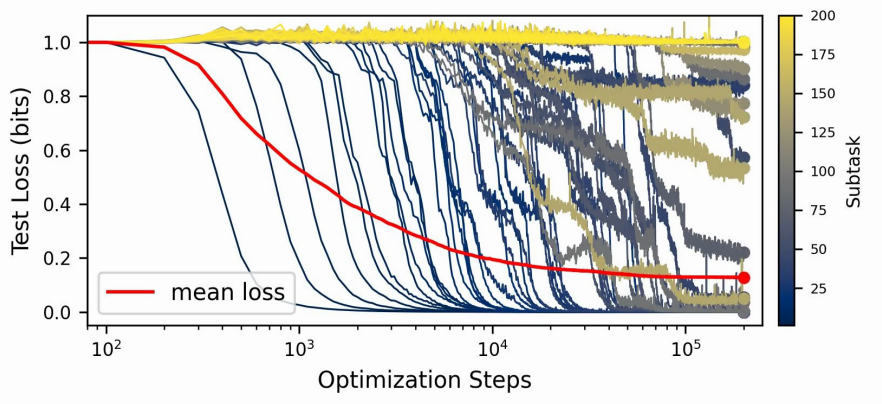
On neural scaling and the quanta hypothesis

Eric J. Michaud

January 12, 2026



This essay first appeared on ericjmichaud.com/quanta, and is also posted on learningmechanics.pub/quanta. This version includes an additional related work section and typo fixes from May-August 2026.

Contents

Introduction

Several years ago, humanity began an experiment. This experiment is now on track to become one of the most expensive ever attempted. We don't yet know what the results of this experiment will be. Certain results could be transformative. The experiment could not only alter the dynamics of wealth and power in human life, but change our basic status as a species: it could put us in contact with alien minds more capable than our own for the first time. Alternatively, some still predict that the experiment will be largely a waste of time and energy. Despite its importance, we don't have a mature theory backing this experiment. And the number of people working on such a theory in public is relatively small.

I am referring to the experiment of scaling up deep neural networks. What happens when we train neural networks, like large language models, with as many parameters as possible, on as much data as possible, with as much compute as possible? A handful of private labs are running this experiment at the moment. They will each spend some tens of billions this year, and possibly hundreds of billions or more over the next few years. Exactly what new capabilities will the next generation of models have? What are the limits of this process? And exactly why is so much data and so much compute needed to train these models in the first place?

Immense resources are staked on these questions.[1] Yet we lack clarity on their answers because *deep learning theory* is still an immature science. The core source of uncertainty is a fundamental one: we don't really know how to think about what neural networks are doing internally. Of course, a huge amount of academic work bears on this topic. This includes empirical work on mechanistic interpretability (Olah et al. 2020; Marks et al. 2024; Wang et al. 2022; Templeton et al. 2024), more theoretical work on kernels (Jacot et al. 2018; Bordelon et al. 2020; Simon et al. 2021), work characterizing which algorithms transformers can express and learn (Merrill et al. 2022; Delétang et al. 2022; Dziri et al. 2023), and much else. However, it still feels early, like we don't have a unified picture yet. I believe a more unified, mature science of deep learning would provide the following:

> (1) A framework for thinking about the internal mechanisms that neural networks learn that explains how they generalize.
>
> (2) An account of how various "engineering" choices, such as architecture, hyperparameters, data, scale, etc., determine which mechanisms networks learn, and how and when they learn them.

So far, the works that arguably come closest to this ideal are models of neural scaling. Such models attempt to describe how neural networks *change* with increasing parameters, data, or training steps, and thus at least implicitly articulate a model of what neural networks do in the first place. Some of these works include Sharma and Kaplan (2020), Bordelon et al. (2020), Hutter (2021), Bahri et al. (2021), and Maloney et al. (2022).

In early 2023, I proposed a model of neural scaling, together with Ziming Liu, Uzay Girit, and Max Tegmark, in a paper called *The Quantization Model of Neural Scaling* (Michaud et al. 2023), which we presented later that year at NeurIPS. This paper was ambitious. It tried to reconcile high-level facts about neural scaling with the way that mechanistic interpretability folks think about neural networks at a very

[1] In a recent interview (Amodei 2025), Dario Amodei estimated that "I would say maybe 20 trillion of capital is on the side of 'accelerate AI as fast as possible'."

low level. It made some strong, though informal, conjectures about the structure of data and the structure of the solutions that neural networks learn. It tried to articulate a more unified picture.

Over the last two years, I have been thinking a lot about that paper and its ideas—[2]about the parts that I am proud of, as well as the limitations of our theory. In this post, I'll share some of these thoughts. I'll first give a refined statement of our theory and its motivation. I'll then discuss a variety of topics related to the theory, including its connection to topics in mechanistic interpretability. I'll also discuss various potential problems with the theory itself and how/whether these problems might be resolved. Ultimately, this discussion will end up being pretty academic. This essay does not give satisfying answers to the most practically relevant questions today about neural scaling. Yet I hope that this discussion will, in the bigger picture, still help make progress on one of the most important open questions today.

What are neural networks doing internally,
and what happens when we scale them up?

Note: People mean different things when they talk about "scaling" these days. In this post, I am just focused on understanding pretraining scaling. But even if posttraining scaling will be more important for improving AI capabilities going forward than scaling pretraining, I suspect that the discussion here will still be relevant, and that further progress on our ability to pretrain on a wide distribution of data will have a role to play in improving AI capabilities over the coming years. Alternatively, if pretraining has essentially plateaued, we ought to be curious about *why* it has and what it means for what comes next.

1 The theory: "quanta" of learning

How does scaling change what neural networks learn? How do larger networks pretrained on more data end up being different from smaller networks trained on less data? We would like a theory of neural scaling that explains both how network *performance* and network *internals* change with scale.

1.1 Background

The most important fact about neural scaling is that network performance, in the aggregate, is predictable. Across domains and network architectures, a large number of studies (Hestness et al. 2017; Rosenfeld et al. 2019; Kaplan et al. 2020; Henighan et al. 2020; Zhai et al. 2022; Hoffmann et al. 2022) have observed that *mean test loss*, taken across the *whole data distribution*, scales smoothly as a function of the resources used to train the network. In particular, mean loss *scales as a power law* with the number of network parameters, the number of samples in the training set, and the number of steps of training. These predictable loss curves are referred to as "neural scaling laws", and since the loss follows a power law, they appear as a straight line on a log-log plot:

[2]All content in this post was authored by me and not by LLMs—I just like em dashes.

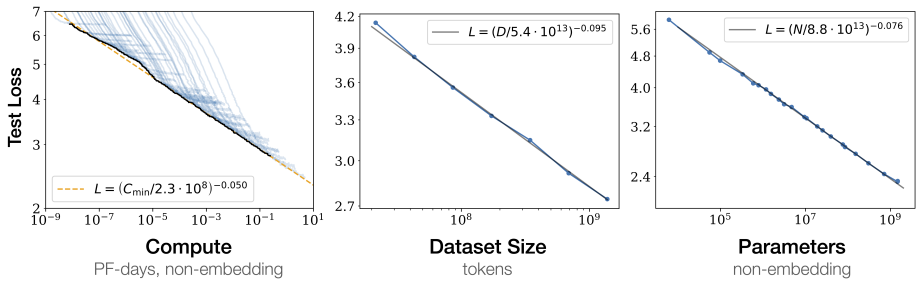


Figure 1 of Kaplan et al. (2020), showing neural scaling laws for language models w.r.t. compute, data, and number of network parameters. When network size is large (not a bottleneck), then loss scales with D, the number of tokens in the training corpus as $L(D) \propto D^{-0.095}$. When data is not the bottleneck, then loss scales with N, the number of (non-embedding) parameters in the network, as $L(N) \propto N^{-0.076}$. Each of these are power laws in D and N respectively. The total compute used to train the network is $C \propto N \times D$, and with an optimal choice of N and D for a given compute budget C, loss scales with compute as a power law too: $L(C) \propto C^{-0.050}$.

It is fascinating that there is this kind of predictable order to neural network performance. As in thermodynamics, where the macroscopic behavior of systems of many particles is predictable (e.g. doubling the temperature of a gas doubles its pressure), the macroscopic properties of neural networks, *systems of many parameters*, vary predictably too. While scaling laws like this are fairly ubiquitous in nature (West 2017), it is striking that modeling human language, a task so complex and perhaps indicative of *intelligence*, would also be so predictable.

The discovery of these scaling laws was also, more practically, a key motivator for Dario Amodei and others at OpenAI to scale up large language models around 2019. As long as the scaling curve has not leveled off, then there are predictable gains to be made *just* by scaling up existing techniques. And so that is roughly what OpenAI did to go from GPT-2 to GPT-3. And just scaling things up unlocked a range of capabilities that were not present in smaller models at the time.

While network performance in the aggregate, as measured by the mean cross-entropy loss, scales predictably, more narrow properties of models, such as their performance on specific tasks, can be harder to predict. Indeed, it seems that for some abilities, larger models can be sharply more capable than smaller models. Such abilities, which large models possess but small models qualitatively do not, are said to "emerge" with scale (Wei et al. 2022). Here are some examples of the scaling behavior for such "emergent" abilities:

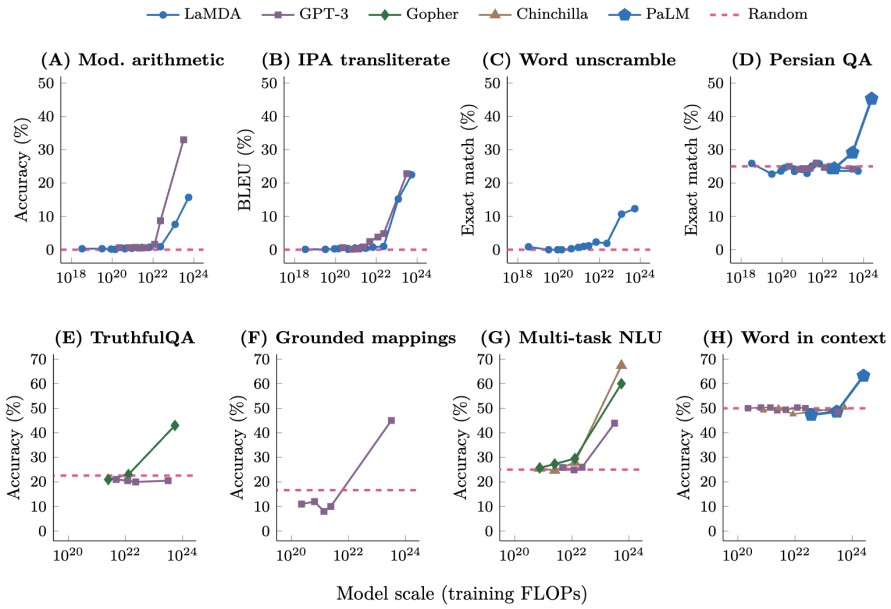


Figure 2 of Wei et al. (2022), showing "emergent" abilities of large language models, where performance is negligible below some threshold of scale and then jumps up after some critical scale.

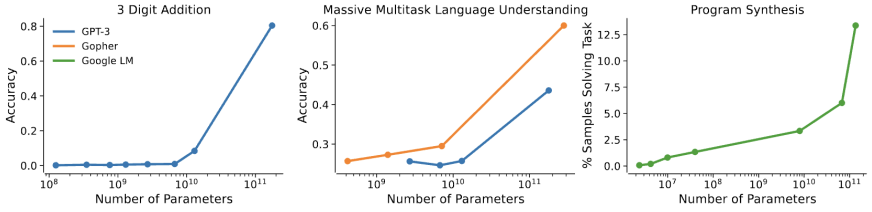


Fig. 2 Three examples of abrupt specific capability scaling described in Section 2.2, based on three different models: GPT-3 (blue), Gopher (orange), and a Google language model (green). **(Left)** 3-Digit addition with GPT-3 [11]. **(Middle)** Language understanding with GPT-3 and Gopher [62]. **(Right)** Program synthesis with Google language models [4].

Figure 2 of Ganguli et al. (2022), showing some additional LLM emergent abilities. This figure compiles data from Brown et al. (2020), Rae et al. (2021), and Austin et al. (2021).

One fun example of an emergent ability was demonstrated in the original GPT-4 release live demo back in March 2023, when Greg Brockman showed that GPT-4 could write a summary of an article using only words starting with the letter 'G'. Greg also gave GPT-3.5 this task, and it failed completely.

Sharp transitions in language model performance can also be observed over the course of training for an individual model. The canonical example of this is the "induction heads" phase change documented in Anthropic's *In-context Learning and Induction Heads* (Olsson et al. 2022). One thing that transformer-based LLMs learn early in training is to be able to copy arbitrary subsequences that occur in the context. So if [A][B] occurred earlier in the context, and the current token is [A], then LLMs are very good at predicting [B] as the next token, for arbitrary [A] and [B]. Anthropic described the circuit that implements this operation (it requires two self-attention layers), and found that it forms early in training, producing a sharp transition in the model's subsequence copying capability:

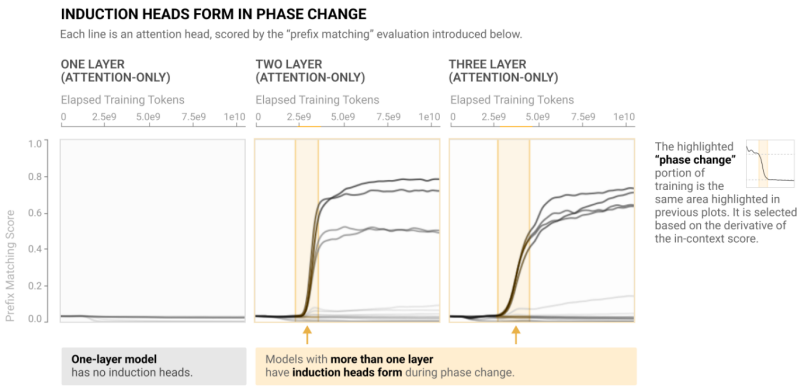
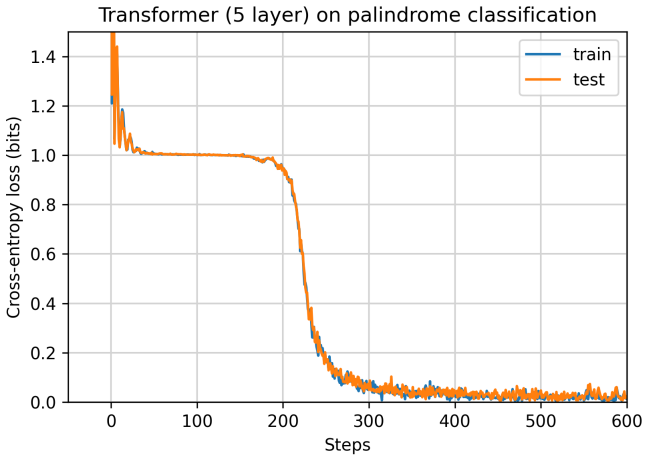


Figure from Olsson et al. (2022) showing that the "prefix matching score" jumps sharply during transformer training, as long as the model has at least two self-attention layers.

These sorts of "phase changes" are somewhat common (though not necessarily universal) when training neural networks on algorithmic tasks (tasks that can be solved by short Turing machines). Neel Nanda documented several of these in his post on grokking (Nanda and Lieberum 2022). One clear example, which I found when playing around with these sorts of tasks, is the particularly sharp transition exhibited by transformers trained to classify whether strings are palindromes, where loss plateaus at 1 bit of cross-entropy and then drops suddenly:



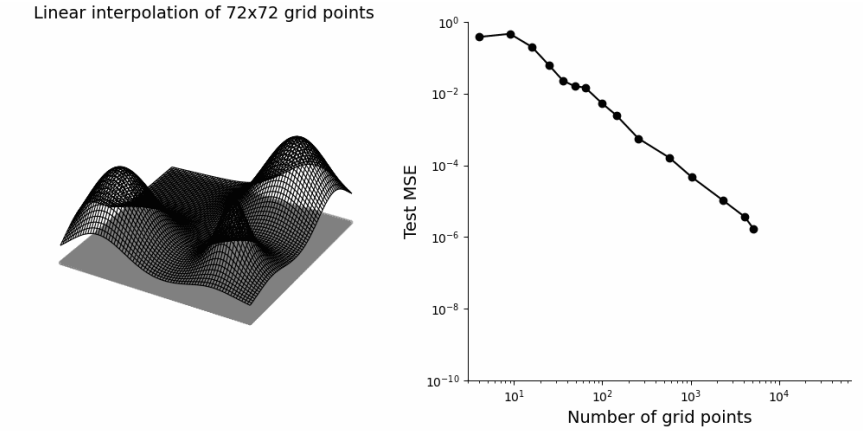
Learning curve for a 5-layer GPT-2 style transformer trained to output whether the sequence seen so far is a palindrome.

It feels like there is some tension between the smoothness of neural scaling laws and the discreteness of emergent abilities. To understand this tension, let us consider another model of neural scaling, that of Sharma and Kaplan (2020).

Sharma and Kaplan propose a simple model of neural scaling. In essence, they make a very general argument from approximation theory. If we think about neural networks as learning to approximate some true target function, then larger networks can approximate this function with higher precision, and scaling them up gradually improves the resolution of this approximation.

To illustrate this principle, below I plot the scaling curve of test error for a piecewise linear approximation to an arbitrarily-chosen function $\mathbb{R}^2 \to \mathbb{R}$. We see (left) that

as we increase the resolution of this piecewise linear approximation, requiring our model to "memorize" the value of the function at more points, that mean squared error eventually settles into smooth power law scaling (right). "Training" points, which the model interpolates between, are shown in the x-y plane below the surface of the function on the left.



A frame from an animated visualization of piecewise-linear function approximation. As resolution increases, mean squared error settles into power-law scaling.

This view can explain scaling laws, and does seem to describe neural scaling on some datasets, but it also seems to conflict intuitively with the more discrete transitions in performance we can observe when training language models. In these discrete transitions, like in the "induction heads" example, the network learns an *algorithm* that allows it to generalize in a manner that is more sophisticated than merely interpolating between training points.

It seems more natural to me to think about large language models as consisting of a large number of computational modules, each implementing some specialized algorithm. This is like in Marvin Minsky's "Society of Mind" picture (Minsky 1986), where minds are thought of as being composed of many individually mindless "agents". If each such algorithm were learned in a discrete transition, like the induction and palindrome examples above, then how could we reconcile this with the smoothness of neural scaling laws?

We will explore the following resolution: that phase changes are indeed real and common, but that each phase transition individually only affects network performance on a small fraction of the data, and so the mean loss can decrease seemingly smoothly as these transitions occur at different times. The discreteness is "averaged out" in the mean loss. We will see that to recover power law scaling of the loss, we must assume that there is a power law deep in the statistics of the data.

This basic way of thinking was suggested in Nanda and Lieberum (2022), mentioned earlier. Neel speculated that "phase changes are everywhere":

> *. . . larger models are made up of many circuits and, though each circuit may form in a phase change, the overall loss is made up out of the combination of many different capabilities (and thus many different circuits). And circuits of different complexity/importance likely form at different points in training. So the overall loss curve is actually the sum of many tiny phase changes at different points in training, and this overall looks smooth and convex. Regularities in loss curves like scaling laws may be downstream of statistical properties of the distribution of*

circuits, which become apparent at large scales.

In our paper on scaling, *The Quantization Model of Neural Scaling* (Michaud et al. 2023), we sharpened up this basic picture into a more precise model of neural scaling, and then performed a variety of experiments to explore whether something like this picture could really be going on. I'll now summarize this work.

1.2 The Quanta Hypothesis

What sort of mind does pretraining produce? What must be learned, to minimize loss in predicting the next token, across all the tokens in all the documents on the internet?

I think the answer is an immense conglomeration of different skills and pieces of knowledge. There is a huge amount of information conveyed in language, and there are also a huge number of different computations that are performed in the processes that have produced the text on the internet. This information, and these algorithms, would ideally be learned by our language models (Gwern 2020; janus 2022).

Let's imagine that we could enumerate all of the pieces of knowledge and algorithms that networks ought to learn. Our core assumption is the following: for any piece of knowledge or algorithm that a network ought to learn, it will either be fully learned or not learned at all. It is not optimal for any network to allocate some capacity to halfway learning some algorithm and some capacity to halfway learning another. Either you succeed in implementing a computational module in the weights or you do not. We call this basic assumption, of discreteness, the "quanta hypothesis" or "quantization hypothesis":

> Networks must learn a variety of modules, each implementing a different algorithm or retrieving a different piece of knowledge. These modules are *discrete*, in the sense that they are either fully learned or not learned at all. We call these the **quanta**.

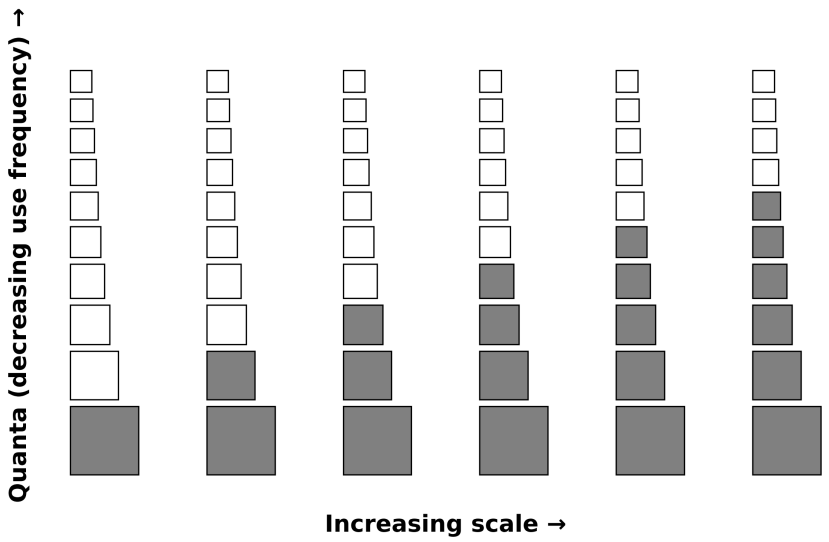
In using this terminology "quanta", we are making an analogy to physics, to Max Planck's assumption in 1900 that energy comes in discrete packets: energy quanta. Here we make a similar assumption that learning is quantized, and comes in discrete chunks.

We just need one more assumption about these quanta, which is that some quanta are much more frequently needed than others. For each "quantum", we can imagine identifying all the tokens across the corpus on which that quantum helps the model do prediction better. Some quanta might be needed on a very large fraction of tokens but others might only be relevant for a few tokens. This seems pretty intuitive—some knowledge is very frequently referenced in text, and other knowledge is quite esoteric, and so rarely needed when predicting the next token. We can imagine that these frequencies could vary over several orders of magnitude. In order to recover neural scaling laws, we need to assume the following:

> The "use frequencies" of the quanta naturally follow a power law.

Let's order the quanta according to their "use frequency". We refer to the set of

quanta, ordered in this way, as the "quanta sequence". Soon we will argue that the effect of scaling, either in parameters or data, is to *simply add more and more quanta to the network in the order of the quanta sequence.* Altogether, this gives an extremely simple picture of neural scaling: scaling simply adds discrete modules, of increasing nicheness, to the network:



Let p_k be the use frequency of the kth quantum, the fraction of tokens on which it influences the model's prediction. Our assumption is that $p_k \propto k^{-(\alpha+1)}$, a power law. Let's assume that adding some quantum to the model reduces the model's loss by an amount Δ, on average, on the tokens on which it influences the model's prediction. On other tokens, that quantum doesn't influence the model's loss. So if we learn quantum k, this lowers the mean loss, across the whole corpus, by $\Delta \cdot p_k$. We assume that this Δ is the same across all quanta.

What if we learned the first n quanta? Then the mean loss is:

$$L(n) = L(0) - \sum_{k=1}^{n} \Delta \cdot p_k,$$

where $L(0)$ is the model's loss when no quanta have been learned. For instance, we could imagine $L(0)$ being the loss of a model that outputs a uniform distribution over the tokens, or which outputs the individual token frequencies, but otherwise has learned no knowledge or sophisticated computational modules. If $p_k = Ak^{-(\alpha+1)}$, we can approximate this sum as an integral, giving:

$$\begin{aligned} L(n) &\approx L(0) - \Delta \cdot A \int_1^n k^{-(\alpha+1)}\, dk \\ &= L(0) - \Delta \cdot A \left[\frac{k^{-\alpha}}{-\alpha}\right]_1^n \\ &= \left(L(0) - \frac{\Delta \cdot A}{\alpha}\right) + \frac{\Delta \cdot A}{\alpha} n^{-\alpha} \end{aligned}$$

and so:

$$\boxed{L(n) \approx L_\infty + Cn^{-\alpha}.}$$

where $C = A\Delta/\alpha$.

It may be unrealistic to assume that Δ is constant for all quanta, but I think as long as the true Δ_k are independent across k then this would just add a bit of noise that would be averaged out and not affect the scaling law much once you get far enough out along the quanta sequence. We have shown that the mean loss scales as a power law in the number of quanta learned n. But the number of quanta learned is an abstract quantity, and neural scaling laws are measured w.r.t. network parameters or dataset size, so we need to do some additional work to argue how these resources influence how many quanta networks can learn:

Parameter scaling. Let's imagine we train a network on plentiful data, so data is not a bottleneck. What will the network learn? Optimally, it will learn as many quanta as it has capacity for, in order of frequency, because the most frequent quanta reduce the mean loss the most. If the network has N parameters, and on average each quantum requires c parameters of capacity to implement, then the number of quanta learned will be roughly $n \approx N/c$. We therefore get scaling w.r.t. network size: $L(N) \approx L_\infty + C(N/c)^{-\alpha}$ so:

$$\boxed{L(N) \approx L_\infty + C_N N^{-\alpha}}$$

where $C_N = Cc^\alpha$. We see then that the neural scaling law w.r.t. parameters has slope $\alpha_N = \alpha$. So our power law in the quanta frequencies with slope $\alpha+1$ translates directly into a power law in the loss with slope α.

Data scaling. Let's imagine we do multi-epoch training with a total of D samples, with a large network, so network capacity is not the bottleneck. How many quanta can we learn? Well, for rare quanta, where $p_k \ll 1/D$, it is unlikely that any tokens involving that quantum will be in the dataset, and so it won't be learned (the training data does not give any incentive/signal to learn it). Let's imagine that, in order to learn some quantum k, τ samples (tokens) involving quantum k must be in the training dataset. Clearly, the early quanta with higher p_k will be more likely to be learned than the later quanta. Roughly, the scale n of the last quantum learned will satisfy: $\tau \approx Dp_n$. So if $p_k = Ak^{-(\alpha+1)}$, then we get $n = (AD/\tau)^{1/(\alpha+1)}$. Plugging this into $L(n)$, we get:

$$\boxed{L(D) \approx L_\infty + C_D D^{-\alpha/(\alpha+1)}}$$

where $C_D = C(\tau/A)^{\alpha/(\alpha+1)}$.

Step scaling. What about scaling in training steps S for single-epoch training? We could use a similar model as above, where a certain number of samples need to be seen in training before a quantum can be learned, which would yield the same scaling law. We also suggest the following model: since the amount that each quantum reduces the training loss follows a power law, perhaps the magnitude of the gradients for learning each quantum also follow a power law, and so SGD would converge at different rates for learning each one. If the "distance" the network has to move is the same for each quantum, and the "velocities" (gradients) in learning each quantum follow a power law, then the time (steps) $S = \frac{\text{distance}}{\text{velocity}}$ needed to learn the nth quantum will be $S \propto 1/p_n \propto n^{\alpha+1}$, and so after S steps, the first n quanta will be learned where $n \propto S^{1/(\alpha+1)}$, giving us:

$$\boxed{L(S) = L_\infty + C_S S^{-\alpha/(\alpha+1)}}.$$

This has the same scaling exponent as our scaling law for multi-epoch data scaling above.

The big picture

This theory appeals to me because, if true, it would give us a natural set of objects to study in trying to understand neural networks: the quanta. These are like the elements of the Periodic Table[3], or, better, like all the genes in an organism's DNA—we could imagine actually enumerating them. For instance, the induction mechanism mentioned above might be the first quantum in the quanta sequence. Somewhere much later along the sequence, at an index we could imagine one day roughly identifying, there'd be a quantum that enables the model to only output words starting with a given letter, which GPT-3.5 did not have the capacity or training time to learn, but which GPT-4 did reach. For each quantum in the sequence, we could ask what that quantum does—what piece of knowledge it represents, or what algorithm it implements. We could identify what data that quantum is helpful in predicting and therefore caused that quantum to develop during training. We could track any quantum's evolution during training, and ask at what scale of compute it tends to be learned.

This theory also provides a straightforward explanation of emergent abilities: they're just the result of new quanta being learned. Furthermore, if we could somehow estimate the "use frequencies" (on the pretraining distribution) of some quanta we haven't yet learned, we could forecast the scale at which those quanta would be learned in future pretraining runs.

We imagine that models are fundamentally *modular*—in some manner, they can be decomposed into parts that can be understood independently. In hoping to understand large networks mechanistically, we could imagine enumerating which quanta that model had successfully learned, and then for any given sample identifying which quanta influenced the model's output on that sample. This is heavily inspired by preexisting ideas in the mechanistic interpretability community, particularly by Chris Olah and his collaborators. Indeed, on Twitter in 2022, Chris Olah said he'd like to see a "more circuits-y" theory of neural scaling (Olah 2022), and I think that the quanta model is one such approach. If the quanta are akin to what Chris calls "circuits", then all we are saying here is that scaling laws arise from an underlying power law distribution over how frequently different circuits are needed by the model. There has been substantial recent progress in the mechanistic interpretability community towards identifying "features" with sparse autoencoders, and I will say a bit about the relationship between features, circuits, and quanta later in this post (Section 2).

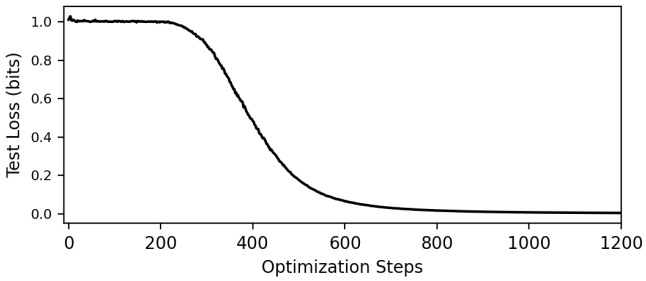
In order to recover power-law neural scaling laws, we had to assume that the "use frequencies" of the quanta follow a power law. Why would this be the case? I'm not sure, but Zipfian distributions are common in text (e.g. word frequencies), so it doesn't seem implausible that there's something a bit more abstract underlying text that's Zipfian. Speculatively, I wonder whether the quanta theory, and this question about use frequencies, are actually gesturing at something quite deep. If our neural networks are learning to reproduce the "rules" (Altman 2024) and processes that generated the data they are trained on, then the quanta hypothesis is really a hypothesis about the processes that produced text: *human minds*. Perhaps our minds are similarly decomposable into parts, and the "use frequencies" are really the frequencies at which these parts—all the ideas that inhabit human minds—were alive and active in the thinking and writing process that has produced our corpus of text across history.

[3]This reference to the periodic table in the context of universality in interpretability comes first from Chris Olah's writing in *Zoom In* (Olah et al. 2020).

1.3 Multitask sparse parity

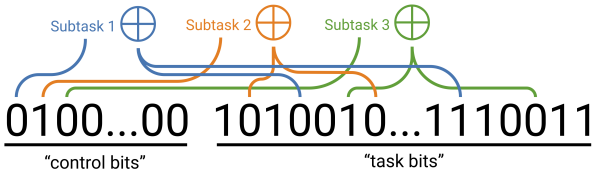
How does this theory make contact with reality? We will first show that the quanta hypothesis roughly describes neural scaling *when the data has the right structure*. Then the question will just be whether natural datasets, like autoregressive language modeling, have this sort of structure.

To do this, we will train neural networks on a synthetic task called multitask sparse parity. This is a binary classification problem on binary strings. Sparse parity had been recently studied by Barak et al. (2022), as a synthetic task where networks exhibit sharp transitions in performance during training. For instance, here is a learning curve for a ReLU MLP with a single hidden layer on sparse parity:

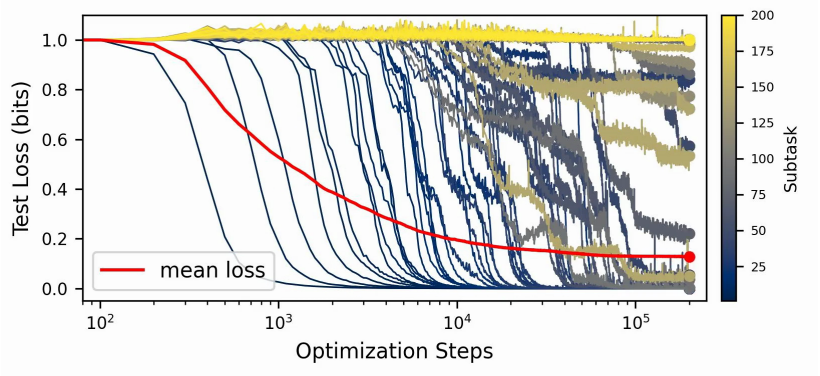


Again, this is a binary classification problem on binary strings. To define the task, some subset of the bits of the string are chosen (e.g., for 100-bit strings, indices $\{17, 53, 89\}$), and then the label of that string is the parity (sum modulo 2) of that subset of bits. To solve the task, the network has to figure out which bits to compute the parity of, just from the labeled training samples. The input strings are drawn from a uniform distribution. Typically, loss plateaus at 1 bit but then drops, giving us a non-convex loss curve.

We will construct a multitask version of sparse parity. To do this, we introduce extra bits to the input, which we call the "control bits". If we want n_{tasks} subtasks, we add n_{tasks} control bits to the beginning of our input strings, with n bits after these called the "task bits", for a total of $n_{\text{tasks}} + n$ bits. For each subtask $i \in \{1, \ldots, n_{\text{tasks}}\}$, we sample a random subset S_i of k bits from the task bits. The n_{tasks} control bits one-hot encode the subtask, so if bit i is a 1, then the label for that string is the parity of the task bits S_i for that subtask. We impose a power law distribution over the subtasks, so the probability that bit i is a 1 (and all other control bits are 0) is $p_i \propto i^{-(\alpha+1)}$.



The training dynamics on this task are really beautiful. We train ReLU MLPs with a single hidden layer using the Adam optimizer. Here is how the loss for each subtask (colored dark blue to yellow), and the mean loss (colored red), evolves during training:



Training dynamics for a network trained on multitask sparse parity (final frame of an animated visualization).

We see here that while the mean loss decreases gradually, the network's loss on each subtask drops more sharply, and these subtasks are learned at different times. The network achieves ≈ 0 loss on the most frequent subtask after 10^3 steps, but learns the less frequent subtasks multiple orders of magnitude later (like after 10^5 steps).

The experiment for the plot above generated batches in an online manner, with a large enough n so that samples were unlikely to be seen multiple times in training, so this experiment was a study in scaling w.r.t. steps in the single-epoch regime. We can also study scaling w.r.t. parameters, by training networks of varying width in the same manner. And we can study scaling w.r.t. data in the multi-epoch regime by training wide networks on finite datasets of varying size and using early-stopping (taking the checkpoint where the test loss was lowest). Here are the combined results from scaling parameters, steps, and data, showing mean loss scaling (top) and scaling broken down by subtask (bottom):

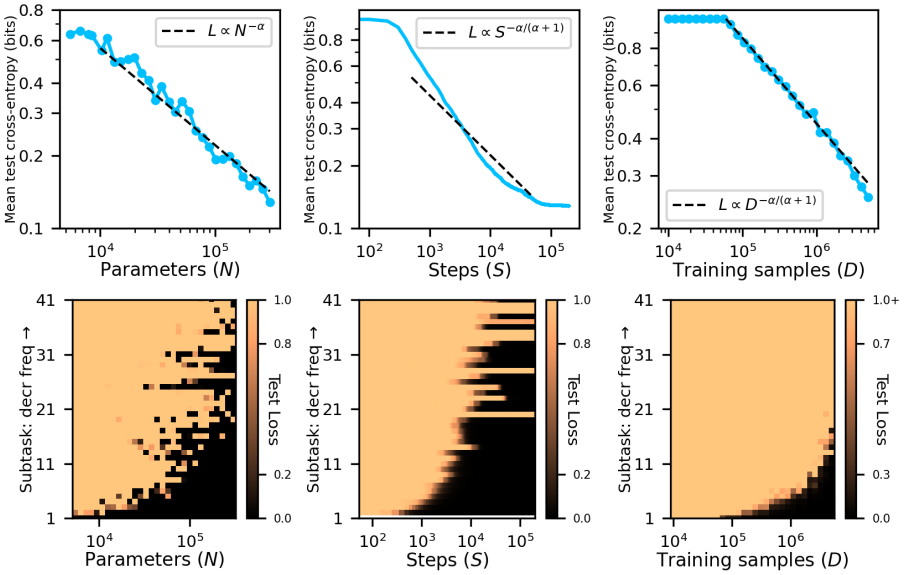


Figure 2 of Michaud et al. (2023), showing scaling behavior on the multitask sparse parity datasets. The top plots show how mean loss scales with parameters (proportional to network width here), steps, and training set size, and the bottom plots show how scaling decomposes by subtask. We see that as we scale up network size, steps, and data, the network achieves ≈ 0 loss on more and more subtasks, roughly in order of their frequency.

In line with the quanta model above, we see that tasks are learned roughly in descending order of frequency in all regimes of scaling. As we train larger and larger networks (left), we get roughly power law scaling in mean loss (top left), and this averages over a bunch of discrete transitions as loss is brought from 1 bit of cross-entropy down to 0 bits of cross-entropy on an increasing number of tasks (bottom left). A similar story applies to multi-epoch data scaling (right).

So the quanta model roughly describes neural scaling on multitask sparse parity:

- **The task decomposes into parts:** our networks must learn a variety of subtasks.
- **Discreteness:** our networks achieve trivial performance (1 bit of cross-entropy) on most tasks for most of training, then transition to good performance (~0 bits of cross-entropy). For most of training (on a log scale), our networks either have learned or not learned a subtask.[4]
- **Scaling:** scaling either data or network size simply increases the number of subtasks learned, roughly in order of frequency.

Before moving on, I will say one thing, which is that the empirical scaling exponents $\alpha_N, \alpha_S, \alpha_D$ that we measure on multitask sparse parity do not always relate precisely to the subtask distribution exponent α in the manner that the quanta theory predicts. For instance, the relation $\alpha_N = \alpha$ does not hold precisely across all α. See Figure 10 of the paper for more on this. Ultimately I'm not sure why this is! Is it some artifact of how we're measuring the scaling exponents? Are we subtly wrong about how scaling N, D and S changes the number of quanta learned n? Later in this post (Section 8) I'll discuss related issues with applying the quanta theory to LLMs—it is possible that these discrepancies with the scaling exponents in multitask sparse parity hold clues for improving the theory more generally.

1.4 Large language model scaling

Does the quanta model describe the dynamics of training and scaling for large language models?

As a starting point, we can look at scaling curves on individual tokens. Here, by "token", I mean a particular token in a particular document. One possibility is that all per-token scaling curves look like discrete transitions, dropping rapidly after some initial plateau. Another possibility is that all curves will look like the mean loss, decreasing smoothly. If the loss on all tokens decreased smoothly like the mean loss, that would be a problem for our theory.

In practice, we find that the situation is more complicated than these two extremes. We study this with the Pythia suite of language models trained by Eleuther AI on The Pile corpus (Biderman et al. 2023). The Pythia models range from the 10s of millions of parameters up to roughly 10 billion parameters, with over 150

[4] I should note however that the paper that brought the sparse parity task to my attention, *Hidden Progress in Deep Learning: SGD Learns Parities Near the Computational Limit* (Barak et al. 2022) actually argues that the underlying learning dynamics aren't discrete at all! In fact, one can track the network's gradual progress internally during training, even if that progress isn't apparent from looking at the loss. I think this does reveal a problem with our theory, which is that we don't have a formal definition of what "discreteness" means in the learning process.

intermediate training checkpoints available for each model. This allows us to study scaling in both parameters and training steps.

Below, I show some examples of curves with different scaling behavior w.r.t. network size (parameters). The overall scaling curve w.r.t. parameters is a smooth power law, which averages over a large number of per-token scaling curves like these:

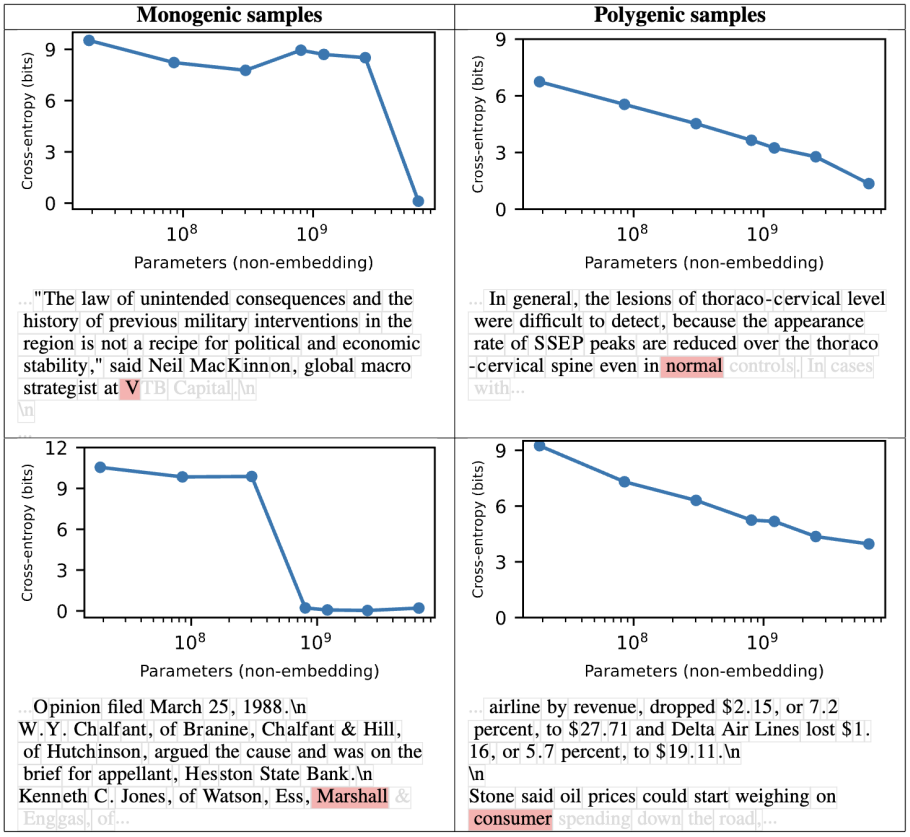


Figure 12: Additional LLM scaling curves on individual samples which exhibit sharp vs smooth improvement. If the Quantization Hypothesis is true for language modeling, then we would interpret samples with sharp drops as "monogenic" and samples with gradual progress as "polygenic".

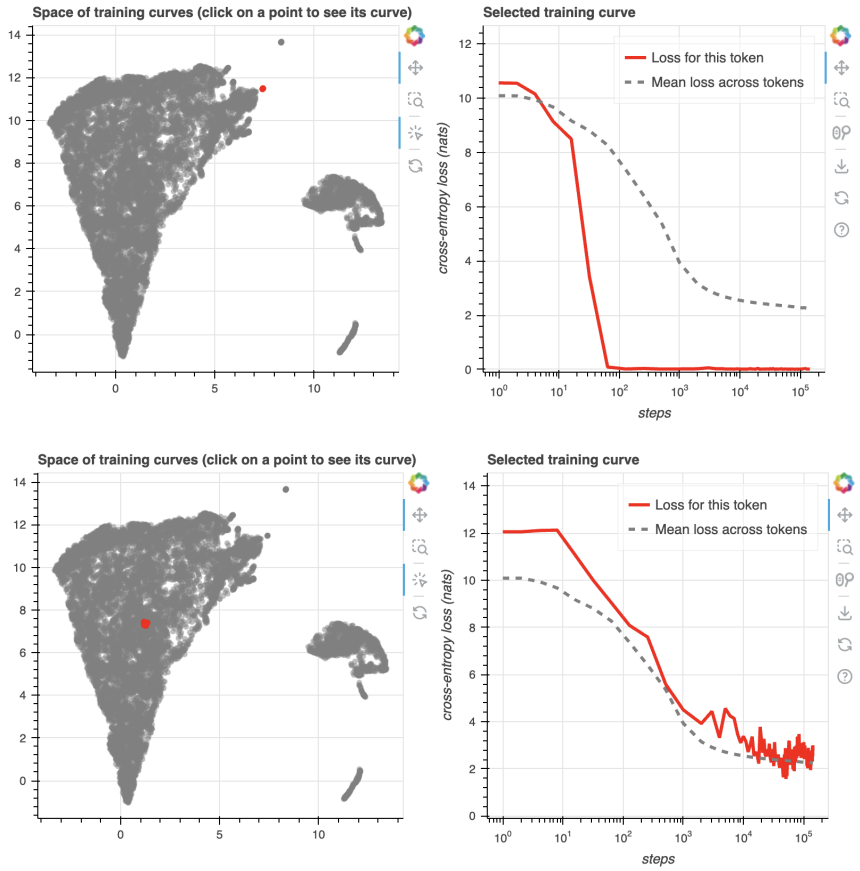
Some cherry-picked per-token scaling curves. The token highlighted in red is the token which the networks are tasked with predicting from the context and which we report the loss on.

If we assume the quanta hypothesis, then samples that have smooth scaling behavior, showing improvement at multiple scales, must be benefitting from multiple quanta that are learned at different scales. We call such samples "polygenic", in an analogy to how polygenic traits are influenced by multiple genes. When samples show improvement at only one scale, we assume they are "monogenic", involving a single quantum.

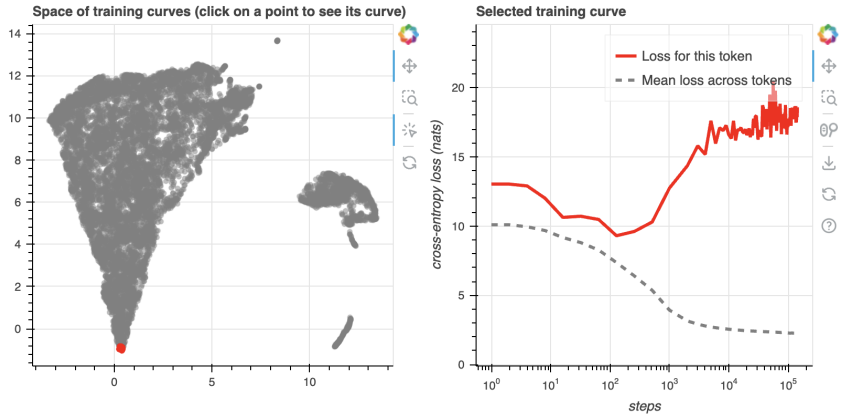
We can also visualize per-token scaling curves over the course of training for a single model. I created some interactive plots for viewing a large number of these curves on my *Space of LLM Learning Curves* (Michaud 2024) post, and which I'd strongly recommend checking out:

`https://ericjmichaud.com/llm-curve-visualization`

What is apparent from looking at a large number of per-token training curves is that the mean loss is averaging over lots of different per-token curves whose behavior is quite diverse across tokens. Some loss curves drop quite sharply early in training, other loss curves drop more gradually, and others exhibit stranger behavior like inverse scaling. Here are some screenshots from that interactive plot:



Per-token training curves for the Pythia models show diverse behavior (1/2). See the interactive plot.



Per-token training curves for the Pythia models show diverse behavior (2/2). Across the interactive plot, we can see phase transitions, inverse scaling, and other behaviors in different regions of the scatter plot.

For the theory, it perhaps would have been ideal if all per-token scaling curves exhibited sharp drops. Instead, we observe that lots of per-token scaling curves have smoother scaling behavior. To reconcile this with the quanta hypothesis, we must assume that most samples are somewhat polygenic. This seems like a reasonable assumption to me—for lots of tokens, it makes sense that lots of different heuristics could be involved in predicting that token. However, I also worry that we are adding epicycles to the theory, and it may instead be more parsimonious to reconsider our basic assumption of discreteness. I will continue with this discussion in the section below on emergence (Section 4), and also in a later discussion on the role of noise in these experiments (Section 6).

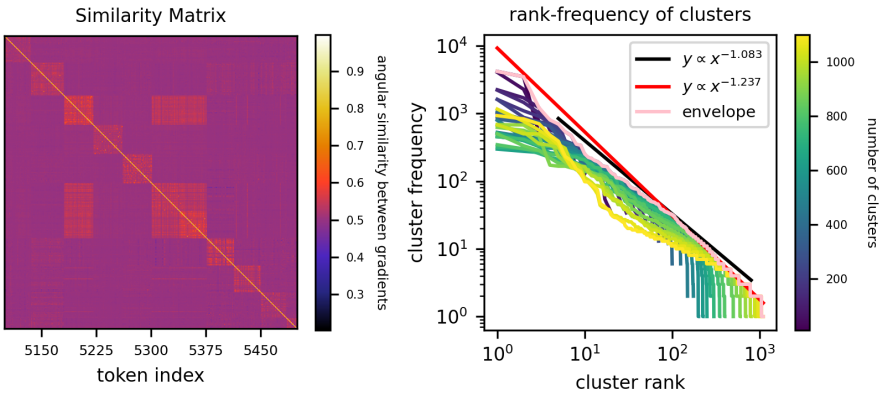
1.5 Discovering quanta

If the task of language modeling does in fact decompose into a large set of atomic skills, what are these skills? What are the quanta in LLMs?

We'd like to automatically discover these quanta. An ideal scheme would enumerate a set of mechanisms in the model and also identify which mechanisms were involved in generating the model's output on each token. We propose a method that is a bit more limited than this. It effectively only identifies one mechanism per token, and so assumes that all tokens are monogenic. Nevertheless, this method still finds some really interesting behaviors automatically.

We applied our method to a small LM, again from the Pythia series, pythia-70m. Then, we filter the tokens—we only consider tokens that pythia-70m predicts confidently and correctly—tokens that the model gets low loss on. If our LLM does not have any mechanisms for predicting some token well, we don't want to attempt to discover such a mechanism. We also filter out tokens that are part of a subsequence that occurred earlier in the context (there are a large number of such tokens that our model gets low loss on due to the induction mechanism mentioned earlier). Then, for a subset of such tokens (we did 10,000 tokens) we compute the gradient of the model's loss on that token w.r.t. its parameters. Lastly, we apply spectral clustering to these gradient vectors. This gives us clusters of tokens that have similar gradients. Our intuition here was that if a model uses similar mechanisms when predicting different samples, the gradients will be similar on those samples.

Spectral clustering computes the cosine similarity between all pairs of gradients. We visualize this matrix of pairwise cosine similarities on the left plot below. If we order the samples according to their cluster, these clusters appear as blocks along the diagonal of this matrix. We show how cluster sizes decay on the right subplot:



Left: a small section of the pairwise similarity matrix between language model gradients. Gradients are approximately orthogonal on most samples, but samples that may use similar mechanisms have gradients with nontrivial similarity. Right: cluster sizes vs. cluster rank (the y axis should say cluster size instead of cluster frequency). Our theory predicts that cluster sizes would decay as a power law.

This method identifies lots of interesting clusters. Here are five samples from a couple of the most interesting clusters:

LLM skill "quanta" auto-discovered in text

Cluster 50: incrementing numerical sequences

```
01- Mi Querencia (Simón Díaz)\n
02- Tonada De Luna Llena (Simón Díaz)\n
03- Sabana (José Salazar/Simón Díaz)\n
04- Caballo Viejo (Simón Díaz)\n
05- Todo Este Campo Es Mío (Simón Díaz)\n
06- La Pena Del Becerrero (Simón Díaz)\n
07
```

```
* * from opening a through road or street for public use across said public
park in the Park of The City of Riverton * * *." (Emphasis supplied.)
Appealing from that order, the city asserts (1) plaintiffs have no standing
or right to maintain the action; (2) that the proposed road was in an unded
icated part of the park; (3) that the proposed road was an access road and
not a through street or part of the city's street system; (4
```

```
4. _Introduction_\n
5. Chapter 1: What Is Trust?\n
6. Chapter 2: Trust Brings Rest\n
7. Chapter 3: Who Can I Trust?\n
8. Chapter 4: The Folly of Self-Reliance\n
9. Chapter 5: Trust God and Do Good (Part 1)\n
10. Chapter 6: Trust God and Do Good (Part 2)\n
11. Chapter 7: At All Times\n
12. Chapter 8
```

```
was achieved. The chosen sites were recorded as: 0 = sound (*n* = 13); 1
= first visible sign of noncavitated lesion seen only when the tooth is dried;
2 = visible noncavitated lesion seen when wet and dry; 3 = microcavitation
in enamel; 4 = noncavitated lesion extending into dentine seen as an under
mining shadow; 5 = small cavitated lesion with visible dentine: less than 50
% of surface; 6
```

```
QCBlockListMsg = 0x0a\n
GetLatestStatusMsg = 0x0b\n
LatestStatusMsg = 0x0c\n
PrepareBlockHashMsg = 0x0d\n
GetViewChangeMsg = 0x0e\n
PingMsg = 0x0f
```

Cluster 100: predicting newlines in line length limited text

```
TO PERFORM QUADRATIC REGRESSION\n
ON THE TI84 GRAPHING CALCULATOR,\n
DETERMINE HOW WELL THE \n
REGRESSION MODEL FITS THE DATA,\n
AND THEN MAKE PREDICTIONS \n
USING THE REGRESSION EQUATION.\n
IN STATISTICS, \n
REGRESSION ANALYSIS INCLUDES\n
ANY TECHNIQUES USED FOR MODELING \n
```

```
# creddump is free software: you can redistribute it and/or modify\n
# it under the terms of the GNU General Public License as published by\n
# the Free Software Foundation, either version 3 of the License, or\n
# (at your option) any later version.\n
#\n
# creddump is distributed in the hope that it will be useful,\n
```

```
<!–\n
/**\n
 * Copyright (c) 2019, The Android Open Source Project\n
 *\n
 * Licensed under the Apache License, Version 2.0 (the "License");\n
 * you may not use this file except in compliance with the License.\n
```

```
 *\n
 Pursuant to 5TH CIR. R. 47.5, the court has determined\n
that this opinion should not be published and is not precedent\n
except under the limited circumstances set forth in 5TH CIR.\n
```

```
children have a lack of maturity and an underdeveloped\n
sense of responsibility, leading to recklessness, impul-\n
sivity, and heedless risk-taking.... Second, children\n
are more vulnerable... to negative influences and\n
outside pressures, including from their family and\n
peers; they have limited contro[1] over their own envi-\n
```

Figure 1 of Michaud et al. (2023).

The token highlighted in red is the token on which the model's loss was backprop'ed on. On the left, we see a cluster of tokens that all involve predicting a number to continue a numerical sequence. This cluster has revealed a basic LLM skill: counting! Even small language models learn this skill, since it is so common to see numbered lists in text on the internet. On the right, we see a separate cluster of newlines being predicted by the model. What these samples have in common is that they are newlines in line-length-limited text. This corresponds to the skill of counting the

length of lines, and then predicting a newline to maintain the length of the previous lines[5]!

These examples are cherry picked. Some of the clusters are totally incoherent, and most of the rest reflect much simpler patterns. Here is an interactive webpage for visualizing all the clusters:

https://quanta-clusters.streamlit.app

While we obviously haven't fully enumerated the quanta in LLMs with these experiments, I still find them super cool. In a totally unsupervised manner, we were able to discover some pretty interesting narrow skills of a language model (incrementing sequences, counting line lengths). The fact that these are relatively large clusters also tells us that these skills are very commonly needed when predicting text on the internet.

In the paper, we also noticed that the envelope of the cluster rank versus cluster size curves, across different choices of n_{clusters}, eventually might follow something like a power law. However, this is super informal, and given the limitations of the clustering method itself, we ought to be cautious about interpreting this as decisive evidence for the theory.

However, we could imagine one day having a much better method for automatically decomposing language models into parts. In some ways, this has been the central focus of mechanistic interpretability research over the last two years, which I'll discuss below (Section 2).

1.6 Related work

Our work built on the work and ideas of many others. It has also been heartening to see people build on our work over the last few years. Here are just some of my favorite prior and subsequent works:

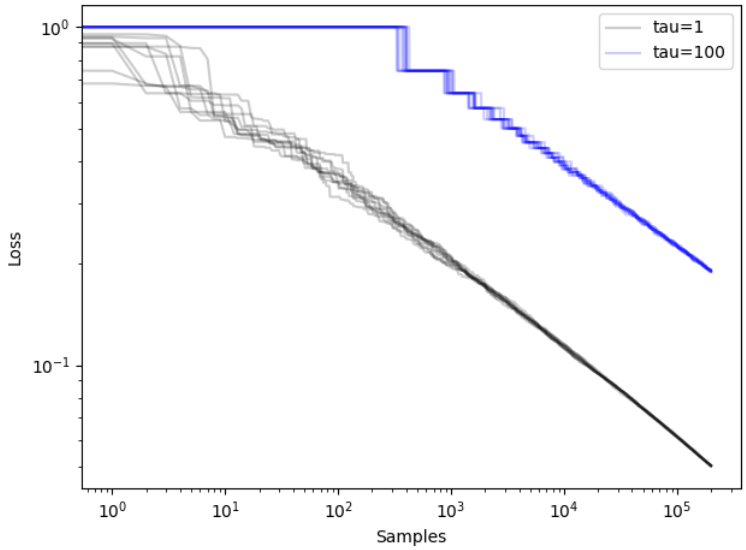
Hutter (2021). Arguably the closest prior work to ours is Marcus Hutter's very nice *Learning Curve Theory* (Hutter 2021). Hutter develops a toy model of data scaling where (1) the learning algorithm must learn a set of discrete "features", (2) a feature is learned if it is present in at least one sample from the training set, where each sample only has one feature, and (3) the features are power-law distributed in frequency. Hutter shows that this toy model gives rise to a power law in the expected loss w.r.t. training set size and training steps via theoretical derivations and toy mathematical experiments.

While Hutter also assumed that the learning problem decomposes into learning a set of discrete things, he considered a simple "tabulation learning algorithm" that simply stores previously seen features. For neural networks, it's not clear a priori that the learning problem would decompose in this way, and I think it was worthwhile for us to place a lot of emphasis on this assumption, name it, and test it empirically. In our multitask sparse parity experiments, we showed that real neural networks can exhibit scaling behavior like that predicted by our model. And we also studied scaling w.r.t. parameters in addition to scaling w.r.t. data. Overall, we were motivated by a set of empirical observations, particularly around real-world network training dynamics

[5]A recent paper from the Anthropic interpretability team (Gurnee et al. 2025) analyzed in great detail how Claude implements the linebreaking behavior (it appears universal between the tiny model we studied and Anthropic's much, much larger model) and found that there is a beautiful structure to how models represent the line length. We discuss this paper again later in the post in the section on sparse autoencoders and feature manifolds (Section 2).

that mechanistic interpretability folks had observed, and I like the empirical focus of our work and the simplicity of its presentation.

One concrete difference between the model of data scaling in Hutter's work and ours is that Hutter assumes that the number of relevant data samples needed to learn some feature is $\tau = 1$, whereas we assume that $\tau > 1$. Having a threshold $\tau > 1$ is likely more realistic for neural networks—these sorts of thresholds, where a particular number > 1 of samples is needed to learn some algorithm that generalizes, has been observed in the literature on grokking (Nanda et al. 2023; Varma et al. 2023; Liu et al. 2022). When I simulate Hutter's model, I find that with $\tau \gg 1$ there is less noise in the learning curves across "runs" (10 runs for each τ) and the features (quanta) are learned in an order that much more closely follows the frequencies of the features:



Saxe et al. (2013). The idea that the training process proceeds in a series of sharp transitions is not new. In the study of deep linear networks, pioneered by Saxe et al. in *Exact solutions to the nonlinear dynamics of learning in deep linear neural networks* (Saxe et al. 2013), one finds that networks learn singular vectors of the input-output correlation matrix in order of their singular value. The "quanta" for linear networks are the singular vectors. When network weights are initialized to be small, the transition times at which these singular vectors are learned become more separated, and the overall loss over training exhibits a series of clear, sharp drops. This setting was further studied by Saxe et al. in *A mathematical theory of semantic development in deep neural networks* (Saxe et al. 2019), who train linear networks on data with hierarchical structure, where the successive learning of singular vectors can be interpreted as learning successively finer conceptual distinctions in that data. So the overall picture of learning in linear networks follows something like the quanta model, with the bonus that the learning dynamics are analytically understandable and the quanta hypothesis is not a conjecture but can be derived (assuming the singular values of the input-output correlation follow a power law). Of course, the big question is whether a conceptually similar story applies to large, nonlinear networks trained on real-world data, and this question is the substance of our paper.

Bordelon et al. (2020). Another very closely related work is *Spectrum Dependent Learning Curves in Kernel Regression and Wide Neural Networks* by Blake Bordelon,

Abdulkadir Canatar, and Cengiz Pehlevan. They show that the generalization error of kernel regression can be written as a sum over "contributions from different eigenmodes", similar to how we ended up writing our loss as a sum over quanta. They also show that the expected error decays as a power law if the kernel's eigenvalues decay as a power law. This is a really nice, rigorous picture, and our model ended up having a similar sort of structure with less rigor but perhaps more ambition. Whereas we worked backwards from empirical observations about neural scaling, Bordelon et al. worked forwards from the mathematics of kernels (though with an assumption about the structure of the data). A big question with these sorts of kernel regression papers is just how well the dynamics of real-world neural networks, particularly LLMs, are described by kernels (Vyas et al. 2022). Blake, Alex Atanasov, and Cengiz have continued in this direction in more recent work (Bordelon et al. 2024).

Simon et al. (2021) and Simon et al. (2023). Similar to Bordelon et al., in *The Eigenlearning Framework: A Conservation Law Perspective on Kernel Regression and Wide Neural Networks* (Simon et al. 2021), Simon et al. analyze kernel regression in terms of the "learnabilities" of different kernel eigenmodes. They find a dynamic where the eigenmodes compete for a finite "learnability" resource that is proportional to the number of training samples. As the number of samples is scaled up, an increasing number of modes transition from being essentially unlearned to essentially learned, in order of their eigenvalues. So again, the quanta model is quite similar to the story of data scaling one can derive mathematically in kernel regression, if one assumes that the kernel eigenvalues follow a power law. I was ignorant of most of the kernels literature when writing our paper, but it's striking that one can derive from first principles a picture similar to the one that we had merely conjectured.

Within a week of our preprint being released in March 2023, Simon et al. released *On the Stepwise Nature of Self-Supervised Learning* (Simon et al. 2023). They study a self-supervised learning task where learning proceeds as a series of "discrete, well-separated steps". This is another task, like multitask sparse parity, that could perhaps capture some of the dynamics, particularly around emergence, we are interested in understanding in real-world networks:

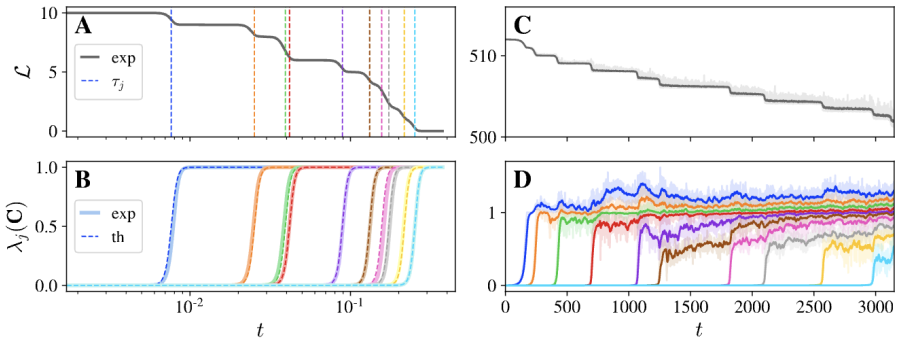


Figure 1 of Simon et al. (2023), showing learning dynamics that proceed as a series of discrete steps.

Bahri et al. (2021). Another important piece of prior work is *Explaining Neural Scaling Laws* by Bahri et al., who essentially try to unify the "approximating a function defined on a manifold" picture of Sharma and Kaplan (2020) with the kernel picture we've just discussed. This conversation is subtle and probably deserves a separate post. But they refer to a result from Weyl (1912) that relates the

eigenspectrum of kernels to the dimension of the manifold they are defined on. Via this argument, there may be a way of unifying the "resolving a function on a manifold" view and the "learning proceeds in a series of steps" view, although it is still possible that much of what real-world LLMs do is somehow not captured by this theory—their experiments are only on vision datasets, and not on language.

Nam et al. (2024). In *An exactly solvable model for emergence and scaling laws in the multitask sparse parity problem*, Nam et al. develop a more formal model of learning on the multitask sparse parity problem. It was great to see this more theoretical study building on our empirical work, and to see the paper accepted at NeurIPS last year.

Neumann and Gros (2024). In *AlphaZero Neural Scaling and Zipf's Law: A Tale of Board Games and Power Laws*, Neumann & Gros attempt to apply the quanta model to explain the neural scaling laws seen in game-playing RL. In particular, they find that the frequencies at which different game states are visited typically follow a power law, and they attempt to connect this power law to the scaling laws their RL setup exhibits. Intriguingly, they find that the games where RL exhibits inverse scaling behavior are also the games where the state distribution exhibits unusual structure where end-game states have high frequency (in contrast to games like chess where particular end-game board states are likely never repeated). This paper has me wondering whether there might be some theory, for LLMs, that unifies pre-training and RL ("reasoning") scaling laws, and perhaps some quanta-like story in terms of frequencies will have a role to play there.

Brill (2024). In *Neural Scaling Laws Rooted in the Data Distribution*, Ari also makes an attempt at unifying the "function approximation on manifold" theories of neural scaling with theories like ours that assume a power-law distribution over discrete subtasks. He does this by proposing a percolation model of data, where different tasks each have some intrinsic value p, the probability that "adjacent pairs" of data elements are "connected". Above some threshold p_c, the data is connected in a single structure and the Sharma and Kaplan (2020) story holds. Below this threshold, a transition happens and the data is broken into lots of clusters (data manifolds) that are power-law distributed in size. Ari then develops a model of neural network parameter scaling, where neural networks allocate different amounts of capacity towards learning functions on the different data clusters. In this model, there turn out to be two regimes, both producing power law scaling. In one regime, the scaling exponent α_N is still determined by the power law distribution exponent α over the data clusters, like in our model. In another regime, the scaling exponent is determined by the dimension of the data manifolds, like in Sharma and Kaplan (2020). Which scaling regime a network is in depends on the distribution over the distinct data manifolds vs. the dimension of those manifolds. This is an interesting picture, and I'd like to see more ambitious work like this thinking about the structure of data.

Braun et al. (2025). In *Interpretability in Parameter Space: Minimizing Mechanistic Description Length with Attribution-based Parameter Decomposition*, Braun et al. are interested in decomposing the parameter space of neural networks into parts. In particular, they're interested in writing the network's parameter vector directly as a sum of parts: $\vec{\theta} = \vec{\theta}_1 + \ldots + \vec{\theta}_m$, where each $\vec{\theta}_i$ implements some distinct "mechanism". They attempt to learn these $\vec{\theta}_i$ so that on any particular sample, some sparse subset of the $\vec{\theta}_i$ are needed to implement the network's behavior on that

sample. This subset is chosen by computing the dot product of the model's gradient on that sample $\nabla_\theta L$ with each mechanism vector $\vec{\theta}_i$ and taking the top components. Once these components are chosen $\vec{\theta}_{i_1}, \ldots, \vec{\theta}_{i_k}$, the model is run with the parameter vector $\sum_{l=1}^{k} \vec{\theta}_{i_l}$. So whereas we clustered gradients in our quanta discovery work, they are doing something closer to sparse coding on gradients. This is great, since it can be applied to "polygenic" samples—it allows for the identification of multiple mechanisms that contribute to a network's output on any given sample. They test the method on toy examples, but I am eager to see it applied to LLMs. In follow-up work, Bushnaq et al. (2025) develop a different parameter decomposition technique that does not use gradient information to select components.

Michaud et al. (2025a). In *On the creation of narrow AI: hierarchy and nonlocality of neural network skills* (Michaud et al. 2025a), with Asher Parker-Sartori and Max Tegmark, I explored an extension of the multitask sparse parity task where there can be hierarchical relationships between subtasks. In what we called "compositional multitask sparse parity", the subtasks vary in difficulty, and some subtasks are much easier to learn after others have been learned first. This means that the quanta for this task aren't independent, and their learning order is not solely determined by their frequency in the data. Instead, the quanta live in a kind of hierarchical skill tree. This sort of structure may be more realistic—humans can only learn some concepts and skills after first learning others—and is a nice generalization of the structure of the quanta model. These sorts of dependencies between subtasks (quanta) were also explored by Ziming Liu et al. in *Physics of Skill Learning* (Liu et al. 2025).

I wish I could discuss all the works that have in some way built on or tested our theory—over a hundred papers have cited our work now. But I'll move on now to a more detailed discussion of various topics that have come up since we released our work. This will include some additional discussion of several other papers.

2 Features, quanta, and sparse autoencoders

Over the last few years, the mechanistic interpretability community has developed many methods for automatically decomposing networks into sparsely activating mechanisms. The most prominent of these methods is the sparse autoencoder. In this section, I'll attempt to say a bit about the relationship between what we called "quanta" and the units that sparse autoencoders discover, "features".

Sparse autoencoders (SAEs) decompose the *activations* of a neural network into a sum of sparsely occurring *features*. SAEs learn a *dictionary* of *feature directions* $\{\hat{f}_i\}_{i=1}^m$ such that activation vectors can be approximated as linear combinations of sparse subsets of these features. SAEs are motivated by two hypotheses. The first is the *linear representation hypothesis*—that neural networks compute a large number of "features" from their input, and these features are represented linearly in the model's activations, meaning that when some feature is represented by the model, the presence of that feature is reflected by shifting the activations in some feature direction. The second assumption is the "Superposition Hypothesis" from Anthropic's *Toy Models of Superposition* (Elhage et al. 2022). If models represent more features m than there are dimensions in the activation space d, then these feature directions can't all be orthogonal, but they can be approximately orthogonal without interfering too much with each other if the features occur sparsely.

Sparse autoencoders try to learn these feature directions $\{\hat{f}_i\}_{i=1}^m$, and also learn an *encoder* $\text{Enc} : \mathbb{R}^d \to \mathbb{R}^m$ which determines how much each feature "fires", if at all,

on each activation vector. The sparse autoencoder loss has a form like:

$$\mathcal{L} = \left\| x - \sum_{i=1}^{m} \text{Enc}(x)_i \hat{f}_i \right\|_2^2 + \lambda \|\text{Enc}(x)_i\|_0$$

which encourages the SAE to reconstruct activations using as few feature directions as possible. Encouragingly, when we look at the situations in which these features fire ($\text{Enc}(x)_i > 0$), we find that they are often consistent, meaningful contexts: they are more *monosemantic* than other units of analysis like individual language model neurons, which fire in more varied and less consistent ways on average than SAE features. Sparse autoencoders trained on language model activations have been scaled up now to millions of features, and many of these features reveal that pretty abstract properties of inputs are being represented by models. For instance, Templeton et al. (2024) reported a feature, in an SAE with 1 million latents, that fires on tokens which introduce syntax errors in code documents.

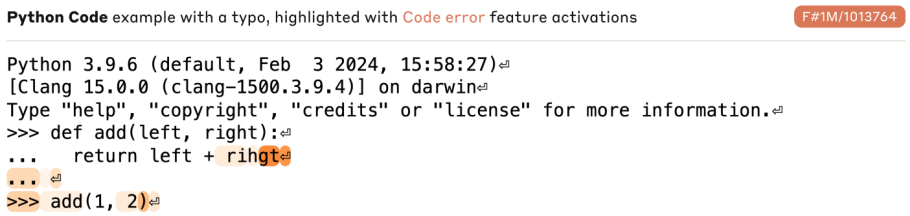
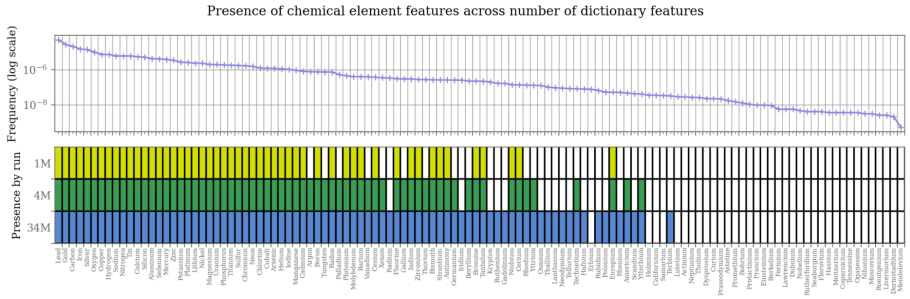


Figure from Scaling Monosemanticity *(Templeton et al. 2024), showing positions where a "Code error" feature fires.*

It is tempting to identify these features with the quanta. For each feature discovered by a sparse autoencoder in the activations of a model, we could imagine identifying that feature with some corresponding mechanism or circuit in the weights of the model which causes that feature to be computed and appear in the activations.

Indeed, the way that sparse autoencoders scale at a very coarse level is suggestive of the quanta model. As we train larger sparse autoencoders, they seem to learn features that activate on increasingly rare concepts in the corpus that the SAE (and LLM) was trained on. Anthropic showed this dynamic specifically for features which activate on references to chemical elements, where larger SAEs learn features for more niche elements. Their plot has very similar scaling dynamics to what we saw in multitask sparse parity (Section 1.3):



Plot from Scaling Monosemanticity *(Templeton et al. 2024), showing whether SAEs of different scales learn features which fire on specific chemical elements. Features for each element appear to be learned roughly in the order of the frequency at which that element is referenced in the corpus that the SAEs are trained on.*

Looking a little closer though, the properties of sparse autoencoder features and the way that sparse autoencoders scale also has disanalogies with our quanta and the quanta model.

One complication is the issue of *feature splitting* (Bricken et al. 2023). Feature splitting is a phenomenon where, if we look carefully at the effect of training larger SAEs, the effect of this scaling is not just to learn additional more niche features but to replace some common features in the smaller SAE with two or more more specific and sparsely activating features. This general dynamic means that sparse autoencoders do not necessarily learn a canonical decomposition of the activations of a network. Instead, it seems like there are many different decompositions of the space, with some decompositions having a different set of features which fire more or less sparsely than the features in other decompositions.

There is another problem with trying to identify the SAE features with quanta that is perhaps more basic: the assumption that motivated SAEs—that the activations of a neural network are composed of 1D linear feature directions—is not the full story. For some features, it seems like instead of the feature taking a scalar value along a line, the feature takes values which lie on manifolds within higher dimensional subspaces of activation space. In order for a sparse autoencoder to reconstruct these features, it will need to use multiple latents either as a basis for the subspace within which the feature manifold lies or to "tile" the manifold with feature directions. In a paper from 2024 led by Josh Engels, *Not All Language Model Features are One-Dimensionally Linear* (Engels et al. 2024), we found a few instances of these sorts of feature manifolds. These include subspaces where activations of a language model on tokens for days of the week ("monday", "tuesday", ...) and months of the year are arranged in a circle, in the correct order. We also found a subspace within which tokens for years are represented as points on a curved manifold:

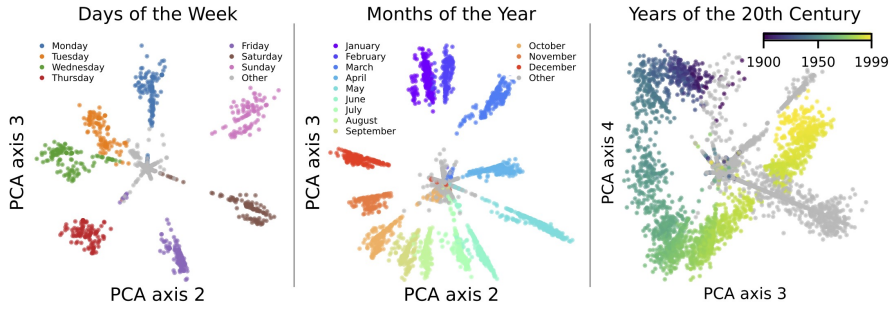


Figure 1 of Engels et al. (2024) showing multi-dimensional language model features. The existence of these features further complicates the question of whether sparse autoencoder features are a canonical set of quanta-like units to decompose neural networks into.

When these multi-dimensional feature manifolds are present in activations, multiple sparse autoencoder latents are used to reconstruct points on that manifold. In some work this year with Liv Gorton and Tom McGrath, *Understanding sparse autoencoder scaling in the presence of feature manifolds* (Michaud et al. 2025b), we investigated whether feature manifolds could cause sparse autoencoders to scale in suboptimal ways. Our basic concern was that if there is a long tail of features in the activations, sparse autoencoders could either improve their loss by further tilting common manifolds or by allocating latents to rare features. In certain situations, sparse autoencoders might continue to tile common feature manifolds with more and more latents instead of learning the rare features. Funnily enough, the math that governs whether this will happen is taken from something very similar to the quanta model. In particular, Brill (2024)'s model of neural scaling mentioned above, where

models approximate functions defined on different feature manifolds with power law frequency, ends up having the mathematical structure that I think ought to describe sparse autoencoder scaling in the presence of feature manifolds in the activations.

While I think the math here is interesting, my guess is that sparse autoencoders are not in the regime where feature manifolds will cause pathological sparse autoencoder scaling in practice. However, again I think the existence of feature manifolds does indeed complicate the problem of whether sparse autoencoder features are individually meaningful quanta-like units of model computation. If we looked at individual sparse autoencoder feature directions, we might miss the forest for the trees.

This point was made by some recent work from the Anthropic interpretability team. That work, *When Models Manipulate Manifolds: The Geometry of a Counting Task* by Wes Gurnee and Emmanuel Ameisen et al. (Gurnee et al. 2025) beautifully brings together many ideas in the literature from the last few years and which we have discussed in this post. In particular, they study the "quanta 100" behavior that our quanta discovery algorithm had found (Section 1.5), which is the skill of predicting newlines at the correct position in line length limited text. They find that the length of the current and previous lines is represented as a point on a helical feature manifold that spirals through a $\sim$ 6-dimensional subspace of the residual stream. They describe how these feature manifolds are manipulated and compared by the model in order to predict the newline at the correct place. Their SAEs (crosscoders (Lindsey et al. 2024) actually) use a family of about 10 features to reconstruct these manifolds, with multiple latents being active at once.

I expect that over time, the interpretability community will slowly be able to explain more sophisticated language model behaviors than this, and that representations with interesting, higher-dimensional geometry will often, though not always, be implicated in such behaviors.

3 On neural scaling and the limits of interpretability

Most people in the interpretability community, even those attempting to do ambitious work on fully reverse-engineering networks, don't seem very interested in the theory of neural scaling. I think this is a mistake—the computational feasibility of ambitious methods like cross-layer transcoders (CLTs) (Ameisen et al. 2025) and weight-sparse transformers (Gao et al. 2025) may depend on the details of how exactly neural scaling changes the internals of the underlying models we seek to explain.

Here, I'll adopt the view that has been repeatedly articulated by Chris Olah and his collaborators, including *Toy Models of Superposition* (Elhage et al. 2022) and Anthropic's work on sparse autoencoders (Bricken et al. 2023), that neural networks perform computation in superposition. In the networks that we train in practice today, the neurons are not monosemantic nor do they fire particularly sparsely. The network weights connecting these neurons are also dense. However, we suppose that our networks are actually simulating a much larger sparse network. In this imaginary "disentangled model", the neurons are monosemantic and sparse, representing a specific concept and only occasionally activating when that concept is present. We might imagine that the connectivities between these neurons in the disentangled model are also mostly sparse—the computation of most features doesn't depend on all other features—though this isn't depicted in Anthropic's diagram:

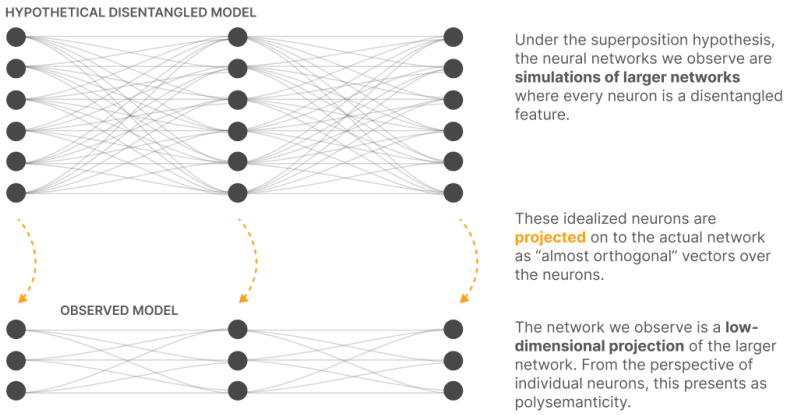
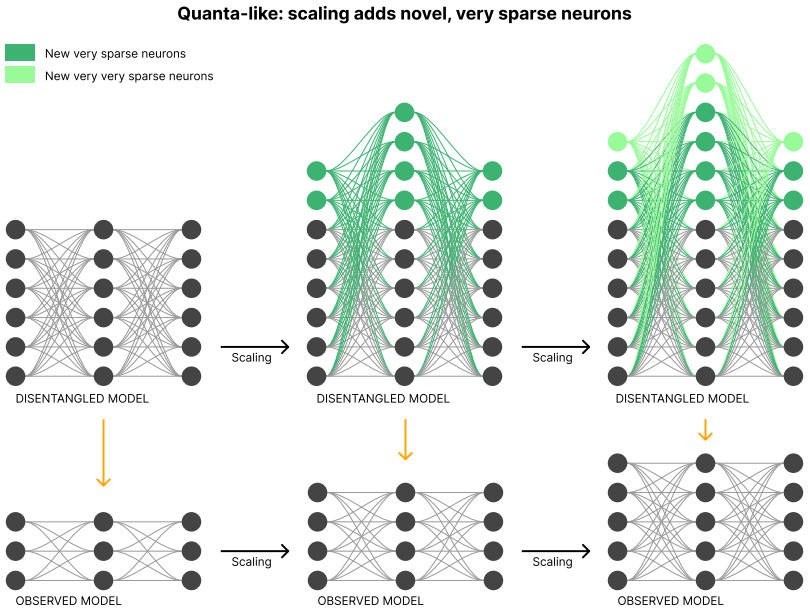


Figure from Anthropic's Toy Models of Superposition *(Elhage et al. 2022) and* Towards Monosemanticity *(Bricken et al. 2023).*

Now, let's imagine that there's a sense in which the disentangled model is *real*. For any given observed model there is a particular associated disentangled model that in some manner describes "what is really going on" in the observed model—the observed model is trying to simulate that particular sparse disentangled model in superposition. Let's assume that for any given observed model, there is a corresponding unique "true" associated disentangled model, up to differences which don't change the computation of the disentangled model, like permuting its neurons.

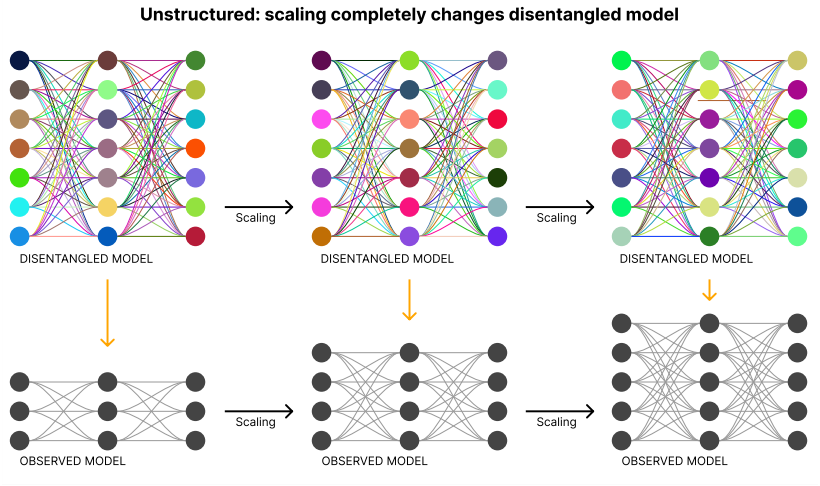
If you believe this idea, then the right way to frame questions about neural scaling, or in general the differences between any two models (Lindsey et al. 2024), is with reference to these disentangled models. There are many different stories we could imagine about how neural scaling changes the disentangled models, with different implications for the difficulty of extracting these models. The goal of ambitious, bottom-up interpretability—sparse autoencoders, cross-layer transcoders, etc.—is to construct part of, or all of, this disentangled model from the observed model. The difficulty of doing this depends on the size and structure of the disentangled models.

One possible story is inspired by the quanta model. If we identify the neurons in the disentangled model with the quanta, then the effect of scaling is to just add new, increasingly sparsely activating neurons to the disentangled model:

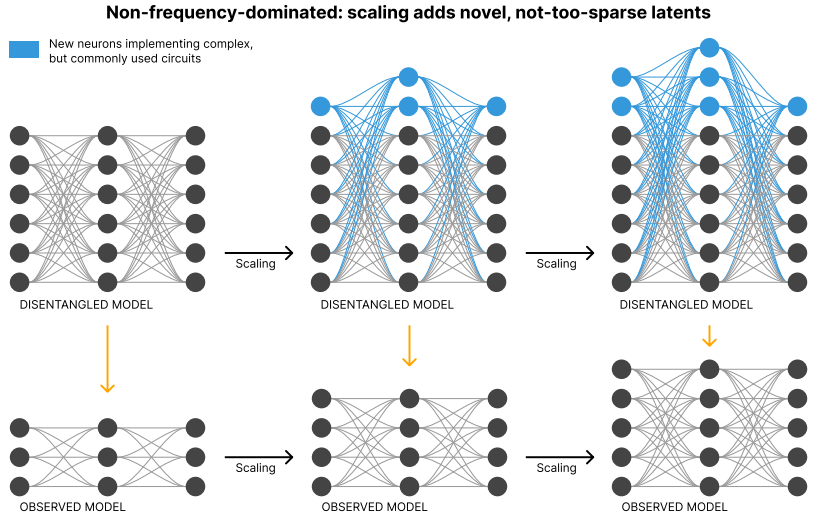


However, this is the worst-case scenario for trying to extract the full disentangled model. As networks are scaled, we would need to train larger and larger sparse networks (e.g. very large cross-layer transcoders or weight-sparse networks) in order to recover the disentangled model. Furthermore, if the new neurons are extremely sparse, we would need to train these CLTs on a very large amount of data, since otherwise the training process might never encounter the situations in which these sparse neurons would fire, and they would then not be learned. When studying a new, scaled up observed model, one could still train CLTs to extract *some* of its disentangled model, but without scaling up one's CLTs appropriately, one would miss the "new" structure in the scaled model (assuming CLTs learn the neurons in the disentangled model in order of their frequency). To the extent that the scaled up model has new concerning behavior, the mechanisms underlying that behavior would be missed by our tools.

There are other possible stories of scaling which are not as bad though. For instance, if scaling changed absolutely everything that was going on in the disentangled model without ultimately using more sparse neurons, then we might not need to scale up CLTs as we study larger models:

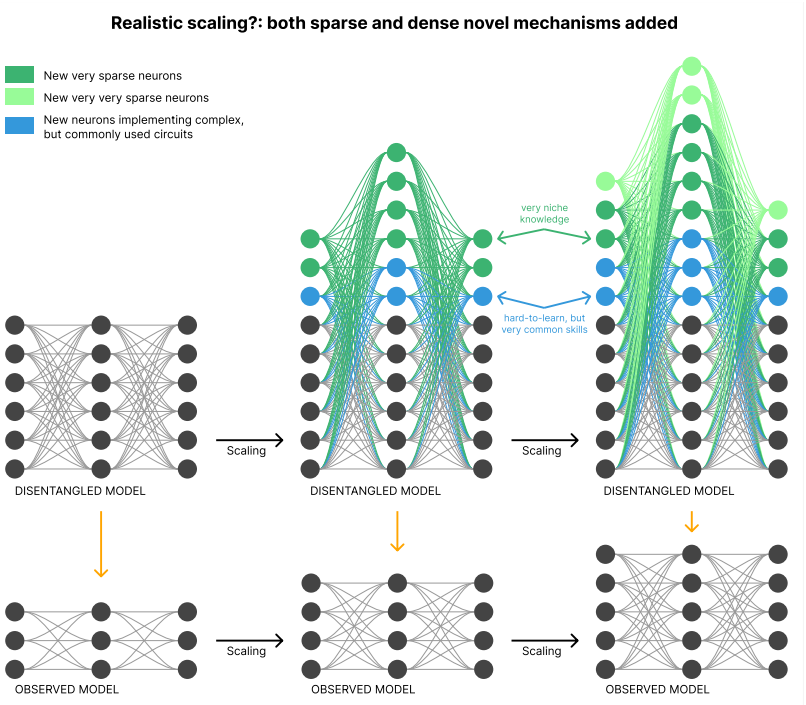


One could imagine another story too where scaling adds novel features, but these features are not increasingly sparse like in the quanta-like story. Instead, they might be part of a more complex circuit which, for whatever reason, is only learned when models are sufficiently scaled up, but which is nevertheless very frequently used by the model:



We did not think about these sorts of scaling dynamics in the quanta model—we said that mechanisms are added only in order of their "use frequency". But I think it's possible that there could be real-world scaling dynamics where commonly-used, hard-to-learn mechanisms are only learned at large scale. In this dynamic, as long as there aren't too many such novel features, we wouldn't need to train CLTs that are too wide, nor would we need to train them on too much data, in order to capture the new mechanisms in scaled up models.

What I suspect is actually going on is a mix of these stories. Scaled up models have both new very sparse mechanisms and new denser mechanisms. While we might miss these very sparse neurons in our SAEs, CLTs, etc., we'd hopefully still be able to efficiently disentangle the novel dense mechanisms. Hopefully these mechanisms, which lie outside of the quanta story, are the safety-relevant ones, since the very sparse ones may be intractable to learn.



It would be great if we had a more mature theory of scaling here which said something about which story we are actually in. I am also curious how RL scaling acts on neural networks. I'd guess that pretraining leans more on the side of adding sparse mechanisms and that RL leans more on the side of adding hard-to-learn dense mechanisms.

I'll say one last thing here, which is a bit different. So far, we've thought about how the structure of the hypothetical disentangled models for our observed models impacts the difficulty of interpretability. But the structure of these disentangled models could have implications far beyond interpretability, and could explain many phenomena we can more directly observe, for instance why some architectures work better than others.

We could imagine that independent of any given network, but instead specific to a *data distribution*, there'd be an associated "perfect disentangled model" that achieves ideal performance on that distribution. When we train a dense finite model, the model tries to simulate as much of the perfect disentangled model, using superposition, as it can. But there are limits to how much of this model it can simulate. When we train larger models, we are able to simulate a larger fraction of this perfect disentangled model.

This perfect disentangled model likely has many types of *structure*. For instance, if many of its neurons can be grouped into clusters, where neurons in two different clusters rarely co-activate on the same sample, then one can simulate the perfect disentangled model with a sparse MoE, and simulate these different groups of neurons with different MoE experts. The ultimate limits of MoEs are then determined by the co-occurrence statistics of the neurons in the perfect disentangled model. Many other facts about neural network training are also probably ultimately attributable to properties of the perfect disentangled model.

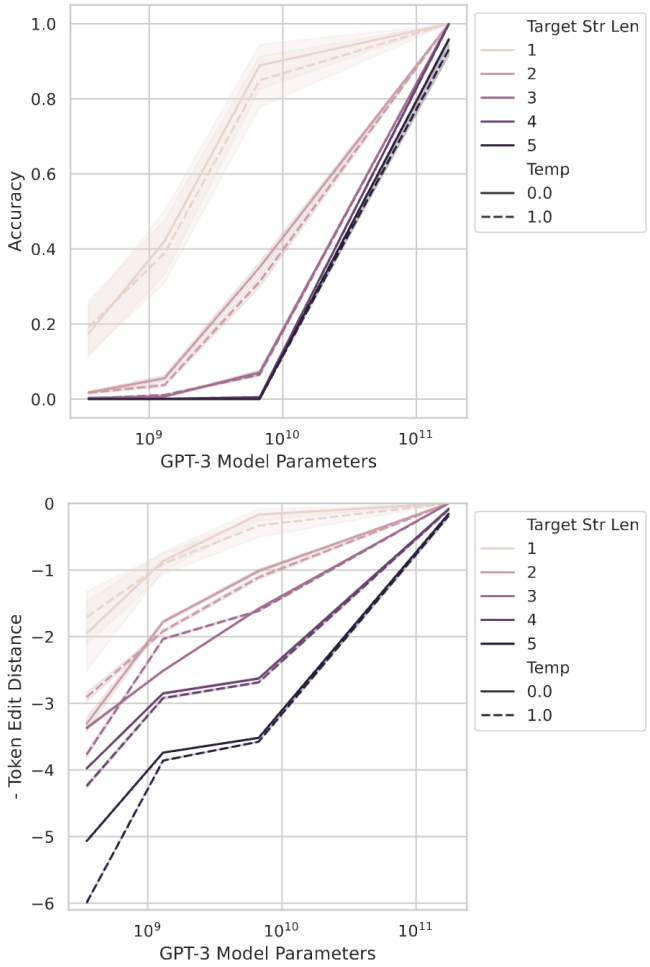
4 Different perspectives on emergent abilities

Our model of neural scaling was motivated, in part, by resolving an apparent tension between the smoothness of neural scaling in the aggregate with the discrete changes in performance seen in "emergent" abilities. However, a now famous paper from Rylan Schaeffer et al. has since suggested that emergent abilities are potentially just artifacts of how we measure model performance and do not necessarily reflect an underlying sharp change in our models due to scaling. This paper, *Are Emergent Abilities of Large Language Models a Mirage?* (Schaeffer et al. 2023) is quite interesting and exhibits results that are challenging for our model of scaling.

Rylan's core point is that abilities that appear "emergent", where models show sharp improvements at certain scales, mostly show up on tasks where performance is measured with a "metric that nonlinearly or discontinuously deforms per-token error rates". Language models output a probability distribution over their token vocabulary, and it turns out that for some metrics, an underlying gradual change to the probabilities that models assign to different tokens can be transformed into something that looks discontinuous.

For example, let's consider the metric of Accuracy. If we are testing whether a language model predicts some particular token correctly, we could define the accuracy as 1 when the correct token is assigned the highest probability by the model, and 0 otherwise. With this metric, we can imagine cases where the outputs of the model could change just barely while totally flipping accuracy from 0 to 1. For instance, consider a model that outputs the correct token with probability 40%, and some other incorrect token with probability 41%, and all other tokens with lower probabilities summing to 19%. Then, as we scale the model, let's imagine that the correct token probability slightly increases to 41%, and the next most likely token decreases to 40%, with the others still summing to 19%. This change has only barely changed the model's output, yet the accuracy has transitioned from 0 to 1.

It is therefore possible that we could measure lots of "emergent abilities" even if scaling truly had an underlying smooth and predictable effect on what models output. And indeed, Rylan showed that for some tasks that seem emergent, changing the choice of metric can reveal some gradual progress occurring with scale. For instance, if one looks at LLM performance on adding two numbers, for sufficiently hard problems, the accuracy (exact match on the model's multi-token output) seems to increase sharply at some scale, whereas if one instead measures the "token edit distance" the model shows some progress towards the correct answer while accuracy is still at ≈ 0:



Taken from Figure 3 of Schaeffer et al. (2023), showing how changing your metric can change whether abilities appear to emerge sharply or gradually.

It is possible in principle that scaling could change model internals in an entirely gradual manner, and we would still measure some "emergent" abilities. There might therefore not be any tension to resolve in the first place, between smooth neural scaling and emergent abilities.

However, I think there is a way of reconciling gradual progress in LLM output distributions with some versions of the quanta theory. As discussed earlier, the loss that LLMs achieve on many tokens seems to improve gradually with scale, and we inferred that those tokens must be polygenic. In Rylan's examples, we can similarly imagine a polygenic story—that many different parallel mechanisms (quanta) are nudging the probability distribution in the correct direction for different reasons, and each of these develops at a different level of scale.

For instance, on arithmetic problems, we could imagine this looking like the following: the simplest mechanism, learned by relatively small models, could involve realizing that the next token is likely a number, and putting more probability on tokens containing numbers. This lowers the loss substantially, but doesn't improve the accuracy by much. At greater scale, other heuristics that partially solve the problem could emerge. For some problem instances, heuristics like "the answer is even" will further nudge the distribution towards the right answer. Eventually, in order to

reduce the loss further, the model will need to finally learn a mechanism for correctly implementing arithmetic, and the accuracy will spike.

I think that a polygenic story like this is likely happening with LLM emergent abilities on multiple choice problems. Consider this plot from the appendix of the original Wei et al. (2022) "Emergent Abilities" paper:

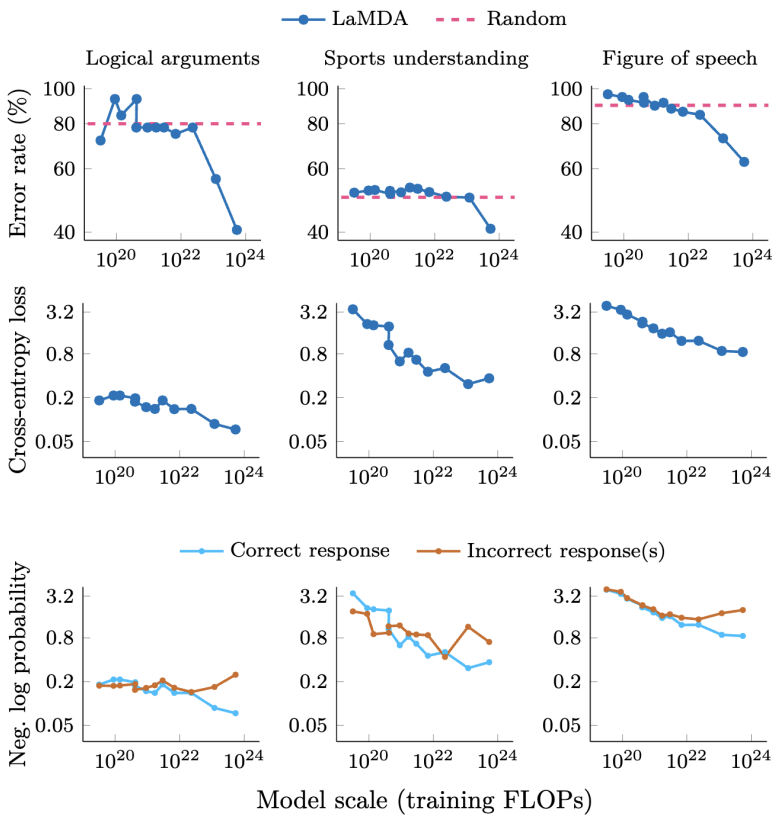


Figure 6 of Emergent Abilities of Large Language Models *(Wei et al. 2022), showing how the probabilities of correct and incorrect answers increase together until an emergent improvement in error rates.*

For each of these multiple-choice "classification" tasks, the accuracy initially plateaus at random-guess accuracy. However, during that plateau, the loss on predicting the correct answer token decreases gradually! So is there some progress being made that is obscured by the choice of metric? In the bottom row, the authors show that the probability the LLMs are assigning to the correct answer is increasing with scale, but the probability of the *incorrect* multiple choice answers are also increasing at the same rate. What I'd guess is happening is that the models are getting better and better at realizing that they are being given a multiple choice problem with some set of possible answers, and nudging probability mass equally towards all of the listed multiple choice answers. This is a great strategy from a loss perspective. Instead of the model being uncertain across the whole token vocabulary, it is just uncertain among the 4 acceptable answers.[6] However, this is very different from actually knowing the answer to these problems! For each task, there is some scale at

[6]If there are 50k tokens in the vocabulary, then outputting a uniform distribution over these tokens corresponds with a cross-entropy loss of about 10 nats. Outputting a uniform distribution over 8 tokens, one of which is correct, corresponds with a loss of about 2 nats. Outputting a uniform distribution over two answers gives a loss of 0.693 nats, or 1 bit. If cross-entropy loss drops

which the correct answer diverges and becomes much more likely than the incorrect answers, and the error rate drops. At this scale, I'd guess that the model has learned some genuinely new mechanism that actually captures the knowledge the benchmark is supposed to measure. The mechanisms learned earlier lowered the loss, but with a strategy that wasn't related to the actual knowledge that is being tested by each benchmark.

Note that similar points were made by Lawrence Chan in the comments on this Alignment Forum post from Fall 2022 (Perez and Nanda 2022), and some other discussion between Gwern and Paul Christiano in those comments touches on the core ideas behind what we ended up calling the quanta hypothesis (induction bumps, the possibility of gradual changes resulting from "a ton of tiny discrete changes", etc.)

In a follow-up paper (Schaeffer et al. 2024), Rylan and several coauthors studied the problem of predicting when abilities will emerge on multiple-choice tasks, but struggled. They explain that: "accurately predicting downstream capabilities requires predicting not just how probability mass concentrates on the correct choice with scale, but also how probability mass fluctuates on the alternative incorrect choices with scale". To me, this explanation seems to miss the possibility that forecasting is hard because of genuine emergence taking place. The gradual improvements in probability mass, on the correct and incorrect tokens, may not reflect progress on learning the true mechanisms we care about.

> Overall, I am unsure whether Rylan's implied "scaling is smooth, emergence is not fundamental" story versus our "emergence is ubiquitous, smoothness is not fundamental" picture is correct. If I had to guess, I'd favor something like a "weak quanta hypothesis", where the overall language modeling task does indeed decompose into lots of subtasks with power-law-distributed importance, but only some, rather than all, of these tasks are learned in a discrete manner. For others, there may be many different ways of solving them that generalize in different ways.

Ideally, I'd like to see a mechanistic study of emergence. For some seemingly emergent ability, what mechanisms are contributing to gradual progress in the model's output distribution, if there is any such gradual progress? When accuracy does eventually spike, can we find new mechanisms clearly present in this larger model that weren't there before? At present, this seems difficult to answer because our ability to do mechanistic interpretability at this level of detail is limited, particularly if each mechanism only has a small influence on model behavior. Also, we don't really have a formal definition of "mechanism" or what it would mean for some mechanism to be "new". Another deep issue here is that if scaling is genuinely smooth, the idea of looking for "new mechanisms" in the first place may not make sense, since the idea of there being clean "mechanisms" at all may not be a correct way of thinking about what neural networks are doing.[7]

below 1 bit, then one can work out quickly that the probability assigned to the correct token must be greater than 50%. With this in mind, I am confused by the loss reported in Wei et al. Figure 6. For "Logical arguments" and "Sports understanding", the cross-entropy drops below 0.69 nats well before the error decreases. For "Logical arguments", the loss is below 0.69 for all models. How is it possible that the accuracy could be low while the loss is so low? There must be something I'm not understanding about the loss here. Perhaps each answer is multiple tokens long, and the loss is being averaged over the whole answer length, rather than just the first token.

[7] This is all horribly confused! There isn't a widely-used, formal definition of "mechanism".

5 Is scaling plateauing?

There has been a lot of discussion over the last year and a half about whether the gains from pretraining scaling have plateaued or "hit a wall". While this may have been more of an OpenAI-specific issue, I think there are nevertheless some interesting questions here, including: (1) will most users notice the improvements from better pretraining at this point? and (2) can pretraining scaling alone give models all the capabilities we want them to have?

In the quanta model, new mechanisms that emerge at scale each change the model's performance on a small fraction of tokens in the pretraining corpus. As one moves further out along the tail of the distribution of quanta, scaling improves performance on a smaller and smaller fraction of the corpus. If the statistics of the quanta "use frequencies" are similar between pretraining and deployment, then scaled-up models will be noticeably different on a smaller and smaller fraction of situations that users find themselves in. At this point in the scaling curve, where models have learned the common knowledge of the early quanta, one needs to be engaged in more esoteric conversations in order to notice the new quanta that have been added, and so most users may not notice the difference from better pretraining.

This dynamic may be further exaggerated by differences in the distribution over what knowledge is needed during deployment vs. pretraining. Chats with most users may be highly skewed towards common knowledge and skills—if we compared a set of random ChatGPT queries with a set of random pretraining documents, I'd guess that the pretraining documents would reference more esoteric knowledge more frequently than the chats in deployment. If this was true, then most users, most of the time, would not notice returns to scale. But particular users (e.g. academics in niche fields) asking about topics in far-out parts of the quanta sequence would notice big improvements from scaling, once pretraining scaling reached those quanta.

> This point can be demonstrated, in a delightfully self-referential way, by asking the models to describe the basic idea of *The Quantization Model of Neural Scaling* by Michaud et al. (2023) without searching the internet. Whenever new models have come out, I've tried them on this prompt, and they have failed until this year: Gemini 3 Pro completely nails it.[8] The "quanta" quanta are just now starting to be learned in pretraining. For niche topics, our best pretrained models this year are qualitatively more knowledgeable than previous models.

While better pretraining continues to add increasingly niche knowledge into models, this does not mean that pretraining alone is sufficient to learn everything that we want our models to learn. It is possible that some desirable circuits are either not incentivized by the pretraining objective or are not expressible in a single forward pass of any reasonably sized model, and therefore would not be learnable during pretraining. We did not consider this sort of dynamic in our work, but it can be easily explored:

Let's imagine that our model "misses" one out of every M quanta. Recall that our quanta have "use frequencies" $p_k = Ak^{-(\alpha+1)}$ and reduce the loss by Δ on the samples where they matter. Learning up to quanta $k = n$ in the quanta sequence,

[8] I forget whether Claude 4 Opus knew about the quanta model, but Claude 4.5 Opus does know it, and nails it about as well as Gemini 3 Pro.

but missing every M quanta, gives a loss of:

$$L(0) - \Delta A \left[\sum_{k=1}^{n} k^{-(\alpha+1)} - \sum_{j=1}^{n/M} (jM)^{-(\alpha+1)} \right]$$

where we've added the effect of every M quanta back into the loss. After integrating, one gets a formula quite similar to the one from before:

$$L'(n) \approx L'_\infty + C' n^{-\alpha}$$

where $L'_\infty = L_\infty + \frac{\Delta \cdot A}{\alpha} M^{-(\alpha+1)}$ and $C' = C \cdot \frac{M-1}{M}$. So if we miss some fraction of the quanta, the scaling exponent will be the same! The curve just slightly shifts, and ultimately levels out at a higher loss floor. This means that quanta can be missed quite subtly. Our current architectures could be failing to learn a large number of the computations which produce the data they are training to predict, and we might not be able to tell.

Interestingly, chain of thought allows the network to perform effectively deeper computation than would be possible in a single forward pass (Pfau et al. 2024), and so RL enables networks to learn these deeper computations.

6 Are sharp transitions just artifacts of noise?

Earlier, we saw that in the Pythia suite of LLMs, as one scales up network size, some individual-token loss curves exhibit sharp drops, and others show smooth scaling. We assumed that these differences reflect something about how many quanta are relevant to doing prediction on each token: loss curves with sharp drops are monogenic, and loss curves with smooth improvements are polygenic.

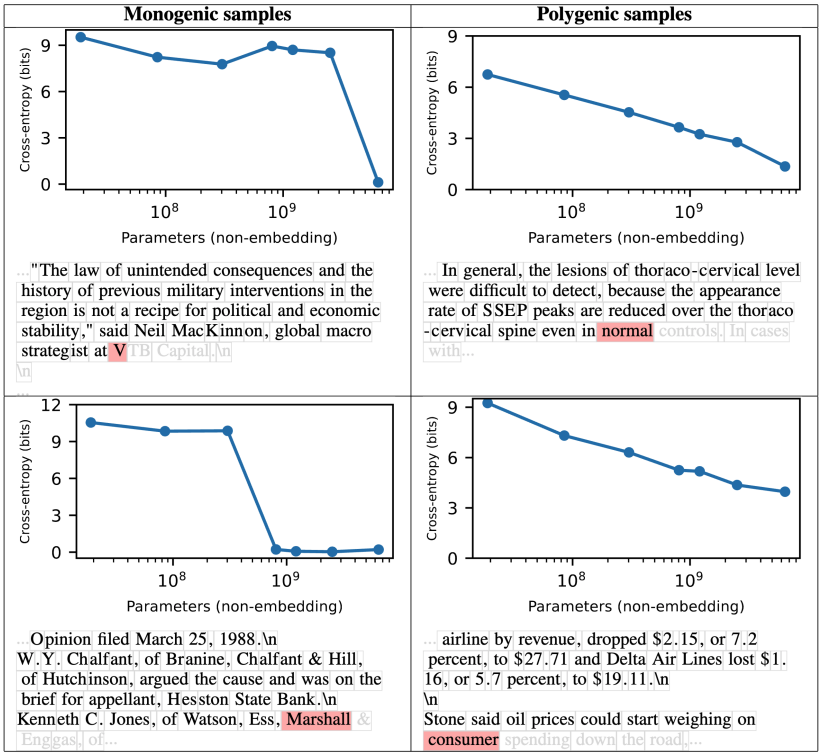
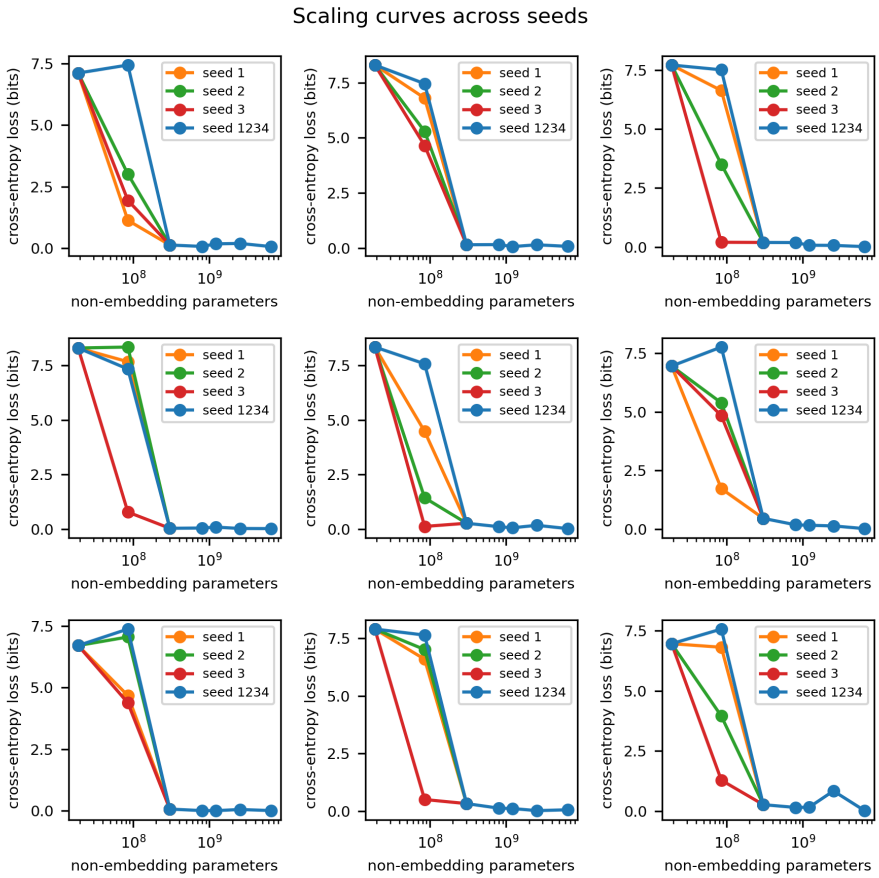


Figure 12 of Michaud et al. (2023).

However, it is worth pointing out that some of these differences could be the result of noise. On many tokens, it could be the case that there is enough variation in a model's loss due to randomness (different seeds) that if we looked at enough tokens, we could find many curves with seemingly sharp drops, just as a result of these fluctuations, even if there is nothing ultimately discrete about the model's performance on those tokens. I thank Naomi Saphra for making this point in an exasperated tweet (Saphra 2023).

In the Pythia sequence, one of the models (160m) was trained with multiple seeds, so we can study this question. In particular, across the corpus, I first looked for tokens where the loss curves exhibit a plateau → drop → plateau behavior with the single default seed, where the drop occurs between the 160m scale and the 410m scale. I only look at curves where pythia-70m and pythia-160m have a loss within 1 bit of each other, and where pythia-410m gets less than 0.5 bits of loss. I then also plot the losses of the 160m model with different seeds. This allows us to get some sense of whether the sharp drop after the 160m scale is intrinsic to the task or not:



Seed-dependence of per-token LLM scaling curves w.r.t. parameters.

We see that on many tokens where the default seed exhibits a sharp drop after the 160m scale, models with different seeds often make partial progress when they reach the 160m scale. It seems that loss on many of these tokens is not actually discrete, and in fact models can make partial progress on them in a seed-dependent way. If you'd like to see more curves like these, here are 2400 of them in a 100 page pdf (5MB).

While models can make partial progress on many tokens, on some of them, the loss curves still seem discrete across all seeds we have. To more firmly settle this question, we'd ideally train hundreds of models with different seeds. If there is some discreteness to the underlying learning problem, then we might expect these loss values to cluster together at particular values for each token. However, if the distribution over losses was almost always uniform or unimodal, that may be a problem for our quanta hypothesis.

7 On joint parameter-data scaling and learning efficiency

In our model of scaling, we independently studied scaling in number of parameters, dataset size, and steps. When scaling parameters, we assumed that data was plentiful, and the only bottleneck on the number of quanta learned was the network's capacity. For data scaling, we assumed the network was arbitrarily large, and the number of quanta learned is determined by whether there are enough data points in the training set to uncover rare quanta. Each scaling regime assumes that the other resources are infinite.

In the paper, we had nothing to say about joint parameter-data scaling: the scaling law $L(N, D)$ when both N and D are finite. A naive application of the quanta model would suggest a joint scaling law like:

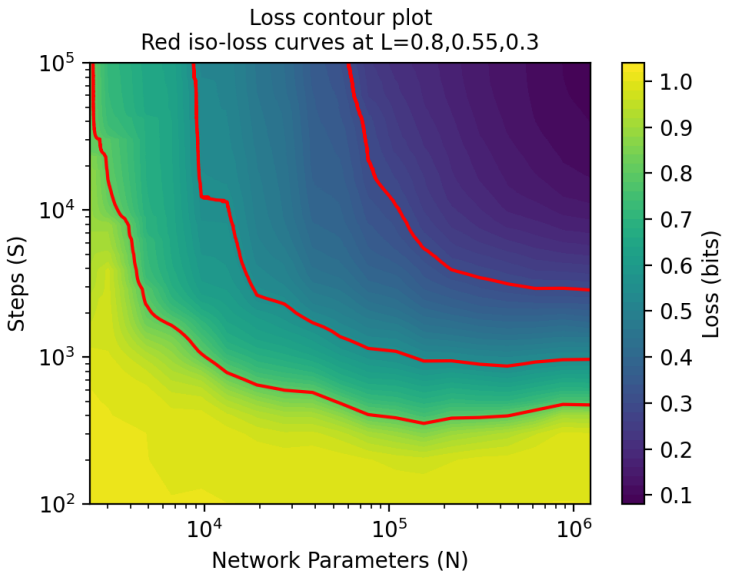
$$L(N, D) = \max\left(L(N), L(D)\right)$$

since our model says that parameters and data independently bottleneck the number of quanta we can learn. In a little more detail, recalling the discussion above (Data scaling), the functional form we'd predict is:

$$L(N, D) = L_\infty + C\left(\min\Big(N/c, (AD/\tau)^{1/(\alpha+1)}\Big)\right)^{-\alpha}$$

which just applies our $L(n) = L_\infty + Cn^{-\alpha}$ scaling law where n is the minimum of the number of quanta learned according to the separate parameter and data scaling stories.

However, this functional form misses a key dynamic in real-world joint scaling, which is that larger networks are more efficient learners. This is a well-known phenomenon in language models. It is also present in our multitask sparse parity networks. Below, I show a loss contour plot for multitask sparse parity, where we find that larger networks achieve a given loss in fewer steps than smaller networks:



Joint parameter-step scaling $L(N, S)$ for multitask sparse parity. We show a few iso-loss curves in red, which indicate that larger networks reach the same mean loss faster than smaller networks. This dynamic, where one can trade off between scaling network size and training steps (or dataset size), is not captured by our model of scaling.

This basic behavior, where one can trade off between data (steps) and parameters, is not compatible with the joint functional form involving max() and min() above. It is however exhibited by the "Chinchilla" functional form:

$$L(N, D) = \frac{A}{N^{\alpha_N}} + \frac{B}{D^{\alpha_D}} + E$$

Interestingly, I think the quanta model makes a prediction about the structure of what the joint functional form ought to be that is different from Chinchilla's. If the overall loss remains just a function of the number of quanta learned $L(n) = L_\infty + Cn^{-\alpha}$, then if N, D or N, S jointly determine the number of quanta learned n, we would expect a functional form like:

$$L(N, D) = L_\infty + (f(N, D))^{-\alpha}$$

where $n = f(N, D)$. For instance, we could imagine something like:

$$L(N, D) = E + \left(\frac{A}{N} + \frac{B}{D^{1/(1+\alpha)}}\right)^{\alpha}.$$

which is similar to the functional form that Kaplan et al. (2020) use in their original "Scaling Laws for Neural Language Models" work (eqn 1.5).

It would be interesting to see whether functional forms like this, which have a global exponent outside of some function of N, D, are a better fit to empirical language model joint parameter-data scaling relationship than the Chinchilla functional form.

Looking forward, I suspect that there is a very nice paper waiting to be written which develops a model of joint parameter, data, and step scaling and which relaxes the quanta model. One possible direction could be to frame the optimization problem as a kind of implicit competition within the network between quickly learning memorizing solutions of low complexity and more slowly learning general solutions

of higher complexity, taking inspiration from Varma et al. (2023)'s explanation of grokking, and invoking minimum-description length arguments. One could aim to explain memorization scaling laws (Morris et al. 2025) too with such a model. Pan et al. (2025) and Ari Brill (Brill 2024) gestured in this direction in some recent work as well.

8 α_N versus α_D and Besiroglu et al.

In our model, $\alpha_N = \alpha$ and $\alpha_D = \alpha/(1+\alpha)$. Does this relationship hold in practice? Below, we plot the scaling exponents α_N and α_D (or possibly α_S) that have been observed in practice across a variety of scaling laws papers. In the quanta model, we'd expect that $\alpha_D = \alpha_N/(1+\alpha_N)$, and we plot this line in black:

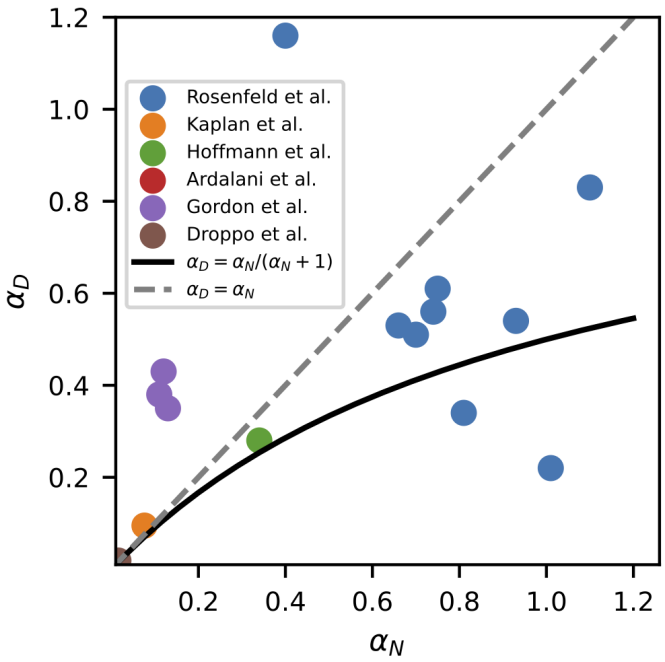


Figure 18 of our paper, showing neural scaling exponents α_N and α_D for a variety of studies in the literature.

This data is kind of a mess! It is certainly not the case that all points lie on the black line. We were encouraged though by the fact that the Chinchilla scaling exponents (green dot) were kinda close to our prediction and had an $\alpha_D < \alpha_N$ as our theory predicted.

Unfortunately, work from Tamay Besiroglu and colleagues (Besiroglu et al. 2024) re-did the scaling law fits from the Chinchilla paper and found a better fit with different exponents which move away from our theory's predictions. The original Chinchilla exponents were $\alpha_N = 0.34$ and $\alpha_D = 0.28$ but Besiroglu et al. estimate $\alpha_N = 0.35$ and $\alpha_D = 0.37$, where α_D is actually higher than α_N! I don't think that this single result kills the quanta model, but I think it further highlights the opportunity to think carefully about alternative models of parameter and data scaling, whether or not one makes quanta-like assumptions.

One possible avenue is to develop a more detailed understanding of optimization. For instance, in some follow-up work called *Physics of Skill Learning* (Liu et al. 2025), Ziming Liu (and I and others) studied how optimization dynamics can influence

neural scaling behavior w.r.t. training steps, and found that there can be dynamics where the number of subtasks (quanta) learned is $n \propto S$ rather than $n \propto S^{1/(1+\alpha)}$ like we assumed in the quanta model, which would result in $\alpha_S = \alpha = \alpha_N$. These sorts of dynamics, if present, would resolve the apparent tension between the quanta model and the observed Besiroglu Chinchilla scaling exponents. Thinking carefully about optimization also may be a route to explaining why larger networks are faster learners and lead to more realistic models of joint parameter-data scaling, as discussed above.

9 On the current culture of neural scaling laws papers

In 1964, the physicist John Platt wrote a paper called *Strong Inference* (Platt 1964). In this paper, Platt observed that some fields of science, like high-energy physics and molecular biology, seemed to make progress much faster than other fields. Platt suggested that these particular fields owed their success to the fact that they practiced "a particular method of doing scientific research" which he called "strong inference". Strong inference involves (1) "devising alternative hypotheses", (2) "devising a crucial experiment... which will, as nearly as possible, exclude one or more of the hypotheses", and (3) "carrying out the experiment so as to get a clean result". While this is supposed to be the normal process of science, Platt points out that this ideal is emphasized and enforced to varying degrees within the culture of different disciplines. He closes by encouraging readers to always privately ask, of their own work, "what experiment could disprove your hypothesis?"

While it is difficult to admit, I don't think that there is a clean experiment that could falsify the quanta hypothesis. If we had a satisfying formal definition of the quanta, or if we could decompose real networks into a set of "true" atomic mechanisms, then we could measure the "use frequencies" of these mechanisms and see whether they indeed followed the same power law we observe from neural scaling. Unfortunately, the quanta remain ethereal.

While I'm being especially hard on myself here, my sense is that this subfield as a whole—on models of neural scaling—struggles with this issue. There are many papers proposing different explanations of neural scaling laws. But these explanations do not always make distinct testable predictions that would allow us to decide between them with a clear experiment.

This may seem like an indictment of the researchers in this field. But I think there is another, kinder explanation for our failure: the field is young, and the problems are hard. We are still in the phase where it is hard to come up with *any* sharp theories for what deep learning is doing.

And yet we must try. The world's future is wrapped up in questions about neural scaling. We owe it a good theory.

Thanks to Uzay Girit, Ziming Liu, Wes Gurnee, Ari Brill, Jamie Simon, Oren Neumann, Srihita Vatsavaya, Daniel Kunin, and William Brandon for reading drafts of this post and for helpful conversations and feedback. All errors remain my own.

References

Platt, John R. (1964). "Strong Inference". In: *Science* 146.3642, pp. 347–353. DOI: 10.1126/science.146.3642.347. URL: https://www.science.org/doi/10.1126/science.146.3642.347.

Minsky, Marvin (1986). *The Society of Mind*. New York: Simon and Schuster.

Saxe, Andrew M., James L. McClelland, and Surya Ganguli (2013). *Exact solutions to the nonlinear dynamics of learning in deep linear neural networks*. arXiv: 1312.6120 [cs.NE]. URL: https://arxiv.org/abs/1312.6120.

Hestness, Joel, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou (2017). *Deep Learning Scaling is Predictable, Empirically*. arXiv: 1712.00409 [cs.LG]. URL: https://arxiv.org/abs/1712.00409.

West, Geoffrey (2017). *Scale: The Universal Laws of Growth, Innovation, Sustainability, and the Pace of Life in Organisms, Cities, Economies, and Companies*. New York: Penguin Press.

Jacot, Arthur, Franck Gabriel, and Clément Hongler (2018). "Neural Tangent Kernel: Convergence and Generalization in Neural Networks". In: *Advances in Neural Information Processing Systems*. URL: https://arxiv.org/abs/1806.07572.

Rosenfeld, Jonathan S., Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit (2019). *A Constructive Prediction of the Generalization Error Across Scales*. arXiv: 1909.12673 [cs.LG]. URL: https://arxiv.org/abs/1909.12673.

Saxe, Andrew M., James L. McClelland, and Surya Ganguli (2019). "A mathematical theory of semantic development in deep neural networks". In: *Proceedings of the National Academy of Sciences* 116.23, pp. 11537–11546. DOI: 10.1073/pnas.1820226116. URL: https://arxiv.org/abs/1810.10531.

Bordelon, Blake, Abdulkadir Canatar, and Cengiz Pehlevan (2020). "Spectrum Dependent Learning Curves in Kernel Regression and Wide Neural Networks". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. Vol. 119. Proceedings of Machine Learning Research. PMLR. URL: https://proceedings.mlr.press/v119/bordelon20a.html.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

Gwern (2020). *The Scaling Hypothesis*. URL: https://gwern.net/scaling-hypothesis.

Henighan, Tom, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish (2020). *Scaling Laws for Autoregressive Generative Modeling*. arXiv: 2010.14701 [cs.LG]. URL: https://arxiv.org/abs/2010.14701.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei (2020). *Scaling Laws for Neural Language Models*. arXiv: 2001.08361 [cs.LG]. URL: https://arxiv.org/abs/2001.08361.

Olah, Chris, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter (2020). "Zoom In: An Introduction to Circuits". In: *Distill*. DOI: 10.23915/distill.00024.001. URL: https://distill.pub/2020/circuits/zoom-in/.

Sharma, Utkarsh and Jared Kaplan (2020). *A Neural Scaling Law from the Dimension of the Data Manifold*. arXiv: 2004.10802 [cs.LG]. URL: https://arxiv.org/abs/2004.10802.

Austin, Jacob, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton (2021). *Program Synthesis with Large Language Models*. arXiv: 2108.07732 [cs.PL]. URL: https://arxiv.org/abs/2108.07732.

Bahri, Yasaman, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma (2021). *Explaining Neural Scaling Laws*. arXiv: 2102.06701 [cs.LG]. URL: https://arxiv.org/abs/2102.06701.

Hutter, Marcus (2021). *Learning Curve Theory*. arXiv: 2102.04074 [cs.LG]. URL: https://arxiv.org/abs/2102.04074.

Rae, Jack W., Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. (2021). *Scaling Language Models: Methods, Analysis & Insights from Training Gopher*. arXiv: 2112.11446 [cs.CL]. URL: https://arxiv.org/abs/2112.11446.

Simon, James B., Madeline Dickens, Dhruva Karkada, and Michael R. DeWeese (2021). *The Eigenlearning Framework: A Conservation Law Perspective on Kernel Regression and Wide Neural Networks*. arXiv: 2110.03922 [cs.LG]. URL: https://arxiv.org/abs/2110.03922.

Barak, Boaz, Benjamin L. Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang (2022). "Hidden Progress in Deep Learning: SGD Learns Parities Near the Computational Limit". In: *Advances in Neural Information Processing Systems*. arXiv: 2207.08799 [cs.LG]. URL: https://arxiv.org/abs/2207.08799.

Delétang, Grégoire, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega (2022). *Neural Networks and the Chomsky Hierarchy*. arXiv: 2207.02098 [cs.LG]. URL: https://arxiv.org/abs/2207.02098.

Elhage, Nelson, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah (2022). *Toy Models of Superposition*. Anthropic. URL: https://transformer-circuits.pub/2022/toy_model/ (visited on 05/04/2026).

Ganguli, Deep, Danny Hernandez, Liane Lovitt, Nova DasSarma, Tom Henighan, Andy Jones, Nicholas Joseph, Jackson Kernion, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Nelson Elhage, Sheer El Showk, Stanislav Fort, Zac Hatfield-Dodds, Scott Johnston, Shauna Kravec, Neel Nanda, Kamal Ndousse, Catherine Olsson, Daniela Amodei, Dario Amodei, Tom Brown, Jared Kaplan, Sam McCandlish, Chris Olah, and Jack Clark (2022). "Predictability and Surprise in Large Generative Models". In: *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. arXiv: 2202.07785 [cs.LG]. URL: https://arxiv.org/abs/2202.07785.

Hoffmann, Jordan, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre (2022). "Training Compute-Optimal Large Language Models". In: *Advances in Neural Information Processing Systems (NeurIPS)*. URL: https://arxiv.org/abs/2203.15556.

janus (2022). *Simulators*. LessWrong. URL: https://www.lesswrong.com/posts/vJFdjigzmcXMhNTsx/simulators.

Liu, Ziming, Ouail Kitouni, Niklas Nolte, Eric J. Michaud, Max Tegmark, and Mike Williams (2022). *Towards Understanding Grokking: An Effective Theory*

of Representation Learning. arXiv: 2205.10343 [cs.LG]. URL: https://arxiv.org/abs/2205.10343.

Maloney, Alexander, Daniel A. Roberts, and James Sully (2022). *A Solvable Model of Neural Scaling Laws*. arXiv: 2210.16859 [cs.LG]. URL: https://arxiv.org/abs/2210.16859.

Merrill, William, Ashish Sabharwal, and Noah A. Smith (2022). "Saturated Transformers are Constant-Depth Threshold Circuits". In: *Transactions of the Association for Computational Linguistics* 10, pp. 843–856. DOI: 10.1162/tacl_a_00493. URL: https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00493/112604/Saturated-Transformers-are-Constant-Depth.

Nanda, Neel and Tom Lieberum (2022). *A Mechanistic Interpretability Analysis of Grokking*. LessWrong. URL: https://www.lesswrong.com/posts/N6WM6hs7RQMKDhYjB/a-mechanistic-interpretability-analysis-of-grokking (visited on 05/04/2026).

Olah, Chris (2022). *Tweet expressing interest in a more circuits-flavored theory of neural scaling*. URL: https://x.com/ch402/status/1576038791151636483 (visited on 05/04/2026).

Olsson, Catherine, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah (2022). *In-context Learning and Induction Heads*. Anthropic. URL: https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html (visited on 05/04/2026).

Perez, Ethan and Neel Nanda (2022). *We May Be Able to See Sharp Left Turns Coming*. AI Alignment Forum. URL: https://www.alignmentforum.org/posts/2AvX8cX47CdwjbkjY/we-may-be-able-to-see-sharp-left-turns-coming (visited on 05/04/2026).

Vyas, Nikhil, Yamini Bansal, and Preetum Nakkiran (2022). *Limitations of the NTK for Understanding Generalization in Deep Learning*. arXiv: 2206.10012 [cs.LG]. URL: https://arxiv.org/abs/2206.10012.

Wang, Kevin, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt (2022). *Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 small*. arXiv: 2211.00593 [cs.LG]. URL: https://arxiv.org/abs/2211.00593.

Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus (2022). "Emergent Abilities of Large Language Models". In: *Transactions on Machine Learning Research*. arXiv: 2206.07682 [cs.CL]. URL: https://arxiv.org/abs/2206.07682.

Zhai, Xiaohua, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer (2022). "Scaling Vision Transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. URL: https://openaccess.thecvf.com/content/CVPR2022/html/Zhai_Scaling_Vision_Transformers_CVPR_2022_paper.html.

Biderman, Stella, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal (2023). "Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling". In: *Proceedings of the 40th International Conference on Machine Learning*. arXiv: 2304.01373 [cs.CL]. URL: https://arxiv.org/abs/2304.01373.

Bricken, Trenton, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah (2023). *Towards Monosemanticity: Decomposing Language Models with Dictionary Learning*. Anthropic. URL: https://transformer-circuits.pub/2023/monosemantic-features (visited on 05/04/2026).

Dziri, Nouha, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi (2023). *Faith and Fate: Limits of Transformers on Compositionality*. arXiv: 2305.18654 [cs.CL]. URL: https://arxiv.org/abs/2305.18654.

Michaud, Eric J., Ziming Liu, Uzay Girit, and Max Tegmark (2023). "The Quantization Model of Neural Scaling". In: *Advances in Neural Information Processing Systems*. arXiv: 2303.13506 [cs.LG]. URL: https://arxiv.org/abs/2303.13506.

Nanda, Neel, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt (2023). "Progress Measures for Grokking via Mechanistic Interpretability". In: *International Conference on Learning Representations*. arXiv: 2301.05217 [cs.LG]. URL: https://arxiv.org/abs/2301.05217.

Saphra, Naomi (2023). *Tweet on broken scaling and statistics*. URL: https://x.com/nsaphra/status/1672234050440798211 (visited on 05/04/2026).

Schaeffer, Rylan, Brando Miranda, and Sanmi Koyejo (2023). "Are Emergent Abilities of Large Language Models a Mirage?" In: *Advances in Neural Information Processing Systems*. arXiv: 2304.15004 [cs.LG]. URL: https://arxiv.org/abs/2304.15004.

Simon, James B., Maksis Knutins, Liu Ziyin, Daniel Geisz, Abraham J. Fetterman, and Joshua Albrecht (2023). *On the Stepwise Nature of Self-Supervised Learning*. arXiv: 2303.15438 [cs.LG]. URL: https://arxiv.org/abs/2303.15438.

Varma, Vikrant, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar (2023). *Explaining Grokking through Circuit Efficiency*. arXiv: 2309.02390 [cs.LG]. URL: https://arxiv.org/abs/2309.02390.

Altman, Sam (2024). *The Intelligence Age*. URL: https://ia.samaltman.com/ (visited on 05/04/2026).

Besiroglu, Tamay, Ege Erdil, Matthew Barnett, and Josh You (2024). *Chinchilla Scaling: A Replication Attempt*. arXiv: 2404.10102 [cs.LG]. URL: https://arxiv.org/abs/2404.10102.

Bordelon, Blake, Alexander Atanasov, and Cengiz Pehlevan (2024). *A Dynamical Model of Neural Scaling Laws*. arXiv: 2402.01092 [cs.LG]. URL: https://arxiv.org/abs/2402.01092.

Brill, Ari (2024). *Neural Scaling Laws Rooted in the Data Distribution*. arXiv: 2412.07942 [cs.LG]. URL: https://arxiv.org/abs/2412.07942.

Engels, Joshua, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark (2024). *Not All Language Model Features are One-Dimensionally Linear*. arXiv: 2405.14860 [cs.LG]. URL: https://arxiv.org/abs/2405.14860.

Lindsey, Jack, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah (2024). *Sparse Crosscoders for Cross-Layer Features and Model Diffing*. Anthropic. URL: https://transformer-circuits.pub/2024/crosscoders/index.html (visited on 05/04/2026).

Marks, Samuel, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller (2024). *Sparse Feature Circuits: Discovering and Editing Interpretable Causal Graphs in Language Models*. arXiv: 2403.19647 [cs.LG]. URL: https://arxiv.org/abs/2403.19647.

Michaud, Eric J. (2024). *The Space of LLM Learning Curves*. URL: https://ericjmichaud.com/llm-curve-visualization/ (visited on 05/04/2026).

Nam, Yoonsoo, Nayara Fonseca, Seok Hyeong Lee, Chris Mingard, and Ard A. Louis (2024). "An Exactly Solvable Model for Emergence and Scaling Laws in the Multitask Sparse Parity Problem". In: *Advances in Neural Information Processing Systems*. arXiv: 2404.17563 [cs.LG]. URL: https://arxiv.org/abs/2404.17563.

Neumann, Oren and Claudius Gros (2024). *AlphaZero Neural Scaling and Zipf's Law: A Tale of Board Games and Power Laws*. arXiv: 2412.11979 [cs.LG]. URL: https://arxiv.org/abs/2412.11979.

Pfau, Jacob, William Merrill, and Samuel R. Bowman (2024). *Let's Think Dot by Dot: Hidden Computation in Transformer Language Models*. arXiv: 2404.15758 [cs.CL]. URL: https://arxiv.org/abs/2404.15758.

Schaeffer, Rylan, Hailey Schoelkopf, Brando Miranda, Gabriel Mukobi, Varun Madan, Adam Ibrahim, Herbie Bradley, Stella Biderman, and Sanmi Koyejo (2024). *Why Has Predicting Downstream Capabilities of Frontier AI Models with Scale Remained Elusive?* arXiv: 2406.04391 [cs.LG]. URL: https://arxiv.org/abs/2406.04391.

Templeton, Adly, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L. Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan (2024). *Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet*. Anthropic. URL: https://transformer-circuits.pub/2024/scaling-monosemanticity/ (visited on 05/04/2026).

Ameisen, Emmanuel, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson (2025). *Circuit Tracing: Revealing Computational Graphs in Language Models*. Anthropic. URL: https://transformer-circuits.pub/2025/attribution-graphs/methods.html (visited on 05/04/2026).

Amodei, Dario (2025). *Anthropic CEO Dario Amodei: AI's Potential, OpenAI Rivalry, GenAI Business, Doomerism*. Interview by Alex Kantrowitz. Big Technology Podcast. URL: https://youtu.be/mYDSSRS-B5U (visited on 05/04/2026).

Braun, Dan, Lucius Bushnaq, Stefan Heimersheim, Jake Mendel, and Lee Sharkey (2025). *Interpretability in Parameter Space: Minimizing Mechanistic Description Length with Attribution-based Parameter Decomposition*. arXiv: 2501.14926 [cs.LG]. URL: https://arxiv.org/abs/2501.14926.

Bushnaq, Lucius, Dan Braun, and Lee Sharkey (2025). *Stochastic Parameter Decomposition*. arXiv: 2506.20790 [cs.LG]. URL: https://arxiv.org/abs/2506.20790.

Gao, Leo, Achyuta Rajaram, Jacob Coxon, Soham V. Govande, Bowen Baker, and Dan Mossing (2025). *Weight-Sparse Transformers Have Interpretable Circuits*. arXiv: 2511.13653 [cs.LG]. URL: https://arxiv.org/abs/2511.13653.

Gurnee, Wes, Emmanuel Ameisen, Isaac Kauvar, Julius Tarng, Adam Pearce, Chris Olah, and Joshua Batson (2025). *When Models Manipulate Manifolds: The Geometry of a Counting Task*. Anthropic. URL: https://transformer-circuits.pub/2025/linebreaks/index.html (visited on 05/04/2026).

Liu, Ziming, Yizhou Liu, Eric J. Michaud, Jeff Gore, and Max Tegmark (2025). *Physics of Skill Learning*. arXiv: 2501.12391 [cs.LG]. URL: https://arxiv.org/abs/2501.12391.

Michaud, Eric J., Asher Parker-Sartori, and Max Tegmark (2025a). *On the Creation of Narrow AI: Hierarchy and Nonlocality of Neural Network Skills*. arXiv: 2505.15811 [cs.LG]. URL: https://arxiv.org/abs/2505.15811.

Michaud, Eric J., Liv Gorton, and Tom McGrath (2025b). *Understanding Sparse Autoencoder Scaling in the Presence of Feature Manifolds*. arXiv: 2509.02565 [cs.LG]. URL: https://arxiv.org/abs/2509.02565.

Morris, John X., Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G. Edward Suh, Alexander M. Rush, Kamalika Chaudhuri, and Saeed Mahloujifar (2025). *How Much Do Language Models Memorize?* arXiv: 2505.24832 [cs.CL]. URL: https://arxiv.org/abs/2505.24832.

Pan, Zhixuan, Shaowen Wang, and Jian Li (2025). *Understanding LLM Behaviors via Compression: Data Generation, Knowledge Acquisition and Scaling Laws*. arXiv: 2504.09597 [cs.AI]. URL: https://arxiv.org/abs/2504.09597.