
Superposition Yields Robust Neural Scaling

Yizhou Liu*, Ziming Liu, and Jeff Gore
 Massachusetts Institute of Technology
 Cambridge, MA 02139

Abstract

The success of today’s large language models (LLMs) depends on the observation that larger models perform better. However, the origin of this neural scaling law — the finding that loss decreases as a power law with model size — remains unclear. Starting from two empirical principles — that LLMs represent more things than the model dimensions (widths) they have (i.e., representations are superposed), and that words or concepts in language occur with varying frequencies — we constructed a toy model to study the loss scaling with model size. We found that when superposition is weak, meaning only the most frequent features are represented without interference, the scaling of loss with model size depends on the underlying feature frequency; if feature frequencies follow a power law, so does the loss. In contrast, under strong superposition, where all features are represented but overlap with each other, the loss becomes inversely proportional to the model dimension across a wide range of feature frequency distributions. This robust scaling behavior is explained geometrically: when many more vectors are packed into a lower dimensional space, the interference (squared overlaps) between vectors scales inversely with that dimension. We then analyzed four families of open-sourced LLMs and found that they exhibit strong superposition and quantitatively match the predictions of our toy model. The Chinchilla scaling law turned out to also agree with our results. We conclude that representation superposition is an important mechanism underlying the observed neural scaling laws. We anticipate that these insights will inspire new training strategies and model architectures to achieve better performance with less computation and fewer parameters.

1 Introduction

The remarkable success of large language models (LLMs) has been driven by the empirical observation that increasing model size, training data, and compute consistently leads to better performance [1–4]. Across a wide range of tasks — including language understanding [1, 5, 6], math [7–10], and code generation [11, 12] — larger models achieve lower loss, higher accuracy, and greater generalization abilities [2, 13]. This consistent trend, known as neural scaling laws, has been observed across multiple model families and architectures, fueling the development of increasingly large models [2–4]. These scaling laws have not only shaped the current strategies for building better models but have also raised fundamental questions about why such simple and universal patterns emerge in complex learning systems.

The scaling of loss with model size plays a central role in both the practical design and the theoretical understanding of large-scale machine learning systems, yet its origin remains inconclusive [3, 14–24]. Various explanations have been proposed, drawing from statistical learning theory and empirical phenomenological models, including improved function or manifold approximation in larger models [14, 15], and enhanced representation or skill learning in larger models [18–21]. In the limit of

*liuyz@mit.edu

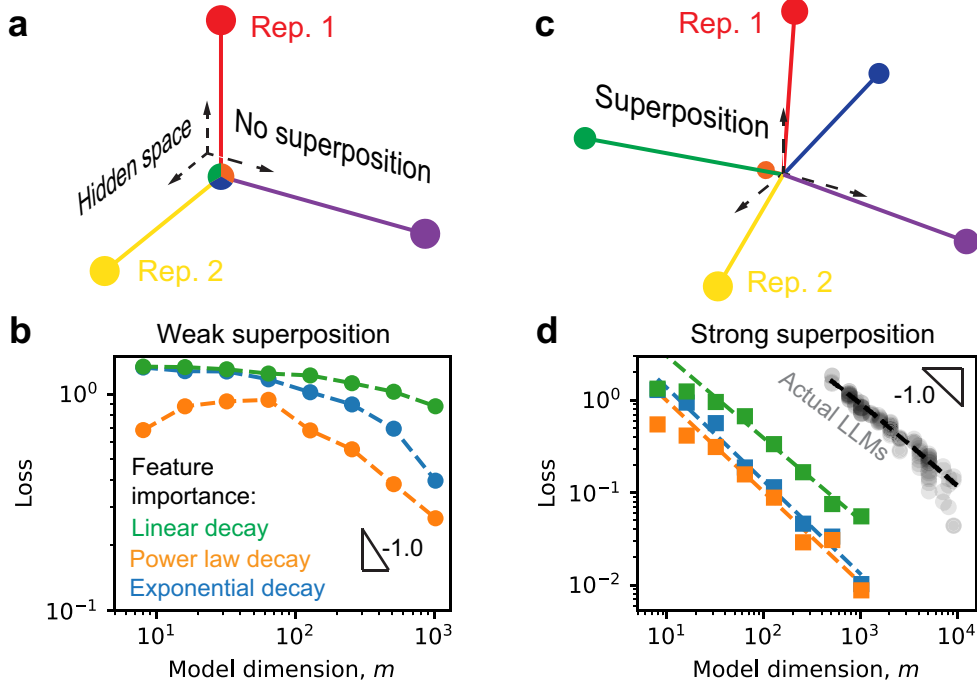


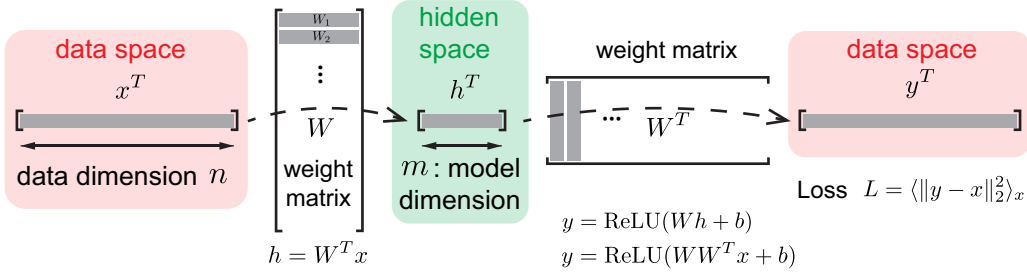
Figure 1: Superposition leads to robust and fast power-law loss decay with model size. (a) Illustration of no superposition where a three-dimensional space can at most represent three features without any interference (overlap). (b) Toy model results in the regime of weak superposition, where we set data dimension $n = 10240$ (number of features). The toy model will be introduced; more details are in Appendix D.1. (c) Illustration of superposition: there are more features than the dimension of the space. (d) The same toy models in the strong superposition regime show lower losses, which are on power laws with model dimension and have exponents close to 1 (color coding same as panel b). The gray points are from actual LLMs, which seem to have a similar power-law relation with exponent ~ 1 .

infinite data, many of these explanations predict a power-law decay of loss with model size, provided the underlying data distribution also follows a power law. However, the predicted scaling exponents depend sensitively on the properties of the data distribution. Moreover, the connection between these mechanistic explanations and the behavior of actual LLMs needs further exploration.

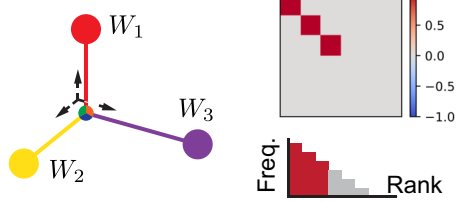
When considering LLMs specifically, it becomes clear that representation learning can be a limiting factor, which is closely related to a phenomenon called **superposition** [25, 26], yet this aspect has not been thoroughly studied. LLMs must learn embedding vectors for tokens, process these representations through transformer layers to predict the next token, and use a final projection (the language model head) to generate the output. Conceptually, fitting functions or manifolds and learning skills or grammars are primarily tasks of the transformer layers, while representation learning is more directly tied to the embedding matrix and the language model head. To represent more than fifty thousand tokens — or even more abstract concepts — within a hidden space of at most a few thousand dimensions, the quality of representations is inevitably constrained, contributing to the final loss. Although models can represent more features than their dimensionality would suggest through a mechanism known as superposition [26], prior work on neural scaling laws has implicitly assumed weak superposition [15–19], which may be less relevant to the regime where LLMs operate. To address this gap, **we systematically study here how representation superposition and data structure together influence the scaling of loss with model size.**

We adopt a toy model construction similar to [26] to study how superposition affects neural scaling laws. In the toy model, representations are learned by recovering data, each composed of multiple latent features. These features in data have different frequencies of occurrence, reflecting their relative importance. In the absence of superposition, only the most frequent features are perfectly represented, while the others are ignored. As illustrated in Figure 1a, the first three of six features are represented in the three-dimensional space without interference, and the remaining three are omitted.

a Toy model of representation learning via data recovery



b No superposition



c Superposition

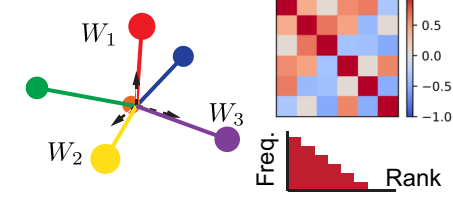


Figure 2: Toy model of superposition. (a) Architecture and loss of the toy model. (b and c) Rows of the weight matrix W , denoted by W_i , are representations for features. (b) No superposition represented the most frequent features, i.e., the first three, without interference (WW^T has zero off-diagonal elements). In the frequency-rank plot, height means frequency, and color means $\|W_i\|_2$. (c) With superposition, more features are represented, while the vectors W_i overlap.

We find that in the weak superposition regime, the scaling of loss with model dimension depends sensitively on how feature frequency decays with rank: the loss follows a power law with model size only if the feature frequencies themselves follow a power law, provided that m is sufficiently large (Figure 1b). By contrast, strong superposition allows many more features to be represented, albeit with overlap in the representation (Figure 1c). In this regime, the model displays a robust behavior: loss scales inversely with model dimension, independent of the data frequency distribution (Figure 1d). Remarkably, we find that actual LLMs follow the same rule. Comparing models with the same dimension but different superposition degrees, the ones with stronger superposition also have a lower loss. **Building on the basic phenomena, we provide the main takeaways:** (i) In the strong superposition regime, loss due to imperfect representation scales inversely with model dimension, largely independent of feature frequency distributions; (ii) Scaling behaviors in the weak superposition regime are brittle and depend on the number of activated but unlearned features, and in the strong superposition regime are robust and can be understood via generic geometric facts; (iii) LLMs exhibit strong superposition and agree quantitatively with our toy model predictions.

The rest of the paper will elaborate on the takeaways. In Section 2, we introduce the toy model, describe the data sampling procedure, and explain how we control the degree of superposition. Section 3 presents the detailed results and quantitative analysis of how superposition and data distribution influence loss scaling. In Section 4, we compare our findings to related works. Finally, Section 5 summarizes our conclusions and discusses limitations and future directions. Codes are available at <https://github.com/liuyz0/SuperpositionScaling>.

2 Methods

To gain understanding of the relationship between superposition and data structure, we need a toy model on learning representation simple enough yet not simpler — two key principles need to be reflected, (i) there are more features to represent than the dimension of the model, and (ii) features occur in data with different frequencies. Later, we will discuss how the loss due to representation studied here may affect the overall final loss in LLMs.

We adopt the toy model of superposition from Anthropic [26] with minor modifications (Figure 2a). Input data $x \in \mathbb{R}^n$ is a vector with data dimension n being the number of atomic (or irreducible)

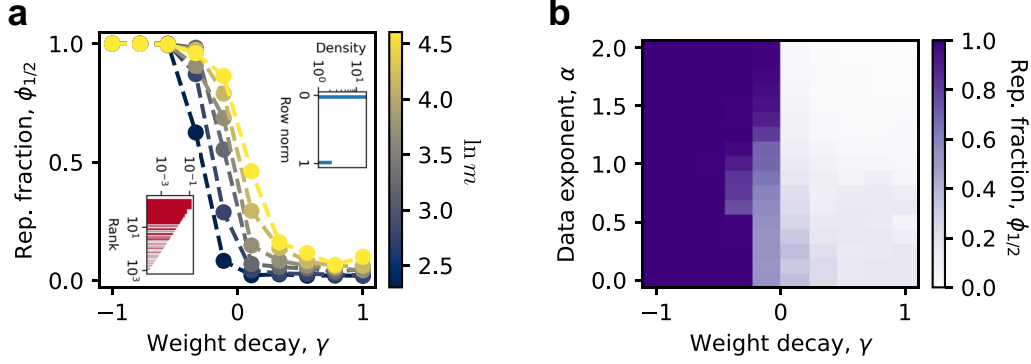


Figure 3: Weight decay can robustly tune the degree of superposition. (a) Given data dimension $n = 1000$ and feature frequency $p_i \propto 1/i^\alpha$ with $\alpha = 1$, we tested model dimension m from 10 to 100 and weight decay from -1 to 1 . The frequency-rank plot shows that important features tend to be represented (color means $\|W_i\|_2$ and colorbar same as Figure 2b). The row norm distribution is bimodal, making $\phi_{1/2}$ a good measure for the fraction represented. Smaller weight decay leads to larger $\phi_{1/2}$ or stronger superposition. (b) After changing α , the ability for small weight decay to reach strong superposition is not changed. We show the result at $m = 39$ here, more in Appendix D.3.

features. Each element x_i in x is interpreted as the activation of this data point at feature i , which is sampled as

$$x_i = u_i v_i, \quad u_i \sim \text{Bernoulli}(p_i) \ \& \ v_i \sim U(0, 2). \quad (1)$$

Here, u_i sampled from a Bernoulli distribution controls whether the feature i is activated, and v_i sampled from a uniform distribution controls the activation strength once feature i is activated. The frequency of feature i to appear in the data is p_i . Without loss of generality, we make the indices of features the same as their frequency or importance rank. The data structure is then about how p_i decreases with rank i . The expected number of activations in one data will be referred to as activation density: $E = \sum_{i=1}^n p_i$. The model learns hidden representations by recovering the data, which cannot be done perfectly because the model dimension m is much smaller than the number of possible features in the data n . The trainable parameters are a weight matrix $W \in \mathbb{R}^{n \times m}$ and a bias vector $b \in \mathbb{R}^n$. The weight matrix embeds data x into a hidden space with dimension m , $h = W^T x$, with $m \ll n$. In practice, we fix n as a large number and change the model dimension m . We use W to read out the embedding, where $y = \text{ReLU}(Wh + b)$. The loss is defined as the difference between the recovered y and the original x , $L = \langle \|y - x\|_2^2 \rangle_x$, where $\langle \cdot \rangle_x$ means average over x distribution.

We can now formally introduce superposition. Note that W_i is the representation of feature i in the hidden space, where we use W_i to denote the i th row of the W matrix, in our toy model, feature i is well represented when the norm $\|W_i\|_2$ is close to 1. No superposition ideally means the first m rows of W form an orthogonal basis (i.e., the first m most important features represented perfectly) and the rest of the rows are zero (i.e., rest of the features ignored or lost), as illustrated in Figure 2b. Superposition means that there are more than m rows in W with norms close to 1 (Figure 2c). Conceptually, superposition is preferred when features are more equal in importance (p_i are more uniform) and when features are more sparse (E/n is smaller) [26].

We found that it is possible to control superposition independently of data properties by modifying the optimizer with a weight decay (or growth) term:

$$W_{i,t+1} = \begin{cases} W_{i,t} - \eta_t \gamma W_{i,t}, & \gamma \geq 0, \\ W_{i,t} - \eta_t \gamma W_{i,t} (1/\|W_{i,t}\|_2 - 1), & \gamma < 0, \end{cases} \quad (2)$$

where η_t is the learning rate and $W_{i,t}$ is the i th row of the weight matrix at step t (vector operations are element-wise). For weight decay $\gamma < 0$, the update corresponds to gradient descent on $(\|W_{i,t}\|_2 - 1)^2$, encouraging unit-norm rows. We implement this in a modified AdamW [27] optimizer with a warm-up and cosine decay learning rate schedule. Training runs for 20k–80k steps with batch sizes of 2048–8192, ensuring models reach saturation. At each step, we sample new data using probabilities p_i . We observe that important features are represented as expected, and row norms of W become bimodal, clustering near 0 and 1 (Figure 3a). This allows us to define the fraction of represented

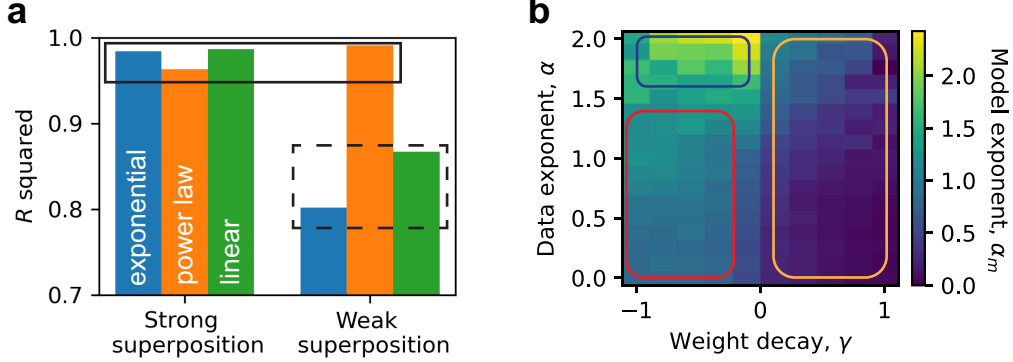


Figure 4: Rich scaling phenomena arise when we change the degree of superposition and data structures. (a) In the same experiments ($n = 10240$) in Figure 1, where $p_i \propto \exp(-i/400)$ (blue), $p_i \propto 1/i^{1.2}$ (yellow), and $p_i \propto n-i$ (green), we use $\gamma = -1$ to reach strong superposition and $\gamma = 0.1$ for weak superposition. R-squared value of the fitting is used to measure how likely the fitted part is a power law (Appendix D.4). R-squared values closer to 1 mean that the data are more similar to a power law. We found that at strong superposition, power laws are robust across different underlying feature distributions. Yet, at weak superposition, only power-law feature frequencies can lead to power-law losses. (b) When changing superposition by weight decay and varying feature frequency decay by α given $p_i \propto 1/i^\alpha$, we found roughly three distinct behaviors. For (b), $n = 1000$ and $m = 10 \sim 100$ (more in Appendix D.4).

features as

$$\phi_{1/2} = |\{i : \|W_i\|_2 > 1/2\}|/n, \quad (3)$$

namely, the fraction of rows with norm larger than $1/2$. We found that weight decay can tune superposition smoothly, with small weight decay γ giving strong superposition, i.e., $\phi_{1/2} \approx 1 \gg m/n$, and large weight decay corresponding to weak superposition, i.e., $\phi_{1/2} \sim m/n$ (Figure 3a). The ability of weight decay to tune superposition is robust to a variety of feature frequency distributions (Figure 3b), where $p_i \propto 1/i^\alpha$ and we control the decay speed of p_i via the data exponent α . Thus, weight decay provides a practical knob for controlling superposition in our study.

Our toy model differs from LLMs in architecture, data, and loss. Since we focus on representations rather than next-token prediction, we omit transformer layers. Conceptually, LLMs map a document to a token, with inputs and outputs in different spaces, while the toy model operates within a single shared space. Despite this, the toy model captures key aspects of language structure through engineered sparsity and feature importance, making its data structure aligned with that of LLMs at a higher level. While LLMs use cross-entropy loss and the toy model uses squared error, we will show that this does not affect the scaling behavior of the loss. Thus, the toy model is a suitable abstraction for studying representation-limited scaling.

3 Results

In the toy model, we found rich phenomena by varying superposition and feature importance. Reviewing the experiments in Figure 1, we set weight decay $\gamma = -1$ to have strong superposition and $\gamma = 0.1$ to have weak superposition. We compute R squared values from linear fits in log-log plots to quantify scaling behavior, assessing how closely the loss follows a power law to model dimension. We can see that at strong superposition, the losses are close to power laws, regardless of the underlying feature frequencies, yet the loss is a power law at weak superposition if the feature frequency p_i is a power law with rank i (Figure 4a). For a systematic scan, we next set $p_i \propto 1/i^\alpha$ and can vary the data exponent α to change how fast p_i decays consistently.² Unless otherwise specified, we set activation density E as 1 by default, whose value will be shown not to affect the scaling. Assuming a power-law form for the final test loss, $L \propto 1/m^{\alpha_m}$, we extract the model exponent α_m from the empirical fit. We fit the loss with a power law in all cases. The fitted α_m reveals how fast losses decay, even in the weak superposition regime where the power law fit is poor (Appendix D.4). Roughly, three distinct patterns emerge: (1) under weak superposition (positive γ , yellow box in

²The word or phrase frequency in natural language follows Zipf’s law, which is a power law.

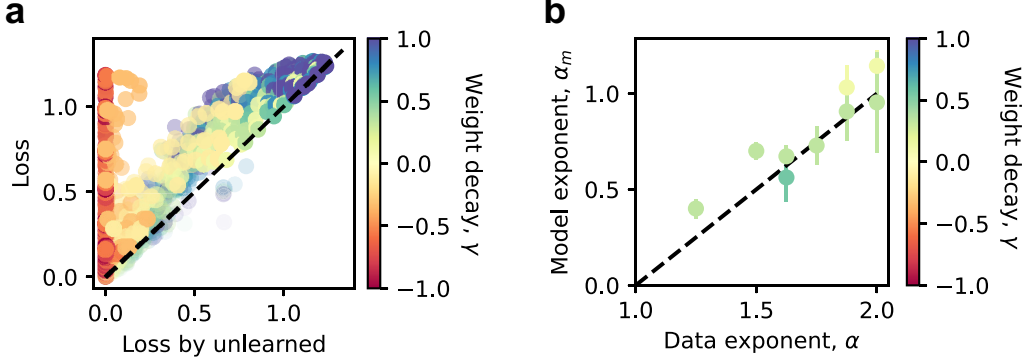


Figure 5: Loss at weak superposition can be well described by the expected number of activated but unlearned features. (a) The actual loss and theory at weak superposition (Equation (4)) agree when weight decay is positive. (b) For those closest to the ideal no superposition case (Appendix D.5), we expect $\alpha_m = \alpha - 1$, which is close to measured values.

Figure 4b), α_m is small, indicating slow loss decay; (2) under strong superposition and a wide range of small data exponents (red box), α_m remains robustly near 1; (3) for strong superposition with large data exponents (blue box), α_m increases with α . By interpreting these three patterns, we aim to understand when loss follows a power law with model dimension, and what determines the exponent when it does.

We begin by analyzing the weak superposition regime. Consider an idealized case where the top $\phi_{1/2}n$ most frequent features are perfectly represented, where $\phi_{1/2}$ is the fraction of represented features. The loss can then be written as:

$$L = \sum_{i=1}^{\phi_{1/2}n} \langle (y_i - x_i)^2 \rangle_x + \sum_{i=\phi_{1/2}n+1}^n \langle (y_i - x_i)^2 \rangle_x = \sum_{i=\phi_{1/2}n+1}^n \langle x_i^2 \rangle_x = \langle v^2 \rangle \sum_{i=\phi_{1/2}n+1}^n p_i. \quad (4)$$

The first equality follows from the definition of loss. The second holds because the first $\phi_{1/2}n$ features are assumed to be perfectly reconstructed, and the remaining features are ignored (i.e., $y_i = 0$ for $i > \phi_{1/2}n$). The final expression uses the definition of $x_i = u_i v_i$, where $v \sim U(0, 2)$, giving $\langle v^2 \rangle = 4/3$. To approximate the remaining sum, we use the integral $\int_{\phi_{1/2}n}^n p_i di$ and refer to this estimate as the loss by unlearned features. We find that, in the weak superposition regime, the actual losses closely match this prediction (Figure 5a). Slight discrepancies — where actual losses are slightly higher — can be attributed to the fact that models do not exactly learn the first $\phi_{1/2}n$ features. In contrast, in the strong superposition regime, the predicted loss by unlearned features is nearly zero, yet the actual loss remains non-zero, indicating additional sources of error. Focusing on cases closest to the ideal no-superposition scenario — where $\phi_{1/2}n = m$ and the first m features are represented — we observe that such cases occur when $\alpha > 1$ and yield a model exponent $\alpha_m \approx \alpha - 1$ (Figure 5b). This matches the theoretical expectation that $\int_m^n p_i di \propto m^{-\alpha+1}$ when $n \gg m$ and $\alpha > 1$. Thus, in the weak superposition regime, loss scaling is well described by the contribution of unlearned features, that is, the total frequency of features not represented by the model.

We can now explain when loss follows a power law in the weak superposition regime. When feature frequencies follow a power law, $p_i \propto 1/i^\alpha$ with $\alpha > 1$, the integral $\int_{\phi_{1/2}n}^n p_i di$ becomes a power law in $\phi_{1/2}$. And if the $\phi_{1/2}$ scales approximately as a power of model size m (ideally it is m/n). Then the loss also scales as a power law in m . Consequently, the deviations from power-law behavior seen in Figure 1b — for non-power-law p_i — are consistent with this explanation. Moreover, when $p_i \propto 1/i^\alpha$ with $\alpha > 1$, we expect the model exponent to be $\alpha_m \approx \alpha - 1$, which matches the relatively small values of α_m observed on the right side of Figure 4b. In summary, the weak superposition regime reveals a strong dependence of loss vs. model dimension scaling on the structure of feature frequencies in the data.

We now turn to the strong superposition regime, where all features are represented. In this regime, the loss arises primarily from the non-zero overlaps between representation vectors. For instance, consider the case where only feature j is activated in the data. Even for $i \neq j$, the output y_j receives

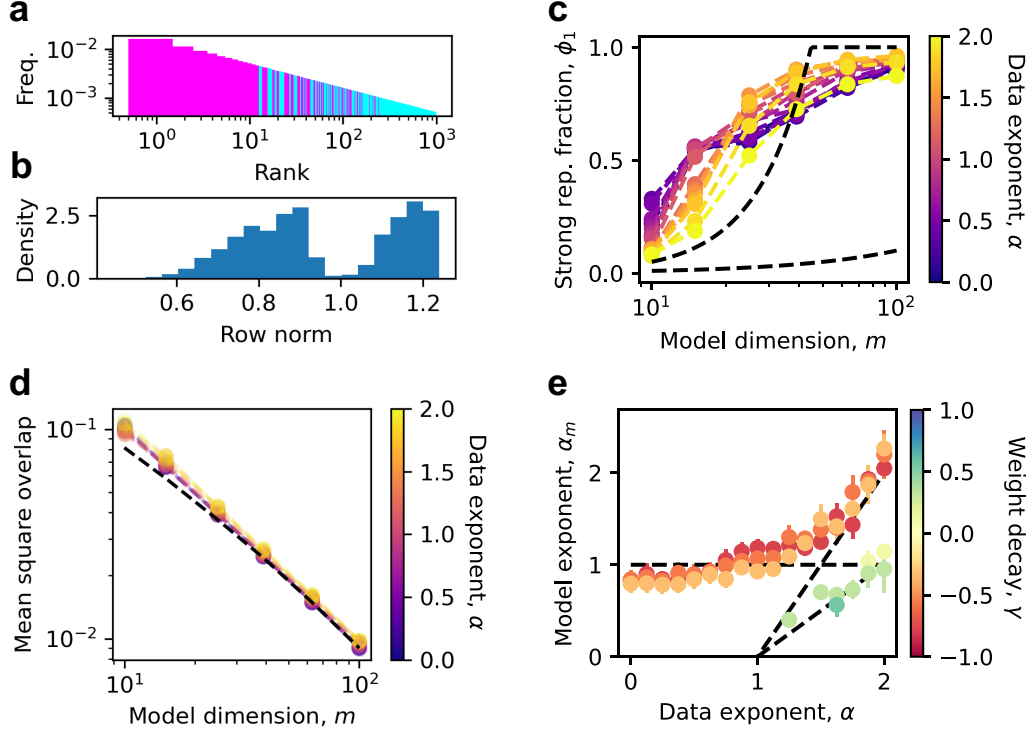


Figure 6: Loss scaling at strong superposition is mainly explained via geometry. (a) The more frequent the features are, the more likely their norms are greater than 1 (purple). (b) The row norm distribution is bimodal around 1. Related results can be found in Appendix D.6. (c) Fraction of strongly represented features ($\|W_i\|_2 > 1$) at strong superposition ($\gamma = -1$) is around $\min\{1, m^2/2n\}$ (fast increasing dashed line), which is much larger than m (slowly increasing dashed line). (d) The strongly represented features have decreasing overlaps as expected, where we plotted all the data when $\gamma < 0$, and the dashed line is κ^2 . (e) At strong superposition, $\alpha_m = 1$ (horizontal line) if the strongly represented features having small overlaps dominate, which is true for a wide range of α . But $\alpha_m = 2(\alpha - 1)$ (fast increasing dashed line) if weakly represented ones with large overlaps dominate, explaining the measured α_m when α is large.

contributions on the order of $W_i \cdot W_j$, leading to a loss that scales as $(W_i \cdot W_j)^2$ due to the squared error objective. The nonlinearity ReLU nonlinearity and bias terms do not affect the scaling behavior (Appendix A.1). Consequently, we study the overlaps between representation vectors when the number of vectors exceeds the dimensionality of the space.

We now provide theoretical expectations for vector overlaps. Consider ν unit vectors $w_i \in \mathbb{R}^m$ with $\nu \geq m$. It can be shown that [28]

$$\max_{i \neq j} |w_i \cdot w_j| \geq \sqrt{\frac{\nu - m}{m(\nu - 1)}} \equiv \kappa. \quad (5)$$

This lower bound, κ , approaches $\sqrt{1/m}$ when $\nu \gg m$. The bound is met when the vectors form an equal angle tight frame (ETF) [29–31], which minimizes interference and appears in contexts such as quantum measurements [32–35] and neural collapse [36–38]. However, ETFs in real space can only exist if $\nu \leq m(m+1)/2$ [29–31]. In our toy model, feature frequencies vary, so exact equal-angle configurations are suboptimal — important features may benefit from smaller overlaps. Nonetheless, when feature frequencies are nearly uniform (e.g., small data exponent α), $\sqrt{1/m}$ remains a good estimate. Although the rows W_i are not strictly unit vectors, they are close enough that the scaling behavior remains valid. In summary, we expect that in the strong superposition regime: (i) Many representation vectors can approximate a mutually orthogonal configuration, with typical overlaps scaling as $\sqrt{1/m}$; (ii) The number of such vectors can be much larger than m but still bounded (e.g., $\sim m^2/2$ for ETFs), so when n is too large, some feature vectors cannot be in such configurations, having larger overlaps.

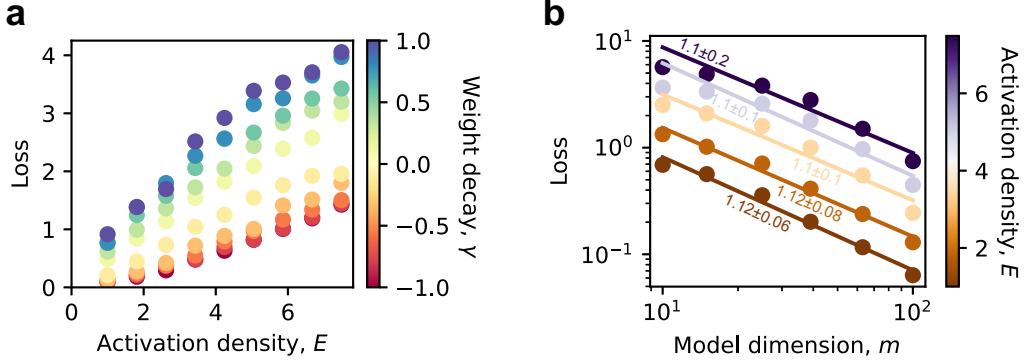


Figure 7: Activation density does not affect scaling exponents in our tests. (a) Loss is roughly proportional to activation density given the degree of superposition ($m = 63$, $n = 1000$). (b) So, E will only affect the coefficient but not the exponent when considering the power law with model dimension. We plot the evidence $\alpha_m \approx 1$ at strong superposition, see more in Appendix D.7.

With these theoretical expectations in hand, we evaluate our trained toy models. The first step is to test for the presence of a nearly orthogonal configuration among representation vectors. A preliminary clue comes from examining the norms of the rows W_i , which exhibit a bimodal distribution: one peak slightly below 1, corresponding to less frequent features, and another above 1, corresponding to the most frequent ones (Figure 6, a and b). This suggests that the most important features are represented in a nearly orthogonal manner, while less important ones exhibit greater overlap. To quantify this, we compute the fraction of rows with norm exceeding 1:

$$\phi_1 = |\{i : \|W_i\|_2 > 1\}|/n, \quad (6)$$

which significantly exceeds m/n and is around the ETF expectation $m^2/2n$ (Figure 6c). For these rows, the mean squared overlaps match the theoretical bound κ^2 well (Figure 6d). Although these vectors do not form a perfect ETF, their configuration is much closer to an ETF than random initialization (Appendix D.6). In contrast, rows with norms less than 1 exhibit larger overlaps on average (Appendix D.6). We refer to rows with $\|W_i\|_2 > 1$ as strongly represented features and those with smaller norms as weakly represented. These results confirm our expectations: strongly represented features attempt to form an approximately orthogonal configuration with typical overlaps scaling as $\sqrt{1/m}$; their number exceeds m a lot but remains bounded, necessitating the presence of weakly represented features with larger overlaps.

In the strong superposition regime, the loss is mainly determined by a weighted sum of squared overlaps between representation vectors. For strongly represented features, where overlaps scale as $\sqrt{1/m}$, the loss contribution scales as $1/m$. In contrast, weakly represented features can have large, constant overlaps. In the worst case, their total contribution is proportional to $\sum_{i=\phi_1 n}^n p_i \sim m^{-2(\alpha-1)}$, where we approximate $\phi_1 n \approx m^2/2$. (See Appendix A.1 for loss decomposition details) If the loss is dominated by the strongly represented features, we expect the overall loss to scale as $1/m$. In the opposite limit — when weakly represented features dominate — the loss scales as $m^{-2(\alpha-1)}$, meaning the effective model exponent becomes $\alpha_m \approx 2(\alpha - 1)$. Although we do not solve the toy model analytically, we hypothesize that as α increases (i.e., feature frequencies become more skewed), optimization prioritizes minimizing the loss of the most frequent features. As a result, the remaining loss is mostly due to the weakly represented ones. Empirically, we find a simple and elegant rule (Figure 6e):

$$\alpha_m \approx \max\{1, 2(\alpha - 1)\}. \quad (7)$$

In short, the optimization selects whether strongly or weakly represented features dominate the loss, whichever leads to faster decay with model size m . We now have a full picture that the origin of $\alpha_m = 1$ lies in the geometric constraints on vector overlaps, and this scaling is robust as long as the feature frequency distribution is not too skewed.

We also studied the effect of the number of expected activated features or activation density E , which was set to 1. By fixing data exponent $\alpha = 1$, which will be shown to be relevant to natural language, we can scan different superposition degrees and activation densities. Since $p_i \leq 1$ is required, which

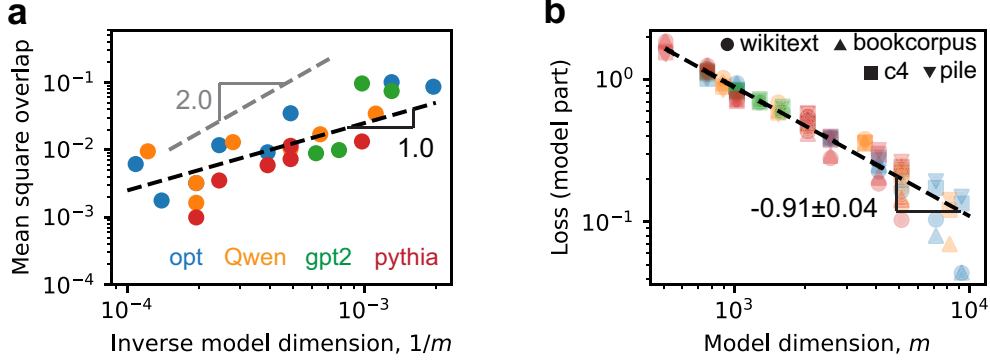


Figure 8: Superposition may explain the neural scaling law observed in actual LLMs. We evaluate four open-sourced model classes, Opt [39], GPT2 [40], Qwen [41], and Pythia [42], which have model sizes from around 100M to 70B (details in Appendix C). (a) We found the mean square overlaps of $W_i/\|W_i\|_2$ roughly follow $1/m$ scaling, where W is the language model head. (b) The model class is reflected by color as panel a, while we use shapes for evaluation datasets [43–46]. The loss related to model size is fitted as a power law, yielding empirical $\alpha_m = 0.91 \pm 0.04$ close to 1 (details in Appendix D.8).

is equivalent to $p_1 \leq 1$, we have $E \leq \sum_{i=1}^n 1/i^\alpha$, setting the upper bound for our scanning. We found that loss is approximately proportional to activation density E (Figure 7a). This fact suggests that the power law exponent should not change, which we confirmed (Figure 7b). Under a controlled superposition degree, activation density linearly increases loss and thus does not affect the scaling exponents in our experiments.

Finally, we explore how the phenomena found in our toy model might be relevant to real LLMs [39–42]. As a simple mapping, we treat tokens as atomic features, with data dimension n equal to the vocabulary size. The model dimension m for LLMs is known. We analyze the language model head, denoted by the weight matrix W . Through the norm and interference distributions of the rows of W , we claim LLMs are in superposition (Appendix D.8). If we measure token frequency, it follows a power law with exponent α close to 1 (Appendix D.8). We conclude, based on the knowledge from toy models, that LLMs operate in a superposition regime, and expect loss to be dominated by small overlaps of the strongly represented $\sim \sqrt{1/m}$. We next calculated the mean squared overlaps of normalized rows $W_i/\|W_i\|_2$, and found they roughly obey $1/m$ scaling (Figure 8a). We argue that cross-entropy loss, given that the overlaps are small in absolute value, can be expanded and approximately scales as the mean square overlaps (Appendix A.2). We therefore expect the loss of representation-limited LLMs to have $1/m$ scaling. LLM losses are close to a linear function of $1/m$ (Appendix D.8). Yet when $m \rightarrow \infty$, the extrapolation of losses does not hit 0. The non-zero intersection can be due to intrinsic uncertainty in language. Increasing model sizes decreases “wrong” interferences but cannot eliminate uncertainty in the data. So, as in previous papers where loss is decomposed into model size part, dataset size part, and a constant [3], we fit our loss values by the following,

$$L = C_m/m^{\alpha_m} + L_{\setminus m}, \quad (8)$$

where the model size part C_m/m^{α_m} is universal (model size is a function of m), and $L_{\setminus m}$ contains loss irrelevant to model size, depending on the evaluation dataset and model class. The fitting yields $\alpha_m = 0.91 \pm 0.04$ (Figure 8b). We inferred from the Chinchilla models [3] that due to model size $N \propto m^{2.52 \pm 0.03}$ (Appendix D.8), $\alpha_m = (2.52 \pm 0.03) \times \alpha_N = 0.88 \pm 0.06$, where $\alpha_N = 0.35 \pm 0.02$ [47] is the power-law exponent of loss with model size. The exponents α_m from LLMs are close to 1.

In conclusion, we presented a comprehensive picture through the toy model, at different superposition degrees and data structures, of when the loss scales as a power law with model dimension or size, and what governs the scaling exponent. We confirmed that LLMs operate in the strong superposition regime. Moreover, LLMs [39–42] quantitatively match toy model predictions: the model exponent α_m is close to 1, and representation overlap scales as $\sqrt{1/m}$. The inferred exponent α_m for Chinchilla models [3] is also close to 1. Superposition is therefore an important mechanism underlying the observed scaling laws.

4 Related works

Neural scaling laws were first characterized empirically [2], demonstrating that for LLMs, the cross-entropy loss improves predictably as a power-law with increased model size (parameters), dataset size, or compute, over multiple orders of magnitude. This finding is built on earlier observations (e.g. [48]) that deep learning performance scales in a smooth power-law fashion with data and model growth. Key works showed the surprisingly universal nature of such scaling behaviors across architectures and tasks [2–4], directing further development of LLMs.

Several heuristic toy models have been proposed to explain neural scaling laws. One common view is that models aim to fit data manifolds or functions, and the scaling exponents depend strongly on the structure of the data [14, 15]. Another group of models assumes the network learns discrete features or skills [18, 19], whose importance follows a power-law distribution, giving results the same as ours in the weak superposition regime. One toy model predicts that loss scales inversely with model width [24], arguing that parameters independently perform the same task with noise, and the scaling follows from the central limit theorem. However, this model applies mainly in the overparameterized regime.

More formal approaches from statistical learning theory still rely on similar heuristics. The scaling behavior depends on how the limits of dataset size and model size are taken. When the dataset is fixed and model size grows to infinity, the system is variance-limited, and loss scales as $1/m$ by central limit theorem arguments [15]. When the dataset size grows to infinity first, the loss scaling enters the resolution-limited regime. In kernel methods, this leads to $\alpha_m = \alpha - 1$ [15–17], consistent with our weak superposition regime. Here, α is the exponent of the power-law decay of kernel eigenvalues, which can be interpreted as abstract feature importance. This regime has also been described as fitting the data manifold [15]. Thus, previous studies mainly address weak superposition or overfitting regimes, while our work reveals new behaviors in the underfitting (where the number of features far exceeds the model dimension) and strong superposition regime, which might be more relevant to LLMs.

Our toy model is Anthropic’s model of superposition [26], with changes in data sampling. The original study explored how data structure influences superposition but did not explicitly control it. Related models have appeared in compressed sensing [49–52] and neural information processing [53, 54], though key differences arise due to distinct contexts and objectives.

5 Discussion

We acknowledge that our work is built on observations of the toy model and shallow theoretical analysis without rigorously solving the toy model. We are thus limited to explaining some deeper questions. For example, Equation (7) is an observation. Although we explained the two extremes, $\alpha_m = 1$ and $\alpha_m = 2(\alpha - 1)$, and provided heuristics for a transition, we cannot predict the exact transition.

Neural scaling laws also include scaling laws with dataset size and with training steps, which we did not study. At each step, a fixed number of new data are used for optimization. So, we expect the scaling with the total data amount and that with training steps will be the same, similar to the results at weak superposition [19]. However, in the strong superposition regime, data or training step scaling is related to angle distribution and how angles between representations evolve, which cannot be easily explained without rigorous solving. Future works with better theoretical analysis of our toy model are needed to answer these detailed or extended questions.

Our analysis suggests LLMs are in the strong superposition regime, but the underlying reasons were not studied. Specifically, LLMs do not need weight growth $\gamma < 0$ to have strong superposition. The use of cross-entropy loss or softmax function may be important — minimizing the interference from an unimportant token prefers to make the representation non-zero with negative overlaps with others. Another contributing factor may be that $m^2/2 > n$ in LLMs, which could make it easier for a nearly mutually orthogonal configuration to form, thereby favoring superposition.

We focused on representation loss in this study, yet LLMs should also have losses due to parsing or processing in the transformer layers. As a conceptual discussion, we imagine that the loss associated with model size can be written as

$$C_m/m^{\alpha_m} = f_m(m) + f_\ell(\ell), \quad (9)$$

where ℓ is the depth of the LLM, f_m and f_ℓ are two functions capturing the loss due to representation and parsing, respectively. We can write the equality because ℓ depends on m in LLMs [39–42]. Given model size N , m and ℓ are constrained (roughly, $N \propto m^2 \ell$). There is an optimal m - ℓ relationship such that the loss $f_m(m) + f_\ell(\ell)$ can be minimized given N [55]. At this optimal m - ℓ relationship, $f_m(m)$ and $f_\ell(\ell)$ should be balanced. One can see that the empirical m - ℓ relationship is not optimal according to [55] but is close. Therefore, following such m - ℓ relation, we expect $f_\ell(\ell)$ to be similar to $f_m(m)$. And if $f_m(m) \sim 1/m$ due to superposition and $f_\ell(\ell)$ is similar, we can measure an empirical $\alpha_m \approx 1$ from data, which is true. So, we conclude that superposition is an important mechanism underlying the loss and scaling laws we see. However, since we do not know the function $f_\ell(\ell)$, we cannot predict the optimal m - ℓ relation or the exponent α_N .

A natural future direction is to study the parsing-limited scaling (i.e, $f_\ell(\ell)$ function). We hypothesize that parsing-limited scaling may resemble the scaling behavior observed when fitting functions or manifolds. The open-source LLMs we examined, as well as those studied in prior neural scaling law papers [2, 3], span model sizes from 100M to 70B parameters, where width (model dimension) varies. However, in the regime from 70B to 700B [39–42], model scaling seems to focus on increasing depth. We therefore expect parsing-limited scaling to become important for models larger than 70B. It is also plausible that the observed scaling of inference time [56] is connected to this parsing-limited regime. The study of neural scaling laws can be refined by distinguishing between width-limited and depth-limited regimes. In each regime, there should be loss decay behaviors with model size, dataset size, and training steps, highlighting the need for further investigation.

Beyond explaining existing phenomena, our results may offer guidance for future LLM development and training strategies. If our framework accounts for the observed neural scaling laws [3], we suggest that this kind of scaling is reaching its limits — not because increasing model dimension is impossible, but because it is inefficient. Recognizing that superposition benefits LLMs, we propose that encouraging superposition could enable smaller models to match the performance of larger ones and make training more efficient. Recent architectures such as nGPT [57], which constrain hidden states and weight matrix rows to lie on the unit sphere, demonstrate improved performance and faster training. We interpret these methods as analogous to our weight growth mechanism, both of which promote superposition. Conversely, weight decay is expected to suppress superposition. Prior work has shown that weight decay in LLMs is used primarily for training stability rather than performance gains. Alternative optimization techniques that stabilize training without weight decay have shown promising results [58], potentially due to enhanced superposition. In summary, promoting strong superposition may be a fruitful direction for the design of both LLM architectures and optimization algorithms.

As a side note, while pre-training loss is a key indicator of model performance, it should not be the sole metric of interest. At the same loss level, LLMs with different degrees of superposition may exhibit differences in emergent abilities such as reasoning or trainability via reinforcement learning [59]. Understanding these differences is an important direction for future research.

In conclusion, we demonstrated that superposition of representations can give rise to fast power law decay of loss with model size — a phenomenon that holds across a wide range of data structures and may help explain the neural scaling laws observed in LLMs [3]. Our results contribute to a deeper understanding of modern artificial intelligence systems. This work also opens rich directions for future research. We hope our insights will support the continued development and training of more capable and efficient LLMs.

References

- [1] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.
- [2] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- [3] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Jack W Rae, Alana Lai, Joel Wang, Katie Millican, Susannah Young, Olivier Tieleman, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

- [4] Tom Henighan, 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. Scaling laws for autoregressive generative modeling, 2020.
- [5] OpenAI. Gpt-4 technical report, 2023. <https://openai.com/research/gpt-4>.
- [6] Haotong Qin, Ge-Peng Ji, Salman Khan, Deng-Ping Fan, Fahad Shahbaz Khan, and Luc Van Gool. How good is google bard’s visual understanding? an empirical study on open challenges. *arXiv preprint arXiv:2307.15016*, 2023. <https://arxiv.org/abs/2307.15016>.
- [7] Aitor Lewkowycz, Pasquale Minervini, Jacob Andreas, et al. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*, 2022. <https://arxiv.org/abs/2206.14858>.
- [8] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Andrew Hartshorn, Elvis Saravia, Andrew Poulton, et al. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022. <https://arxiv.org/abs/2211.09085>.
- [9] Stephen Wolfram. Wolframalpha as the computation engine for gpt models, 2023. <https://www.wolfram.com/wolfram-alpha-openai-plugin>.
- [10] Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 2024. <https://www.nature.com/articles/s41586-024-00001-2>.
- [11] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. <https://arxiv.org/abs/2107.03374>.
- [12] GitHub. Github copilot: Your ai pair programmer, 2022. <https://github.com/features/copilot>.
- [13] Jack W Rae, Sebastian Borgeaud, Trevor Cai, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021. <https://arxiv.org/abs/2112.11446>.
- [14] Utkarsh Sharma and Jared Kaplan. Neural scaling laws: A low-dimensional perspective. *Journal of Machine Learning Research*, 23(228):1–29, 2022.
- [15] Yasaman Bahri, Ethan Dyer, Jaehoon Lee, Jared Kaplan, and Surya Ganguli. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 2024.
- [16] Brandon Bordelon, Abdurrahman Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression. In *International Conference on Machine Learning*, pages 1024–1034. PMLR, 2020.
- [17] Alexander Maloney, Daniel A Roberts, and James Sully. A solvable model of neural scaling laws. In *Advances in Neural Information Processing Systems*, volume 35, pages 29112–29125, 2022.
- [18] Marcus Hutter. Learning curves for language modeling. *arXiv preprint arXiv:2110.02861*, 2021.
- [19] Eric Michaud, Ethan Liu, Kyle McDonell, Henry Lin, and Max Tegmark. Quantization and emergence in neural scaling. In *Advances in Neural Information Processing Systems*, 2023.
- [20] Ziming Liu, Yizhou Liu, Eric J Michaud, Jeff Gore, and Max Tegmark. Physics of skill learning. *arXiv preprint arXiv:2501.12391*, 2025.
- [21] David Hernandez, Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, and Tom Brown. Scaling laws for mechanistic interpretability. *Transformer Circuits Thread*, 2022.
- [22] Daniel Brill. Toward a unifying theory of neural scaling. *arXiv preprint arXiv:2401.12447*, 2024.

- [23] Stefano Spigler, Mario Geiger, and Matthieu Wyart. Asymptotic learning curves of kernel methods: Beyond the teacher–student setting. *Advances in Neural Information Processing Systems*, 33:21538–21549, 2020.
- [24] Jinyeop Song, Ziming Liu, Max Tegmark, and Jeff Gore. A resource model for neural scaling law, 2024.
- [25] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy, 2018.
- [26] Nelson Elhage, 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. Toy models of superposition. *Transformer Circuits Thread*, 2022.
- [27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.
- [28] L. Welch. Lower bounds on the maximum cross correlation of signals. *IEEE Transactions on Information Theory*, 20(3):397–399, 1974.
- [29] Peter G Casazza and Gitta Kutyniok. A survey of finite frames: Theory and applications. *Springer*, 2012.
- [30] Thomas Strohmer and Robert W Heath Jr. Grassmannian frames with applications to coding and communication. *IEEE Transactions on Information Theory*, 49(10):2934–2947, 2003.
- [31] Matthew Fickus and Dustin G Mixon. Real and complex equiangular tight frames. *arXiv preprint arXiv:1201.3155*, 2012.
- [32] Joseph M Renes, Robin Blume-Kohout, A J Scott, and Carlton M Caves. Symmetric informationally complete quantum measurements. *Journal of Mathematical Physics*, 45(6):2171–2180, 2004.
- [33] Yizhou Liu and John B. DeBroy. Relating measurement disturbance, information, and orthogonality. *Phys. Rev. A*, 104:052216, Nov 2021.
- [34] Yizhou Liu and Shunlong Luo. Quantifying unsharpness of measurements via uncertainty. *Phys. Rev. A*, 104:052227, Nov 2021.
- [35] Yizhou Liu, Shunlong Luo, and Yuan Sun. Total, classical and quantum uncertainties generated by channels. *Theoretical and Mathematical Physics*, 213(2):1613–1631, 2022.
- [36] Lu Huang, Vardan Papyan, and Thomas Strohmer. Neural collapse: A phenomenon in the terminal phase of deep learning. *arXiv preprint arXiv:2008.08186*, 2020.
- [37] Vardan Papyan, X.Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.
- [38] Gilad Tirer and Raja Giryes. Extended neural collapse: Explicit solutions and generalization guarantees. In *Advances in Neural Information Processing Systems*, volume 35, pages 24225–24238, 2022.
- [39] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022.
- [40] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

- [41] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- [42] Stella Biderman, 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. Pythia: A suite for analyzing large language models across training and scaling, 2023.
- [43] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.
- [44] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.
- [45] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [46] Yukun Zhu, Ryan Kiros, Richard S Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *arXiv preprint arXiv:1506.06724*, 2015.
- [47] Tamay Besiroglu, Ege Erdil, Matthew Barnett, and Josh You. Chinchilla scaling: A replication attempt. *arXiv preprint arXiv:2404.10102*, 2024.
- [48] Joel Hestness, Sharan Narang, Niki Ardalani, Greg Diamos, Heewoo Jun, Hossein Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yan Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.
- [49] David L Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [50] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.
- [51] Richard G Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 24(4):118–121, 2007.
- [52] Madhu S Advani and Surya Ganguli. Statistical mechanics of optimal convex inference in high dimensions. *Physical Review X*, 6(3):031034, 2016.
- [53] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.
- [54] Behtash Babadi and Haim Sompolsky. Sparseness and expansion in sensory representations. *Neuron*, 83(5):1213–1226, 2014.
- [55] Yoav Levine, Noam Wies, Or Sharir, Hofit Bata, and Amnon Shashua. The depth-to-width interplay in self-attention. *arXiv preprint arXiv:2006.12467*, 2020.
- [56] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.
- [57] Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. ngpt: Normalized transformer with representation learning on the hypersphere. *arXiv preprint arXiv:2410.01131*, 2024.

- [58] Yizhou Liu, Ziming Liu, and Jeff Gore. Focus: First order concentrated updating scheme. *arXiv preprint arXiv:2501.12243*, 2025.
- [59] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

A Theoretical analysis

A.1 Toy model loss

We provide a simple analysis for toy model loss scaling. The expected loss in the weak superposition regime is well explained by Equation (4). We do not need to repeat it here.

In the strong superposition regime, we consider an even special data sampling, where data x is sampled such that each data point has and only has one activated feature. The frequency for feature i to be activated is still p_i . After determining which feature is activated, say i , we still sample x_i as v_i from $U(0, 2)$. This sampling is different from the experiments. Yet, since we learned that activation density does not affect scaling exponent (Figure 7), we expected this analysis to predict at least the scaling exponent. Under the assumptions, we have

$$L = \sum_{i=1}^n p_i \left\langle \sum_{j \neq i} \text{ReLU}^2(W_j \cdot W_i v_i + b_j) + (\text{ReLU}(W_i \cdot W_i v_i + b_i) - v_i)^2 \right\rangle_{v_i}. \quad (10)$$

We are unable to solve for the optimal W and b such that this loss L is minimized. Yet, it is easy to see that we want $\|W_i\|_2^2$ to be close to 1, $W_j \cdot W_i$ to be as small as possible, and b_j to be small negative values of the same order of magnitude as $W_i \cdot W_j$, such that the interference terms $\text{ReLU}(W_j \cdot W_i v_i + b_j)$ may vanish and the recovered feature value $\text{ReLU}(W_i \cdot W_i v_i + b_i)$ can be close to the real one v_i .

We now review our proposed configuration, consisting of strongly represented and weakly represented features. The first $\phi_1 n$ most important features are considered to be strongly represented, whose overlap with any other representation scales as $\sqrt{1/m}$. The rest of the features are weakly represented and squeezed into a small angle such that they have small overlaps with the strongly represented, while they can have large overlaps with each other. With such a configuration, the first $\phi_1 n$ terms in the summation of Equation (10) will scale as $1/m$ since each term $\langle \dots \rangle_{v_i}$ scales as $1/m$. The rest of the terms in Equation (10), in the worst scenario that the terms $\langle \dots \rangle_{v_i}$ do not decrease obviously with m , will be proportional to $\sum_{i=\phi_1 n}^n p_i$. If the strongly represented features dominate, we can have $1/m$ scaling for the loss. On the contrary, when the weakly represented dominate, we expect the loss to have scaling like $\sum_{i=\phi_1 n}^n p_i$. More specifically, if $p_i \sim 1/i^\alpha$ ($\alpha > 1$) and we use $m^2/2$ to approximate $\phi_1 n$, the loss scales as $1/m^{2(\alpha-1)}$.

A.2 Cross-entropy loss

We provide the reason why cross-entropy loss also scales as squared overlaps. We consider the last hidden state after going through the normalization layer is $W_i/\|W_i\|_2$, such that the output should be the i th token. By constructing such an example, we ignore possible loss due to parsing but focus on the loss just due to representation. The loss from this data point is

$$L = -\ln \frac{e^{\|W_i\|_2}}{\sum_j e^{W_i \cdot W_j / \|W_i\|_2}} = \ln \left[1 + \sum_{j \neq i} e^{W_i \cdot W_j / \|W_i\|_2 - \|W_i\|_2} \right]. \quad (11)$$

We assume that $W_i \cdot W_j / \|W_i\|_2$ is much smaller than 1 since we know the overlap scale as $1/m$. We then approximate the loss via Taylor expansion

$$L = \ln \left[1 + (n-1)e^{-\|W_i\|_2} + \sum_{j \neq i} [W_i \cdot W_j / \|W_i\|_2 + (W_i \cdot W_j / \|W_i\|_2)^2 / 2] e^{-\|W_i\|_2} \right]. \quad (12)$$

In the first thought, the summation $\sum_{j \neq i} W_i \cdot W_j$ should be zero since there are positive and negative overlaps distributed evenly if the vectors span the whole space. But in language, one sentence can

have different continuations, connecting different tokens. For example, both putting “cats” or “dogs” after “I like” are legit. The existence of data “I like” then will tend to squeeze different tokens closer to each other. The summation $W_i \cdot W_j$ should be a small positive constant $\epsilon_{D,i}$ related to the correlation in data. The reason we keep the second-order term is clear now as they are the lowest order terms related to model sizes. We keep expanding the \ln function and have

$$L = (n-1)e^{-\|W_i\|_2} + \frac{\epsilon_{D,i}e^{-\|W_i\|_2}}{\|W_i\|_2} + \frac{1}{2} \sum_{j \neq i} \left(\frac{W_i \cdot W_j}{\|W_i\|_2} \right)^2 e^{-\|W_i\|_2}. \quad (13)$$

The part related to the model size is mainly

$$L_m = \frac{1}{2} \sum_{j \neq i} \left(\frac{W_i \cdot W_j}{\|W_i\|_2} \right)^2 e^{-\|W_i\|_2}. \quad (14)$$

In this construction, one can see that once $\|W_i\|_2$ is sufficiently large, the loss can be arbitrarily low, which does not happen in reality. The reason is still related to the intrinsic uncertainty in language data. If one sentence can have different continuations, we need in the hidden space, a region that can lead to large probabilities over different tokens. However, when the norm is too large, one will find that the hidden space is sharply separated — each hidden state yields high probability only on one token. We then expect the norm $\|W_i\|_2$ to be as large as possible such that $ne^{-\|W_i\|_2}$ is small while $\|W_i\|_2$ is upper bounded by intrinsic data uncertainty. Therefore, $\|W_i\|_2$ should not depend on model size much (verified in Appendix D.8). The loss related to model size L_m then scales as $1/m$ since the cosine similarity scales as $\sqrt{1/m}$ and L_m is related to the squared cosine similarity in the lowest order approximation.

B Toy model training

In this Appendix, we explain how we trained the toy models and obtained raw data. There are two classes of toy models trained. The first one is a large toy model with data dimension $n = 10240$, which is reported in Figure 1 and Figure 4 to show scaling behavior across around two orders of magnitude. The other toy model class is small toy models fixing $n = 1000$, such that we can scan more hyperparameters. Figures 3, 4, 5, 6, and 7 use small toy models.

B.1 Large toy models

We implemented a neural network experiment to study the scaling of feature representation and recovery. The toy model is defined as a two-layer neural network with ReLU activation (see Figure 2).

The hyperparameters are give as follows.

- Data dimension n : 10240
- Model dimension m : Varied exponentially from 2^3 to 2^{10}
- Batch size: 2048 (tested up to 8192, which does not affect final loss)
- Total training steps: 20000 (tested up to 80000, which does not affect final loss)
- Learning rate: Initially set to 0.02, scaled according to hidden dimension
- Weight decay: -1.0 for strong superposition, and 0.1 for weak superposition
- Device: Training performed using one V100 GPU, with floating-point precision (FP32)

Data points x were synthetically generated at each training step according to Equation (1) to simulate feature occurrence frequencies. We considered three distributions with activation density $E = 1$:

- Exponential: $p_i \propto e^{-i/400}$
- Power-law: $p_i \propto i^{-1.2}$
- Linear: $p_i \propto n - i$

We employed the AdamW optimizer with distinct learning rates and weight decay settings for the weight matrix W and bias vector b . Specifically, for weight matrix W , learning rate was scaled as $\text{lr} \times (8/m)^{0.25}$ with specified weight decay. And for bias vector b , a learning rate of $2.0/m$ was used with no weight decay. A cosine decay learning rate schedule with a warm-up phase (5% of total steps) was implemented. At each training step, input data batches were dynamically generated based on the selected probability distribution. The final test loss is calculated across newly sampled data with a size being 100 times the batch size.

The model and optimizer were compiled and executed on a CUDA-enabled GPU for efficient training. After training, weight matrices W and training losses were stored and analyzed.

Final outputs, including weight matrices and training loss histories, were saved in PyTorch format for subsequent analysis and visualization.

This setup provided a structured exploration of feature representation scaling under varying dimensions and distributions, crucial for understanding superposition and scaling laws in neural networks.

The code can be found in `exp-17.py`.

B.2 Small toy models

We conducted numerical simulations using a neural network model designed for feature recovery. The objective was to analyze the model’s behavior across various conditions involving feature frequency skewness (controlled by data exponent α), model dimensions, and weight decay parameters.

In the small toy models reported in Figures 3, 4, 5, and 6, we set the hyperparameters as

- Feature dimension n : Fixed at 1000.
- Hidden dimension m : Varied logarithmically between 10 and 100, across 6 distinct sizes.
- Batch size: 2048.
- Training steps: 20000 steps for each condition.
- Learning rate: Initialized at 1×10^{-2} , dynamically adjusted using cosine decay scheduling with a warm-up phase of 2000 steps.
- Weight decay: Explored systematically from -1.0 to 1.0, in increments of 0.22 approximately (10 discrete values).
- Data exponent α : Ranged linearly from 0 to 2, with 17 discrete steps.

In Figure 7, we fix data exponent $\alpha = 1$ while scan 9 activation densities linearly from 1 to the maximal value $\sum_{i=1}^n 1/i$. All other settings are the same.

Synthetic data was generated for each batch based on a power-law probability distribution, defined as:

$$p_i \propto \frac{1}{i^\alpha} \quad \text{where } i \in \{1, 2, \dots, n\}$$

with the condition $\sum_i p_i = E$. Each element of the batch data x was randomly activated based on this probability, then scaled by a uniform random value between 0 and 2.

At each training step, input batches were regenerated, and the learning rate was updated following the cosine decay schedule described above.

The training performance was evaluated using Mean Squared Error (MSE) loss computed between the network output and input batch data at every step. Final weights were saved for further analysis. The final test loss is calculated across newly sampled data with a size being 100 times the batch size.

The simulations were performed in parallel using 96 CPU cores, where each core executed one distinct parameter combination defined by the weight decay and data exponent values. Or, in Figure 7, the parameter combination is defined by weight decay and activation density values.

Loss histories and trained weight matrices were saved separately for post-experiment analysis. Files were systematically indexed to indicate the corresponding experimental parameters. This detailed setup facilitated a comprehensive investigation of model behavior under diverse training and data distribution conditions.

The code can be found in `exp-10.py`, `exp-10-3.py`, and `exp-15.py`.

C LLM evaluation

C.1 Overlap analysis

We analyzed the row overlaps of the language model head weight matrices among various large language models (LLMs) to investigate the geometric properties of their hidden spaces.

We selected models from the following families, varying widely in parameter count:

- OPT (from OPT-125m to OPT-66b)
- Qwen2.5 (from 0.5B to 72B)
- GPT-2 (GPT2, GPT2-Medium, GPT2-Large, GPT2-XL)
- Pythia (from 70m to 12B)

Weights were downloaded directly from Hugging Face model repositories. For each model, the weight matrix or language modeling head was normalized by its row norms:

$$W_i \leftarrow \frac{W_i}{\|W_i\|_2 + \epsilon}, \quad \epsilon = 10^{-9},$$

where ϵ is for numerical stability.

We computed the pairwise absolute cosine overlaps between all normalized vectors using batch-wise computations for efficiency. The overlap between embedding vectors W_i and W_j is given by:

$$\text{overlap}(W_i, W_j) = \left| \frac{W_i \cdot W_j}{\|W_i\|_2 \|W_j\|_2} \right|.$$

To handle large embedding matrices efficiently, overlaps were computed in batches (size of 8192 vectors).

We calculated two key statistics for the overlaps within each model:

- Mean Overlap: The average of absolute overlaps for all unique vector pairs:

$$\text{mean_overlap} = \frac{\sum_{i < j} \text{overlap}(W_i, W_j)}{n(n-1)/2}$$

- Overlap Variance: Calculated as:

$$\text{variance_overlap} = \frac{\sum_{i < j} (\text{overlap}(W_i, W_j) - \text{mean_overlap})^2}{n(n-1)/2}$$

From these values, we can calculate mean square overlaps as $\text{mean_overlap}^2 + \text{variance_overlap}$.

The calculations were accelerated using GPU resources (CUDA-enabled) to efficiently handle computations involving extremely large matrices.

Results including mean overlaps, variances, and matrix dimensions were recorded for comparative analysis across model sizes and architectures.

The code is in `overlap-0.py`.

C.2 Evaluation loss

This experiment aims to evaluate multiple large language models (LLMs) efficiently using model parallelism and dataset streaming techniques. The models were assessed on standard text datasets to measure their predictive performance systematically.

Models were selected from Hugging Face and evaluated using a model-parallel setup:

- OPT series
- Qwen2.5 series

- GPT-2 series
- Pythia series

We used the following publicly available datasets for evaluation:

- Wikitext-103: Standard English language modeling dataset.
- Pile-10k: A subset of The Pile, designed for diverse textual data.
- C4: Colossal Clean Crawled Corpus, containing large-scale web text.
- BookCorpus: Large-scale collection of books used for unsupervised learning.

Datasets were streamed directly, efficiently sampling 10000 text segments with a maximum sequence length of 2048 tokens ($\sim 2 \times 10^7$ tokens).

Texts from datasets were tokenized using the respective model-specific tokenizers. Tokenization involved truncation and manual padding to uniform batch lengths. Specifically, padding tokens were assigned an ID of 0, and label padding utilized a special token (-100) to ensure they did not contribute to loss computations.

Each model was loaded using Hugging Face’s `AutoModelForCausalLM` with model parallelism enabled, allowing the evaluation of large models that exceed single-GPU memory limits. Evaluations were conducted in batches, employing a `DataLoader` with a custom collate function for optimized memory use.

The model’s predictive performance was assessed by computing loss values internally shifted by the Hugging Face library, suitable for causal language modeling.

Model parallelism was implemented to efficiently distribute computations across multiple GPUs, leveraging CUDA-enabled hardware.

For each model and each dataset, we run one evaluation and save the evaluation losses.

Random seeds and deterministic sampling ensured reproducible dataset selections, though explicit seed settings were noted as commented options within the implementation.

Evaluation results, including loss metrics and potentially intermediate model states, were systematically stored for detailed post-analysis.

The code is in `cali-1.py`.

C.3 Token frequency

The purpose of this analysis is to compare token frequencies generated by different tokenizers across several widely-used textual datasets. Understanding these frequencies helps in assessing the representational capacity and efficiency of tokenizers used by various large language models.

We considered the same four datasets mentioned for LLM evaluation. And we use four different tokenizers from the four model classes we evaluated.

Each tokenizer processed textual data from the specified datasets, streaming data directly to efficiently handle large-scale inputs. A target of 1,000,000 tokens per tokenizer-dataset pair was set to ensure sufficient statistical representativeness.

For each dataset-tokenizer combination:

1. Text samples were streamed directly from the datasets.
2. Text was tokenized without adding special tokens (e.g., EOS).
3. Token frequencies were counted and accumulated until the target token count (1 million tokens) was reached.
4. Token frequencies were saved as JSON files for subsequent detailed analyses.

Token frequency data was systematically stored for each tokenizer and dataset combination, enabling comparative analyses of token distributions. The data files provide foundational insights into tokenizer efficiency and coverage across diverse textual domains.

The code is in `token-freq-0.py`.

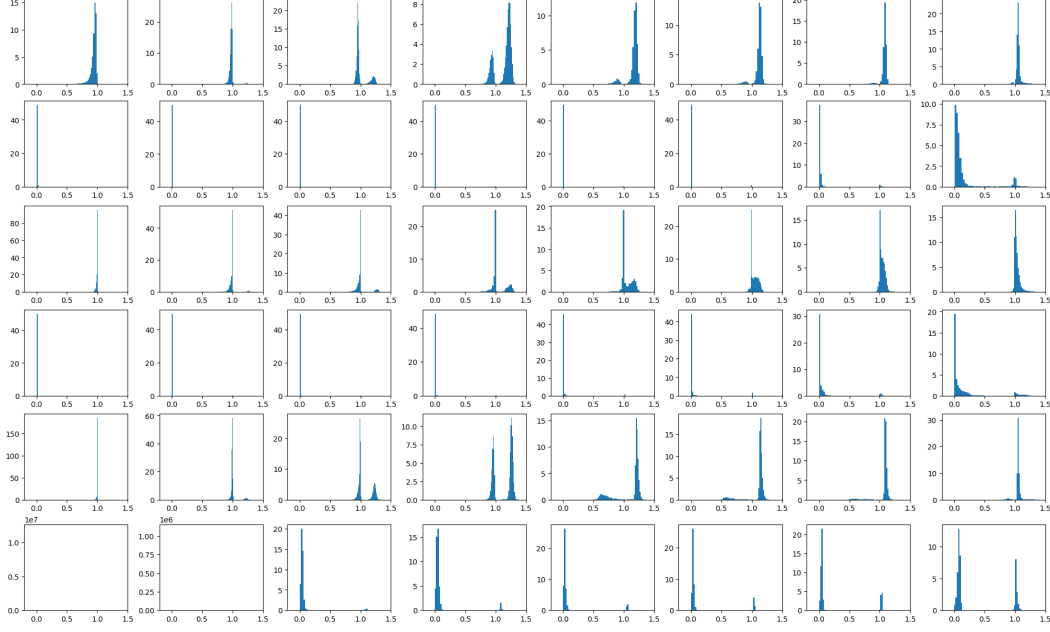


Figure 9: Row norm distributions for large toy models. There are 6 rows of panels. Rows 1 and 2 correspond to the power-law feature frequency. Rows 3 and 4 correspond to the exponential feature frequency. Rows 5 and 6 correspond to the exponential feature frequency. In the two rows that correspond to the same feature frequency, the upper row is at strong superposition and the lower one is at weak superposition. The 8 columns from left to right correspond to different model dimensions m from small to large.

D Figure details and supplementary results

Here, we show how to process the raw data obtained from toy models or LLMs to generate results seen in the main text. Some supplementary analysis is also conducted to support the main text arguments.

D.1 Figure 1

The toy models reported in Figure 1 are large toy models with data dimension $n = 10240$ explained in Appendix B.1. After obtaining the final losses, we directly plot them with respect to the model dimension m .

When we are fitting the loss in log-log as a line, we choose the linear part to fit. If the loss versus model dimension curve is obviously not a line, we fit the whole curve as a line and output the R^2 value as a measure of how non-linear it is. Specifically, we fit the last five points for the power-law decay feature case in the weak superposition regime (yellow data in Figure 1b). Other cases in the weak superposition regime are fitted to a line with all data. In the strong superposition regime, when feature frequency decreases as a power law or as a linear function, we fit the data as a line starting from the third point (yellow and green in Figure 1d). And for exponential decay feature frequencies, we fit all the data with a line. In the strong superposition regime, the measure model exponent α_m are close to 1: 1.01 ± 0.05 (exponential decay), 1.0 ± 0.1 (power-law decay), and 0.89 ± 0.05 (linear decay).

The LLM data are copied from Figure 8b, with slope -0.91 ± 0.04 being close to 1 as well. We will explain details about Figure 8 later.

We also output the weight matrix W for these large toy models (Figure 9). They follow the same pattern that in the weak superposition regime, row norms are bimodal and are either close to 0 or 1, making 0.5 a good separation point for measuring how many features are represented. And in the strong superposition regime, the row norms are distributed near 1, and 1 is a good separation point for the two peaks, i.e., the peak greater than 1 refers to strongly represented features which are more important, and the peak smaller than 1 corresponds to the weakly represented.

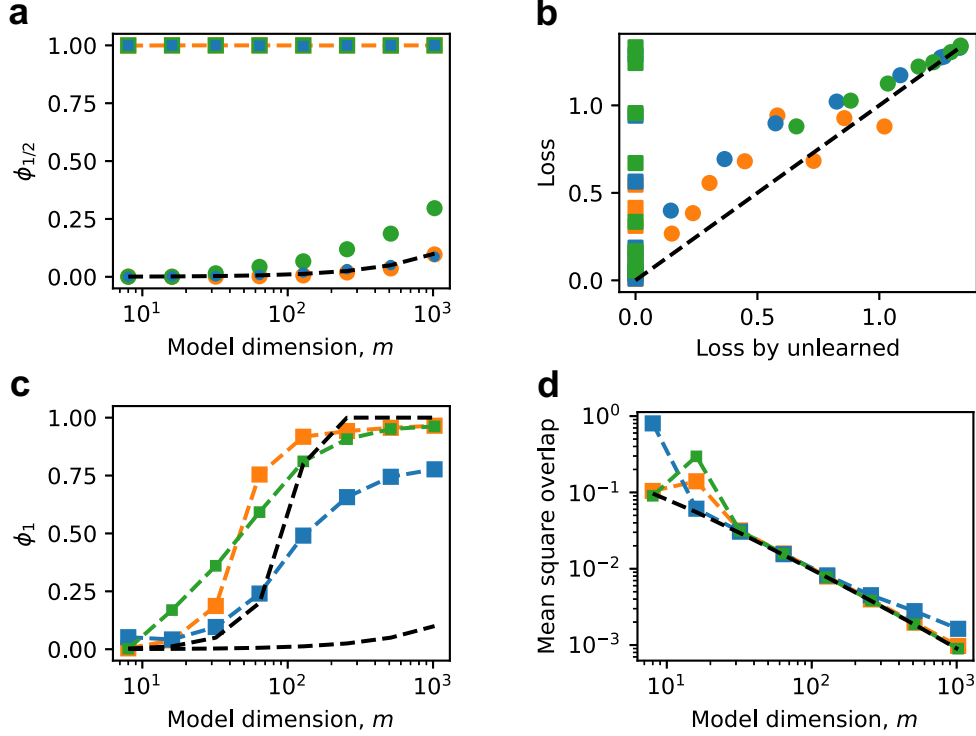


Figure 10: Large toy models agree with theoretical expectations. We use blue, yellow, and green for exponential, power-law, and linear feature frequencies, respectively. Dots correspond to the weak superposition regime, and squares to the strong superposition regime. (a) The fraction of represented features is 1 for strong superposition, and is close to m/n (black dashed line). (b) The expected number of activated but unlearned features well describes the loss at weak superposition. The dashed line is where the actual loss is the same as the predicted one. (c) In the strong superposition regime, the number of strongly represented features is much larger than m but bounded by some value around $m^2/2$. The slowly growing dashed line is m/n , and the fast growing dashed line is $\min\{1, m^2/2n\}$. (d) The mean squared overlap of strongly represented features is close to κ^2 , which is close to $1/m$, given that the number of strongly represented features is much larger than m .

We can analyze the large toy model in the same way as what has been done in Figure 5 and 6. The fraction of represented features $\phi_{1/2}$ is calculated, which is 1 in the strong superposition regime, while it is close to m/n in the weak superposition regime (Figure 10a).

With the measured $\phi_{1/2}$, we can estimate the loss due to unlearned features, $\langle v^2 \rangle \sum_{i=\phi_{1/2}n}^n p_i$. This loss by unlearned features agrees well with the actual loss in the weak superposition regime (Figure 10b).

In the strong superposition regime, the fraction of strongly represented features is calculated, agreeing with the expectation that the number of strongly represented features is much larger than m but bounded by some value around $m^2/2$ (Figure 10c).

At the end, we see that the mean square overlap of the strongly represented is close to the characterized value κ^2 (Figure 10d), which scales as $1/m$ since the number of the strongly represented is much larger than m .

D.2 Figure 2

Figure 2 introduced the toy model and the concept of superposition without real data. The W matrix we used to show superposition in Figure 2c is obtained by optimizing the square of off-diagonal terms of the normalized W , i.e., each row is normalized to have norm 1 first.

D.3 Figure 3

In Figure 3, we reported results from the trained small toy models with data dimension $n = 1000$, whose details are in Appendix B.2.

We showed results at data exponent $\alpha = 1$ in Figure 3a. The inner panels are obtained at $m = 100$. We showed that the more frequent features tend to have larger norms or to be better represented. And the norm distribution is very bimodal. We here show that it is true that the norm is around 1 or 0 for various α and model sizes m and degrees of superposition (Figures 11 and 12). The fraction of represented, $\phi_{1/2}$, can be calculated directly.

We next showed the $\phi_{1/2}$ as a function of α and γ at $m = 39$. Here, we provide the same plot but at different m (Figure 13). The pattern is robust, suggesting weight decay is a good tool to change the degree of superposition.

D.4 Figure 4

Figure 4a reports the R^2 values from the fitting introduced in Appendix D.1, where the raw data comes from training large toy models (Appendix B.1). When we are fitting the loss in log-log as a line, we choose the linear part to fit. If the loss versus model dimension curve is obviously not a line, we fit the whole curve as a line and output the R^2 value as a measure of how non-linear it is. Specifically, we fit the last five points for the power-law decay feature case in the weak superposition regime (yellow data in Figure 1b). Other cases in the weak superposition regime are fitted to a line with all data. In the strong superposition regime, when feature frequency decreases as a power law or as a linear function, we fit the data as a line starting from the third point (yellow and green in Figure 1d).

And from the raw data of small toy models (Appendix B.2), we can fit the model exponent α_m directly and plot it as a function of γ and α as in Figure 4b.

The fitting in Figure 4b does not care whether the loss versus model dimension curve is a power law or not. We provide the R squared values for the fitting here (Figure 14). The closer R squared values are to 1, the better the data can be thought to be a power law (a line in log-log plot). In the strong superposition regime, the R squared values suggest the data are close to be power-law. While in the weak superposition regime, data may not be power-law, especially when γ is too large. When α is smaller than 1, it is not a power law in theory. The R squared values are not too small since the loss decay is very slow, and a line in log-log plot is still a good approximation. When $\alpha > 1$ and $\gamma \approx 1$, the number of represented features can be smaller than m or even non-increasing. Too large weight decay still makes the configuration of the representation be in no superposition. However, it destroys some feature representations that can exist, making the configuration far from the ideal case where m features are represented. So, we may not see power laws when weight decay is too strong.

D.5 Figure 5

In Figure 5a, we plot the raw losses from small toy model experiments (Appendix B.2). The “loss by unlearned” is approximated by the integral, which is

$$L = \begin{cases} \frac{\phi_{1/2}^{1-\alpha} - 1}{1 - n^{1-\alpha}} n^{1-\alpha}, & \alpha \neq 1, \\ -\frac{\ln \phi_{1/2}}{\ln n}, & \alpha = 1. \end{cases}$$

To quantify how much the learned weight matrix deviates from the ideal no superposition structure, we construct a reference matrix and compute a norm difference. Specifically, we first create an n -by- n zero matrix called base, and then insert an identity matrix of size m in its top-left corner. This padded identity matrix serves as a reference for the perfect recovery of the first m features. We then compute the matrix product WW^T from the learned weights and compare it to this reference using the matrix 2-norm. The resulting value reflects the ambiguity or interference in the learned representations. We store this norm in the ambiguity tensor at the location indexed by the current task and model width. Given a weight decay and data exponent, we have 6 ambiguity values since we have 6 m values. We calculate maximum ambiguity among these 6 models, and choose the 9 cases

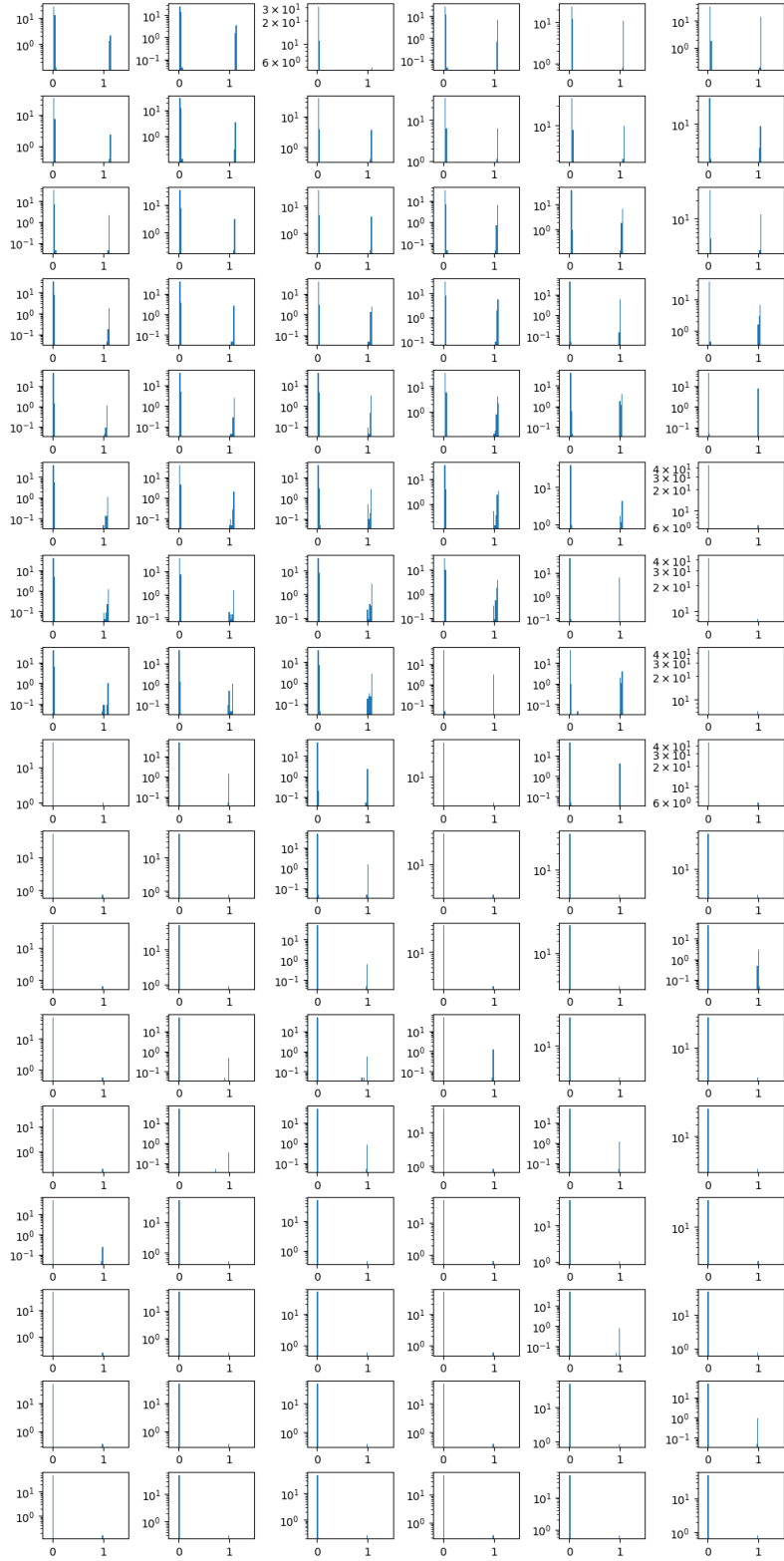


Figure 11: Row norm distribution at weak superposition ($\gamma = 0.55$) shows that the rows either are close to zero or have norm close to 1, making 0.5 a good separation. The 17 rows of panels from top to down correspond to 17 α from 0 to 2. And the 6 columns from left to right correspond to 6 m from 10 to 100.

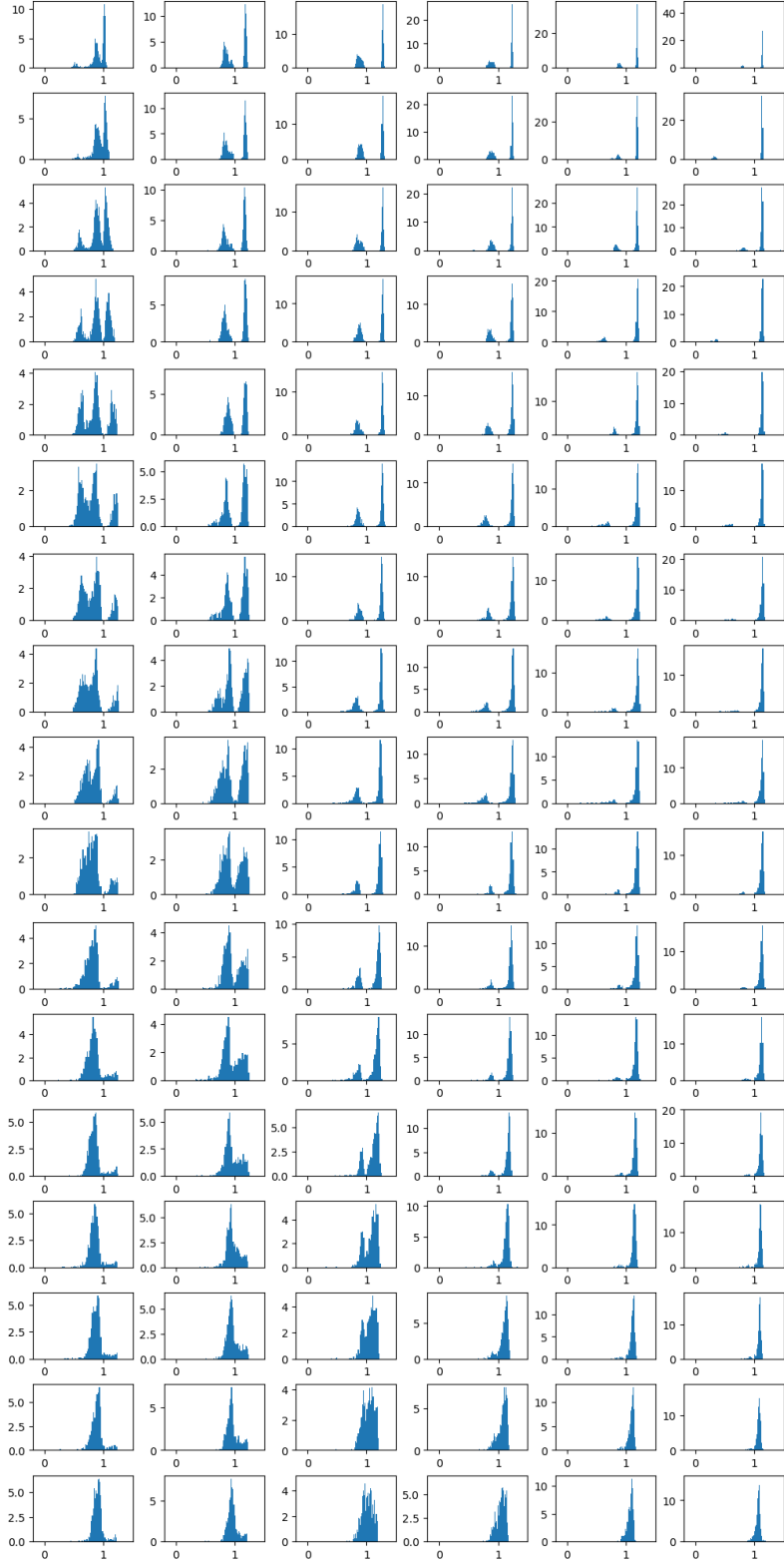


Figure 12: Row norm distribution at strong superposition ($\gamma = -0.55$) shows that the rows have norm close to 1. And density at 1 is very low, making 1 a good separation point for two groups of row norms. The 17 rows of panels from top to down correspond to 17 α from 0 to 2. And the 6 columns from left to right correspond to 6 m from 10 to 100.

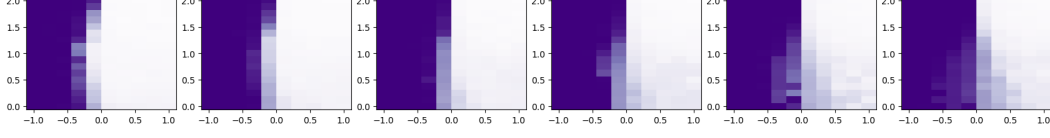


Figure 13: Fraction of represented features as a function of γ (x-axis) and α (y-axis). The 6 columns from left to right correspond to 6 m from 10 to 100. The colorbar is the same as the main text (Figure 3b) where purple means 1.

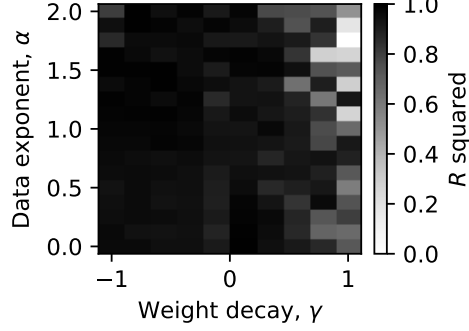


Figure 14: R squared values for fitting loss as a power law with model dimension. Data are from the small toy models with data dimension $n = 1000$.

with the smallest maximum ambiguity to plot in Figure 5b. One can see that when weight decay is near 0.5, the models are closest to the ideal no superposition case where the first m features are represented perfectly. Smaller weight decay may not be sufficient to eliminate superposition, and larger weight decay can suppress features that, in principle, can be represented perfectly.

D.6 Figure 6

Figure 6 studies the results from small toy models (Appendix B.2) focusing on the strong superposition regime. For the strongly represented fraction, ϕ_1 , we can directly compute based on the definition and the obtained weight matrices (Figure 6, a, b, and c). Here, we provide more data to show that 1 is a natural separation point in norm to determine which are strongly represented and which are weakly represented (Figure 12).

Once select the rows with norm greater than 1, we can calculate their mean square overlap based on normalized rows $W_i / \|W_i\|_2$ (Figure 6d). We argue that after training, the vectors will be more

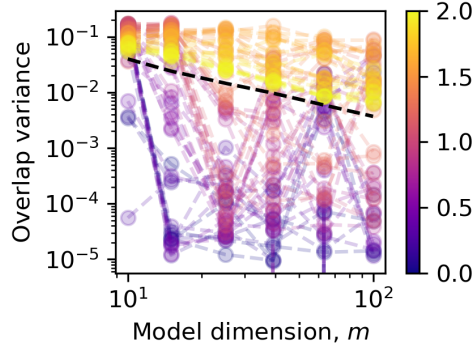


Figure 15: The overlap variance of the strongly represented is smaller than that of random vectors. We calculate the absolute overlaps of the strongly represented and can calculate the variance. We plot variance versus model dimension for different α (denoted by color) and weight decays (all the negative weight decays). The dashed line is obtained by sampling $m^2/2$ Gaussian random vectors, normalizing them to have norm 1, calculating the absolute overlaps, and calculating the variance of the overlaps.

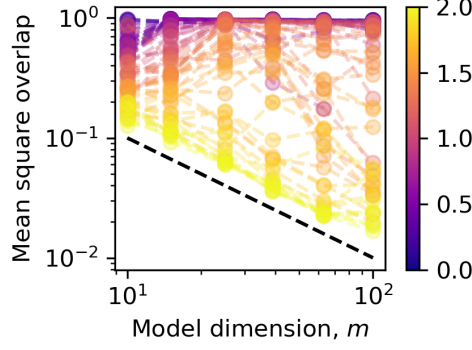


Figure 16: The weakly represented features can have large overlaps as expected, where we plotted all the data when $\gamma < 0$, and the dashed line is $1/m$. The colorbar corresponds to α in the same way as the main text.

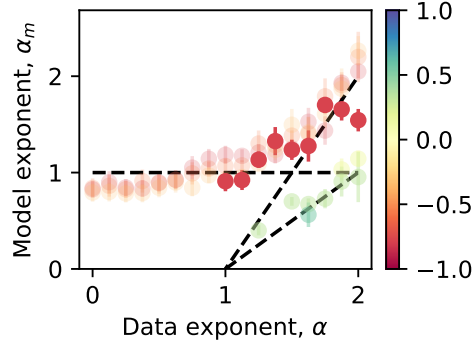


Figure 17: Small toy models with m from 50 to 150 (such that $m^2/2 > n$) in the strong superposition regime yield similar α_m around 1 when α is small and a slightly smaller α_m (smaller than $2(\alpha - 1)$) when α is large. We copied Figure 6e and made the points transparent for comparison. The non-transparent points are from small toy models with m from 50 to 150. $\alpha_m = 1$ is the horizontal line, $\alpha_m = 2(\alpha - 1)$ is the fast increasing dashed line, and $\alpha_m = \alpha - 1$ is the slowly increasing dashed line.

similar to ETFs than to random initialization. This is studied via the variance of overlaps. ETFs, in theory, have zero variance. We find that the majority of the overlap variances are much smaller than the random initialization, especially when features have similar frequencies, which agrees with the expectation (Figure 15). The cases where the actual variance is greater than that of the random vectors have large α , roughly correspond to the cases where α_m deviates from 1 — ETF-like configuration no longer dominates. This is intuitive that when α is too large, the heterogeneity of overlaps will become large — it is better to let more frequent features occupy larger angle space. We argue that the large variance at large α does not mean the configuration tends to be random, but tends to be something more closely related to the frequency distribution of the features.

As a related side note, our explanations based on the strongly and weakly represented features capture the basic trend that when α is getting large, the more important features will have larger angle space and the loss decay will be more related to the data exponent. However, this theory is oversimplified, where the strongly represented all have small overlaps and the weakly represented all have large overlaps. The real situation may be more like the angle occupied by one feature decreases continuously as the frequency decreases. As suggested by Figure 15, overlap variance within the strongly represented is greater when α is larger. To be more precise about the overlap distribution as well as the exponent α_m when α is large, we cannot use simple theoretical expectations like ETFs but have to solve the toy model.

We also provide evidence that overlaps of the weakly represented (Figure 16). We see that all the mean square overlaps are larger than $1/m$ instead of being on the line $1/m$. And some values are around 1 and not decreasing much with increasing model size m , supporting our picture that the less frequent features are squeezed into a region and have large overlaps.

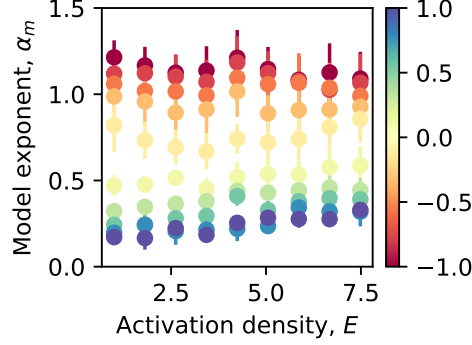


Figure 18: Model exponent is robust to activation density at different levels of superposition. The colorbar encodes weight decay as in the main text.

After fitting α_m of the trained small toy models (Appendix B.2), we plotted the α_m corresponding to the second to the fourth smallest weight decays in Figure 6e. We also copied from Figure 5b and plotted the ideal weak superposition case in Figure 6e. One question we had is that if $m^2/2$ is always greater than n , in our analysis, all vectors can be strongly represented, what should α_m be? We trained the small toy models again as in Appendix B.2 but with m from 50 to 150. We found that in the strong superposition regime, the α_m is still around 1 when α is smaller than 1.5, and α_m still increases a little while smaller than $2(\alpha - 1)$ when α is larger than 1.5 (Figure 17). When $m^2/2n > 1$ is always true, the vectors can be put into a configuration where all overlaps are small and scale as $\sqrt{1/m}$, such that α_m should be closer to 1. However, as mentioned before, our picture that the strongly represented have nearly uniform absolute overlaps is oversimplified. In the real situation, more frequent features have smaller or even faster decaying overlaps. Therefore, when α is too large, a weighted sum of squared overlaps, weighting the more frequent features more, can decrease faster than the average decaying speed $1/m$. Again, we need to solve the toy model faithfully to uncover the rigorous relation between α_m and α and argue the robustness of α_m from theory.

D.7 Figure 7

Once obtaining the small toy models scanning activation density and keeping $\alpha = 1$, we can plot the loss as a function of activation density E in Figure 7. The linear fitting is also straightforward. We chose one γ to show in the main text. Here, we present the whole picture that, with any weight decay tested, the model exponent is robust to the change of activation density (Figure 18).

D.8 Figure 8

After obtaining the overlaps as described in Appendix C.1, we directly plot the raw data in Figure 8a. The data are quite noisy, and we did not fit the data with a line.

We argued that the LLMs are in the strong superposition regime since all tokens are represented. Figure 19 shows a typical row norm distribution of LLM (opt, 125M parameters [39]). We showed the mean, minimum, and maximum row norms of all the LLMs studied in Figure 20. From the non-zero minimum norms and the fact $n \gg m$, we confirm LLMs are in strong superposition. As mentioned in the analysis in Appendix A.2, we argue that the row norm of LLMs should not depend on m but controlled more by the intrinsic data property of language, which is also verified to be valid (Figure 20).

We obtain the evaluation loss of each model on each dataset as described in Appendix C.2. We fit our loss values by the formula,

$$L = C_m/m^{\alpha_m} + L_{\setminus m},$$

where C_m/m^{α_m} is universal and $L_{\setminus m}$ is a constant depending on the dataset and model class. There are in total 16 different $L_{\setminus m}$ since we have 4 different model classes and 4 datasets. In our fitting model, there are in total 18 parameters. We use Adam to minimize the mean square error between the predicted loss by the above function and the real loss. All losses obtained are used in optimization. The code is in `nonlinearfit-3.ipynb`.

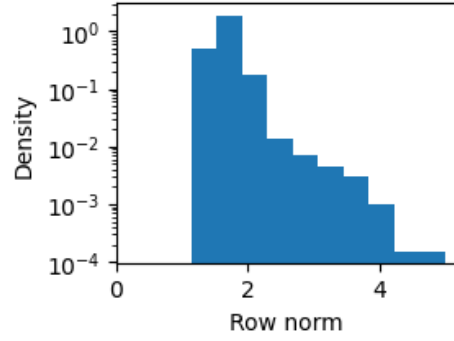


Figure 19: Row norm distribution of the language model head of OPT-125M.

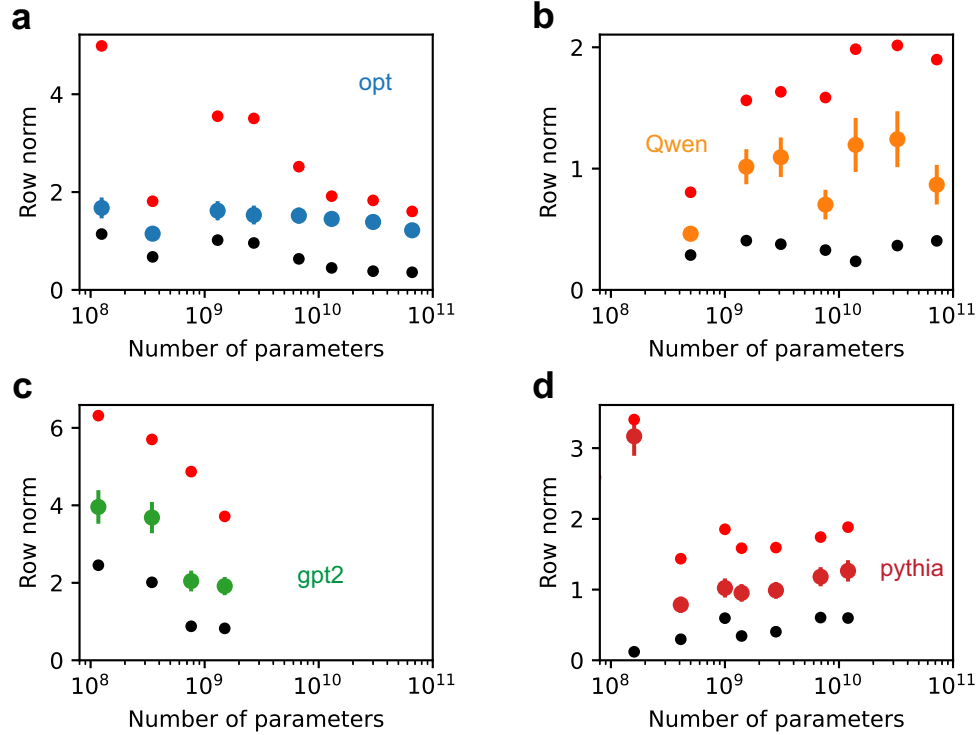


Figure 20: LLMs are in strong superposition based on the non-zero norms of the representation vectors. (a) OPT models. (b) Qwen2.5 models. (c) GPT-2 models. (d) Pythia models. The dots with error bars are mean values, and the error bar is the standard deviation. Red small points are the maximum values, and dark small points are the minimum values. The mean norm value as a characteristic row norm does not depend on model size much as expected (Appendix A.2).

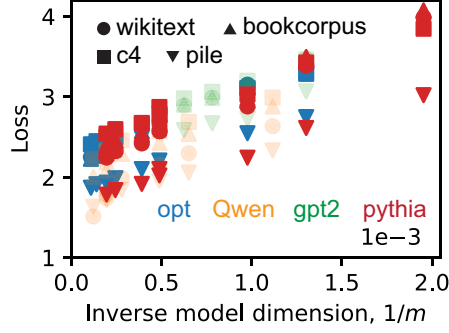


Figure 21: Raw evaluation losses as a function of inverse model dimension.

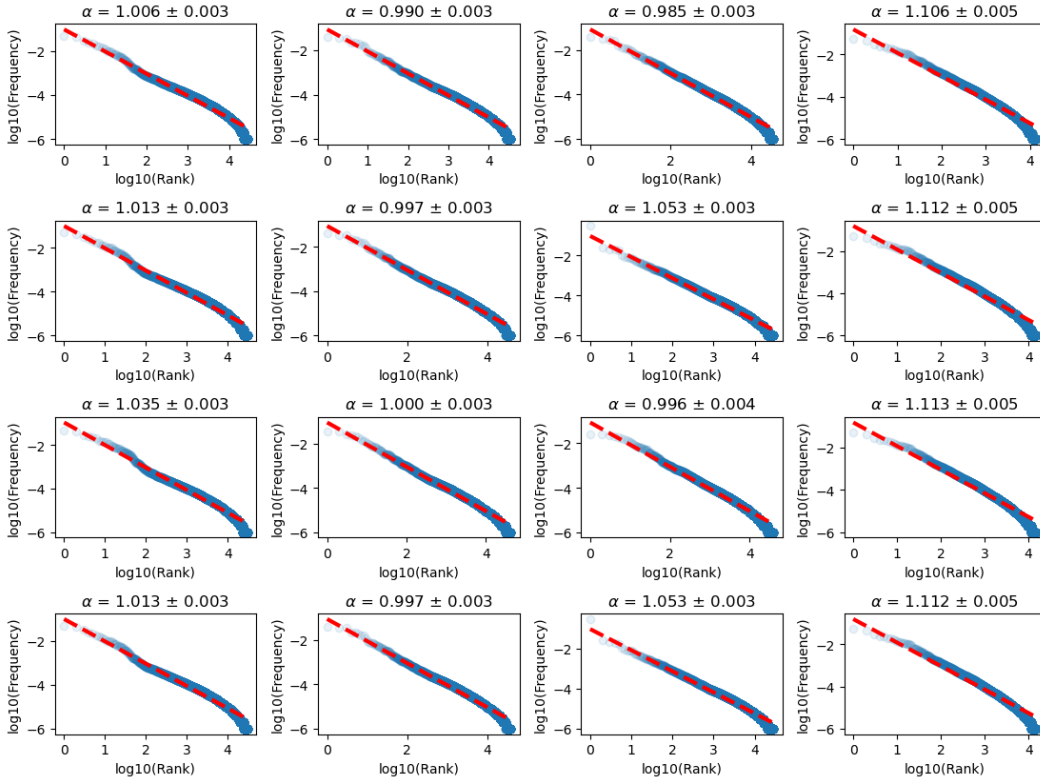


Figure 22: Taking tokens as atomic features, their frequencies indeed follow a power law, and the measured data exponent α is close to 1. The four rows from top to bottom correspond to four tokenizers, Pythia, OPT, Qwen2.5, and GPT-2, respectively. And the four columns from left to right correspond to four datasets analyzed, wikitext, C4, the Pile, and Bookcorpus, respectively.

We provide the raw data, losses, as a function of $1/m$ (Figure 21). The losses look like a line with $1/m$ in one model class and with the same dataset. And the slope of the line seems to be universal. These two points support us in proposing the formula above, where C_m/m^{α_m} is universal. The intersections are different depending on the dataset and model class, corresponding to different $L_{\setminus m}$.

We obtained the token frequencies as described in Appendix C.3. Given the raw data, we sort the token frequency and obtain the frequency-rank plot. We sample 1000 (this number does not matter once it is large, 10000 gives the same result) points uniformly in the $\log_{10}(\text{Rank})$, and fit the frequency-rank as a power law, or a line in log-log plot. Results show that the data exponent fitted α is close to 1 regardless of the dataset or the tokenizer (Figure 22).

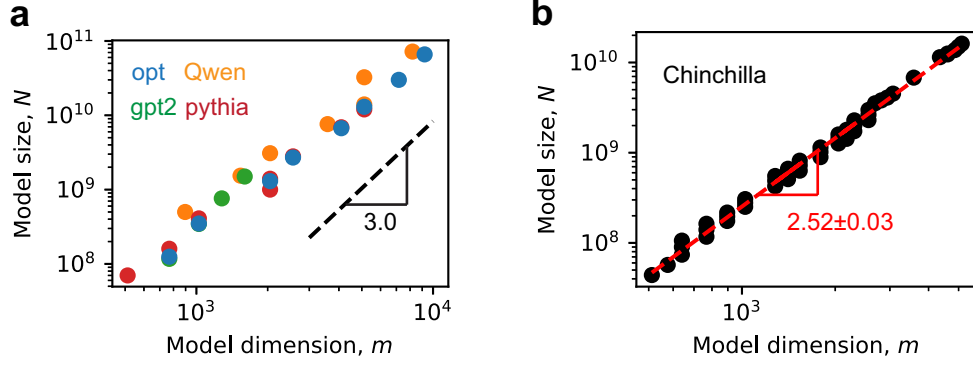


Figure 23: The model size is approximately a power law with model dimension. (a) The four model classes we analyzed [39–42]. (b) The Chinchilla models [3].

We study the relationship between model dimension m and model size N (number of parameters). For the four open-sourced models we analyzed [39–42], we can see that $N \sim m^3$, especially when m is large. If we fit the N - m relation by a power law while assuming a universal exponent but different coefficients depending on the model class, we obtain an exponent of 2.51 (Figure 23a). For the Chinchilla model reported in [3], we find N is also close to a power law with the model dimension, and the fitted exponent is 2.52 ± 0.03 (Figure 23b).