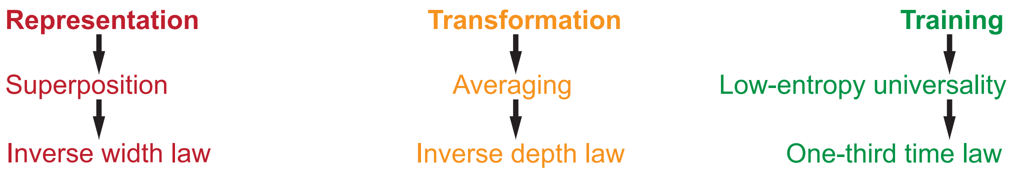
Neural Scaling Laws Trilogy: Representation, Transformation, and Training

Yizhou Liu
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
liuyz@mit.edu

Abstract

The success of large language models (LLMs) partly relies on the neural scaling laws—performance of LLMs continues to improve as their model size and training dataset size scale up. In this blog, we combine our three papers on the origins of neural scaling laws to provide a comprehensive picture: (i) LLMs need to represent many more features than their widths, which is called superposition, and the geometric interference of representations leads to a part of loss $\sim m^{-1}$, where m is the width; (ii) Transformer layers mainly reduce error by ensemble averaging, yielding a part of loss $\sim \ell^{-1}$, where ℓ is the depth; (iii) softmax and cross-entropy lead to the power-law scaling of loss $\sim \tau^{-1/3}$ in matching low-entropy next-token distributions, closely related to universality in statistical physics, where τ is the time of training dynamics and $\tau \propto D$ (dataset size) for online learning. (i) and (ii) together lead to the loss scaling $N^{-1/3}$ under the optimal shape, where $N \approx 12m^2\ell$ is the number of parameters asymptotically. With (iii), we further have $N \propto D$ and the loss $\sim C^{-1/6}$ (compute $C \approx 6ND$) at the compute-optimal frontier. Our results agree well with the measured scaling exponents from actual LLMs. Based on time scaling (iii), we argue that LLM training is not limited by the amount of data, but the quality of data and the number of steps. Combining all, we emphasize that the power laws originate from high-level architecture and data properties, e.g., softmax and a low entropy of target distributions, producing exponents robust to other details like data or task structures. Moving forward, we predict the improvement of scaling should come from architectural innovations and offer our most optimistic estimation: Loss $\sim C^{-1/2}$ for compute-optimal training (cubic speedup). Finally, we discuss why the loss value is important for LLMs, what it cannot capture, and when we should stop pretraining. We hope this trilogy can help with the ongoing development of efficient and interpretable LLMs.[1]



[1] See website version at https://liuyz0.github.io/blog/2026/NSLT

1 Why are LLMs large

Performance depends strongly on scale.—As indicated by the name, one of the most important features of large language models (LLMs) is being large. In early 2020, roughly 3 years before the release of ChatGPT, OpenAI reported neural scaling laws: Transformer model performance measured by cross-entropy loss scales as a power law with model size, dataset size, and the amount of compute [1]. Larger models and longer training lead to better performance.

Scaling laws are predictive.—Following these findings, in mid 2020, GPT-3 had reached 175 billion parameters [2], while the largest model size in [1] was 1.5 billion. In GPT-4 technical report [3], power-law scaling fitted from smaller models (with compute at most 10^{-4} of GPT-4's) could accurately predict the performance of GPT-4. Following the scaling laws to scale up, Transformer models, without qualitative architectural changes, invented for translation at first [4], can speak fluently as humans, rendering the Turing test largely beside the point.

Artificial intelligence systems centered around LLMs, arguably the greatest breakthrough of our era, are therefore built on the effectiveness of neural scaling laws: LLMs became larger and larger as performance continued to improve in a predictable way, leading to surprising emergent behaviors.

2 We cannot scale up pyramids to reach the moon

Powerful but empirical.—Although the neural scaling laws can fit the experimental data well and can be used to predict the performance of larger models, they are intrinsically empirical, naturally leading to concerns and hopes. For instance, we may wonder whether the scaling laws are only valid within a certain range of model sizes or dataset sizes, beyond which they may not hold. We are also curious about whether better power-law exponents or even exponential scaling can be achieved by changing the architecture or training methods. To answer these questions, we need to understand the origins of neural scaling laws.

Motivations to understand.—At first, trying and doing—rather than thinking deeply—is often how completely new worlds are opened up. The Great Pyramid of Giza, built around 2600 BC, was the tallest human-made structure for over 3800 years, which was a magnificent miracle. We would believe that a little understanding of geometry like the scaling between the volume and the height could help the ancient Egyptians to build larger and larger pyramids after the first one. A deeper understanding of mechanics and material can lead to totally different architectures, which are much higher and much more stable. In our case, we want to understand the neural scaling laws to answer the questions above, enabling more efficient models.

Concrete and quantitative phenomena can be the window to a greater world. After building a variety of structures with different materials and shapes, at some point, humans could realize that the universal principle all tall structures fight against is gravity. Then, they could possibly build something qualitatively different, like rockets, which can reach the moon. Scaling up pyramids cannot reach the moon, yet understanding the fundamental principles behind them can. We hope that understanding the neural scaling laws can also be a window to fundamental principles of intelligence, which not only benefits LLMs, but may also lead to qualitatively different things that we cannot even imagine now.

And of course, there is always curiosity—a motivation independent of practical implications. With all these wishes, we therefore decided to study neural scaling laws.

Quantitative puzzles.—We sought to focus on the Chinchilla scaling laws [5], which provide a concrete description of how cross-entropy loss depends on model size N (number of parameters) and dataset size D (number of training tokens):

$$L = \frac{c_N}{N^{\alpha_N}} + \frac{c_D}{D^{\alpha_D}} + L_0. \tag{1}$$

The cross-entropy can be written as the Kullback-Leibler divergence between the predicted and target distributions plus the entropy of the target distribution. So, there is an irreducible loss floor L_0 containing at least the mean entropy of next-token distributions. Early scaling-law fits implicitly assumed $L_0 = 0$ [1], leading to biased exponent estimates. The Chinchilla analysis accounted for this irreducible term and found more reliable exponents, $\alpha_N \approx 0.34$ and $\alpha_D \approx 0.28$. Combining these scalings with the compute budget $C \approx 6ND$ [1, 5] implies that the compute-optimal frontier, where loss is minimized with given C via balancing N and D, satisfies $N \propto D \propto C^{1/2}$ roughly. Subsequent works reported similarly that $N \propto D$ is compute-optimal [6], supporting the Chinchilla scaling laws. **We therefore focused on the following concrete questions**: Why does the loss obey power laws in N and D, and why do the exponents take the values $\alpha_N \approx 0.34$ and $\alpha_D \approx 0.28$?

3 Representation, transformation, and training

LLMs need to represent many concepts or features (i.e., representation) and learn the correlation or interaction between them to predict what probably comes next (i.e., transformation). Even after training to optimality, both representation and transformation can still be limited by model size. For simplicity, we call the loss from imperfect representation due to limited model size as the **representation loss**, and the loss from imperfect transformation due to limited model size as the **transformation loss**. With finite training, which is imperfect, the loss further increases, which is related to the dataset size scaling. We therefore can imagine three contributors to the loss and study them one by one.

Representation.—We first isolated and explored the representation loss [7]. We analyzed a toy model with only embedding and unembedding [8] without transformation. When models try to represent more features that are necessary than the width m (or embedding dimension, or model dimension) they have, which is a phenomenon called **superposition**, the geometric interference of representations leads to errors and a part of loss $\sim m^{-1}$. Although we cannot estimate the number of necessary features for LLMs, our study of token embeddings showed that the mean overlap keeps decreasing as predicted without a signature of saturation, suggesting that the error due to interference is significant compared to the signal due to actual correlation, or the number of necessary features is much larger than the current widths. We therefore predicted a part $\sim m^{-1}$ in loss.

Transformation.—We next focused on the transformation loss [9]. We analyzed hidden states evolution across layers and found that LLMs do not use their layers in a compositional way, but update the hidden states incrementally. A further analysis of incremental updates via toy residual networks showed that loss can scale as ℓ^{-3} if the model tries to discretize a smooth dynamical process with ℓ layers or ℓ^{-1} if different layers are similar and reduce error by ensemble averaging. In either way, we predicted a power-law part with the number of layers or depth ℓ.

Training.—Finally, we studied the training dynamics [10]. We found that the minimal ingredients that are relevant to LLMs and sufficient to produce similar power-law training behaviors are the softmax function and the cross-entropy loss—the key components of the language model head. We showed that the non-linearities intrinsically lead to power-law loss and gradients when learning low-entropy distributions like the next-token distributions, mirroring the concept of **universality** in statistical physics, which ultimately lead to the power-law scaling of loss $\sim \tau^{-1/3}$ where τ is the time of optimization dynamics and is related to the number of steps via an integral over the learning rate schedule. For online learning, we further have $\tau \propto D$ approximately, leading to a power-law dataset size scaling of loss as $D^{-1/3}$.

A semi-empirical formula.—Conceptually, we gathered: (i) With no error due to transformation or training, the loss due to imperfect representation is related to width as $\sim m^{-1}$; (ii) With perfect representation and training, the loss due to imperfect transformation is mainly a power law with depth ℓ; (iii) With sufficiently large model size, the loss converges as $\tau^{-1/3}$. To combine all, we proposed a formula summing the three power laws in the same spirit as the Chinchilla scaling laws:

$$L = \frac{c_m}{m^{\alpha_m}} + \frac{c_\ell}{\ell^{\alpha_\ell}} + \frac{c_\tau}{\tau^{\alpha_\tau}} + L_0. \tag{2}$$

In reality, imperfect representation may also affect the transformation and training, leading to cross terms. Yet, the cross terms are of higher order, and asymptotically with large m, ℓ, and τ, the three power laws should dominate, and the above formula can be a good approximation. We therefore reached this semi-empirical formula, where we had theoretical insights for each power law and proposed the summation based on heuristics and previous empirical laws.

Fitting the data.—We next tested the effectiveness of the above formula and whether actual exponents agree with our predictions. We used the data reconstructed from Chinchilla scaling laws [11]. Since for each data point, we only had information about D but not τ, yet we had $\tau \propto D$ approximately, we used the following for fitting,

$$L = \frac{c_m}{m^{\alpha_m}} + \frac{c_\ell}{\ell^{\alpha_\ell}} + \frac{c_D}{D^{\alpha_D}} + L_0, \tag{3}$$

with $\alpha_D \approx \alpha_\tau$. Since the formula was expected to be accurate only asymptotically, we fitted the ~ 200 data points with the lowest loss values, reducing standard errors of fitting [9]. We found that the above formula can fit the data well (Figure 1, a-c) with fitted exponents $\alpha_m = 0.98 \pm 0.08$, $\alpha_\ell = 1.2 \pm 0.3$, and $\alpha_D = 0.30 \pm 0.01$. The relative error between the empirical loss and the fitted model prediction is 0.4% on average, supporting the effectiveness of our proposed formula.

The fitted exponents are consistent with our predictions. We predicted $\alpha_m = 1$ due to superposition and geometric interference. We expected the layers to approximate smooth dynamics or reduce error by ensemble averaging through hidden state evolution. The fitted exponent $\alpha_\ell \approx 1$ is consistent with our prediction and further narrows down the possibility to the ensemble averaging case. Finally, we predicted $\alpha_D \approx 1/3$, which is close to our fitted value 0.30 and also the Chinchilla value $\alpha_D = 0.28$. Since $\tau \propto D$ is not exact, we further evaluated checkpoints of Pythia models, where τ can be obtained. Fitting the loss with τ directly (Figure 1d), we found the part of loss curves from different models due to insufficient training collapse and $\alpha_\tau \approx 1/3$, supporting the proposed loss decomposition formula and our prediction of the exponent. We conclude that the three leading terms with clear theoretical mechanisms can capture the loss scaling of actual LLMs well.

Neural scaling laws: superposition, averaging, and universality.—To summarize, our neural scaling laws fundamentally contain three power laws with model width m, depth ℓ, and dynamic time τ, respectively. Limited by model size, representation suffers from **superposition**, leading to a part of loss $\sim m^{-1}$. Limited by model size, transformation cannot be perfect and models reduce the error by **ensemble averaging**, yielding a part of loss mainly $\sim \ell^{-1}$. Limited by finite training and power-law dynamics due to the **universality** in softmax and cross-entropy, the loss has an extra part scaling as $\sim \tau^{-1/3}$. Supported by empirical data, the loss scaling can be well captured by a combination of the three power laws:

$$L = \frac{c_m}{m} + \frac{c_\ell}{\ell} + \frac{c_\tau}{\tau^{1/3}} + L_0. \tag{4}$$

Explaining the Chinchilla scaling laws.—Having established our understanding of neural scaling, we now try to explain the Chinchilla scaling laws. Given the total number of parameters $N \approx 12m^2\ell$ for large m and ℓ, the optimal relationship between m and ℓ given N will be $m \propto \ell$ with a fixed coefficient, near which the power laws with width and depth can be summed into one term $\sim N^{-1/3}$. The Chinchilla scaling has $\alpha_N = 0.34$ close to this prediction. We therefore argue that the observed scaling with N is a consequence of our representation and transformation parts of loss and the fact that models are trained near the optimal shape.

In the standard pre-training regime with fixed batch size and a single pass over data, the dynamic time τ is proportional to the dataset size D fixing the type of learning rate schedule and the maximum learning rate. In reality, different models may use slightly different maximum learning rates, making $D \propto \tau$ only approximately correct. Therefore, our theory predicts a term that scales as $D^{-1/3}$, which is close to the fitted $\alpha_D = 0.28$ in Chinchilla scaling laws. We therefore argue that the dataset size scaling is a consequence of the training dynamics and the online training process.

Combining the arguments above, we have

$$L = \frac{c_N}{N^{1/3}} + \frac{c_D}{D^{1/3}} + L_0, \tag{5}$$

near the optimal shape and with online training. We can therefore explain why the Chinchilla scaling laws have the power-law form and why the exponents are close to $1/3$.

Fixing the compute budget $C \approx 6ND$, the optimal-compute frontier minimizes loss by balancing N and D, which satisfies $N \propto D$ with a fixed coefficient, agreeing with the Chinchilla paper [5]. At the frontier, the loss is then

$$L = \frac{c_C}{C^{1/6}} + L_0. \tag{6}$$

We can therefore explain the empirical condition $N \propto D$ for the compute-optimal frontier and the loss scaling at the frontier.

We conclude that we have identified the three most fundamental contributors to the neural scaling: representation superposition, ensemble averaging in transformation, and universality of training

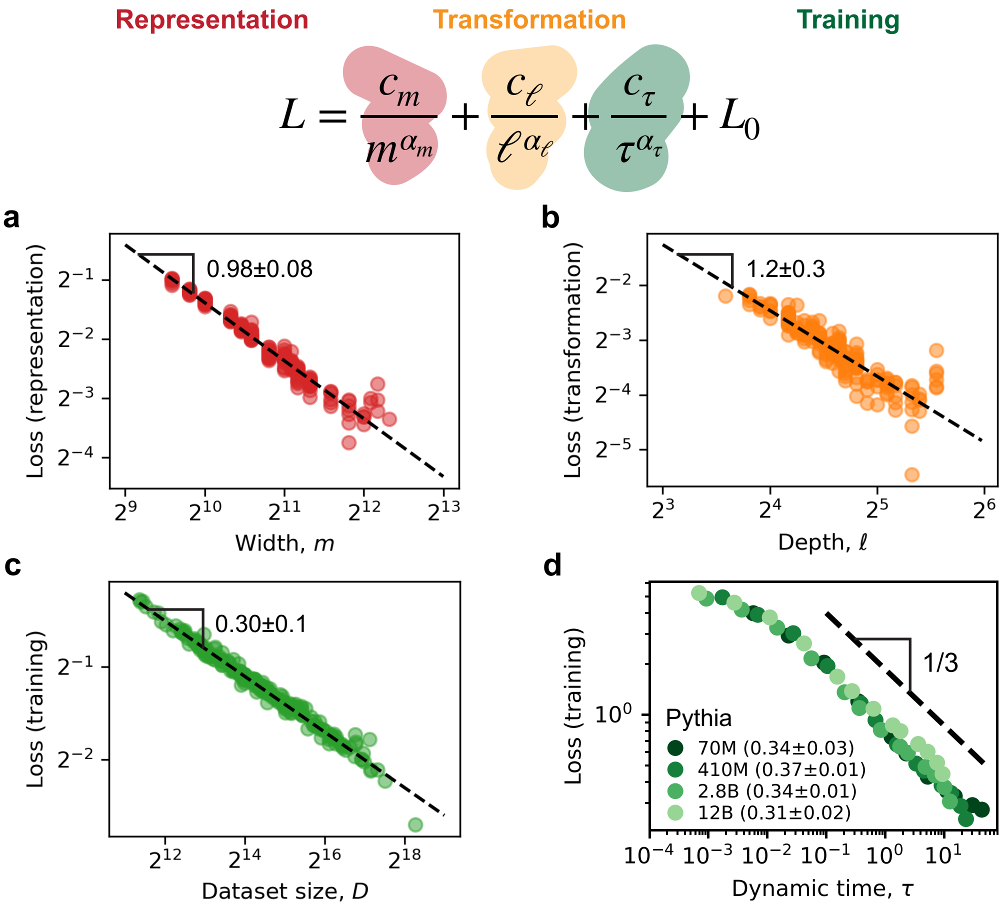


Figure 1: The proposed neural scaling formula, summing the three principal power laws, can well describe the empirical data. Details of Chinchilla data fitting are in [9]. (a-c) Loss from Chinchilla models [5, 11] as power laws with width m, depth ℓ, and dataset size D. (a) After fitting, we subtract the fitted depth and time power laws and the irreducible loss from the empirical loss, and plot the remaining part, which is expected to be the representation part, with m. (b-c) Similarly, we can plot the transformation part and the training part with ℓ and D, respectively. Each part does look like a power law, and the fitted exponents agree with our theory expectations. (d) We further fitted the training part with the dynamic time τ directly [10], which is accessible in Pythia models [12]. The loss curves from different models collapse, supporting the proposed formula. And the fitted exponents shown in the parentheses are all close to $1/3$, supporting our theory.

dynamics. Although future works are needed to study the interactions between these mechanisms, a summation of the power laws from the three mechanisms, which should be leading terms, can already describe the data well and explain previous empirical findings.

4 Neural scaling laws: past and present

After answering our primary questions about the origins of neural scaling laws, we next compare our findings with previous understanding, better locating our results in the space of theories.

Shape does not matter, does it?.—Empirical scaling laws typically report the scaling with the total number of parameters N, since it was observed that the shape or ratio between width m and depth ℓ has minor impact on the loss [1]. However, our theory suggests that the loss is a function of m and ℓ separately, instead of just N. We argue that the shape does matter (e.g., fixing N and making $m = 1$ certainly affects the loss), yet near the optimal shape, the loss is robust to the shape. With our fitted coefficients, we found that asymptotically, the optimal shape is $m/\ell \approx 70$, while m/ℓ from around 30 to 180 lead to a loss increase of less than 5% (compared to the reducible part of the loss at compute-optimal frontier with the same N). Our theory actually agrees with previous empirical findings, yet further clarifies the role of shape and the reason why it has minor impact on loss.

More training steps, rather than more data.—The framing that loss scales with dataset size as a power law makes people think that more new data are needed for better performance, raising the concern that we may run out of data in the world. Our theory suggests that the true gradients vanish as a power law, leading to power-law training dynamics, and finally a power law with dataset size due to online training. We therefore conclude that as long as we can estimate the gradients at each step well, which requires high data quality, high diversity, and a large batch size, the loss can continue to improve by scaling up the number of steps, not necessarily the amount of new data. Beyond the pretrain loss, the performance of LLMs on downstream tasks essentially also depends on the quality and diversity of data, not the amount of data. We argue that the quality and diversity of pretrain dataset are more important, and once the dataset size is sufficiently large to have such quality and diversity, we merely need to scale up training steps.

Architecture or data.—The majority of previous theories attribute neural scaling laws to power laws in the data. Heuristically, there are various features and skills in the data to learn [13]. If we assume that more important or frequent features are learned first, a power-law distribution of feature importance or frequencies can lead to power-law training. And if we assume model size limits the number of features that can be learned, the power law in data will also lead to the power-law loss with model size. Such intuitions are correct for linear or effective linear models trained with the mean squared error (MSE) [14]. Features are mathematically eigenmodes of some data covariance matrix. Importance is reflected in the eigenvalues, which control the exponential convergence rate of the corresponding eigenmodes. A power-law distribution of eigenvalues can then lead to power-law training. And linear models have no superposition [7], learning only the first N eigenmodes with N parameters and yielding power-law loss with model size.

However, for non-linear models with softmax, in learning peaked distributions, the loss and gradients behave completely differently from those under MSE, leading to power-law dynamics independent of the data distribution [10]. For non-linear models with many more features than their widths, superposition is preferred [7, 8], leading to m^{-1} scaling due to geometry robust to a range of data distributions [7]. For transformation, we have the ℓ^{-1} scaling from averaging and central limit theorem [9], which is a result of the residual connection and not related to power-law structure in data. We emphasize that our mechanisms, superposition, averaging, and universality, depend on **high-level architecture and data properties**, e.g., the use of softmax and the low entropy of target distributions, producing exponents **independent of other details** like power-law structures in data.

To have a heuristic comparison, we argue that linear models with MSE are baskets trying to contain solid bricks (features). The baskets can only contain a limited number of bricks, and the bricks are learned in a power-law order (Figure 2a). LLMs, on the other hand, are non-linear models with softmax—more like a bottle trying to compress different kinds of gases (features). The bottle can have all kinds of gases (superposition), and the speed of learning is limited by the valve (softmax and cross-entropy) rather than gas fractions (Figure 2b).

Our theory therefore suggests new interpretations of neural scaling laws, which may have important implications.

5 Future of neural scaling laws

With the understanding of the current neural scaling laws, we can now speculate on how better scaling might be achieved. The high-level properties of language are not expected to change. Yet, we have insights about architectural innovations that may bypass the identified bottlenecks of current scaling.

Compositionality.—Based on our understanding, transformation mainly reduces error by averaging, which is the most inefficient yet robust way to make use of multiple layers. For more efficient transformation, we may need to encourage the layers to differ. One way we can imagine is to encourage early exit, such that early layers focus on low-level features and handle easy inputs, while later ones focus on high-level features and are used if necessary. If the layers can be compositional, the best case is that loss can scale exponentially with depth, where the characteristic depth is related to the characteristic height of the grammar parsing tree of language (imagining probabilistic context-free

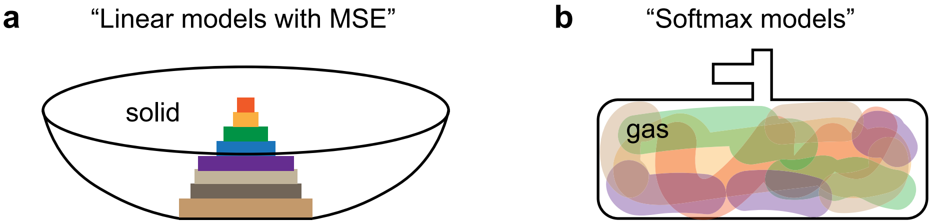


Figure 2: A linear model with MSE loss requires power laws in data to have power laws with model size and training time, which is not true for non-linear models with softmax (when the output distributions are peaked). (a) Linear models are like baskets trying to contain solid bricks (features). The baskets can only contain a limited number of bricks (no superposition [7]), and the bricks are put in an order set by exponential convergence rates of features. (b) LLMs are non-linear models with softmax function, which are more like a bottle trying to compress different kinds of gases (features). The bottle can have all kinds of gases (superposition [7]), and the speed of learning is limited by the valve (softmax)—everything is power-law converging as the loss and gradients are power laws.

languages). To solve complex problems, more depth is desired, yet this can be achieved through extended reasoning at inference time rather than by increasing architectural depth. We then hope that, by making use of compositionality, the loss can be

$$L = \frac{c_m}{m} + \frac{c_\tau}{\tau^{1/3}} + L_0 = \frac{c_N}{N^{1/2}} + \frac{c_\tau}{\tau^{1/3}} + L_0, \tag{7}$$

with depth being a fixed constant. In this hypothetical scenario, at the optimal-compute frontier, we would have $D \propto N^{3/2}$ and[2]

$$L = \frac{c_C}{C^{1/5}} + L_0. \tag{8}$$

Less brute-force in searching.—The training part of loss is a power law with exponent $1/3$ because we need to output a low-entropy distribution, which requires large logits and leads to power-law vanishing gradients. The distribution is low-entropy as it is defined on the whole vocabulary yet the number of relevant tokens is small. A lot of tokens are indeed irrelevant to a specific context. If we can make use of the structure of words, e.g., first identifying the cluster of tokens most likely to be relevant, we can output a distribution on a smaller set of tokens, increasing the entropy and lowering the magnitude of logits. The best case for high-entropy distribution prediction is exponential convergence [10]. We are less confident about the speedup from this approach as effectively we are still searching over the whole vocabulary. Yet, if we can achieve this exponential speedup, the training time would become a large constant, and the loss would be

$$L = \frac{c_m}{m} + \frac{c_\ell}{\ell} + L_0. \tag{9}$$

In this hypothetical scenario, at the optimal-compute frontier, we would have $N \propto \ell$ and

$$L = \frac{c_N}{N^{1/3}} + L_0 = \frac{c_C}{C^{1/3}} + L_0. \tag{10}$$

The most optimistic prediction.—We have outlined our vision of how transformation and training can be improved. The representation due to superposition seems to be more fundamental, and currently we have no good idea to render it more efficient. Our most optimistic prediction is based on combining the two speedups above, i.e., compositionality and less brute-force in searching, which could lead to the loss scaling as

$$L = \frac{c_m}{m} + L_0 = \frac{c_N}{N^{1/2}} + L_0 = \frac{c_C}{C^{1/2}} + L_0, \tag{11}$$

with depth and training time being fixed constants. We therefore hypothesize a cubic speedup (compared to the current $C^{-1/6}$ scaling for the compute-optimal) as the best scenario, which is fundamentally constrained by representation.

[2] We use c_N and c_C to represent the coefficients of the power laws, which are not constants throughout the blog and can take different values as we change the architecture and the optimal-compute frontier.

6 Beyond the loss scaling

So far, we have explained the origins of the neural scaling laws, and studied the implications of the mechanisms for the loss scaling. We next discuss the limitations of loss scaling and the implications of the mechanisms beyond the loss scaling.

Meaning of loss.—Pretrain loss is important as a low loss is a necessary condition for good performance. The loss value captures how well the model can recite and imitate human texts. A model with good representation and transformation can certainly have a low loss. Yet, a low loss may not be sufficient for good performance, especially for downstream tasks beyond next-token prediction.

Limitation of loss scaling.—A focus on loss scaling may then ignore task-specific performance. For instance, increasing width and depth may lead to the same low loss, yet the wider model may be better at knowledge-intensive tasks while the deeper model may be better at reasoning-intensive tasks. Two models can reach the same test loss value on two different data distributions, while their performance can be very different. A single loss value is coarse-grained, and we may need fine-grained metrics to evaluate specific abilities and change the architecture and training data distributions for certain tasks.

When to stop.—Since the pretrain loss cannot capture specific abilities, we should stop pretraining when we care about downstream tasks, or when the models are able to learn the abilities through post-training. This time point to stop, heuristically, is when the model can speak fluently. Imagine that autoregressive language generation is like a chaotic dynamical system, where a small error at each step can lead to a large deviation after many steps, we hypothesize that based on the desired context length to generate, we can estimate the desired loss value. Once this desired context length is long enough for the model to learn reasoning, the model no longer needs to speak the correct texts only based on memory and intuitions, but also by self-evaluation and correction. We therefore also hypothesize that there may be a critical loss value fundamentally related to reasoning and signaling the end of pretraining.

Mechanisms are helpful beyond loss scaling.—We found the mechanisms, i.e., superposition, averaging, and universality, through the study of scaling laws, while they may also be helpful for other aspects of LLMs. Superposition reveals the fundamental limitation of representation, causing difficulties for both efficiency and interpretability. However, superposition may also be a source of creativity. Certain geometric interference constitutes noise in some contexts but signal in others. Since reinforcement learning can only strengthen existing behaviors, superposition may help to learn new skills, creating new connections between features. Averaging in transformation is inefficient, yet robust. And our proposal to encourage compositionality may help with efficiency and interpretability—each layer can have a clearer function. Our finding of universality due to softmax and cross-entropy emphasizes the critical and special role of specific non-linearities used. Interpretability research may benefit from a more detailed understanding of the consequences of softmax specifically, e.g., how representation geometry is shaped by softmax. We hope that the mechanisms we found provide fine-grained insights into LLMs, helpful for understanding LLMs beyond the loss scaling.

7 At the end

In summary, we studied the origins of neural scaling laws observed in LLMs, and found that the loss is dominated by three fundamental power laws, i.e., an inverse width law, an inverse depth law, and a one-third training time law, which are due to representation superposition, ensemble averaging in transformation, and universality in training dynamics arising from non-linearities, respectively.

We emphasize that these power laws depend on high-level architecture and data properties, and are not sensitive to other details like power-law structures in data. With these insights, we know these three power laws can keep decreasing as models scale up, i.e., will not plateau at non-zero values at any finite scale. Moreover, the insights enable us to propose directions of architectural innovations for better scaling. Finally, we discussed the limitations of loss scaling and the implications of the three mechanisms beyond the loss scaling.

We anticipate that the understanding of neural scaling can help with the ongoing development of efficient and interpretable LLMs, and may also open a window to fundamental principles of intelligence.

References

[1] 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.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[5] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

[6] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[7] Yizhou Liu, Ziming Liu, and Jeff Gore. Superposition yields robust neural scaling. *arXiv preprint arXiv:2505.10465*, 2025.

[8] 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.

[9] Yizhou Liu, Sara Kangaslahti, Ziming Liu, and Jeff Gore. Inverse depth scaling from most layers being similar. *arXiv preprint arXiv:2602.05970*, 2026.

[10] Yizhou Liu, Ziming Liu, Cengiz Pehlevan, and Jeff Gore. Universal one-third time scaling in learning peaked distributions. *arXiv preprint arXiv:2602.03685*, 2026.

[11] Tamay Besiroglu, Ege Erdil, Matthew Barnett, and Josh You. Chinchilla scaling: A replication attempt. *arXiv preprint arXiv:2404.10102*, 2024.

[12] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[13] Eric Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. *Advances in Neural Information Processing Systems*, 36:28699–28722, 2023.

[14] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024.