Michaud et al. (2024) yields a simple model of emergent capabilities, since a capability involving only one quantum would appear abruptly at the scale that it’s learned. Michaud et al. (2024), borrowing genetics terms, called a prediction problem “monogenic” if it involves only one quantum and “polygenic” if influenced by multiple quanta. They examined model scaling curves for individual natural language prediction tasks, finding mostly smooth scaling curves, but with some exhibiting abrupt performance improvements at particular scales. Michaud et al. (2024) identified smooth scaling curves with polygenic behaviors and abrupt ones with monogenic behaviors.

Our theory’s subcritical regime broadly accords with Michaud et al. (2024), but with additional complicating considerations. First, polygenic behavior occurs if a prediction problem is compatible with multiple subtasks due to noisy, masked, or ambiguous input, making the optimal behavior a mixture distribution. Second, because many subtasks are learned in parallel, both monogenic and polygenic prediction problems can manifest smooth scaling curves. Third, performance on a problem can indeed improve abruptly at the smallest model size a relevant quantum is learned. However, cluster functions are modeled smoothly, so even monogenic problems might not yield noticeable discontinuities. Finally, violations of general-purpose learning (e.g. exploitation of memorized surface-level patterns) have undefined scaling behavior, possibly compatible with abrupt performance improvements.

<sup>8</sup>In this work, we also invoked “emergence” in an unrelated context to describe how power-law-distributed cluster structure appeared in the data distribution without being explicitly assumed.

## L. Chinchilla Scaling Law

Fig. 10 shows the relationship between model width and number of parameters in the Chinchilla model family, using the architectural hyperparameters reported by [Hoffmann et al. \(2022, Table A9\)](#). We fit the parametric relation  $d_{\text{model}} = CP^\beta$ , obtaining best-fit parameters  $C = 0.57 \pm 0.05$  and  $\beta = 0.387 \pm 0.004$ .

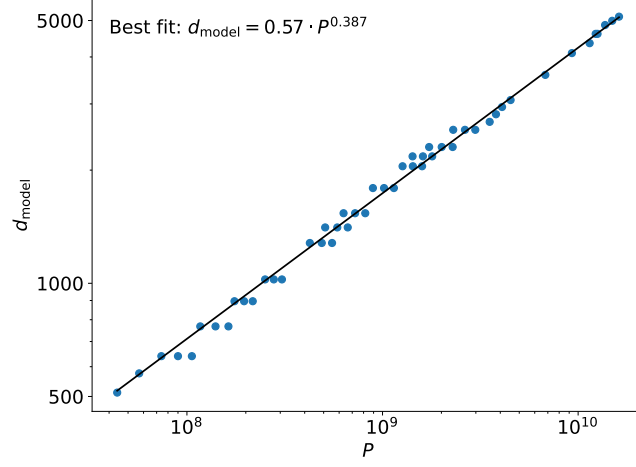


Figure 10. Relation between model width  $d_{\text{model}}$  and parameters  $P$  for various sizes of the Chinchilla model family. Data points are those reported by [Hoffmann et al. \(2022, Table A9\)](#).