An exactly solvable model for emergence and scaling laws

Yoonsoo Nam*[a], Nayara Fonseca*[a], Seok Hyeong Lee[b], and Ard Louis[a]

[a]Rudolf Peierls Centre for Theoretical Physics, University of Oxford
[b]Center for Quantum Structures in Modules and Spaces, Seoul National University

Abstract

Deep learning models can exhibit what appears to be a sudden ability to solve a new problem as training time (T), training data (D), or model size (N) increases, a phenomenon known as emergence. In this paper, we present a framework where each new ability (a skill) is represented as a basis function. We solve a simple multi-linear model in this skill-basis, finding analytic expressions for the emergence of new skills, as well as for scaling laws of the loss with training time, data size, model size, and optimal compute (C). We compare our detailed calculations to direct simulations of a two-layer neural network trained on multitask sparse parity, where the tasks in the dataset are distributed according to a power-law. Our simple model captures, using a single fit parameter, the sigmoidal emergence of multiple new skills as training time, data size or model size increases in the neural network.

1 Introduction

Large language models (LLMs) can exhibit rapid (on a log scale) transitions where they acquire the ability (or skill) to solve a new task as the number of parameters, the training dataset size, or the amount of training time is scaled up. This phenomenon has been dubbed *emergence* [1–4], and has attracted a lot of recent attention. It motivates the costly drive to train ever larger models on ever larger datasets, in the hope that new skills will emerge. It also motivates theoretical studies with the goal that LLMs can be made more predictable and safer in the future. While the concept of emergence has been critiqued on the grounds that the sharpness of the transition to acquiring a new skill may be sensitive to the measure being used [5], the observation that important new skills are learned for larger models that appear not to be present in smaller ones is robustly established. These results raise many challenging questions such as: What triggers the appearance of complex new skills as models scale up? Can we predict when new skills will be acquired? These questions are complicated by difficulties in formally defining skills or capabilities [6], and by our general lack of understanding of the internal representations of deep neural networks [7].

Another widely observed property of deep learning models is that the loss improves predictably as a power-law in the number of data points or number of model parameters or simply in the amount of compute thrown at a problem. Such phenomenon are called neural scaling laws [8, 9], and have been widely observed across different architectures and datasets [10–15]. While the scaling exponents can depend on these details, the general phenomenon of scaling appears to be remarkably robust. This raises many interesting questions such as: What causes the near-universal scaling behaviour? How does the continuous scaling of the loss relate to the discontinuous emergence of new skills?

A challenge in answering the questions raised by the phenomena of emergence and scaling laws arises from the enormous scale and expense of training cutting-edge modern LLMs, which are optimized for commercial applications, and not for answering scientific questions about how they work. One way that progress can be made is to study simpler dataset/architecture combinations that are more tractable. The current paper is inspired in part by recent work in this direction that proposed studying emergence in learning the sparse parity problem [16, 17], which is easy to define, but known to be computationally hard. In particular, Michaud et al. [17] introduce the multiple unique sparse parity

*These authors contributed equally; {yoonsoo.nam,nayara.fonsecadesa}@physics.ox.ac.uk.

problem – where tasks are distributed in the data through a power-law distribution of frequencies – as a proxy for studying emergence and neural scaling in LLMs. For this data set, the authors were able to empirically measure and schematically derive scaling laws as a function of training steps (T), parameters (N), and training samples (D). They also directly observed the emergence of new skills with increasing T, showing how smooth neural scaling laws can arise by averaging over many individual cases of the emergence of new skills.

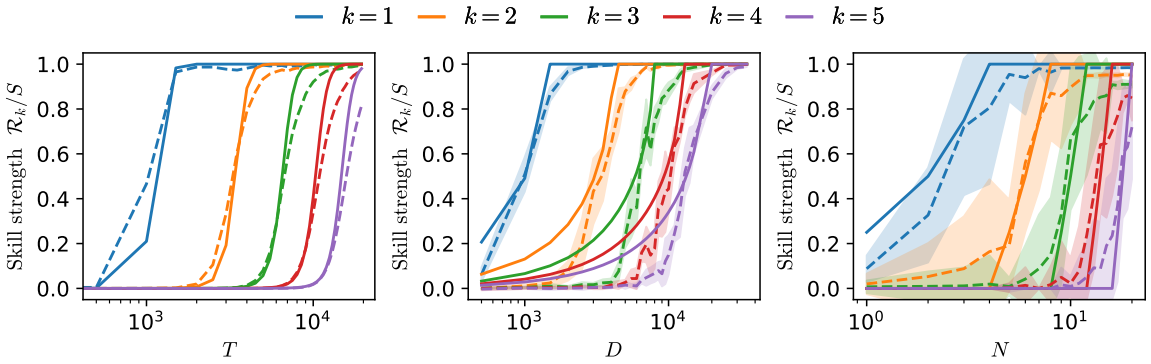
Here we define a basis of orthogonal functions for the multitask sparse parity problem. Each basis function corresponds to a skill that can be learned, and their respective frequencies are distributed following a power law with exponent $\alpha + 1$. We then propose a simple multilinear expansion in these orthogonal functions that introduces a layered structure reminiscent of neural networks and gives rise to the stage-like training dynamics [18]. With our simple model, we can analytically calculate full scaling laws, including pre-factors, as a function of data exponent α, T, D, N, and optimal compute C. Our simple model can, with just one parameter calibrated to the emergence of the first skill, predict the ordered emergence of multiple skills in a 2-layer neural network. We summarize our contributions:

1. *Skills as basis functions.* We establish a framework for investigating emergence by representing skills as orthogonal functions that form a basis in function space (Section 2). We apply our methods to controlled experiments on the multitask sparse parity dataset.

2. *Multilinear model.* We propose an analytically tractable model that is expanded in the basis of skill functions, and is multilinear with respect to its parameters so that it possesses a layerwise structure (Section 3). The multilinear nature of the model produces non-linear dynamics, and the orthogonal basis decouples the dynamics of each skill.

3. *Scaling laws.* We employed both intuitive (Section 4, Section 5, and Appendix D) and rigorous (Appendix J) derivations of scaling laws for our multilinear model, relating the model's performance to training time (T), dataset size (D), number of parameters (N), and optimal compute ($C = N \times T$). We show that the scaling exponents for these factors are $-\alpha/(\alpha+1)$, $-\alpha/(\alpha+1)$, $-\alpha$, $-\alpha/(\alpha+2)$, respectively, where $\alpha+1$ is the exponent of the power-law (Zipfian) input data.

4. *Predicting emergence.* We demonstrate that our multilinear model captures the skill emergence of a 2-layer NN for varying training time, dataset size, and number of trainable parameters. Our results show that the multilinear model, calibrated only on the first skill, can predict the emergence of subsequent skills in the 2-layer NN, see Fig. 1 and Section 6.



(a) Time Emergence (b) Data Emergence (c) Parameter Emergence

Figure 1: **Predicting Emergence.** The skill strength $\mathcal{R}_k$, defined as the k^{th} coefficient if a model is expanded in the basis of the skill functions (g_ks), measures how well the k^{th} skill is learned, and is plotted against (a) time T, (b) data set size D, and (c) number of parameters N. $\mathcal{R}_k$ is normalized by the target scale S such that $\mathcal{R}_k/S = 1$ means zero skill loss. The dashed lines show the abrupt growth – emergence – of 5 skills for a 2-layer NN (Appendix I) trained on the multitask sparse parity problem with data power-law exponent $\alpha = 0.6$. Solid lines are the predictions (Eqs. (37), (40) and (44), respectively) from our multilinear model calibrated on the first skill (blue) only.

Skill idx (I)	Control bits	Skill bits (X)	y	$M(i,x)$	$g_1(i,x)$	$g_2(i,x)$	...	$g_{n_s}(i,x)$
1	10000000	110011000001010	S	[1,1,0]	1	0	...	0
1	10000000	100110010100011	$-S$	[0,1,0]	-1	0	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2	01000000	001101010110101	$-S$	[0,0,1]	0	-1	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n_s	00000001	001001001001100	$-S$	[1,1,1]	0	0	...	-1

Table 1: **Multitask sparse parity dataset and skill basis functions.** The control bits n_s are one-hot vectors encoding specific parity tasks, indexed in the first column. The frequency of the distinct parity tasks follows a rank-frequency distribution with an inverse power law relation. The skill bits are binary strings. The y column shows the target scale S multiplied by the resulting parity computed from $m = 3$ bits (highlighted in colors), which form the subset $M(i,x)$. The last columns show the values of the skill basis functions $g_k(i,x)$, defined in Eq. (2).

Related works. Focusing on data scaling, Hutter [19] develops a model with a discrete set of features. Under the assumption of a Zipfian distribution of features, this model demonstrates that the error decreases as a power law with increasing data size. In a related vein, Michaud et al. [17] propose a model of neural scaling laws in which the loss is decomposed into a sum over 'quanta'. Their model aims to reconcile the apparent discrepancy between loss metrics' regular power-law scaling and novel capabilities' abrupt development in large-scale models. Various other models for neural scaling laws have been proposed in recent research, including connecting neural scaling exponents to the data manifold's dimension [20] and their relation with kernels [21], proposing solvable random-feature models [22, 23], and developing data scaling models using kernel methods [24–26]. Closely related to the study of neural scaling laws is the understanding of emergent abilities in large language models. Several studies [1–4] document examples of such emergent abilities. [1] Arora and Goyal [28] propose a framework for the emergence of tuples of skills in language models, in which the task of predicting text requires combining different skills from an underlying set of language abilities. Okawa et al. [29] demonstrate that a capability composed of smoothly scaling skills will exhibit emergent scaling due to the multiplicative effect of the underlying skills' performance. Other works related to the skill acquisition include Yu et al. [30], who introduce a new evaluation to measure the ability to combine skills and develop a methodology for grading such evaluations, and Chen et al. [31], who formalize the notion of skills and their natural acquisition order in language models. Throughout this work, we consider the infinite data regime, such that the optimizer only sees 'fresh' samples at each iteration step, and there is no distinction between training and test losses. This contrasts with the grokking phenomenon [32], which also exhibits sigmoid-shape curves but is related to a discrepancy between the model's train and test behavior.

2 Setup

In this section, we define the multitask sparse parity problem under the mean-squared error (MSE) loss. We represent skills as orthogonal functions and measure their strength in a model by calculating the linear correlation between the model output and the skill basis functions. For a comprehensive list of notations used in this work, refer to the **glossary** in Appendix A.

Multitask sparse parity problem. In the sparse parity problem, n_b skill bits are presented to the model. The target function is a parity function applied to a fixed subset of the input bits. The model must detect the relevant $m < n_b$ sparse bits and return the parity function on this subset ($M(i,x)$, see Table 1). Michaud et al. [17] introduced the **multitask** sparse parity problem by introducing n_s unique sparse parity variants – or skills – with different sparse bits (for a representation, see Table 1).

[1] We note that Schaeffer et al. [5] have argued that many of these examples may be artifacts of the evaluation metric (see also [3, 4, 27]). Our work only considers continuously optimized measures (such as MSE loss) instead of hard threshold measures (like accuracy) that may artificially enhance the sigmoid-shaped curves.

Each skill is represented in the n_s control bits as a one-hot string, and the model must solve the specific sparse parity task indicated by the control bits (for more details, see Appendix B.1).

The n_s skills (random variable $I \in \{1, 2, \ldots, n_s\}$) follow a power law (Zipfian[2]) distribution $\mathcal{P}_s$, and the skill bits (random variable $X \in \{0,1\}^{n_b}$) are uniformly distributed. Because $\mathcal{P}_s$ and $\mathcal{P}_b$ are independent, the input distribution $\mathcal{P}(I, X)$ follows a product of two distributions:

$$\mathcal{P}_s(I = i) := \frac{i^{-(\alpha+1)}}{\sum_j^{n_s} j^{-(\alpha+1)}}, \qquad \mathcal{P}_b(X = x) := 2^{-n_b}, \qquad \mathcal{P}(I, X) := \mathcal{P}_s(I)\mathcal{P}_b(X). \tag{1}$$

We denote $A = \left(\sum_{j=1}^{n_s} j^{-(\alpha+1)}\right)^{-1}$ so that $\mathcal{P}_s(i) := Ai^{-(\alpha+1)}$.

Skill basis functions. We represent the k^{th} skill as a function $g_k : \{0,1\}^{n_s+n_b} \to \{-1, 0, 1\}$ that returns the parity ($\{-1, 1\}$) on the k^{th} skill's sparse bits if $i = k$, but returns 0 if the control bit mismatches that of the k^{th} skill ($i \neq k$):

$$g_k(i, x) := \begin{cases} (-1)^{\sum_j M_j(i,x)} & \text{if } i = k \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where $M : \{0,1\}^{n_s+n_b} \to \{0,1\}^m$ is the map that selects the relevant sparse bits for the i^{th} skill (Table 1). Note that different skill functions have 0 correlation as the supports of skills functions are mutually exclusive:

$$g_k(i, x)g_{k'}(i, x) = \delta_{i,k}\delta_{k,k'}. \tag{3}$$

The target function. The target function is a sum over n_s skill functions multiplied by a target scale S:

$$f^*(i, x) := S \sum_{k=1}^{n_s} g_k(i, x). \tag{4}$$

The target scale S is the norm of the target function ($\mathbf{E}_{I,X}\left[f^*(I, X)f^*(I, X)\right] = S^2$). Note that the skill functions serve as 'features' or countable basis for describing the target function as in [19].

Loss. We use MSE loss For analytic tractability:

$$\mathcal{L} := \frac{1}{2}\mathbf{E}_{X,I}\left[\left(f^*(I, X) - f(I, X)\right)^2\right], \tag{5}$$

where f is the function expressed by a given model. We define the skill loss $\mathcal{L}_k$ as the loss when only the k^{th} skill is given, formulated as

$$\mathcal{L}_k := \frac{1}{2}\mathbf{E}_X\left[\left(f^*(I = k, X) - f(I = k, X)\right)^2\right]. \tag{6}$$

Then the total loss can be expressed as a sum of $\mathcal{L}_k$ weighted by their skill frequencies $\mathcal{P}_s(I = k)$:

$$\mathcal{L} = \frac{1}{2}\sum_{k=1}^{n_s} \mathcal{P}_s(I = k)\mathbf{E}_X\left[\left(Sg_k(I = k, X) - f(I = k, X)\right)^2\right] \tag{7}$$

$$= \sum_{k=1}^{n_s} \mathcal{P}_s(I = k)\mathcal{L}_k. \tag{8}$$

Skill strength. The skill strength or the linear correlation between the k^{th} skill (g_k) and a function expressed by the model at time T (f_T) is

$$\mathcal{R}_k(T) := \mathbf{E}_X\left[g_k(I = k, X)f_T(I = k, X)\right]. \tag{9}$$

[2]The literature on Zipf's law is vast. In some traditions Zipf is synonymous with a power law. In others Zifp only refers to frequency-rank plots with exponent 1. We will mainly use the more general power law terminology.

The skill strength $\mathcal{R}_k$ is the k^{th} coefficient if a model is expanded on the basis of the skill functions (g_ks). The skill strength, like the test loss, **can be accurately approximated** (see Appendix I.2). The skill loss $\mathcal{L}_k$ (Eq. (6)) can be expressed by the skill strength and the norm of the learned function for $I = k$:

$$\mathcal{L}_k(T) = \frac{1}{2}\mathbf{E}_X\left[(Sg_k(I=k,X) - f_T(I=k,X))^2\right] \tag{10}$$

$$= \frac{1}{2}\left(S^2 + \mathbf{E}_X\left[f_T(I=k,X)^2\right] - 2S\mathcal{R}_k(f_T)\right). \tag{11}$$

The skill loss becomes 0 if and only if $f_T(I=k,X) = Sg_k(I=k,X)$.

Experimental setting. We experiment with a 2-layer fully connected neural network (NN) with ReLU activations. The NN receives the $n_s + n_b$ bits as inputs and outputs a scalar ($\{0,1\}^{n_s+n_b} \to \mathbb{R}$). In most of the experiments, the NN is trained with stochastic gradient descent (SGD) with width 1000, using $n_s = 5$, $m = 3$, and $n_b = 32$, unless otherwise stated. See Appendix I for details.

3 Multilinear Model

We propose a simple multilinear model – multilinear with respect to the parameters – with the first N most frequent skill functions $g_k(i,x)$ as the basis functions (features):

$$f_T(i,x;a,b) = \sum_{k=1}^{N} a_k(T)b_k(T)g_k(i,x), \tag{12}$$

where $a, b \in \mathbb{R}^N$ are the parameters. The model has built-in skill functions g_k – which transform control bits and skill bits into the parity outputs of each skill – so the model only needs to scale the parameters to $a_k b_k = S$. The multilinear structure (Fig. 2(a)) is similar to the layered structure of NNs and gives rise to the stage-like training dynamics (Section 4) different from that of linear models. [3] A similar model and its dynamics have been studied by Saxe et al. [18] in the context of linear neural networks; see Appendix B.2 for details.

Note that $a_k(T)b_k(T)$ is equivalent to the skill strength $\mathcal{R}_k$ in Eq. (9):

$$a_k(T)b_k(T) = \mathbf{E}_X\left[g_k(I=k,X)f_T(I=k,X)\right] = \mathcal{R}_k(T), \tag{13}$$

For the multilinear model, we can express the skill loss in Eq. (10) as a function of $\mathcal{R}_k$:

$$\mathcal{L}_k(T) = \frac{1}{2}(S - \mathcal{R}_k(T))^2. \tag{14}$$

Assuming that we are training the model on D samples from $\mathcal{P}(I,X)$, the dynamics of each skill ($\mathcal{R}_k$) is **decoupled** because g_ks' supports are **mutually exclusive** (Eq. (3)). The empirical loss on D samples can be decomposed into the sum of empirical skill losses, which only depends on their respective $\mathcal{R}_k$ (see Appendix C.1):

$$\boxed{\mathcal{L}^{(D)} = \sum_{k=1}^{n_s} \mathcal{L}_k^{(D)} = \frac{1}{2D}\sum_{k=1}^{n_s} d_k(S - \mathcal{R}_k)^2,} \tag{15}$$

where d_k is the number of samples of the k^{th} skill (i.e. number of samples (i,x) with $g_k(i,x) \neq 0$). Now, we can independently solve each skill's dynamics under gradient descent.

[3] Note that if we reparameterize $a_k b_k$ as a single parameter, the model becomes a linear model but with different dynamics. See Fig. 10 and Appendix G for further discussion on the similarities and differences between linear and multilinear models.

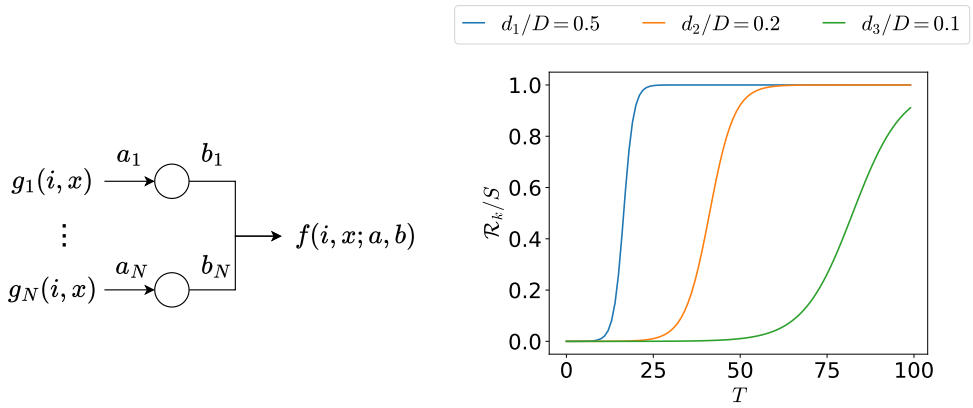
Decoupled dynamics of the multilinear model. *Assuming small and positive initialization ($0 < \mathcal{R}_k(0) \ll S$), the k^{th} skill strength ($\mathcal{R}_k(T)$) follows*

$$\boxed{\frac{\mathcal{R}_k(T)}{S} = \frac{1}{1 + \left(\frac{S}{\mathcal{R}_k(0)} - 1\right) e^{-2\eta \frac{d_k}{D} ST}},} \tag{16}$$

where η is the learning rate, D is the number of training samples, and d_k is the number of samples of the k^{th} skill (i.e. number of samples (i, x) with $g_k(i, x) \neq 0$).

Proof See Appendix C.1. ■

The skill strength in Eq. (16) is a sigmoid function in time and smaller d_k/D, which becomes $\mathcal{P}_s(I = k)$ for $D \to \infty$, results in a delayed growth (Fig. 2(b)). For the connection to linear neural networks [18], see Fig. 7 in Appendix B.2.



(a) Multilinear model illustration

(b) Decoupled dynamics

Figure 2: **Multilinear model. (a):** An illustration of the multilinear model which is multilinear in terms of parameters, generating a layerwise structure. The model has the skill functions g_ks as basis functions. **(b):** The dynamics of the multilinear model are decoupled and each skill strength ($\mathcal{R}_k$) shows a sigmoidal growth in time. Note that skills with lower frequency have a more delayed growth.

4 Stage-like training: intuitive derivation of the scaling laws

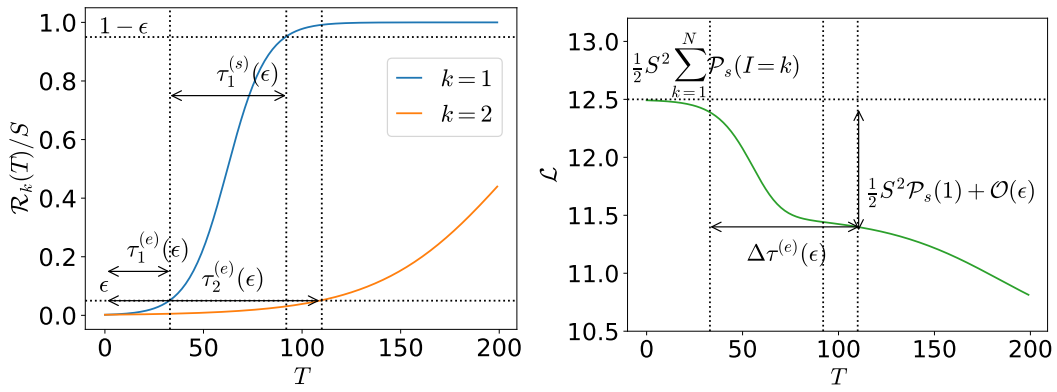
In this section, we define stage-like training – one skill is completely learned before the next skill initiates learning (Fig. 2(b)) – show under what conditions it occurs, and provide an example of how stage-like training results in the time scaling law. This section offers intuition on how the layerwise structure shared by NNs and our model can result in empirically observed scaling laws. Readers may skip this section as the scaling laws for our model (Section 5) can be shown without the stage-like training assumption. Still, it provides intuition for the discussion on NN dynamics in Section 6 and explains how the model in [17] may arise from the NN dynamics.

In Fig. 2(b), we observe the stage-like training in which one skill saturates (reaches $\mathcal{R}_k/S \approx 1$) before the next skill initiates its emergence. To quantify this behavior, we define two intervals for each skill (see Fig. 3(a)):

- The emergent time $\tau_k^{(e)}(\epsilon)$: the time for $\mathcal{R}_k/S$ to reach ϵ;
- The saturation time $\tau_k^{(s)}(\epsilon)$: the time for $\mathcal{R}_k/S$ to saturate from ϵ to $1 - \epsilon$.

Using the dynamics equation (Eq. (16)) and that $d_k/D \to \mathcal{P}_s(k)$, the emergent time and saturation time of the k^{th} skill becomes

$$\tau_k^{(e)}(\epsilon) = \frac{1}{2\eta \mathcal{P}_s(k) S} \ln\left(\frac{\frac{S}{\mathcal{R}_k(0)} - 1}{\frac{1}{\epsilon} - 1}\right) \propto k^{\alpha+1}, \qquad \tau_k^{(s)}(\epsilon) = \frac{1}{\eta \mathcal{P}_s(k) S} \ln\left(\frac{1}{\epsilon} - 1\right) \propto k^{\alpha+1}. \tag{17}$$



(a) Emergent and saturation time

(b) Loss change between emergences

Figure 3: **Stage-like training.** The multilinear model is trained on the multitask sparse parity problem with $\alpha = 0.6$. **(a):** Skill strength of the model as a function of time. The emergent time $\tau_k^{(e)}(\epsilon)$ is the time required for the k^{th} skill to reach $\mathcal{R}_k/S = \epsilon$. The saturation time $\tau_k^{(s)}(\epsilon)$ is the time required for $\mathcal{R}_k/S$ to saturate from ϵ to $1-\epsilon$. The model shows stage-like training if the emergent time interval $\tau_{k+1}^{(e)}(\epsilon) - \tau_k^{(e)}(\epsilon)$ is larger than the saturation time $\tau_k^{(s)}(\epsilon)$ for sufficiently small ϵ (0.05 in the figure). **(b):** The loss as a function of time for the same system as (a). For stage-like training, the change in the loss for the k^{th} emergence is $\mathcal{L}_k + \mathcal{O}(\epsilon)$ and the interval for the next emergence is $\Delta\tau^{(e)}(\epsilon) = \tau_{k+1}^{(e)}(\epsilon) - \tau_k^{(e)}(\epsilon)$.

For sufficiently small initialization ($\mathcal{R}_k(0) \ll S$), we get a **stage-like** training:

$$\tau_k^{(s)}(\epsilon) < \tau_{k+1}^{(e)}(\epsilon) - \tau_k^{(e)}(\epsilon), \qquad \epsilon \ll 1. \tag{18}$$

In other words, the model finishes learning (saturating) the k^{th} skill before starting to learn (emerging) the next skill (Fig. 3(a)). The emergent interval between the k and $k+1$ skills relative to the $\tau_k^{(e)}(\epsilon)$ is

$$\frac{\Delta T}{T} = \frac{\tau_{k+1}^{(e)}(\epsilon) - \tau_k^{(e)}(\epsilon)}{\tau_k^{(e)}(\epsilon)} = \frac{(k+1)^{\alpha+1} - k^{\alpha+1}}{k^{\alpha+1}} \tag{19}$$

$$= (\alpha+1)k^{-1} + \mathcal{O}(k^{-2}). \tag{20}$$

Accordingly, at $\tau_k^{(e)}(\epsilon)$, all skills with index up to but not including k have saturated ($\mathcal{R}_{i<k} \approx S$), or equivalently $\mathcal{L}_{i<k} \approx 0$ (Eq. (14)). The total loss, the sum of $\mathcal{L}_k$ weighted by $\mathcal{P}_s(k) \propto k^{-(\alpha+1)}$ (Eq. (8)), becomes $\sum_{j=k}^{\infty}\mathcal{P}_s(I=j)S^2/2$ (see Fig. 3(b)). The saturation of the k^{th} skill results in a loss difference of $\mathcal{P}_s(I=k)S^2/2$. Thus, we obtain

$$\frac{\Delta\mathcal{L}}{\mathcal{L}} \approx \frac{\mathcal{P}_s(I=k)}{\sum_{j=k}^{\infty}\mathcal{P}_s(I=j)} = -\frac{k^{-(\alpha+1)}}{\sum_{j=k}^{\infty} j^{-(\alpha+1)}} \approx -\frac{k^{-(\alpha+1)}}{\int_k^{\infty} j^{-(\alpha+1)}dj} \tag{21}$$

$$= -\alpha k^{-1} + \mathcal{O}(k^{-2}). \tag{22}$$

Assuming $k \gg 1$ and combining Eq. (22) and Eq. (20) to the largest order, we have the equation for the power law with exponent $-\alpha/(\alpha+1)$ in Fig. 4(a, i):

$$\frac{\Delta\mathcal{L}}{\mathcal{L}} = -\frac{\alpha}{\alpha+1}\frac{\Delta T}{T}. \tag{23}$$

When the condition $k \gg 1$ is no longer met, typically for small T, the model deviates from the power-law, as evident in the top left region of Fig. 4(a,i). This behavior is further illustrated by the reversed-sigmoidal shape of $\mathcal{L}$ in Fig. 3(b).

If the stage-like training holds for any resource (e.g., time, data, or parameters), the scaling law can be derived using the ratio of change in loss per skill (Eq. (22)) and the ratio of change with respect to

the resource (given by the emergent time in Eq. (20)). Although the stage-like training dynamics may not occur in real-world scenarios, studying the layerwise dynamics and the stage-like training behavior that emerges from the power-law input distribution ($\mathcal{P}_s$) offers insights into the emergence of skills in NNs that possess this layerwise structure.

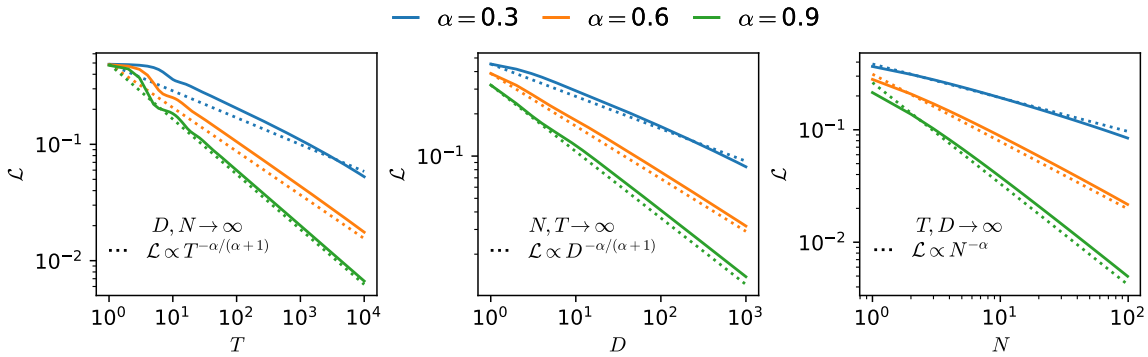
5 Scaling laws

In this section, we derive the scaling laws of our multilinear model (Section 3) for time (T), data (D), parameters (N) and optimal compute (C). For analytical tractability, we define compute as $C := T \times N$ [23]. Table 2 shows a summary of the scaling laws. Note that we achieve the same scaling laws as in Hutter [19] for D and in Michaud et al. [17] for T, D, and N. Assuming $0 < \alpha < 1$, the exponents are consistent with the small power-law exponents reported in large-scale experiments, see, e.g., [9, 14, 33].

Using Eqs. (7), (14) and (16), we have the loss as a function of time (T), data (D), parameters (N), and the number of observations for each skill $[d_1, \cdots, d_{n_s}]$:

$$\mathcal{L} = \frac{S^2}{2} \sum_{k=1}^{N} \mathcal{P}_s(k) \frac{1}{\left(1 + \left(\frac{S}{\mathcal{R}_k(0)} - 1\right)^{-1} e^{2\eta \frac{d_k}{D} ST}\right)^2} + \frac{S^2}{2} \sum_{k=N+1}^{n_s} \mathcal{P}_s(k). \tag{24}$$

Under suitable assumptions (for example, we take $D, N \to \infty$ for the T scaling law), we can use Eq. (24) to derive how $\mathcal{L}$ scales respect to T, D, N, and C (Table 2). Figure 4 shows the empirical scaling laws for T, D, and N, while Fig. 5 shows the empirical scaling for optimal compute C where model sizes (N) and training times (T) are chosen optimally to minimize the loss for given compute budget (C). Details of the intuitive derivations for the scaling laws are provided in Appendix D, while a rigorous approach (including error bounds) is presented in Appendix J.



(a) Time Scaling (b) Data Scaling (c) Parameter Scaling

Figure 4: **Scaling laws.** The learning curve ($\mathcal{L}$ is the MSE loss) of the multilinear model (solid) and the respective power-law (dotted) for (a) time T, (b) data D, and (c) parameters N. Lower left legends show the condition and exponent for each scaling law. For the derivation of the exponents of the scaling laws, see Appendix D. For a rigorous derivation including the prefactors, see Appendix J.

5.1 Time scaling law

As previously discussed, when the initialization is sufficiently small, we can derive the time scaling law in Eq. (23) using the stage-like training condition in Eq. (18). However, for our model, it is possible to derive the time scaling law without relying on the stage-like assumption. We first assume the time as the bottleneck and take $N, D \to \infty$. By using Eq. (14) and the decoupled dynamics of the skill loss (Eq. (16)), each skill loss $\mathcal{L}_k$ becomes a function **solely dependent** on $k^{-(\alpha+1)}T$. This establishes a relationship between the derivatives of the skill loss with respect to k and T.

Bottleneck	Time	Data	Parameter	Exponent
Time (T)	T	∞	∞	$-\alpha/(\alpha+1)$
Data (D)	∞	D	∞	$-\alpha/(\alpha+1)$
Parameter (N)	∞	∞	N	$-\alpha$
Compute (C)	$C^{(\alpha+1)/(\alpha+2)}$	∞	$C^{1/(\alpha+2)}$	$-\alpha/(\alpha+2)$

Table 2: **Summary of the scaling laws**. The leftmost column shows the bottleneck of the scaling law. The middle three columns show the resource values in terms of the bottleneck (either taken to infinity or proportional to the bottleneck). The last column shows the scaling exponent for the loss as power-law of the bottleneck where $\alpha+1$ is the exponent of the Zipfian input data (Eq. (1)). For a detailed derivation, see Appendix D.

$$\mathcal{L}_k = \frac{S^2}{2\left(1+\left(\frac{S}{\mathcal{R}_k(0)}-1\right)^{-1} e^{2\eta A k^{-(\alpha+1)} ST}\right)^2}, \qquad \frac{d\mathcal{L}_k}{dT} = -\frac{k}{(\alpha+1)T}\frac{d\mathcal{L}_k}{dk}, \tag{25}$$

where $d_k/D \to \mathcal{P}_s$ for $D \to \infty$ and A is the normalization constant for the distribution of skills $\mathcal{P}_s$ in Eq. (1). For the total loss, we approximate the sum in Eq. (8) by an integral and differentiate with respect to T, which can be integrated by parts using Eq. (25) to give the scaling law:

$$\frac{d\mathcal{L}}{dT} \approx \int_1^\infty A k^{-(\alpha+1)} \frac{d\mathcal{L}_k}{dT} dk = -\frac{1}{(\alpha+1)T}\int_1^\infty A k^{-\alpha}\frac{d\mathcal{L}_k}{dk}dk \approx -\frac{\alpha}{(\alpha+1)}\frac{\mathcal{L}}{T}, \tag{26}$$

where the last approximation requires large T. We can rearrange the equation above to obtain the time scaling law with exponent $-\alpha/(\alpha+1)$ as in Fig. 4(a). For details of the derivation and the finite correction of the scaling law for small α, see Appendix D.2.

5.2 Data scaling law

The data scaling law assumes $T \to \infty$ and $N \to \infty$ with data as the bottleneck. From the dynamics of the multilinear model (Eq. (16)) and the relationship between skill loss and skill strength (Eq. (14)), we can show that our model is a one-shot learner:

One shot learner. *Given that $N > k$, $T \to \infty$, and d_k is the number of samples from the training set with $g_k(i,x) \neq 0$, the k^{th} skill loss after training is*

$$\mathcal{L}_k(\infty) = \begin{cases} 0 & : d_k > 0 \\ (S - \mathcal{R}_k(0))^2/2 \approx S^2/2 & : d_k = 0. \end{cases} \tag{27}$$

Proof Follows trivially from Eq. (15), see Appendix C.2. ■

Our model requires only one sample from the k^{th} skill to learn such a skill, similar to how language models are few-shot learners at inference. [4] The model can one-shot learn a skill since it has g_k as the basis functions, and the dynamics among different skills are decoupled. A similar one-shot learner has been studied in Hutter [19] where the error depends on a single 'observation' of a feature.

Because the k^{th} skill loss **only depends** on d_k (number of observations for the k^{th} skill), we can calculate the expectation of the skill loss for D data points from $P_{observed}(k|D)$ or the probability that $d_k > 0$:

$$P_{observed}(k|D) = 1 - (1 - \mathcal{P}_s(k))^D, \quad \mathbf{E}_D[\mathcal{L}_k] = \frac{1}{2}S^2(1 - P_{observed}(k|D)), \tag{28}$$

where the expectation $\mathbf{E}_D$ is over all possible training sets of size D. Using Eq. (8) and Eq. (28), the total loss is

$$\mathbf{E}_D[\mathcal{L}] = \frac{1}{2}S^2\sum_{k=1}^\infty (1-\mathcal{P}_s(k))^D \mathcal{P}_s(k) \approx \frac{1}{2}S^2 A\int_1^\infty \left(1 - Ak^{-(\alpha+1)}\right)^D k^{-(\alpha+1)}dk, \tag{29}$$

[4] Few-shot learning is typically discussed in the context of models that have undergone pre-training (see, e.g. [1]). We speculate that expanding in the basis g_k in our framework can model aspects of the pre-training process.

where A is the normalization constant for $\mathcal{P}_s(k)$ in Eq. (1). We can express the loss difference for an additional data point $\Delta\mathcal{L} = \mathbf{E}_{D+1}[\mathcal{L}] - \mathbf{E}_D[\mathcal{L}]$ as an integral over k:

$$\Delta\mathcal{L} \approx -\frac{1}{2}\alpha S^2 A \int_1^\infty \left(1 - Ak^{-(\alpha+1)}\right)^D k^{-2(\alpha+1)} dk. \tag{30}$$

We can calculate $\Delta\mathcal{L}$ with integration by parts and express the terms with $\Delta\mathcal{L}$ and $\mathbf{E}_D[\mathcal{L}]$ to obtain the following relationship for $D \gg 1$:

$$\frac{\Delta\mathcal{L}}{\mathbf{E}_D[\mathcal{L}]} \approx -\frac{\alpha}{(\alpha+1)}\frac{\Delta D}{D}. \tag{31}$$

The equation above is the data scaling law with exponent $-\alpha/(\alpha+1)$ (Fig. 4(b)), which agrees with previous works, see, e.g., [17, 19, 26]. For the details of the derivation, see Appendix D.3.

5.3 Parameter scaling law

The parameter scaling law assumes $T \to \infty$ and $D \to \infty$, with the parameters $N < n_s$ as the bottleneck. Because our model is a one-shot learner (Eq. (27)), learning of the k^{th} skill **only depends** on the existence of g_k in the model; the model with $[g_1, \cdots, g_N]$ will learn all $k \leq N$ skills with $\mathcal{L}_k = 0$.

Equivalence between a basis function and a skill. *Given $T, D \to \infty$ and if the multilinear model has the N most frequent skill functions as a basis,*

$$\mathcal{L}_k(\infty) = \begin{cases} 0 & : k \leq N \\ S^2/2 & : k > N. \end{cases} \tag{32}$$

Proof Follows trivially from Eq. (15), see Appendix C.3. ■

Using Eq. (32) and Eq. (7), we can express the total loss as function of N:

$$\mathcal{L} \approx \frac{S^2}{2}\int_{N+1}^\infty Ak^{-(\alpha+1)}dk \propto (N+1)^{-\alpha}. \tag{33}$$

By approximating $N \approx N+1$ for $N \gg 1$, we obtain the power law with exponent $-\alpha$ (Fig. 4(c)).

5.4 Scaling law for compute optimal performance

Assuming $D \to \infty$ and ignoring the constant factors, we approximate the loss in Eq. (24) as

$$\mathcal{L} \approx T^{-\alpha/\alpha+1} + N^{-\alpha}, \tag{34}$$

where we approximate the first term in Eq. (24) (first N skills bottlenecked by time) with the time scaling law and the second term in Eq. (24) ($n_s - N$ unlearned skills bottlenecked from the lack of basis functions) with the parameter scaling law. By Lagrangian multiplier, we can show that optimal compute C is achieved when $T \propto N^{\alpha+1}$. Intuitively, the optimal allocation $T \propto N^{\alpha+1}$ is when T is just enough to learn the first N skills as the emergent and saturation times of the N^{th} skill is proportional to $T^{\alpha+1}$ (Eq. (17)). Plugging in the relationship between T and N for optimal C in Eq. (34), we get

$$\mathcal{L} \propto C^{-\alpha/(\alpha+2)}. \tag{35}$$

In Fig. 5, we plot the loss as the function of C for various N and observe that optimal C follows the scaling law (Eq. (35)). The derivation for Eq. (35) from Eq. (34) is provided in Appendix D.4, but the justification of Eq. (34) requires a more formal approach, which is given in Appendix J (for relevant theorems, see Theorems 1 to 3; for the summary, see Corollaries 3 and 4).

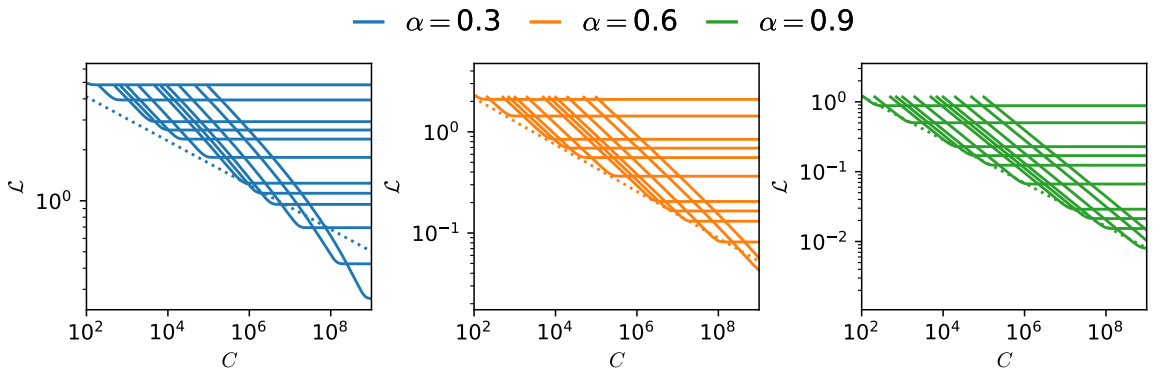


Figure 5: **Scaling law for optimal compute.** The solid lines are the learning curves of the multilinear model as a function of compute $C = T \times N$ with varying parameters N from 10^1 (top plateau) to 10^4 (bottom plateau). A tradeoff between N and T exists for fixed C: smaller N trains faster but plateaus after learning all N skills while larger N can achieve smaller loss at a slower pace. The dotted lines show the optimal loss for given compute C, which follows a power law with exponent $-\alpha/(\alpha+2)$. Note that the solid lines plateau after intersecting with the dotted lines, indicating that the optimal allocation of T and N for C is to have T just large enough to fit N skills. For the discussion on $\alpha = 0.3$ where the optimal C for the model decays faster than the power-law, see Appendix D.2.

6 Predicting Emergence

In the previous section, we have shown that our simplified model satisfies scaling laws for time, data points, and parameters, which match the exponents observed in Michaud et al. [17]. In deriving the scaling law, we used emergence or an abrupt change in $\mathcal{L}_k$: *(i)* decoupled sigmoidal saturation (Eq. (25)) for the time scaling law, *(ii)* one-shot learning (Eq. (27)) for data scaling law, and *(iii)* the one-to-one relationship between basis functions and skills for the parameter scaling law (Eq. (32)). In this section, we analyze the emergence of a 2-layer fully connected NN (Section 2) and discuss to what degree the emergence in NNs can be described with our model.

Because NNs lack the built-in g_k, we extend our model to approximate the emergence in NNs. The extended models will keep their multilinearity and decoupling among the skills, resulting in the same scaling laws, but will require an extra parameter that must be calibrated (fit). We calibrate our model from an NN trained on one skill ($n_s = 1$) and use it to predict the emergence of all skills for the $n_s = 5$ setup (Fig. 6).

6.1 Time emergence

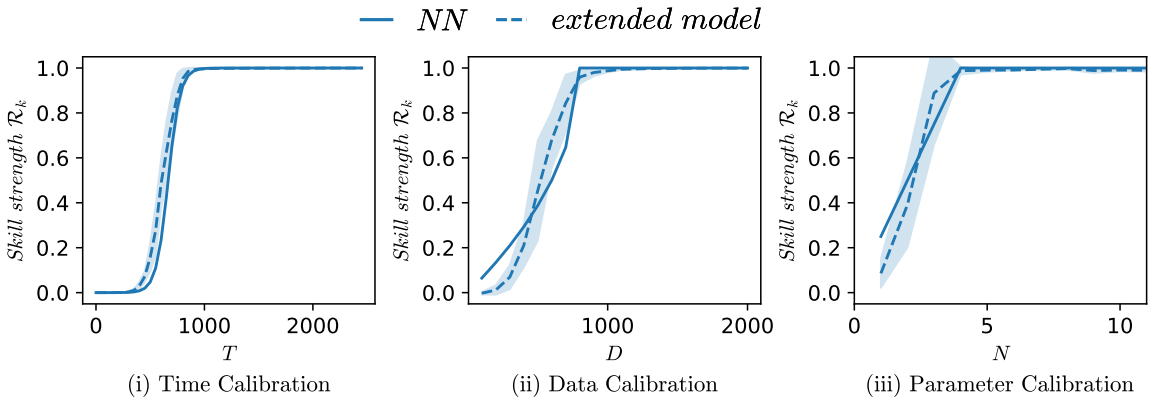
In our multilinear model, the orthonormal basis g_k results in decoupled dynamics for each skill (Eq. (15)), and the layerwise structure – the product of parameters $a_k b_k$ – leads to abrupt saturation of each skill strength (Eq. (16)). NNs share the layerwise structure but lack a fixed orthonormal basis g_k and thus the decoupled dynamics for each skill; NNs must 'discover' g_k (up to a scaling factor) during training before/while saturating each skill (i.e., $\mathcal{R}_k/S \approx 1$).

Extended model. To address this difference in scaling between our model and NNs, we extend the model by multiplying our basis g_k by a constant $\mathcal{B} > 0$:

$$f_T(i,x;a,b) = \sum_{k=1}^{N} a_k(T) b_k(T) \mathcal{B} g_k(i,x), \qquad \mathcal{B} > 0. \tag{36}$$

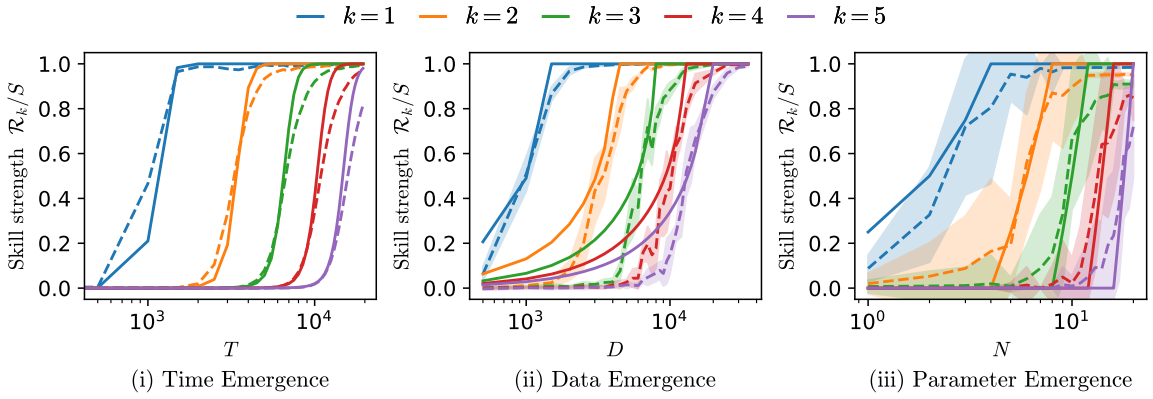
The calibration constant $0 < \mathcal{B} < 1$ adjusts the slower dynamimcs resulting from NN 'discovering' (feature-learning) g_k, which rescales the dynamics in T:

$$\frac{\mathcal{R}_k(T)}{S} = \frac{1}{1 - \left(\frac{S}{\mathcal{R}_k(0)} - 1\right) e^{-2\eta \mathcal{P}_s (I=k) \mathcal{B}^2 S T}}. \tag{37}$$



(i) Time Calibration (ii) Data Calibration (iii) Parameter Calibration

(a) Calibration on the first skill



(i) Time Emergence (ii) Data Emergence (iii) Parameter Emergence

(b) Emergence prediction for $n_s = 5$ system (Fig. 1 repeated)

Figure 6: **Calibration and prediction on emergence. (a):** The calibration of the extended multilinear model (solid) on the 2-layer NN (dashed) for $n_s = 1$ system. For the calibrated parameters, we have $\mathcal{B}^2 = 1/22$ for time (Eq. (37)), $D_c = 800$ for data (Eq. (40)), and $N_c = 4$ for parameters (Eq. (44)). **(b):** The emergence prediction of the multilinear model (solid) compared to the emergence of 2-layer NN (dashed) for $n_s = 5$ system (Fig. 1 repeated). For the NN, we repeated the experiments 50 times from random initialization to calculate the standard deviation.

In Fig. 6(a,i), we observe that the extended model with $\mathcal{B}^2 = 1/22$ fits the NN trained on one skill (i.e., $n_s = 1$). In Fig. 6(b,i), we observe that the extended model with $\mathcal{B}^2 = 1/22$ accurately predicts each skill's time of emergence and reasonably well the time of saturation, suggesting that emergence in NNs can be described by gradient descent dynamics of a simple layerwise model.

6.2 Data point emergence

In Section 5, we derived the data scaling law with the model's single-shot learning ability. One-shot learning is only possible because the model has g_ks as an orthogonal basis. NNs, without g_ks as a fixed basis, must 'discover' g_k, which requires multiple samples from the k^{th} skill.

Extended model. To make our model a D_c-shot learner, we extend it by replacing g_k with the $e_{k,l}$ basis:

$$f_T(i, x; a, B) = \sum_{k=1}^{N} a_k(T) \sum_{l=1}^{D_c} B_{k,l}(T) e_{k,l}(i, x), \tag{38}$$

where the matrix $B \in \mathbb{R}^{N \times D_c}$ is an extension of $b \in \mathbb{R}^N$ in Eq. (12), D_c is a fixed scalar, and $e_{k,l}(i, x) : \{0, 1\}^{n_s+n_b} \to \mathbb{R}$ are functions with the following properties:

$$\mathbf{E}_{X|I=k} \left[e_{k,l} e_{k,l'} \right] = \delta_{ll'}, \qquad e_{k,l}(I \neq k, x) = 0, \quad \sum_{l=1}^{D_c} \frac{1}{\sqrt{D_c}} e_{k,l} = g_k. \tag{39}$$

The first property states that e_ks, when $I = k$, are orthonormal in X. The second property asserts that, similar to g_k (Eq. (2)), $e_{k,l}$ is non-zero only when $I = k$, and fitting of the k^{th} skill only occurs among $e_{k,l}$s: the skills are still decoupled. The third property states that g_k can be expressed using $e_{k,l}$.

Because the extended model decouples the learning of each skill, we can express $\mathcal{L}_k$ with d_k, analogous to Eq. (27). For the k^{th} skill, the extended model overfits when there are fewer observations (d_k) compared to the dimension of the $e_{k,l}$ basis (D_c), and fits g_k when $d_k \geq D_c$: thus our model is a D_c shot learner.

D_c shot learner. *If we initialize the extended model in Eq. (38) with sufficiently small initialization and if Eqs. (39) are satisfied, then the skill strength after training ($T \to \infty$) on D datapoints is*

$$\mathcal{R}_k(\infty) = \begin{cases} S\left(1 - \sqrt{1 - d_k/D_c}\right) & : d_k < D_c \\ S & : d_k \geq D_c. \end{cases} \tag{40}$$

The number d_k is the number of samples in the training set for the k^{th} skill (i.e. datapoints with $g_k(i, x) \neq 0$).

Proof See Appendix E.3. ■

Using Eq. (40), we can calculate the emergence of $\mathcal{R}_k/S$ as a function of D. Note that Eq. (40) reduces to Eq. (27) when $D_c = 1$ and is similar to the model in [17] in that, to learn a skill, the model requires a certain number of samples from the skill.

The derivation of Eq. (40) follows trivially from the dynamics of the extended model (Eq. (16)) and well-known results in linear/kernel regression [26, 34–37]. To be more specific, the model finds the minimum norm solution as if we performed ridgeless regression on g_k with basis functions $[e_{k,1}, \cdots e_{k,D_c}]$. See Appendix E.3 for details.

In Fig. 6(a, ii), we observe that the NN saturates at $D \approx 800$ (solid line) for $n_s = 1$ set up. In Fig. 6(b, ii), we observe that our extended model with $D_c = 800$ approximates the data emergence for the consecutive skills, suggesting that the NN discovers g_k when it observes D_c samples from the k^{th} skill.

6.3 Parameter emergence

Since our model has g_ks as basis functions, adding a g_N (2 parameters) results in learning the N^{th} skill (Eq. (32)). A 2-layer NN with N parameters – the width of the hidden layer (number of nodes)

– cannot express g_N with a single node (hidden neuron); it requires multiple hidden nodes to express a single skill (Fig. 6(a,iii)). For this experiment, Adam [38] was used, instead of SGD, to increase the chance of escaping the near-flat saddle points induced by an insufficient number of parameters. [5]

Extended model. To compensate for the need for multiple nodes in expressing one skill, we extend our model similarly to Eq. (38). Because the number of parameters is now a bottleneck, we ensure the model has N basis functions ($e_{k,l}$s).

$$f_T(i,x;a,B) = \sum_{k=1}^{q-1}\sum_{l=1}^{N_c} a_k(T)B_{k,l}(T)e_{k,l}(i,x) + \sum_{l'=1}^{r} a_q(T)B_{q,l'}(T)e_{q,l'}(i,x), \tag{41}$$

where N_c is the number of basis functions needed to express a skill, quotient q is $\lfloor (N-1)/N_c \rfloor + 1$,[6] and remainder r is such that $(q-1)N_c + r = N$. In short, the N basis functions are

$$[e_{1,1},\ \cdots,\ e_{1,N_c},\ e_{2,1},\ \cdots,\ e_{q,r}]. \tag{42}$$

Similar to Eq. (39), the basis functions satisfy the following properties

$$\mathbf{E}_{X|I=k}\left[e_{k,l}e_{k,l'}\right] = \delta_{ll'}, \quad e_{k,l}(I \neq k, x) = 0, \quad \sum_{l=1}^{N_c} \frac{1}{\sqrt{N_c}} e_{k,l} = g_k. \tag{43}$$

N_c basis functions for a skill. *For the extended model in Eq. (41), the skill strength at $T, D \to \infty$ for a given N becomes*

$$\mathcal{R}_k(\infty) = \begin{cases} 0 & : k > q \\ S\frac{r}{N_c} & : k = q \\ S & : k < q\,. \end{cases} \tag{44}$$

Proof See Appendix E.4. ■

We can derive Eq. (44) because the basis functions $[e_{k,1}, \cdots, e_{k,N_c}]$ for $k < q$ can express g_k (Eq. (43)) but $[e_{q,1}, \cdots, e_{q,r}]$ cannot express g_q when $r < N_c$.

In Fig. 6(a, iii), the extended model (Eq. (41)) with $N_c = 4$ fits the $n_s = 1$ set up where N is the hidden-layer width of the NN. The $N_c = 4$ model leads to good prediction for the $n_s = 5$ setup in Fig. 6(b, iii). The results suggest that an NN, while lacking the ordering of basis functions (Eq. (42)), prefers to use the hidden neuron in fitting more frequent skills. The 'preference' toward frequent skills is in agreement with Fig. 6(b,i) where NN learns more frequent skills first.

6.4 Limitations of the multilinear model

Our extended multilinear model, with the decoupled dynamics for each skill, predicts the time, data, and parameter emergence with a single calibration. However, the dynamics of a simple model with strong assumptions such as in-built g_k differ from the more complex dynamics of NNs that lack such assumptions.

Time emergence. We note that the NN and the multilinear model emerge at similar instances, but the NN takes longer to saturate fully. This is because, for a given skill, the dynamics of the NN is not one sigmoidal saturation but a sum of **multiple** sigmoidal dynamics with different saturation times. To express the parity function, the NN must use multiple hidden neurons, and the skill strength can be divided into the skill strength from each neuron[7] whose dynamics follow a sigmoidal saturation. Because of the non-linearity and the function it expresses, each neuron is updated at different rates, and the slowly saturating neurons result in a longer tail in comparison to our multilinear model. For an example, see Fig. 9 in Appendix F.

[5] We are free to use any optimizer as long as it preserves the stage-like training or the order in which the skills are learned.

[6] $\lfloor a/b \rfloor$ is the quotient or $\text{int}(a/b)$.

[7] This is possible because of the linearity of the last layer. See Appendix F for the definition of skill strength per neuron.

Data point emergence. Our extended model (Eq. (40)) deviates from the NN when $d_k \ll D_c$: NN shows a more abrupt change in $\mathcal{R}_k$ as a function of D. This is because our model asserts strict decoupling among the skills: even a few d_k will contribute to learning g_k from $e_{k,l}$. This differs from the NN, which lacks strict decoupling among the samples from different skills. We speculate that because NNs can perform benign [39] or tempered [40] overfitting, they treat a few data points from less frequent skills as 'noise' from more frequent skills: requiring more samples to learn the infrequent skills.

Parameter emergence. Note that Fig. 6(b, iii) has high variance compared to other emergence plots in Fig. 6(b); this is because NN sparsely, over many repeated trials, use the hidden neurons to learn less frequent skills over more frequent ones (See Fig. 11 in Appendix H for an example of such outliers). Because the 'preference' of NNs toward more frequent skills is not as strict as in our model, we speculate that initial conditions (ones that ease the learning of less frequent skills) play a role in creating outliers.

7 Discussion and Conclusion

This paper established an explicit setting to investigate emergence by representing skills as orthogonal functions. We proposed a tractable multilinear model—with the skill functions g_k as the basis—that shows emergence and scaling laws. We additionally extended the model to predict emergence in a 2-layer NN.

In our multilinear model, the skill functions (as basis functions) decouple the loss, so each skill loss evolves independently of the others (Eq. (15)). As a consequence, for time, skill learning dynamics are decoupled (Eq. (16)); for data, they depend only on that specific skill's observation (Eq. (27)); for the parameter, learning of the N^{th} skill depends only on g_N (Eq. (32)); these are the main properties we use to derive the scaling laws.

Given the importance of g_k in our model, especially in decoupling the skills, it is puzzling why our model can describe emergence in NNs, which lack the g_ks as the basis. NNs, with the ability to feature-learn [41, 42] (for recent studies on feature learning, see e.g. [43–48]), will eventually fit the target function, but it need not learn each g_k in a decoupled manner. For example, an NN is free to learn any linear combination of g_k in any order independent of the skill frequency. We speculate that the layerwise structure – which leads to a stage-like training, along with the significant differences in skill frequency (Zipfian), prompts the NN to learn the most frequent g_k with limited resources (time, data, or parameters): an effective decoupling of skills.

We can interpret the skill functions as features – the functions useful in describing the target function [41] – and the multilinear model as a 'feature-learned' model already equipped with the g_ks. In this interpretation, the resemblance between emergence in NNs and the multilinear model (Section 6) may inform us about the relationship between feature learning and emergence [6]; the feature learning of the k^{th} skill – learning to express g_k – may have occurred well before $\mathcal{R}_k$ saturates (and $\mathcal{L}_k$ decreases). Investigating the role of features (and feature learning) in emergence is left for future work.

While much work remains to understand LLMs, it is encouraging that a simple model can replicate emergence and that an NN, by its layerwise structure, allows decoupled approximations. Our proposal sheds light on the relationship between model scale and performance and lays the foundation for further theoretical investigations into the emergence of complex skills in large-scale models.

We, similar to many previous investigations (see e.g., [17, 19]), used a simple data structure with a power-law distribution. This begs the question of how well these results hold for more realistic datasets. In future research, we plan to explore setups involving natural language processing tasks to establish a tighter connection between 'skills' exhibited by LLMs and our theoretical framework. One approach is to design a task where the skill functions g_k are directly relevant to language tasks, such as translation with Zipfian word frequencies. This extension is natural since our current work only utilizes the orthogonality and frequency of g_k, without relying on the fact that they are parity functions. By validating our theoretical findings in language tasks using transformer-like models, we aim to contribute to a broader understanding of how neural networks acquire and exhibit complex behaviors.

8 Acknowledgements

NF acknowledges the UKRI support through the Horizon Europe guarantee Marie Skłodowska-Curie grant (EP/X036820/1). SL was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No.2020R1A5A1016126). We thank Charles London, Zohar Ringel, and Shuofeng Zhang for their helpful comments.

References

[1] 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.

[2] Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, et al. Predictability and surprise in large generative models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1747–1764, 2022.

[3] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint:2206.04615*, 2022.

[4] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint: 2206.07682*, 2022.

[5] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems*, 36, 2023.

[6] Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, Benjamin L. Edelman, Zhaowei Zhang, Mario Günther, Anton Korinek, Jose Hernandez-Orallo, Lewis Hammond, Eric Bigelow, Alexander Pan, Lauro Langosco, Tomasz Korbak, Heidi Zhang, Ruiqi Zhong, Seán Ó hÉigeartaigh, Gabriel Recchia, Giulio Corsi, Alan Chan, Markus Anderljung, Lilian Edwards, Yoshua Bengio, Danqi Chen, Samuel Albanie, Tegan Maharaj, Jakob Foerster, Florian Tramer, He He, Atoosa Kasirzadeh, Yejin Choi, and David Krueger. Foundational challenges in assuring alignment and safety of large language models. *arXiv preprint: 2404.09932*, 2024.

[7] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[8] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint:1712.00409*, 2017.

[9] 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:2001.08361*, 2020.

[10] Jonathan S Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. *arXiv preprint: 1909.12673*, 2019.

[11] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint:2010.14701*, 2020.

[12] Mitchell A Gordon, Kevin Duh, and Jared Kaplan. Data and parameter scaling laws for neural machine translation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5915–5922, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[13] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.

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

[15] Gregor Bachmann, Sotiris Anagnostidis, and Thomas Hofmann. Scaling mlps: A tale of inductive bias. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

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

[18] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *Proceedings of the International Conference on Learning Representations 2014*, 2014. arXiv:1312.6120.

[19] Marcus Hutter. Learning curve theory. *arXiv preprint:2102.04074*, 2021.

[20] Utkarsh Sharma and Jared Kaplan. Scaling laws from the data manifold dimension. *Journal of Machine Learning Research*, 23(9):1–34, 2022. arXiv:2004.10802.

[21] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *arXiv preprint:2102.06701*, 2021.

[22] Alexander Maloney, Daniel A Roberts, and James Sully. A solvable model of neural scaling laws. *arXiv preprint:2210.16859*, 2022.

[23] Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. A dynamical model of neural scaling laws. *arXiv preprint:2402.01092*, 2024.

[24] Stefano Spigler, Mario Geiger, and Matthieu Wyart. Asymptotic learning curves of kernel methods: empirical data versus teacher–student paradigm. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124001, 2020.

[25] Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *International Conference on Machine Learning*, pages 1024–1034. PMLR, 2020.

[26] Hugo Cui, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová. Generalization error rates in kernel regression: The crossover from the noiseless to noisy regime. *Advances in Neural Information Processing Systems*, 34:10131–10143, 2021.

[27] Boaz Barak. Windows on theory blog: Emergent abilities and grokking: Fundamental, mirage, or both? *https://windowsontheory.org/2023/12/22/emergent-abilities-and-grokking-fundamental-mirage-or-both/*, 2023.

[28] Sanjeev Arora and Anirudh Goyal. A theory for emergence of complex skills in language models. *arXiv preprint:2307.15936*, 2023.

[29] Maya Okawa, Ekdeep S Lubana, Robert Dick, and Hidenori Tanaka. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task. *Advances in Neural Information Processing Systems*, 36, 2024.

[30] Dingli Yu, Simran Kaur, Arushi Gupta, Jonah Brown-Cohen, Anirudh Goyal, and Sanjeev Arora. Skill-mix: A flexible and expandable family of evaluations for ai models. *arXiv preprint:2310.17567*, 2023.

[31] Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! a data-driven skills framework for understanding and training language models. *Advances in Neural Information Processing Systems*, 36, 2023.

[32] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv:2201.02177*, 2022.

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

[34] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature communications*, 12(1):2914, 2021.

[35] Arthur Jacot, Berfin Simsek, Francesco Spadaro, Clément Hongler, and Franck Gabriel. Kernel alignment risk estimator: Risk prediction from training data. *Advances in Neural Information Processing Systems*, 33:15568–15578, 2020.

[36] Ouns El Harzli, Bernardo Cuenca Grau, Guillermo Valle-Pérez, and Ard A Louis. Double-descent curves in neural networks: a new perspective using gaussian processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11856–11864, 2024.

[37] James B Simon, Madeline Dickens, and Michael R DeWeese. A theory of the inductive bias and generalization of kernel regression and wide neural networks. *arXiv e-prints*, pages arXiv–2110, 2021.

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint:1412.6980*, 2014.

[39] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[40] Neil Mallinar, James Simon, Amirhesam Abedsoltan, Parthe Pandit, Misha Belkin, and Preetum Nakkiran. Benign, tempered, or catastrophic: Toward a refined taxonomy of overfitting. *Advances in Neural Information Processing Systems*, 35:1182–1195, 2022.

[41] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[42] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[43] Greg Yang and Edward J Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pages 11727–11737. PMLR, 2021.

[44] Alexander Atanasov, Blake Bordelon, and Cengiz Pehlevan. Neural networks as kernel learners: The silent alignment effect. *arXiv preprint arXiv:2111.00034*, 2021.

[45] Arthur Jacot, Eugene Golikov, Clément Hongler, and Franck Gabriel. Feature learning in l_2-regularized dnns: Attraction/repulsion and sparsity. *Advances in Neural Information Processing Systems*, 35:6763–6774, 2022.

[46] Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *Advances in Neural Information Processing Systems*, 35:32240–32256, 2022.

[47] Inbar Seroussi, Gadi Naveh, and Zohar Ringel. Separation of scales and a thermodynamic description of feature learning in some cnns. *Nature Communications*, 14(1):908, 2023.

[48] Hugo Cui, Luca Pesce, Yatin Dandi, Florent Krzakala, Yue M Lu, Lenka Zdeborová, and Bruno Loureiro. Asymptotics of feature learning in two-layer networks after one gradient-step. *arXiv preprint arXiv:2402.04980*, 2024.

[49] Irina Gennad'evna Shevtsova. Sharpening of the upper bound of the absolute constant in the berry–esseen inequality. *Theory of Probability and Its Applications*, 51(3):549–553, 2007.

[50] Hugh L Montgomery and Robert C Vaughan. *Multiplicative Number Theory I: Classical Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2007.

A Glossary

A	Normalization constant for $\mathcal{P}_s$ such that $\mathcal{P}_s(k) = Ak^{-(\alpha+1)}$
T	Time or step
D	Number of data points
N	Number of parameters (skill basis functions in the model)
C	The computation cost $T \times N$
n_s	The number of skills in the multitask sparse parity problem
I	Random variable of the control bits
X	Random variable of the skill bits
$\mathcal{P}_s$	Probability of skills (control bits)
$\mathcal{P}_b$	Probability of skill bits
S	The target scale or the norm of the target function
$\mathcal{R}_k$	Skill strength of the k^{th} skill (Eq. (9))
$\mathcal{L}$	Total (generalization) loss
$\mathcal{L}^{(D)}$	Empirical loss for D samples
$\mathcal{L}_k$	Skill loss of the k^{th} skill (Eq. (10))
d_k	Number of observation of the k^{th} skill (i.e. number of training points (i, x) with $g_k(i, x) \neq 0$)
f^*	Target function $f^* : \{0, 1\}^{n_s+n_b} \to \{-S, S\}$ (Eq. (4))
g_k	The k^{th} skill basis function $g_k : \{0, 1\}^{n_s+n_b} \to \{-1, 0, 1\}$ (Eq. (2))

Control bits	Skill bits	y
10000000000	1100010000010 10	1
01000000000	0101001000010 00	0
00100000000	0011010101101 01	1
⋮	⋮	⋮
00000000001	1000100010011 00	1

Table 3: Representation of the multitask sparse parity as presented in [17]. The control bits are one-hot vectors encoding a specific parity task. The frequency of the different tasks follows a Zipfian distribution. In this example, there are $n_s = 10$ tasks, and skill bits are length $n_b = 15$. The y column is the resulting parity computed from $m = 3$ bits (highlighted in colors). The multitask dataset provides a controlled experimental setting designed to investigate skills.

B Background

In the following, we describe the multitask sparse parity dataset as in [17], and the nonlinear dynamics of two-layer linear networks as in [18].

B.1 Multitask sparse parity

The sparse parity task can be stated as follows: for a bit string of length n_b, the goal is to determine the parity (sum mod 2) of a predetermined subset of m bits within that string. The **multitask** sparse parity [17] extends this problem by introducing n_s unique sparse parity variants in the dataset. The input bit strings have a length of n_s+n_b. The first n_s bits function as indicators by assigning a specific task. The frequency of the distinct parity tasks follows a rank-frequency distribution with an inverse power law relation (Zipfian distribution). The last n_b bits are uniformly distributed. This sets a binary classification problem $\{0,1\}^{n_s+n_b} \to \{0,1\}$ where only a single bit of the initial n_s bits is nonzero. In Table 3, the many distinct parity tasks represent different skills. [8]

The proposal in [17] aims to reconcile the regularity of scaling laws with the emergence of abilities with scale using three key hypotheses: (i) skills, represented as a finite set of computations, are distinct and separate; (ii) these skills differ in their effectiveness, leading to a ranking based on their utility to reduce the loss; and (iii) the pattern of how frequently these skills are used in prediction follows a Zipfian power-law distribution. Interestingly, the multitask problem has a consistent pattern across scaling curves: each parity displays a distinct transition, characterized by a sharp decrease in loss at a specific scale of parameters, data, or training step. Such a sudden shift occurs after an initial phase of no noticeable improvement, leading to reverse sigmoid-shaped learning curves. Michaud et al. [17] empirically show that for a one-hidden-layer neural network with ReLU activation, trained using cross-entropy loss and the Adam optimizer, these transitions happen at different scales for distinct tasks. This results in a smooth decrease in the overall loss as the number of skill levels increases.

B.2 Nonlinear dynamics of linear neural network

Saxe et al. [18] have solved the exact dynamics for two-layer linear neural networks with gradient descent[9] under MSE loss (Fig. 7(a)). The dynamics decompose into independent modes that show sigmoidal growth at different timescales (Fig. 7(c)). The setup assumes orthogonal input features $X \in \mathbb{R}^{d_1}$ and input-output correlation matrix $\Sigma \in \mathbb{R}^{d_1 \times d_3}$ for target output $f^*(X) \in \mathbb{R}^{d_3}$:

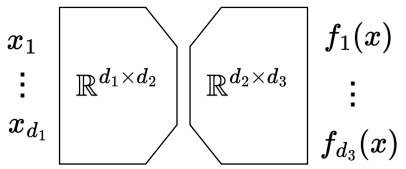
$$\mathbf{E}_X \left[X_i X_j\right] = \delta_{ij}, \qquad \Sigma = \mathbf{E}_X \left[X f^{*T}(X)\right] \tag{45}$$

By performing SVD (singular value decomposition) on $\Sigma = U\Lambda V$, the target function $f^* : \mathbb{R}^{d_1} \to \mathbb{R}^{d_3}$ becomes:

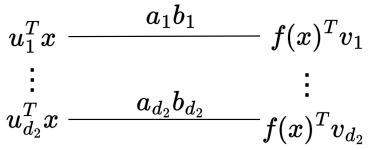
$$f^*(x) = \sum_{k=1}^{d_2} v_k \lambda_k u_k^T x, \qquad U\Lambda V = \mathbf{E}_X \left[X f^*(X)^T\right] \tag{46}$$

[8] Note that here we follow the even/odd parity convention used in [17], i.e., $\{0,1\}$, instead of $\{1,-1\}$ as used in the main text.

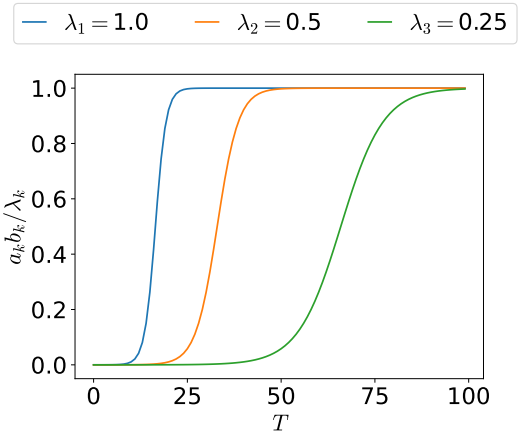
[9] To be specific, it is under gradient flow or the continuous limit of full batch gradient descent.



(a) Linear neural network



(b) Independent modes



(c) Dynamics of modes

Figure 7: **Nonlinear dynamics of linear neural networks.** (a) A two-layer undercomplete neural network, which is a multiplication of two matrices, where $d_2 < d_1$ and $d_2 < d_3$. (b) The d_2 independent modes of dynamics for linear neural network (Eq. (47)). The product of parameters $a_k b_k$ are learnable parameters and vectors u_k, v_k are obtained from SVD of the input-output correlation matrix Σ (Eq. (45)). (c) The temporal evolution of $a_k b_k$ under gradient descent, which follows a sigmoidal growth (Eq. (48)). Note that smaller λ_k – the singular value of Σ – results in delayed saturation of $a_k b_k$.

where $u_k \in \mathbb{R}^{d_1}$,$v_k \in \mathbb{R}^{d_3}$ are the row vectors of U, V and $\lambda_k \in \mathbb{R}$ are the singular values of Λ of the input-output correlation matrix Σ.

The dynamics of a two-layer undercomplete linear neural network (the width of the hidden layer is smaller than the width of the input and output) equals that of the following model:

$$v_k^T f(x; a, b) = a_k b_k u_k^T x \qquad k \in \{1, 2, \cdots, d_2\}. \tag{47}$$

where $a_k, b_k \in \mathbb{R}$ are the parameters. Note that Eq. (47) are d_2 decoupled functions $v_k^T f(x) : \mathbb{R}^{d_1} \to \mathbb{R}$ (Fig. 7(b)). Assuming small and positive initialization ($0 < a_k(0) b_k(0) \ll \lambda_k$), the dynamics of Eq. (47) under gradient descent with learning rate η can be solved analytically; the product of parameters $a_k b_k$ grows sigmoidally with saturation time proportional to λ_k^{-1} (Fig. 7(c)):

$$\frac{a_k(T) b_k(T)}{\lambda_k} = \frac{1}{1 + \left(\frac{\lambda_k}{a_i(0) b_i(0)} - 1 \right) e^{-2\eta \lambda_k t}}. \tag{48}$$

As this is a linear neural network, the sigmoid growth is the result of the nonlinear dynamics of learning (non-linear activations are not present). Saxe et al. [18] demonstrate that multilinear models show delayed sigmoidal growth of different modes and show that Eq. (48) agrees with the empirical dynamics of a two-layer neural network with tanh activations.

C Derivation of the multilinear model

In this section, we provide derivation for our multilinear model. The two corollaries for data and parameters (Corollaries 1 and 2) follow from the decoupled dynamics of NN (Lemma 1).

C.1 Decoupled dynamics of the multilinear model

Lemma 1. *Let the multilinear model Eq. (12) be trained with gradient on D i.i.d samples for the setup in Section 3 (input distribution: Eq. (1), target function: Eq. (4), and MSE loss: Eq. (5)). Let $k \leq N$ be a skill index in the multilinear model and the input distribution ($k \leq n_s$). Then assuming the following initialization $a_k(0) = b_k(0)$ and $0 < a_k(0)b_k(0) < S$, the dynamics of the k^{th} skill strength ($\mathcal{R}_k$) is*

$$\mathcal{R}_k(T) = \frac{S}{1 + \left(\frac{S}{\mathcal{R}_k(0)} - 1\right) e^{-2\eta S \frac{d_k}{D} T}} \tag{49}$$

and the skill loss is

$$\mathcal{L}_k(T) = \frac{S^2}{2\left(1 + \left(\frac{S}{\mathcal{R}_k(0)} - 1\right)^{-1} e^{2\eta S \frac{d_k}{D} T}\right)^2}. \tag{50}$$

where η is the learning rate and d_k is the number of observations with $g_k(I = k, x^{(j_k)}) \neq 0$.

Proof For $j = 1, \cdots, D$, denote $(i^{(j)}, x^{(j)})$ be the j^{th} data point in the training set. Then the empirical loss for D datapoints is given as

$$\mathcal{L}^{(D)} = \frac{1}{2D} \sum_{j=1}^{D} \left(f^*(i^{(j)}, x^{(j)}) - f(i^{(j)}, x^{(j)})\right)^2. \tag{51}$$

We note that

$$\begin{aligned}
\left(f^*(i^{(j)}, x^{(j)}) - f(i^{(j)}, x^{(j)})\right)^2 &= \left(\sum_{k=1}^{n_s} (S - a_k b_k) g_k(i^{(j)}, x^{(j)})\right)^2 \\
&= (S - a_{i^{(j)}} b_{i^{(j)}})^2 g_{i^{(j)}}(i^{(j)}, x^{(j)})^2 \\
&= (S - a_{i^{(j)}} b_{i^{(j)}})^2,
\end{aligned}$$

as $g_i(i, j) \in \{1, -1\}$ and $g_k(i, j) = 0$ for $i \neq k$. So if we denote d_k the number of data points with $i^{(j)} = k$, then we can conclude

$$\mathcal{L}^{(D)} = \frac{1}{2D} \sum_{j=1}^{D} (S - a_{i^{(j)}} b_{i^{(j)}})^2 = \frac{1}{2D} \sum_{k=1}^{n_s} d_k (S - a_k b_k)^2, \tag{52}$$

which is the decoupled loss in the main text (Eq. (15)). Using the gradient descent equation and Eq. (52), we obtain

$$\frac{da_k}{dt} = -\eta \frac{d\mathcal{L}_D}{da_k} \tag{53}$$

$$= -\eta \frac{d_k}{D} b_k (a_k b_k - S). \tag{54}$$

Likewise, we can obtain the equation for b_k as

$$\frac{db_k}{dt} = -\eta \frac{d_k}{D} a_k (a_k b_k - S). \tag{55}$$

Because of symmetry between a and b (See Appendix B.2 or [18]), assuming $a_k(0) = b_k(0)$, and $a_k(0)b_k(0) > 0$ results in $a_k(T) = b_k(T)$ for all T. The equation for $\mathcal{R}_k = a_k b_k$ is

$$\frac{d\mathcal{R}_k}{dt} = -\eta \frac{da_k}{dt} b_k + a_k \frac{db_k}{dt} = -\eta \frac{d_k}{D} (b_k^2 + a_k^2)(a_k b_k - S) \tag{56}$$

$$= -2\eta \frac{d_k}{D} \mathcal{R}_k (\mathcal{R}_k - S). \tag{57}$$

Assuming $a_k(0)b_k(0) < S$, we can solve the differential equation to obtain

$$\mathcal{R}_k(T) = \frac{S}{1 + \left(\frac{S}{\mathcal{R}_k(0)} - 1\right) e^{-2\eta S \frac{d_k}{D} T}}. \tag{58}$$

The equation for $\mathcal{L}_k$ follows from Eq. (14). ■

C.2 One-shot learner

Corollary 1. *For the setup in Lemma 1, the k^{th} skill loss ($\mathcal{L}_k$) at $T, N \to \infty$ is*

$$\mathcal{L}_k(\infty) = \begin{cases} 0 & : d_k > 0 \\ (S - \mathcal{R}_k(0))^2/2 \approx S^2/2 & : d_k = 0, \end{cases} \tag{59}$$

where d_k is the number of k^{th} skill's observation.

Proof The corollary follows directly from Lemma 1. By taking $T, N \to \infty$,

$$\mathcal{R}_k(\infty) = \begin{cases} d_k > 0 : & S \\ d_k = 0 : & \mathcal{R}_k(0) \end{cases} \tag{60}$$

We obtain the result by using the relationship between $\mathcal{R}_k$ and $\mathcal{L}_k$ in Eq. (14). ■

C.3 Equivalence between a basis function and a skill

Corollary 2. *Let the multilinear model Eq. (12) be trained with gradient on D i.i.d samples for the setup in Section 3 (input distribution: Eq. (1), target function: Eq. (4), and MSE loss: Eq. (5)). Assume $a_k(0) = b_k(0)$, $0 < a_k(0)b_k(0) < S$, and that the model has N most frequent skills as basis functions. Then $\mathcal{R}_k$ for the $k^{th} \leq ns$ skill at $T, D \to \infty$ is*

$$\mathcal{L}_k(\infty) = \begin{cases} 0 & : k \leq N \\ S^2/2 & : k > N \end{cases} \tag{61}$$

Proof The corollary follows directly from Lemma 1. By taking $T, D \to \infty$,

$$\mathcal{R}_k(\infty) = \begin{cases} k \leq N : & S \\ k > N : & \mathcal{R}_k(0) \end{cases} \tag{62}$$

We obtain the result by using the relationship between $\mathcal{R}_k$ and $\mathcal{L}_k$ in Eq. (14) and $\mathcal{R}_k(0) \ll S$. ■

D Detailed derivation of the scaling laws

This section provides a detailed derivation of the scaling laws up to a rigor common in physics and engineering. For example, we approximate the Riemann sum as integral or treat k, the number of skills, as a differentiable parameter. For more general and rigorous derivations and discussions, see Appendix J.

D.1 Summary of the scaling laws

Bottleneck	Time	Data	Parameter	Exponent
Time (T)	T	∞	∞	$-\alpha/(\alpha+1)$
Data (D)	∞	D	∞	$-\alpha/(\alpha+1)$
Parameter (N)	∞	∞	N	$-\alpha$
Compute (C)	$C^{(\alpha+1)/(\alpha+2)}$	∞	$C^{1/(\alpha+2)}$	$-\alpha/(\alpha+2)$

D.2 Time scaling law

With the dynamics of $\mathcal{R}_k$ solved as Eq. (16), we can calculate the skill loss $\mathcal{L}_k = (S - \mathcal{R}_k(T))^2/2$ (given as Eq. (14)) as follows:

$$\mathcal{L}_k = \frac{S^2}{2\left(1 + \left(\frac{S}{\mathcal{R}_k(0)} - 1\right)^{-1} e^{2\eta \frac{d_k}{D} ST}\right)^2}. \tag{63}$$

Noting that $d_k/D \to \mathcal{P}_s$ as $D \to \infty$, where $\mathcal{P}_s = Ak^{-(\alpha+1)}$, we have

$$\mathcal{L}_k = \frac{S^2}{2\left(1 + \left(\frac{S}{\mathcal{R}_k(0)} - 1\right)^{-1} e^{2\eta A k^{-(\alpha+1)} ST}\right)^2}. \tag{64}$$

This is a function of $k^{-(\alpha+1)}T$ only, suggesting the **decoupling** of dynamics for each skill. Thus,

$$\frac{d\mathcal{L}_k}{dT} = -\frac{k}{(\alpha+1)T}\frac{d\mathcal{L}_k}{dk}. \tag{65}$$

Using Eq. (8) and taking $N, n_s \to \infty$ at the same rate[10], we can approximate the loss as integral instead of a sum over k:

$$\mathcal{L} \approx \lim_{N\to\infty} \int_1^N Ak^{-(\alpha+1)} \mathcal{L}_k dk, \tag{66}$$

where A is the normalization constant for $\mathcal{P}_s$. We can differentiate the loss and use Eq. (65) to express the equation in terms of k:

$$\frac{d\mathcal{L}}{dT} = \lim_{N\to\infty} \int_1^N Ak^{-(\alpha+1)} \frac{d\mathcal{L}_k}{dT} dk = -\lim_{N\to\infty} \frac{1}{(\alpha+1)T} \int_1^N Ak^{-\alpha} \frac{d\mathcal{L}_k}{dk} dk. \tag{67}$$

Integrating by parts, we obtain

$$\frac{d\mathcal{L}}{dT} = -\lim_{N\to\infty} \frac{1}{(\alpha+1)T} \left[Ak^{-\alpha}\mathcal{L}_k\right]_1^N - \lim_{N\to\infty} \frac{\alpha}{(\alpha+1)T} \int_1^N Ak^{-(\alpha+1)} \mathcal{L}_k dk \tag{68}$$

$$= -\lim_{N\to\infty} \mathcal{O}\left(N^{-\alpha}\frac{1}{T}\right) + \mathcal{O}\left(\frac{1}{Te^T}\right) - \frac{\alpha}{(\alpha+1)T}\mathcal{L}. \tag{69}$$

The first term goes to 0 as $N \to \infty$ and the second term goes to 0 exponentially faster compared to the last term for $T \gg 1$, which leads to the scaling law:

$$\frac{d\mathcal{L}(T)}{\mathcal{L}(T)} = -\frac{\alpha}{\alpha+1}\frac{dT}{T}. \tag{70}$$

Finite N correction for small α. In Fig. 8, we observe that our model with $\alpha = 0.1$ deviates from the expected power law with exponent $-\alpha/(\alpha+1)$. The deviation can be explained by the antiderivative term in Eq. (68):

$$\lim_{N\to\infty} \left[\frac{1}{2(\alpha+1)} \frac{S^2 A}{\left(1 + \frac{1}{S/\mathcal{R}_k(0) - 1} e^{2\eta S A k^{-(\alpha+1)} T}\right)^2} \frac{k^{-\alpha}}{T}\right]_1^N = \lim_{N\to\infty} \left(\mathcal{O}\left(N^{-\alpha}\frac{1}{T}\right) - \mathcal{O}\left(\frac{1}{Te^T}\right)\right). \tag{71}$$

The second term ($k = 1$) goes to 0 faster than $\mathcal{O}(T^{-1})$ for sufficiently larger T but the first term ($k = N$) may not decay fast enough for finite N and sufficiently small α. For example, $N = 50,000$ and $\alpha = 0.1$ leads to $N^{-\alpha} \approx 0.3$, which is not negligibly small.

[10] We take N and n_s to ∞ at the same rate because we do not want the parameters to be a bottleneck in this set up.

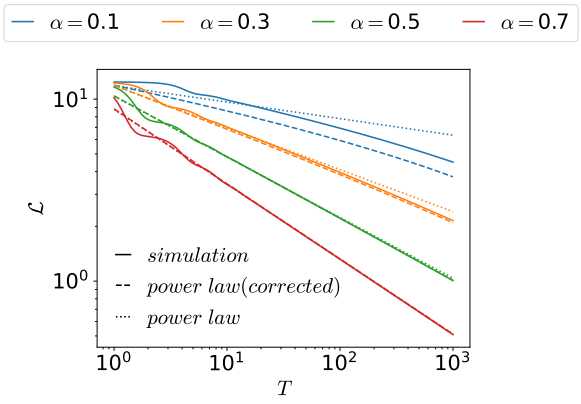


Figure 8: **Scaling law and corrected predictions.** A simulation of our multilinear model with $N = 50,000$ (solid), a scaling law with exponent $-\alpha/(\alpha+1)$ (dotted), and a corrected scaling law considering finite N (dashed, Eq. (72)). The finite N corrected scaling law better predicts the dynamics, especially for smaller α.

Assuming finite N and small α such that the first term in Eq. (71) is non-negligible, we can rewrite Eq. (68) as

$$\frac{d\mathcal{L}}{dT} \approx -\frac{\alpha}{(\alpha+1)}\frac{\mathcal{L}+\mathcal{L}_C}{T}, \qquad \mathcal{L}_C \approx S^2 A N^{-\alpha}/2\alpha, \tag{72}$$

where we assumed a small initialization $S/\mathcal{R}_k(0) \gg 1$ and sufficiently large number of parameters $N^{\alpha+1} \gg T$ to approximate $\mathcal{L}_C$. Because the total loss at initialization is $\mathcal{L}(0) = S^2/2$, $\mathcal{L}_C$ is non-negligible compared to the loss for sufficiently small α. Thus considering $\mathcal{L}_C$, we obtain the corrected power-law which better approximates the time scaling law (dashed lines in Fig. 8). For a rigorous and comprehensive analysis of the time scaling law, see Theorem 2 and Theorem 3 in Appendix J.

D.3 Data scaling law

In this section, we show the data scaling law in more detail. Using the one-shot learning property (Eq. (27)), the probability of observing the k^{th} skill (Eq. (28)), and the decomposition of the loss into skill losses (Eq. (7)), the expected loss for D datapoints is

$$\mathbf{E}_D\left[\mathcal{L}\right] = \sum_{k=1}^{\infty} S^2 \mathcal{P}_s(I=k)(1 - P_{observed}(k)) \tag{73}$$

$$= S^2 A \sum_{k=1}^{\infty} k^{-(\alpha+1)} \left(1 - \mathcal{P}_s(I=k)\right)^D \tag{74}$$

$$\approx S^2 A \int_1^{\infty} k^{-(\alpha+1)} \left(1 - Ak^{-(\alpha+1)}\right)^D dk, \tag{75}$$

where A is the normalization constant such that $\mathcal{P}(I=k) = Ak^{-(\alpha+1)}$. The difference in the loss $\Delta\mathcal{L} = \mathbf{E}_{D+1}\left[\mathcal{L}\right] - \mathbf{E}_D\left[\mathcal{L}\right]$ is

$$\Delta\mathcal{L} = S^2 A \int_1^{\infty} k^{-(\alpha+1)} \left(1 - Ak^{-(\alpha+1)}\right)^D \left(\left(1 - Ak^{-(\alpha+1)}\right) - 1\right) dk \tag{76}$$

$$= -S^2 A^2 \int_1^{\infty} k^{-2(\alpha+1)} \left(1 - Ak^{-(\alpha+1)}\right)^D dk. \tag{77}$$

We can integrate $\Delta\mathcal{L}$ by parts.

$$\begin{aligned}\Delta\mathcal{L} &= \left[-\frac{S^2Ak^{-\alpha}}{(\alpha+1)(D+1)}\left(1-Ak^{-(\alpha+1)}\right)^{D+1}\right]_1^\infty \\ &\quad - \frac{S^2A\alpha}{(\alpha+1)(D+1)}\int_1^\infty k^{-(\alpha+1)}\left(1-Ak^{-(\alpha+1)}\right)^{D+1}dk \\ &\approx \mathcal{O}\left((1-\mathcal{P}_s(1))^{D+1}\right) - \frac{S^2A\alpha}{(\alpha+1)(D+1)}\int_1^\infty k^{-(\alpha+1)}\left(1-Ak^{-(\alpha+1)}\right)^{D}\left(1-Ak^{-(\alpha+1)}\right)dk \\ &\approx -\frac{\alpha}{(\alpha+1)(D+1)}\mathbf{E}_D\left[\mathcal{L}\right] + \frac{\alpha}{(\alpha+1)(D+1)}\Delta\mathcal{L}.\end{aligned}$$

In the second line, the first term goes to 0 for $D \gg 1$. In the last line, we used the expression for $\Delta\mathcal{L}$ (Eq. (76)) and $\mathbf{E}_D\left[\mathcal{L}\right]$ (Eq. (73)). Rearranging the equation above and using that $D \gg 1$, we obtain

$$\frac{\Delta\mathcal{L}}{\mathbf{E}_D\left[\mathcal{L}\right]} = -\frac{\alpha}{1+(\alpha+1)D} \approx -\frac{\alpha}{(\alpha+1)}\frac{1}{D} \tag{78}$$

$$= -\frac{\alpha}{(\alpha+1)}\frac{\Delta D}{D}. \tag{79}$$

where in the last line, $\Delta D/D = 1/D$ as the change in the number of data points relative to D is one.

D.4 Optimal compute scaling law

We start from Eq. (24) with $D \to \infty$

$$\mathcal{L} \approx \int_1^N Ak^{-(\alpha+1)}\mathcal{L}_k dk + \lim_{n_s\to\infty}\frac{S^2}{2}\int_N^{n_s} Ak^{-(\alpha+1)}dk. \tag{80}$$

We can use Eq. (72) to calculate the first term and integrate the last term to get

$$\mathcal{L} \approx (\mathcal{L}(0)+\mathcal{L}_C)T^{-\alpha/(\alpha+1)} - \mathcal{L}_c + \frac{S^2A}{2\alpha}N^{-\alpha} \tag{81}$$

$$\approx \mathcal{O}(T^{-\alpha/(\alpha+1)}) + \mathcal{O}(N^{-\alpha}). \tag{82}$$

where we used that $\mathcal{L}(0) \gg \mathcal{L}_C$ and $S^2A/(2\alpha) - \mathcal{L}_C > 0$. Intuitively, the approximation shows the tradeoff between T – when increased, decreases the loss of the first N skills – and N – when increased, decreases the loss at sufficiently large T – for fixed compute C. For a comprehensive analysis of the approximation above, see Appendix J.

Removing the irrelevant constant terms,

$$\mathcal{L} = T^{-\alpha/(\alpha+1)} + N^{-\alpha}. \tag{83}$$

We can use the method of Lagrangian multiplier to obtain

$$-\frac{\alpha}{\alpha+1}T^{-\alpha/(\alpha+1)-1} + \lambda N = 0 \tag{84}$$

$$-\alpha N^{-(\alpha+1)} + \lambda T = 0 \tag{85}$$

$$NT - C = 0, \tag{86}$$

where λ is the Lagrange multiplier and C is compute. We can solve the above set of equations to obtain $T^{\alpha+1} \propto N$ and plug it in Eq. (83) to get

$$\mathcal{L} \propto C^{-\alpha/(\alpha+2)}. \tag{87}$$

The derivation is similar to that of [23]. For a rigorous derivation of the optimal compute scaling law, see Corollary 4 and Appendix J.

E Derviation of the extended multilinear model

E.1 Gradient flow in the extended multilinear model

Lemma 2. *Let the extended multilinear model Eq. (38) be trained with gradient on D i.i.d samples for the setup in Section 3 (input distribution: Eq. (1), target function: Eq. (4), and MSE loss: Eq. (5)). For the skill index $k \leq N$ be a skill index in the multilinear model, let the feature matrix $\Phi \in \mathbb{R}^{D_C \times d_k}$ for the k^{th} skill be*

$$\Phi_{lj} = e_{k,l}(i^{(j)} = k, x^{(j)}), \tag{88}$$

and SVD on $\Phi = USV$. Assuming that the system is overparametrized ($d_k < D_c$), the gradient on $B_k \in \mathbb{R}^{D_C}$ is contained in the column space of semi-orthogonal matrix $U \in \mathbb{R}^{D_C \times d_k}$:

$$UU^T \frac{dB_k}{dt} = \frac{dB_k}{dt}. \tag{89}$$

Proof Similar to Lemma 1, the total loss can be decomposed into each skill such that the dynamics of $B_{k,l}$ relies only on d_k observations of the k^{th} skill:

$$\mathcal{L}_D = \frac{1}{2D} \sum_{k=1}^{n_s} \sum_{j=1}^{D} \left(f^*(i^{(j)}, x^{(j)}) - f(i^{(j)}, x^{(j)}) \right)^2 \tag{90}$$

$$= \frac{1}{2D} \sum_{k=1}^{n_s} \sum_{j_k=1}^{d_k} \left(Sg_k(k, x^{(j_k)}) - \sum_{l=1}^{D_C} a_k B_{k,l} e_{k,l}(k, x^{(j_k)}) \right)^2 \tag{91}$$

$$= \frac{1}{2D} \sum_{k=1}^{n_s} \sum_{j_k=1}^{d_k} \left(\sum_{l=1}^{D_C} (\frac{S}{\sqrt{D_C}} - a_k B_{k,l}) e_{k,l}(k, x^{(j_k)}) \right)^2. \tag{92}$$

In the second line, we used Eq. (39) that $e_{k,l}(I \neq k, x) = 0$ and the orthogonality of g_k (Eq. (3)). In the last line, we used Eq. (39) that $g_k = {D_C}^{-1/2} \sum_l e_{k,l}$. We can find the gradient descent equation of $B_{k,l}$ from Eq. (92):

$$\frac{dB_{k,l}}{dt} = -\eta \sum_{j=1}^{d_k} \frac{1}{D} \left[a_k e_{k,l}(k, x^{(j)}) \sum_{l'=1}^{D_C} (a_k B_{k,l'} - \frac{S}{\sqrt{D_C}}) e_{k,l'}(k, x^{(j)}) \right], \tag{93}$$

which in the matrix form is

$$\frac{dB_k}{dt} = -\frac{\eta a_k}{D} \Phi \Phi^T \left(B_k a_k - \frac{\vec{S}}{\sqrt{D_C}} \right), \tag{94}$$

where D_c dimensional vectors B_k and $\vec{S}$ are $[B_{k,1}, \cdots, B_{k,D_C}]$ and $[S, \cdots, S]$ respectively. It illustrates that $\frac{dB_k}{dt}$ is contained in $\mathrm{im}(\Phi)$, which is contained in $\mathrm{im}(U)$ (immediate from $\Phi = USV$). As $UU^T(Uz) = U(U^TU)z = Uz$, UU^T acts as identity on image of U, showing that $UU^T \frac{dB_k}{dt} = U^T \frac{dB_k}{dt}$. ■

E.2 Conserved quantity of extended multilinear model

Lemma 3. *In the setup of Lemma 2, $a_k^2 - |B_k|^2$ is conserved over time.*

Proof We can use Eq. (92) to find the equation for a_k:

$$\frac{da_k}{dt} = -\eta \sum_{j=1}^{d_k} \frac{1}{D} \left[\sum_{l=1} B_{k,l} e_{k,l}(k, x^{(j)}) \sum_{l'=1}^{D_C} (a_k B_{k,l'} - \frac{S}{\sqrt{D_C}}) e_{k,l'}(k, x^{(j)}) \right], \tag{95}$$

which in the matrix form is

$$\frac{da_k}{dt} = -\frac{\eta}{D} B_k^T \Phi\Phi^T \left(B_k a_k - \frac{\vec{S}}{\sqrt{D_C}} \right). \tag{96}$$

Then

$$a_k \frac{da_k}{dt} = -\frac{\eta a_k}{D} B_k^T \Phi\Phi^T \left(B_k a_k - \frac{\vec{S}}{\sqrt{D_C}} \right) \tag{97}$$

$$= B_k^T \frac{dB_k}{dt}, \tag{98}$$

where we used Eq. (94) in the last line. Thus, $a_k^2 - |B_k|^2$ is conserved during the dynamics. ■

E.3 D_C shot learner

Proposition 1. *Let the setup be as that in Lemma 2. Suppose that $a_k(T)$ is eventually bounded away from zero, i.e. there exists $\delta > 0$ and $M > 0$ such that $T > M \Rightarrow |a_k(T)| \geq \delta$. Also assume that $U^\perp$-component of $B_k(0)a_k(0)$ and $B_k(0)S$ is negligible. Then the skill strength $\mathcal{R}_k$ is*

$$\mathcal{R}_k(\infty) = \begin{cases} d_k < D_c : & S\left(1 - \sqrt{1 - d_k/D_c}\right) \\ d_k \geq D_c : & S \end{cases} \tag{99}$$

Proof First, we show that $\frac{d\mathcal{L}}{dt} \leq 0$ with equality only holding when the gradient is 0.

$$\frac{d\mathcal{L}}{dt} = \frac{d\mathcal{L}}{da}\frac{da}{dt} + \sum_i^{D_c} \frac{d\mathcal{L}}{db_i}\frac{db_i}{dt} \tag{100}$$

$$= -\frac{d\mathcal{L}}{da}\frac{d\mathcal{L}}{da} - \sum_i^{D_c} \frac{d\mathcal{L}}{db_i}\frac{d\mathcal{L}}{db_i} \leq 0. \tag{101}$$

Its equality holds only when

$$\frac{d\mathcal{L}}{da} = \frac{da}{dt} = 0 \quad \text{and} \quad \frac{d\mathcal{L}}{db_i} = \frac{db_i}{dt} = 0 \tag{102}$$

We first show that both a_k and B_k are bounded throughout whole dynamics. As

$$\mathcal{L}_k = \left| \Phi \left(B_k a_k - \frac{\vec{S}}{\sqrt{D_C}} \right) \right|^2 \geq \sigma^2 \left| UU^T \left(B_k a_k - \frac{\vec{S}}{\sqrt{D_C}} \right) \right|^2 \tag{103}$$

for σ^2 the smallest nonzero eigenvalue of $\Phi\Phi^T$, where $\Phi = USV$. This shows

$$UU^T \left(B_k a_k - \frac{\vec{S}}{\sqrt{D_C}} \right) \tag{104}$$

is bounded, so $UU^T B_k a_k$ is bounded. Meanwhile, in Lemma 2, we showed $(1 - UU^T)\frac{dB_k}{dt} = 0$, so $(1 - UU^T)B_k a_k$ is bounded. This shows $B_k a_k$ is bounded. As $a_k - |B_k|^2$ is constant (Lemma 3) and $|B_k a_k| = |a_k||B_k|$ is bounded, this shows both a_k and $|B_k|$ are bounded.

The dynamics moving in some bounded region always has at least one accumulation point, which we denote as p. We will show that $\frac{d\mathcal{L}_k}{dt} = 0$ at p. The function $\mathcal{L}_k(t)$ in t is decreasing differential function which is positive. We also note $\frac{d^2\mathcal{L}_k(t)}{dt^2}$ is globally bounded, as it can be expressed in polynomial expression in (a_k, B_k) and we showed $(a_k(t), B_k(t))$ is bounded. From Taylor's theorem one can obtain

$$\inf \mathcal{L}_k(t) \leq \mathcal{L}_k(t_1 + t_2) \leq \mathcal{L}_k(t_1) + t_2 \frac{d\mathcal{L}_k}{dt}(t_1) + \frac{t_2^2}{2} M \tag{105}$$

for $M = \sup |\frac{d^2\mathcal{L}_k(t)}{dt^2}|$. Choosing $t_2 = -\frac{d\mathcal{L}_k}{dt}(t_1)M^{-1}$ shows that

$$\mathcal{L}_k(t_1) - \frac{1}{2M}\left(\frac{d\mathcal{L}_k}{dt}(t_1)\right)^2 \geq \inf \mathcal{L}_k(t) \tag{106}$$

and letting $t_1 \to \infty$ here gives

$$\lim_{t_1\to\infty} \frac{1}{2M}\left(\frac{d\mathcal{L}_k}{dt}(t_1)\right)^2 \leq \lim_{t_1\to\infty} (\mathcal{L}_k(t_1) - \inf \mathcal{L}_k(t)) = 0 \tag{107}$$

so $\frac{d\mathcal{L}_k}{dt} \to 0$ as $t \to \infty$. Meanwhile, as p is accumulation point of (a_k, B_k), $\frac{d\mathcal{L}_k}{dt}(p)$ is accumulation point of $\frac{d\mathcal{L}}{dt}(a_k(t), b_k(t))$. As $\lim_{t\to\infty} \frac{d\mathcal{L}_k}{dt}(t) = 0$, the only accumulation point of $\frac{d\mathcal{L}_k}{dt}(t)$ is zero, which shows that $\frac{d\mathcal{L}_k}{dt}(p) = 0$.

We have seen that $a_k^2 - |B_k|^2$ and $(I - UU^T)B_k$ are conserved in our dynamics. A quantity conserved in dynamics should also be conserved at p, so $p = (a, B)$ should satisfy the following conditions.

- $a^2 - |B|^2 = a_k(0)^2 - |B_k(0)|^2$ (Lemma 3)
- $(I - UU^T)B = (I - UU^T)B_k(0)$ (Lemma 2)
- $\frac{d\mathcal{L}_k}{dt}(a, B) = 0$, or equivalently the gradient is 0 at p

We will solve for p satisfying those three conditions. The third condition is equivalent to that

$$aUU^T\left(Ba - \frac{\vec{S}}{\sqrt{D_C}}\right) = 0. \tag{108}$$

As $a_k(T)$ is eventually bounded away from zero, we have $a \neq 0$, so

$$UU^T\left(Ba - \frac{\vec{S}}{\sqrt{D_C}}\right) = 0. \tag{109}$$

It follows that

$$B = UU^TB + (I - UU^T)B = UU^T\frac{\vec{S}}{\sqrt{D_C}}a^{-1} + (I - UU^T)B_k(0) \tag{110}$$

and substituting to first condition gives

$$a^2 - \frac{1}{a^2}\left|UU^T\frac{\vec{S}}{\sqrt{D_C}}\right|^2 - \left|(I - UU^T)B_k(0)\right|^2 = a_k(0)^2 - |B_k(0)|^2. \tag{111}$$

This is equivalent to a quadratic equation in a^2, and has a following solution of

$$a^2 = \sqrt{\left|UU^T\frac{\vec{S}}{\sqrt{D_C}}\right|^2 + \frac{(a_k(0)^2 - |UU^TB_k(0)|^2)^2}{4}} + \frac{a_k(0)^2 - |UU^TB_k(0)|^2}{2}. \tag{112}$$

This shows there are two candidates for p, with a given as two square roots of Eq. (112) and B determined from a by Eq. (110). It is impossible for $\mathcal{L}_k(t)$ to have accumulation points both in regions $a > 0$ and $a < 0$, as it would imply $a_k(t) = 0$ happens infinitely many often, contradicting that a_k is eventually bounded away from zero. Thus it follows that $\mathcal{L}_k(t)$ can only have one accumulation point. As dynamics having unique accumulation point should converge, it follows that

$$(a, B) = (a_k(\infty), B_k(\infty)). \tag{113}$$

One can check that the $U^\perp$-component of $B_k(\infty)a_k(\infty)$ is given as

$$(I - UU^T)B_k(\infty)a_k(\infty) = (I - UU^T)B_k(0)a \tag{114}$$

and this is bounded by $|(1-UU^T)B_k(0)|(S+a_k(0))$, so by our assumption this is negligible. Thus, we find that $B_k(\infty)a_k(\infty)$ is the pseudo-inverse solution, which is also found by the linear model with $e_{k,l}$ as basis functions. Using the result from kernel regression [26, 34–37] we have

$$\mathcal{L}_k = \frac{S^2}{2}(1-\frac{d_k}{D_C}). \tag{115}$$

Applying Eq. (14), we find the result. ■

E.4 N_c basis functions for a skill

Proposition 2. *Let the extended multilinear model Eq. (38) (but we change the notation D_c to N_c) be trained with gradient on $D \to \infty$ i.i.d samples for the setup in Section 3 with $n_s \to \infty$ (input distribution: Eq. (1), target function: Eq. (4), and MSE loss: Eq. (5), initialization: that of Proposition 1). For a model with the following finite N basis functions*

$$[e_{1,1},\ \cdots,\ e_{1,N_c},\ e_{2,1},\ \cdots,\ e_{q,r}], \tag{116}$$

where quotient $q = \lfloor (N-1)/N_c \rfloor + 1$ and remainder r is such that $(q-1)N_c + r = N$. The skill strength at $T \to \infty$ becomes

$$\mathcal{R}_k(\infty) = \begin{cases} k > q: & 0 \\ k = q: & S\frac{r}{N_c} \\ k < q: & S. \end{cases} \tag{117}$$

Proof Because we have $D \to \infty$ and $[e_{k,1}, \cdots e_{k,N_c}]$ can express g_k (Eq. (43)), it is trivial to show that $\mathcal{R}_k(\infty) = S$ for $k < q$. For $k = q$, the gradient descent dynamics (Eq. (94)) leads to

$$\frac{dB_k}{dt} = -\frac{\eta a_k}{D}\Phi\Phi^T\left(B_k a_k - \frac{S}{\sqrt{N_c}}\right) \tag{118}$$

where the matrix $\Phi \in \mathbb{R}^{r\times d_k}$ and vector $B_k \in \mathbb{R}^r$ are the feature matrix(Eq. (88)) and parameters for the k^{th} skill. As $D \to \infty$, $\Phi\Phi^T$ becomes a rank r identity matrix:

$$\lim_{D\to\infty}\frac{1}{D}(\Phi\Phi^T)_{ll'} = \mathbf{E}_X\left[e_{k,l}(k,X)e_{k,l'}(k,X)\right] = \delta_{l,l'}. \tag{119}$$

Plugging the identity matrix in $\Phi\Phi^T$,

$$\frac{dB_{k,l}}{dt} = -\eta a_k\left(B_{k,l}a_k - \frac{S}{\sqrt{N_c}}\right). \tag{120}$$

Assuming the initialization in Proposition 1, we can show that $a_k(\infty)B_{k,l}(\infty) = S/\sqrt{N_c}$ for $l \le r$. The skill strength $\mathcal{R}_k(\infty)$ is

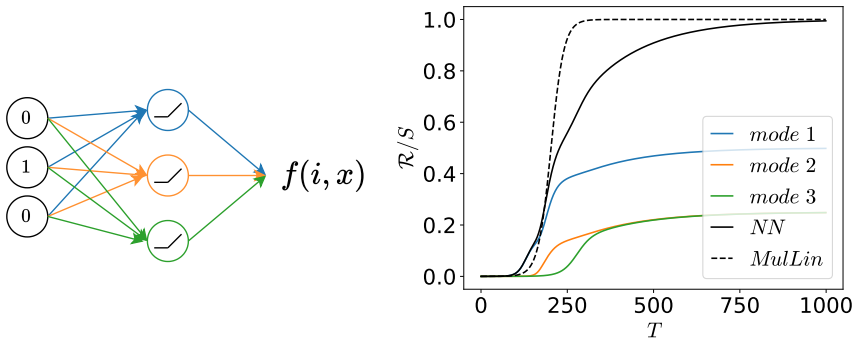
$$\mathcal{R}_k(\infty) = \sum_{l=1}^{r}\frac{S}{\sqrt{N_c}}\mathbf{E}_X\left[e_{k,l}(k,X)g_k(k,X)\right], \tag{121}$$

$$= S\frac{r}{N_c}, \tag{122}$$

where we used Eq. (43) for the linear correlation between $e_{k,l}$ and g_k. ■

F The tail of dynamic emergence

In this section, we discuss an example for the time emergence case (Fig. 6(b, i)) in which the saturation of skill in an NN consists of multiple saturating 'modes' as in Fig. 9.



(a) neuron modes for a parity function (b) mode/skill strength

Figure 9: **Modes in NN.** A 2-layer FCN with ReLU activation with a width of 3 and weight sharing (Eq. (125)) is trained to fit the parity function. **(a):** The skill strength $\mathcal{R}$, because of the last layer's linearity, can be decomposed into skill strength from each hidden neuron or each 'mode' (shown in different colors, Eq. (130)). **(b):** The skill strength for each mode follows a near-sigmoidal curve with different emergent/saturation times (colors) whose sum results in the total skill strength (solid black). Note that different saturation times of each mode result in a deviation from the prediction of the multilinear model with $\mathcal{B}^2 = 1/3$ (dashed black).

Task. We assume an input $X \in \mathbb{R}^{3\times 8}$ (note that we are not using X as a random variable) that is all 8 possible inputs for dimension 3 bits. The output is the parity function scaled by S.

$$X = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}, \qquad Y = \begin{pmatrix} S & -S & -S & S & -S & S & S & -S \end{pmatrix} \tag{123}$$

NN. We assume a 2-layer width 3 NN with ReLU activation with the input dimension 3 (Fig. 9(a)). The NN has 16 parameters, but to simplify the argument, we use weight sharing so NN has only 4 parameters:

$$f(x;\alpha,\beta,\gamma,c) = w^T\sigma(Wx+b)+c \tag{124}$$

where σ is the ReLU activation and W, b, w are

$$W = \begin{pmatrix} -\alpha & \alpha & -\alpha \\ -\beta & \beta & -\beta \\ \gamma & -\gamma & \gamma \end{pmatrix}, \qquad b = \begin{pmatrix} 0 \\ \beta \\ -\gamma \end{pmatrix}, \qquad w = \begin{pmatrix} -2\alpha \\ \beta \\ \gamma \end{pmatrix}. \tag{125}$$

Modes. It is easy to see that $\alpha = \beta = \gamma = \sqrt{2S}$ and $c = -S$ leads to the target parity function. We note that one parameter except c (i.e. α, β, γ) maps to one neuron or a mode (colors in Fig. 9(a)). We define the first mode $f^{(1)}$ as

$$f^{(1)}(x) = w_1\sigma(W_1^Tx+b_1) = -2\alpha^2\sigma(x_2-x_1-x_3) \tag{126}$$

$$= -2\alpha^2 h_1(x), \qquad h_1(x) := \sigma(x_2-x_1-x_3), \tag{127}$$

where w_1, b_1 are the first entry of w, b respectively and W_1 is the first row of W. Note that $f^{(1)}(x)$ takes a form similar to the multilinear model (Eq. (12)) but with h_1 as the respective basis. We define $f^{(2)}, f^{(3)}$ similarly, and the sum of modes becomes the NN:

$$f(x) = \sum_{q=1}^{3} f^{(i)}(x) + c, \tag{128}$$

which resembles the multilinear model with different skills.[11]

[11] Note that the parity function or the target function corresponds to 'one' skill.

Mode strength. Analogous to the skill strength in Eq. (9), we define mode q's strength $\mathcal{R}^{(q)}$ as

$$\mathcal{R}^{(q)} = \frac{1}{8S^2} Y^T f^{(q)}(X), \tag{129}$$

where $f^{(q)}(X) = [f^{(q)}(X_1), \cdots, f^{(q)}(X_8)]$ and X_j are the j^{th} column of X. By the linearity of the expectation,

$$\mathcal{R} = \sum_{q=1}^{3} \mathcal{R}^{(q)}. \tag{130}$$

Note that constant c always has zero correlation (inner product) to the target function (Y).

Analysis. The dynamics of each mode $\mathcal{R}^{(q)}(x)$ differs from that of the multilinear model (Eq. (16)) because $h_q(x)$ often depends on the parameter, and the dynamics are no longer decoupled among each mode. Nevertheless, each mode follows a sigmoid-like growth (Fig. 9(b)). We note that each mode has a different saturation time scale or is updated at different frequencies. A mode with a longer time scale leads to a longer 'tail' of saturation as discussed in the main text.

Update frequency. Because of the non-linearity, the amount of gradient each mode receives is different. We can explicitly calculate the gradient each parameter receives:

$$\frac{d\alpha^2}{dt} = 2\eta\alpha^2(-S - (-2\alpha^2 + 2\beta^2 + c)) \tag{131}$$

$$\frac{d\beta^2}{dt} = -\eta\beta^2(S - (-2\alpha^2 + 5\beta^2 + 5c)) \tag{132}$$

$$\frac{d\gamma^2}{dt} = -\eta\gamma^2(S - (\gamma^2 + c)) \tag{133}$$

$$\frac{dc}{dt} = -\eta(2\alpha^2 - 5\beta^2 - \gamma^2 - 8c). \tag{134}$$

We immediately notice that c will grow the fastest for small initialization ($\alpha, \beta, \gamma, c \ll 1$) because it saturates exponentially while other parameters saturate sigmoidally. Considering that S is always the largest term and c saturate to S quickly, we notice that the saturation is in the order of α^2 ($\approx 2S + 2c \approx 4S$), β^2 ($\approx -S + 5c \approx 4S$), and $\gamma^2 (\approx 2S)$. We observe that our crude approximation holds in Fig. 9(b): the first (α) and the second (β) modes saturate at similar timescale, while the third mode (γ) requires approximately twice the time for saturation.

G Connection to a linear model

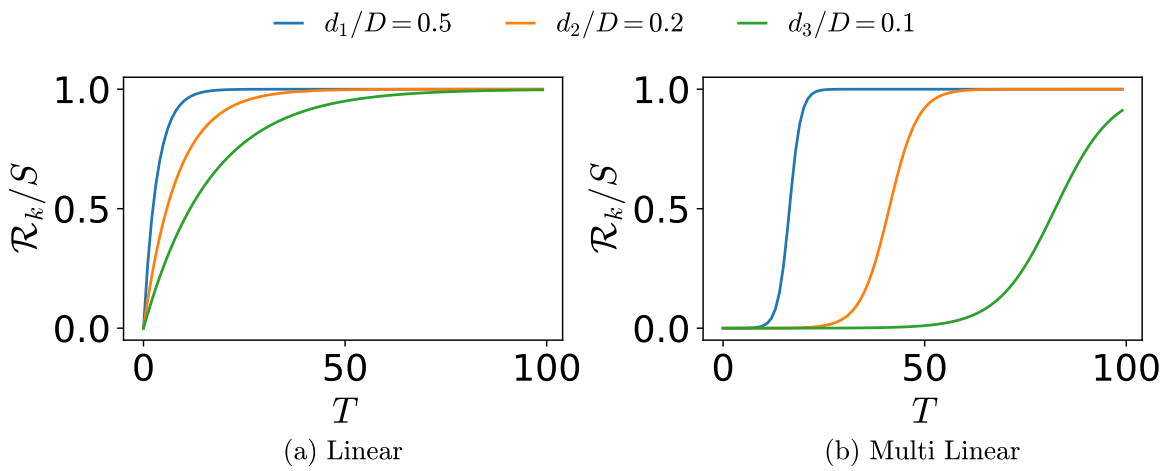
In this section, we will discuss the role of multilinearity in describing time emergence or stage-like training. [12] A linear model with g_k as the basis functions is

$$f_T(i, x; w) = \sum_{k=1}^{N} w_k(T) g_k(i, x), \tag{135}$$

where we made the model linear by replacing $a_k b_k$ with w_k. The dynamics of the linear model under gradient flow is

$$\mathcal{R}_k(T) = w_k(T) = S(1 - e^{-\eta \frac{d_k}{D} T}), \tag{136}$$

where we assumed $w_k(0) = 0$. The linear model follows an exponential saturation of the skill strength in contrast to the sigmoidal saturation of the multilinear model (Fig. 10(a)), and cannot describe the sigmoidal time emergence behavior of a 2-layer NN in Fig. 1(a).



(a) Linear (b) Multi Linear

Figure 10: **Dynamics of linear and multilinear model. (a):** Skill strength dynamics of the linear model (Eq. (136)) **(b):** Skill strength dynamics of the multilinear model (Eq. (16)). For the linear model, $\mathcal{R}_k$ emerges from $T = 0$ for all $d_k/D > 0$: obstructing the stage-like training. For the multilinear model, $\mathcal{R}_k$ shows a delayed emergence depending on d_k/D: allowing the stage-like training and describing the sigmoidal time emergence in Fig. 1(a).

However, the linear model in Eq. (136) – because of it has the skills functions as the basis functions – can predict T, D, N scaling laws in Section 5 and data and parameter emergence in Section 6. For the time scaling law, we recover the relationship between $d\mathcal{L}_k/dT$ and $d\mathcal{L}_k/dk$ in Eq. (25) because $\mathcal{R}_k(T)$ is a function of $\frac{d_k}{D}T$ only (where $d_k/D = \mathcal{P}_s(k)$ for $D \to \infty$). For the data scaling law, we recover Eq. (27) because each w_ks is decoupled. For the parameter scaling law, we recover Eq. (32) trivially. The data and parameter emergence in Section 6 can be obtained from the linear model in Eq. (135) if we extend the model analogous to Eqs. (38) and (41). The equivalence can be shown by Lemma 2 which states that the multilinear model finds the minimum norm solution: the solution that the linear model finds in a ridgeless regression setup.

[12] For a discussion on linear neural networks, see Appendix B.2 instead.

H Samples for parameter emergence

$k=1$	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
$k=2$	4.5	0.95	0.95	0.95	0.96	0.96	0.04	0.96	0.96	0.95
$k=3$	0.6	0.0	0.72	0.90	0.92	0.64	0.88	0.8	0.58	0.52
$k=4$	0.0	0.78	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$k=5$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 11: **Samples of skill strength $\mathcal{R}_k/S$.** The table shows the skill strength at $N = 10$ for 10 different runs of the parameter emergence experiment (Fig. 6(b, iii)). Note that the variance of $\mathcal{R}_k/S$ is amplified by the outliers – shaded columns – that learn a less frequent skill at the cost of a more frequent skill (second column) or fail to learn a skill (seventh column).

I Methods

I.1 2-Layer neural network

We trained a two-layer (one hidden layer) fully connected neural network with ReLU activation. All parameters of NN were initialized with a Gaussian distribution with a standard deviation of 0.001. The input dimension of the model was $n_s + n_b = 5 + 32$ where n_s is the length of control bits (number of skills) and n_b is the length of the skill bits. Each skill has $m = 3$ mutually exclusive sparse bits that are used to express the skill function. The target scale was $S = 5$. The model was trained with SGD without momentum and no weight decay (the exception is the parameter emergence experiment where Adam with learning rate 0.001 and weight decay of 5×10^{-5} was used to escape the local minima). For data and parameter emergence experiments, the learning rate was halved every $50,000$ steps.

At every 20 steps, the skill strength $\mathcal{R}_k(t)$ (Eq. (9)) were measured using $20,000$ i.i.d samples from the k^{th} skill.[13] To mimic the infinite parameter $N \to \infty$, we used the model of width 1000 (for the hidden layer). To mimic the infinite time $T \to \infty$, we trained for 5×10^5 steps where each step had the batch size of 4000. To mimic $D \to \infty$, we sampled new data points for every batch. The details are given in the following table.

Name	Values
width	1000
learning rate	0.05
initialization standard deviation	0.001
activation	ReLU
batch size	4000
steps	500,000
target scale	5
number of skill bits	32
number of skills	5

I.2 Measurement of skill strength

The skill strength $\mathcal{R}_k$ is a simple linear correlation between the learned function f – function expressed by NN – and g_k for $\mathcal{P}_b$ given $I = k$. We approximate the expectation over X by taking the mean over $20,000$ i.i.d samples from $\mathcal{P}_b$ for the k^{th} skill:

$$\mathcal{R}_k = \mathbf{E}_X[f(k, X)g_k(k, X)] \approx \frac{1}{20000} \sum_{j=1}^{20000} f(k, x^{(j)})g_k(k, x^{(j)}), \tag{137}$$

where the notation $x^{(j)}$ denotes the j^{th} sample.

[13] Note that except the data scaling law experiment, the training set size is infinite.

J Rigorous derivation of the scaling laws

In Appendix D, we discussed the scaling laws in simplified settings, favoring intuition over mathematical rigor. Building upon the intuitive understanding developed in Appendix D, we now turn our attention to a rigorous analysis of the scaling laws. In this section, we will derive general scaling laws by considering a comprehensive set of parameters and variables. Our goal is to establish the conditions under which these scaling laws hold and to quantify the associated error terms. By explicitly analyzing the error terms, this section aims to provide a rigorous assessment of the validity and limitations of our scaling law estimates.

Large resource	Condition	Scaling law	Constant	Statement
$D \gg T^3$	$N^{\alpha+1} = o(T)$	$\mathcal{L} = N^{-\alpha}$	Theorem 1	Theorem 1
$D \gg NT^2, T^3$	$N^{\alpha+1} \gg T$	$\mathcal{L} = T^{-\alpha/(\alpha+1)}$	Theorem 4	Theorems 2 and 3
$D \gg T^3$	$N^{\alpha+1} \approx T$	$\mathcal{L} = C^{-\alpha/(\alpha+2)}$	Corollary 5	Corollary 4
$T \gg D(\log D)^{1+\epsilon}$	$N^{\alpha+1} = o(D)$	$\mathcal{L} = N^{-\alpha}$	Theorem 5	Theorem 5
$T \gg D(\log D)^{1+\epsilon}$	$N^{\alpha+1} \gg D$	$\mathcal{L} = D^{-\alpha/(\alpha+1)}$	Theorem 5	Theorem 5

Table 4: **Summary of scaling law and their conditions.** The leftmost column indicates the condition for the 'large resource' – large enough to be treated as infinity, while the second column is the condition between the other two resources for the scaling law (third column). The last two columns show where the statement for the prefactor constant (e.g. $\mathcal{A}$ for scaling law $\mathcal{L} = \mathcal{A}N^{-\alpha}$) and the scaling law (with the assumptions and explicit error terms) are given.

J.1 General set up, repeated

We go back to most general settings possible. Our starting point is Eq. (50), which describes the dynamics of $\mathcal{R}_k$ and $\mathcal{L}_k$ valid for $k \leq N$:

$$\mathcal{L}_k = \frac{S^2}{2\left(1 + \left(\frac{S}{\mathcal{R}_k(0)} - 1\right)^{-1} e^{2\eta \frac{d_k}{D} ST}\right)^2} \tag{50}$$

We do not use skills for indices $k > N$ in our model, but we can still denote

$$\mathcal{R}_k = 0 \quad \text{and} \quad \mathcal{L}_k = \frac{S^2}{2}. \tag{138}$$

For $\mathcal{P}_s(k) = Ak^{-\alpha-1}$, the total loss is given as

$$\mathcal{L} = \sum_{k=1}^{n_s} \mathcal{P}_s(k)\mathcal{L}_k = \sum_{k=1}^{N} \mathcal{P}_s(k)\mathcal{L}_k + \sum_{k=N+1}^{n_s} \mathcal{P}_s(k)\frac{S^2}{2}. \tag{139}$$

When n_s, N, T are all set, their dependency with the data is only determined by the statistics d_k, the number of data with $i^{(j)} = k$. We assumed that $(i, x) \in I \times \{0, 1\}^{n_d}$ was collected as random samples with i following the Zipfian distribution of size n_s and exponent $\alpha + 1$, or equivalently $P(i = k) = \mathcal{P}_s(k) = Ak^{-\alpha-1}$ for $1 \leq k \leq n_s$. Then $(d_1, \cdots, d_{n_s})$ is a vector denoting number of occurrences in D independent sampling from that distribution. It follows that d_i follows binomial distribution $B(D, \mathcal{P}_s(k))$.

In this complete perspective, our loss is dependent on all of those parameters and variables

$$\mathcal{L} = \mathcal{L}(n_S, \mathcal{D}, \mathcal{R}_{init}, N, T) \tag{140}$$

where $\mathcal{R}_{init} = (\mathcal{R}_1(0), \cdots, \mathcal{R}_N(0))$ denotes the vector representing initial condition. We will also simply denote $r_k = \mathcal{R}_k(0)$. We will not assume much on r_k, but we absolutely need $0 < r_k < S$ for dynamics to hold, and we also should have

$$\sum_{k=1}^{n_s} \mathcal{P}_s(k) r_k^2 = \mathbf{E}[f(0)^2] \ll S^2. \tag{141}$$

We will not impose any particular distribution on $\mathcal{R}_{init}$. Instead, we will try to identify sufficient conditions on r_k for our desired result to hold, and those conditions will differ by the situation we are considering. For example, in Theorems 2 and 3 where we prove time scaling law $\mathcal{L} = \Theta(T^{-\alpha/(\alpha+1)})$ for large enough D and bottleneck T, we only require $0 < r_k < S/2$. However, the exact constant depends on the distribution of r_k, and figuring out the explicit constant seems to be only feasible when we fix $r_k = r$ as in Theorem 4.

J.2 Estimates for large D

We will first consider the situation where D becomes the 'large resource' so that its effect on the loss function is negligible. The number of data d_k follows binomial distribution $B(D, \mathcal{P}_s(k))$, so d_k/D converges to $\mathcal{P}_s(k)$ for large enough D. So taking the limit of $\mathcal{L}$ when we let $D \to \infty$ has effect of replacing d_k/D by $\mathcal{P}_s(k)$ in the expression of $\mathcal{L}$. We will establish an explicit inequality comparing the difference between $\mathcal{L}$ and this limit.

Lemma 4. *For a function $F : \mathbb{R} \to \mathbb{R}$ with its total variation $V(F)$ bounded, we have*

$$\left| \mathbf{E}_{\mathcal{D}} \left[F(\frac{d_k}{D}) \right] - \mathbf{E}_{z \sim \mathcal{N}(\mathcal{P}_s(k), \mathcal{P}_s(k)(1-\mathcal{P}_s(k))/D)} \left[F(z) \right] \right| < \frac{V(F)}{\sqrt{D}\sqrt{\mathcal{P}_s(k)(1-\mathcal{P}_s(k))}} \tag{142}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes normal distribution of mean μ and variance σ^2.

Proof This is just application of the Berry-Esseen inequality (with constant 1, see [49] for modern treatment) applied to d_k following binomial distribution $B(D, \mathcal{P}_s(k))$. ■

Lemma 5. *Let $F : \mathbb{R} \to \mathbb{R}$ be a C^2 function such that F'' is bounded. Then we have*

$$\left| \mathbf{E}_{z \sim \mathcal{N}(\mathcal{P}_s(k), \mathcal{P}_s(k)(1-\mathcal{P}_s(k))/D)} \left[F(z) \right] - F(\mathcal{P}_s(k)) \right| \leq \frac{\mathcal{P}_s(k)(1-\mathcal{P}_s(k))}{2D} \sup |F''|. \tag{143}$$

Proof First we apply Taylor's theorem to show that

$$|F(z) - F(\mathcal{P}_s(k)) - F'(\mathcal{P}_s(k))(z - \mathcal{P}_s(k))| \leq \frac{(z - \mathcal{P}_s(k))^2}{2} \sup |F''|. \tag{144}$$

Taking expectation when z follows normal distribution $\mathcal{N}(\mathcal{P}_s(k), \mathcal{P}_s(k)(1 - \mathcal{P}_s(k))/D$ gives

$$\begin{aligned}
|\mathbf{E}_z \left[F(z) - F(\mathcal{P}_s(k)) \right]| &= |\mathbf{E}_z \left[F(z) - F(\mathcal{P}_s(k)) - F'(\mathcal{P}_s(k))(z - \mathcal{P}_s(k)) \right]| && (145)\\
&\leq \mathbf{E}_z \left[|F(z) - F(\mathcal{P}_s(k)) - F'(\mathcal{P}_s(k))(z - \mathcal{P}_s(k))| \right] && (146)\\
&\leq \mathbf{E}_z \left[\frac{(z - \mathcal{P}_s(k))^2}{2} \sup |F''| \right] && (147)\\
&= \frac{\mathcal{P}_s(k)(1 - \mathcal{P}_s(k))}{2D} \sup |F''|. && (148)
\end{aligned}$$

■

Proposition 3. *We have*

$$\left| \mathbf{E}_{\mathcal{D}} \left[\mathcal{L}_k \right] - \frac{S^2}{2 \left(1 + \left(\frac{S}{r_k} - 1 \right)^{-1} e^{2\eta \mathcal{P}_s(k) ST} \right)^2} \right| < \frac{2^\alpha S^2}{\sqrt{D \mathcal{P}_s(k)}} + \frac{4 S^4 \eta^2 T^2 \mathcal{P}_s(k)}{D}. \tag{149}$$

Proof Consider the function $F : \mathbb{R} \to \mathbb{R}$ given as

$$F(z) = \frac{S^2}{2 \left(1 + \left(\frac{S}{r_k} - 1 \right)^{-1} e^{2\eta STz} \right)^2}. \tag{150}$$

This function is monotone decreasing and C^2 on whole domain, and its supremum and infimum are given as

$$\sup F = \lim_{z\to-\infty} F(z) = \frac{S^2}{2} \quad \text{and} \quad \inf F = \lim_{z\to\infty} F(z) = 0. \tag{151}$$

This implies that

$$V(F) = \sup F - \inf F = \frac{S^2}{2}. \tag{152}$$

Also, we will show that F'' is globally bounded. We first calculate

$$F''(z) = -4S^3 r_k (1-\frac{r_k}{S})^2 \eta^2 T^2 \frac{e^{2\eta STz}(1-\frac{r_k}{S}-\frac{2r_k}{S}e^{2\eta STz})}{\left(1-\frac{r_k}{S}+\frac{r_k}{S}e^{2\eta STz}\right)^4}. \tag{153}$$

We consider the following inequalities

$$e^{2\eta STz} \le \frac{S}{r_k}\left(1-\frac{r_k}{S}+\frac{r_k}{S}e^{2\eta STz}\right) \tag{154}$$

$$\left|1-\frac{r_k}{S}-\frac{2r_k}{S}e^{2\eta STz}\right| \le \left|1-\frac{r_k}{S}\right| + \frac{2r_k}{S}e^{2\eta STz} < 2\left(1+\frac{r_k}{S}(e^{2\eta STz}-1)\right) \tag{155}$$

to show that

$$|F''(z)| < 4S^3 r_k (1-\frac{r_k}{S})^2 \eta^2 T^2 \frac{\frac{2S}{r_k}\left(1-\frac{r_k}{S}+\frac{r_k}{S}e^{2\eta STz}\right)^2}{\left(1-\frac{r_k}{S}+\frac{r_k}{S}e^{2\eta STz}\right)^4} < 8S^4\eta^2T^2 \tag{156}$$

for all z. Thus we can apply both Lemma 4 and Lemma 5 to this function F and we have

$$\begin{aligned}\left|\mathbf{E}_{\mathcal{D}}\left[F(\frac{d_k}{D})\right] - F(\mathcal{P}_s(k))\right| <& \frac{V(F)}{\sqrt{D}\sqrt{\mathcal{P}_s(k)(1-\mathcal{P}_s(k))}} + \frac{\mathcal{P}_s(k)(1-\mathcal{P}_s(k))}{2D}\sup|F''| \\ <& \frac{S^2}{2\sqrt{D}\sqrt{\mathcal{P}_s(k)(1-\mathcal{P}_s(k))}} + \frac{4\mathcal{P}_s(k)S^4\eta^2T^2}{D} \\ <& \frac{2^\alpha S^2}{\sqrt{D\mathcal{P}_s(k)}} + \frac{4\mathcal{P}_s(k)S^4\eta^2T^2}{D}\end{aligned} \tag{157}$$

where the last line follows from that we always have

$$1-\mathcal{P}_s(k) \ge 1-\mathcal{P}_s(1) = \frac{2^{-(\alpha+1)}+\cdots+n_s^{-(\alpha+1)}}{1+2^{-(\alpha+1)}+\cdots+n_s^{-(\alpha+1)}} > \frac{2^{-(\alpha+1)}}{1+2^{-(\alpha+1)}} > \frac{1}{2^{2(\alpha+1)}}. \tag{158}$$

■

Lemma 6. *For any integer N and $\sigma \ge 1/2$ and $\sigma \ne 1$, we have*

$$\sum_{k=1}^{N} k^{-\sigma} = \zeta(\sigma) + \frac{N^{1-\sigma}}{1-\sigma} + O(N^{-\sigma}) \tag{159}$$

where ζ is Riemann zeta function (defined over whole complex plane except 1 via analytic continuation). In addition,

$$\sum_{k=1}^{N} k^{-1} = \log N + \gamma + O(N^{-1}) \tag{160}$$

where $\gamma = 0.5772156649...$ is Euler's constant.

Proof See Corollary 1.15 of [50], or other analytic number theory textbooks. ■

Proposition 4. *(Large D approximation) We have*

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] - \sum_{k=1}^{N} \mathcal{P}_s(k) \frac{S^2}{2\left(1 + \left(\frac{S}{r_k} - 1\right)^{-1} e^{2\eta \mathcal{P}_s(k) ST}\right)^2} - \sum_{k=N+1}^{n_s} \mathcal{P}_s(k) \frac{S^2}{2} \tag{161}$$

$$= O\left(S^2 D^{-1/2} f_\alpha(N) + S^4 \eta^2 T^2 D^{-1}\right) \tag{162}$$

where

$$f_\alpha(N) = \begin{cases} 1 & \text{if } \alpha > 1 \\ \log N & \text{if } \alpha = 1 \\ N^{(1-\alpha)/2} & \text{if } \alpha < 1. \end{cases} \tag{163}$$

The constant on the O term only depends on α.

Proof From the description of $\mathcal{L}$ in Eq. (139), we have

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] - \sum_{k=1}^{N} \mathcal{P}_s(k) \frac{S^2}{2\left(1 + \left(\frac{S}{r_k} - 1\right)^{-1} e^{2\eta \mathcal{P}_s(k) ST}\right)^2} - \sum_{k=N+1}^{n_s} \mathcal{P}_s(k) \frac{S^2}{2} \tag{164}$$

$$= \sum_{k=1}^{N} \mathcal{P}_s(k) \left(\mathbf{E}_{\mathcal{D}}[\mathcal{L}_k] - \frac{S^2}{2\left(1 + \left(\frac{S}{r_k} - 1\right)^{-1} e^{2\eta \mathcal{P}_s(k) ST}\right)^2} \right). \tag{165}$$

We apply Proposition 3 to give

$$\sum_{k=1}^{N} \mathcal{P}_s(k) \left(\mathbf{E}_{\mathcal{D}}[\mathcal{L}_k] - \frac{S^2}{2\left(1 + \left(\frac{S}{r_k} - 1\right)^{-1} e^{2\eta \mathcal{P}_s(k) ST}\right)^2} \right) < \sum_{k=1}^{N} \mathcal{P}_s(k) \left(\frac{2^\alpha S^2}{\sqrt{D \mathcal{P}_s(k)}} + \frac{4 S^4 \eta^2 T^2 \mathcal{P}_s(k)}{D} \right). \tag{166}$$

Each of these sum involving $\mathcal{P}_s(k)$ is bounded as

$$\sum_{k=1}^{N} \mathcal{P}_s(k)^2 < \left(\sum_{k=1}^{N} \mathcal{P}_s(k) \right)^2 < 1 \tag{167}$$

and

$$\sum_{k=1}^{N} \sqrt{\mathcal{P}_s(k)} < \sum_{k=1}^{N} k^{-(\alpha+1)/2} = O(f_\alpha(N)) \tag{168}$$

which follows from Lemma 6. Combining those two gives

$$\sum_{k=1}^{N} \mathcal{P}_s(k) \left(\frac{2^\alpha S^2}{\sqrt{D \mathcal{P}_s(k)}} + \frac{S^4 \eta^2 T^2 \mathcal{P}_s(k)}{D} \right) = O\left(S^2 D^{-1/2} f_\alpha(N) + S^4 \eta^2 T^2 D^{-1}\right). \tag{169}$$

■

While Proposition 4 holds for any D, it becomes only meaningful if the resulting error terms are less than the main term we desire. We will revisit this when the exact main term is found, and determine the sufficient size of D for error terms to become small enough.

J.3 Estimates for not too small n_s

We next discuss the effect of n_s. When $n_s \to \infty$ heuristically, then intuitively we have $\mathcal{P}_s(k) \to k^{-(\alpha+1)}/\zeta(\alpha+1)$. We will discuss the difference between when we regard n_s as ∞ and when we do not.

Proposition 5. *The following equations hold:*

$$A^{-1} = \sum_{k=1}^{n_s} k^{-(\alpha+1)} = \zeta(\alpha+1) - \frac{n_s^{-\alpha}}{\alpha} + O(n_s^{-\alpha-1}) \tag{170}$$

$$\mathcal{P}_s(k) = \frac{k^{-\alpha-1}}{\zeta(\alpha+1)} \left(1 + \frac{n_s^{-\alpha}}{\alpha\zeta(\alpha+1)} O(n_s^{-\alpha-1})\right) \tag{171}$$

$$\sum_{k=N+1}^{n_s} \mathcal{P}_s(k) = \frac{N^{-\alpha} - n_s^{-\alpha}}{\alpha\zeta(\alpha+1)} + O(N^{-\min(\alpha+1,2\alpha)}) \tag{172}$$

All implied constants on O only depend on α.

Proof The first statement Eq. (170) follows from substituting $\sigma = \alpha + 1$ in Lemma 6. As $\mathcal{P}_s(k) = Ak^{-(\alpha+1)}$, the second statement Eq. (171) immediately follows. If we substitute $n_s = N$ into Eq. (170) and calculate differences between them, we obtain

$$\sum_{k=N+1}^{n_s} k^{-\alpha-1} = \frac{N^{-\alpha} - n_s^{-\alpha}}{\alpha} + O(N^{-\alpha-1}). \tag{173}$$

Thus we have

$$\sum_{k=N+1}^{n_s} \mathcal{P}_s(k) = A \sum_{k=N+1}^{n_s} k^{-(\alpha+1)} = \frac{N^{-\alpha} - n_s^{-\alpha}}{\alpha\zeta(\alpha+1)} + O\left(N^{-\alpha-1} + (N^{-\alpha} - n_s^{-\alpha})n_s^{-\alpha}\right). \tag{174}$$

Regardless of the size of n_s, We always have

$$(N^{-\alpha} - n_s^{-\alpha})n_s^{-\alpha} \leq \left(\frac{N^{-\alpha}}{2}\right)^2 = \frac{N^{-2\alpha}}{4} \tag{175}$$

so the third statement Eq. (172) follows. ■

We go back to description of total loss given in Eq. (139) as

$$\mathcal{L} = \sum_{k=1}^{N} \mathcal{P}_s(k)\mathcal{L}_k + \sum_{k=N+1}^{n_s} \mathcal{P}_s(k)\frac{S^2}{2} \tag{139}$$

and we take its expectation in $\mathcal{D}$. Proposition 4 suggests that its limit when $D \to \infty$ is given as

$$\lim_{D\to\infty} \mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \sum_{k=1}^{N} \mathcal{P}_s(k) \frac{S^2}{2\left(1 + \left(\frac{S}{r_k} - 1\right)^{-1} e^{2\eta\mathcal{P}_s(k)ST}\right)^2} + \sum_{k=N+1}^{n_s} \mathcal{P}_s(k)\frac{S^2}{2}. \tag{176}$$

Denote

$$\mathscr{L}_1 = \sum_{k=1}^{N} \mathcal{P}_s(k) \frac{S^2}{2\left(1 + \left(\frac{S}{r_k} - 1\right)^{-1} e^{2\eta\mathcal{P}_s(k)ST}\right)^2} \tag{177}$$

$$\mathscr{L}_2 = \sum_{k=N+1}^{n_s} \mathcal{P}_s(k)\frac{S^2}{2}. \tag{178}$$

We discuss the effect of n_s in $\mathscr{L}_1$ and $\mathscr{L}_2$, by comparing limit of $\mathscr{L}_1$ and $\mathscr{L}_2$ when $n_s \to \infty$ and their original values.

- For the term $\mathscr{L}_1$, the change of letting n_s as finite value from $n_s \to \infty$ has effect of multiplying T by $1+n_s^{-\alpha}/(\alpha\zeta(\alpha+1))$, and multiplying whole $\mathscr{L}_1$ by $1+n_s^{-\alpha}/(\alpha\zeta(\alpha+1))$. It can be equivalently put as

$$\mathscr{L}_1(n_s,N,T) = \left(1+\frac{n_s^{-\alpha}}{\alpha\zeta(\alpha+1)}+O(n_s^{-\alpha-1})\right)\mathscr{L}_1\left(\infty,N,T\left(1+\frac{n_s^{-\alpha}}{\alpha\zeta(\alpha+1)}+O(n_s^{-\alpha-1})\right)\right). \tag{179}$$

  We always have $n_s > N$ and $N \to \infty$ eventually, so if dependency of $\mathscr{L}_1$ with respect to T is at most polynomial order then change of main term of $\mathscr{L}_1$ is negligible. We can't establish exact statements yet without the descriptions of size of $\mathcal{L}_1$.

- The term $\mathscr{L}_2$ only depends on N and n_s, not on T. Applying Proposition 5 (especially Eq. (172)) gives

$$\mathscr{L}_2(n_s,N,T) = \frac{N^{-\alpha}-n_s^{-\alpha}}{\alpha\zeta(\alpha+1)}\frac{S^2}{2}+O(N^{-\min(\alpha+1,2\alpha)}S^2) \tag{180}$$

  When n_s grows faster than N then $n_s^{-\alpha}$ part is totally negligible, and when n_s has same order as N then $n_s^{-\alpha}$ affects the constant for main term of $\mathscr{L}_2$. Things might get little complicated when $n_s = N + o(N)$, where $N^{-\alpha} - n_s^{-\alpha} = o(N^{-\alpha})$ can happen then.

- Comparing size of $\mathscr{L}_1$ and $\mathscr{L}_2$ mainly depends on time. The term $\mathscr{L}_2$ is fixed, and $\mathscr{L}_1$ decreases as T increases. For $T = \infty$ we have $\mathscr{L}_1 = 0$, so $\mathscr{L}_2$ having order $N^{-\alpha}$ dominates (this proves scaling law for N of exponent α), so restriction on n_s becomes quite substantial. For small T and large N where size of $\mathscr{L}_2$ is small, we can expect restriction on n_s is less substantial. For example, in the extreme case $N = \infty$, we have $\mathscr{L}_2 = 0$, and n_s does not matter at all (except that of course it should satisfy $n_s \geq N$).

For such reasons, it is hard to quantify exact conditions for n_s such that error terms are controlled, unless we specify relative growth of (N,T). However, $n_s = \omega(N)$ suffices to assure that setting $n_s = \infty$ has zero effect on the main term. We will not worry about n_s in this setting anymore too, and come back to this at the very end to determine enough n_s.

J.4 Estimating main terms

We assume $D = \infty$ and $n_s = \infty$ – virtually implying that $d_k/D = \mathcal{P}_s(k)$ and $\mathcal{P}_s(k) = k^{-\alpha-1}/\zeta(\alpha+1)$ (calculated by rule of $n_s = \infty$). We decomposed our main term into

$$\lim_{n_s\to\infty}\lim_{D\to\infty}\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \mathscr{L}_1+\mathscr{L}_2 \tag{181}$$

where

$$\mathscr{L}_1 = \sum_{k=1}^{N}\mathcal{P}_s(k)\frac{S^2}{2\left(1+\left(\frac{S}{r_k}-1\right)^{-1}e^{2\eta\mathcal{P}_s(k)ST}\right)^2} \tag{182}$$

and

$$\mathscr{L}_2 = \sum_{k=N+1}^{\infty}\mathcal{P}_s(k)\frac{S^2}{2}. \tag{183}$$

By Proposition 5, $\mathcal{L}_2$ is determined almost completely as

$$\mathscr{L}_2 = \frac{S^2N^{-\alpha}}{2\alpha\zeta(\alpha+1)}+O(N^{-\alpha-1}). \tag{184}$$

Now focus on $\mathscr{L}_1$. For

$$F(z) = \frac{S^2}{2\left(1+\left(\frac{S}{r_k}-1\right)^{-1}e^{2\eta STz}\right)^2} \tag{185}$$

(note: it really depends on r_k so it is correct to write F_k, but for convenience we will keep using F.) one can express $\mathscr{L}_1$ as

$$\mathscr{L}_1 = \sum_{k=1}^{N}\mathcal{P}_s(k)F(\mathcal{P}_s(k)). \tag{186}$$

Lemma 7. *Let $F(z)$ be defined as Eq. (185).*

1. *(Estimate for large z) We have*

$$0 \le F(z) \le \frac{(S-r_k)^2}{2}\min\left(1, \frac{S^2}{r_k^2}e^{-4\eta STz}\right). \tag{187}$$

2. *(Estimate for small z) For $z \ge 0$, we have*

$$\frac{(S-r_k)^2}{2} - \frac{8\eta S^3 T}{27}z \le F(z) \le \frac{(S-r_k)^2}{2}. \tag{188}$$

Proof

1. The left side is obvious. For the right side, $F(z) \le (S-r_k)^2/2$ follows from noting that $F(0) = \frac{(S-r_k)^2}{2}$ and proving $F'(z) \le 0$, and $F(z) \le \frac{(S-r_k)^2}{2}\frac{S^2}{r_k^2}e^{-4\eta STz}$ follows from just replacing $1 + \left(\frac{S}{r_k}-1\right)^{-1}e^{2\eta STz}$ in the denominator of F by $\left(\frac{S}{r_k}-1\right)^{-1}e^{2\eta STz}$.

2. For the left side, it suffices to show $-F'(z) \le \frac{8\eta S^3T}{27}$. One can calculate

$$F'(z) = -2S^2 r_k(1-\frac{r_k}{S})^2\eta T\frac{e^{2\eta STz}}{\left(1+\frac{r_k}{S}(e^{2\eta STz}-1)\right)^3} \tag{189}$$

and

$$F''(z) = -4S^3 r_k(1-\frac{r_k}{S})^2\eta^2T^2\frac{e^{2\eta STz}(1-\frac{r_k}{S}-\frac{2r_k}{S}e^{2\eta STz})}{\left(1+\frac{r_k}{S}(e^{2\eta STz}-1)\right)^4} \tag{190}$$

so F has unique inflection point at

$$1-\frac{r_k}{S}-\frac{2r_k}{S}e^{2\eta STz} = 0 \quad \Rightarrow \quad e^{2\eta STZ} = \frac{1}{2}\left(\frac{S}{r_k}-1\right) \tag{191}$$

and this point is where $-F'(z)$ obtains maximum. Substituting this to the expression of $F'(z)$ gives $-F'(z) = \frac{8\eta S^3T}{27}$.

■

Our threshold for distinguishing two approximation methods will be set as $z = z_0 = (\zeta(\alpha+1)\eta ST)^{-1}$, where both two error terms are bounded by $O(S^2)$. The constant $\zeta(\alpha+1)$ is set to make later calculations much easier. Applying Lemma 7 gives

$$\mathscr{L}_1 = \sum_{k=1}^{N}\mathcal{P}_s(k)F(\mathcal{P}_s(k)) \tag{192}$$

$$= \sum_{1\le k\le N, \mathcal{P}_s(k)<z_0}\frac{(S-r_k)^2}{2}\mathcal{P}_s(k) + O\left(\eta S^3T\sum_{1\le k\le N,\mathcal{P}_s(k)<z_0}\mathcal{P}_s(k)^2 + S^2\sum_{1\le k\le N,\mathcal{P}_s(k)>z_0}\mathcal{P}_s(k)\min\left(1,\frac{S^2}{r_k^2}e^{-4\eta ST\mathcal{P}_s(k)}\right)\right). \tag{193}$$

Denote

$$\mathscr{M} = \sum_{1\le k\le N,\mathcal{P}_s(k)<z_0}\frac{(S-r_k)^2}{2}\mathcal{P}_s(k) \tag{194}$$

$$\mathscr{E}_1 = \eta S^3T\sum_{1\le k\le N,\mathcal{P}_s(k)<z_0}\mathcal{P}_s(k)^2 \tag{195}$$

$$\mathscr{E}_2 = S^2\sum_{1\le k\le N,\mathcal{P}_s(k)>z_0}\mathcal{P}_s(k)\min\left(1,\frac{S^2}{r_k^2}e^{-4\eta ST\mathcal{P}_s(k)}\right). \tag{196}$$

Proposition 6. *Suppose that there exists $0 < r < \sqrt{S}$ such that $r \le r_k < S/2$ for all k. In the decomposition of*

$$\lim_{n_s\to\infty}\lim_{D\to\infty}\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \mathscr{M} + \mathscr{L}_2 + O(\mathscr{E}_1 + \mathscr{E}_2) \tag{197}$$

given as above, we have the following bound.

1. *If $(\eta ST)^{1/(\alpha+1)} > N$, then*

$$\mathscr{L}_2 = \frac{S^2N^{-\alpha}}{2\alpha\zeta(\alpha+1)} + O(S^2N^{-\alpha-1}) \tag{198}$$

$$\mathscr{M} = \mathscr{E}_1 = 0 \tag{199}$$

$$\mathscr{E}_2 = O\left(S^2(\log(S/r))^{\alpha/(\alpha+1)}(\eta ST)^{-\alpha/(\alpha+1)}\right) \tag{200}$$

2. *If $(\eta ST)^{1/(\alpha+1)} < N$, then*

$$\mathscr{L}_2 + \mathscr{M} = \Theta\left(S^2\sum_{k>(\eta ST)^{1/(\alpha+1)}}\mathcal{P}_s(k)\right) = \Theta(S^2(\eta ST)^{-\alpha/(\alpha+1)}) \tag{201}$$

$$\mathscr{E}_1 = O\left(S^2(\eta ST)^{-\alpha/(\alpha+1)}\right) \tag{202}$$

$$\mathscr{E}_2 = O\left(S^2(\log(S/r))^{\alpha/(\alpha+1)}(\eta ST)^{-\alpha/(\alpha+1)}\right) \tag{203}$$

Here all constants in O and Θ terms are absolute with respect to η, S, T, N. (They may depend on α.)

Proof We first note that the condition $\mathcal{P}_s(k) < z_0 = (\zeta(\alpha+1)\eta ST)^{-1}$ is equivalent to

$$\mathcal{P}_s(k) < z_0 = (\zeta(\alpha+1)\eta ST)^{-1} \Leftrightarrow k^{-\alpha-1} < \frac{1}{\eta ST} \Leftrightarrow k > (\eta ST)^{1/(\alpha+1)}. \tag{204}$$

Thus we can rephrase the descriptions of terms as

$$\mathscr{M} = \sum_{(\eta ST)^{1/(\alpha+1)}<k\le N}\frac{(S-r_k)^2}{2}\mathcal{P}_s(k) \tag{205}$$

$$\mathscr{E}_1 = \eta S^3T\sum_{(\eta ST)^{1/(\alpha+1)}<k\le N}\mathcal{P}_s(k)^2 \tag{206}$$

$$\mathscr{E}_2 = S^2\sum_{k\le\min((\eta ST)^{1/(\alpha+1)},N)}\mathcal{P}_s(k)\min\left(1, \frac{S^2}{r_k^2}e^{-4\eta ST\mathcal{P}_s(k)}\right). \tag{207}$$

Applying Proposition 5 easily shows that

$$\mathscr{L}_2 = \frac{S^2N^{-\alpha}}{2\alpha\zeta(\alpha+1)} + O(S^2N^{-\alpha-1}). \tag{208}$$

For $\mathscr{M}$ and $\mathscr{E}_1$, we will consider them by dividing two cases depending on whether $(\eta ST)^{1/(\alpha+1)} > N$ or $(\eta ST)^{1/(\alpha+1)} < N$. If $(\eta ST)^{1/(\alpha+1)} > N$, then the condition $(\eta ST)^{1/(\alpha+1)} < k \le N$ is never satisfied, so $\mathscr{M} = \mathscr{E}_1 = 0$. Now suppose $(\eta ST)^{1/(\alpha+1)} < N$. We first note that

$$\mathscr{L}_2 + \mathscr{M} = \sum_{(\eta ST)^{1/(\alpha+1)}<k\le N}\frac{(S-r_k)^2}{2}\mathcal{P}_s(k) + \sum_{k>N}\frac{S^2}{2}\mathcal{P}_s(k). \tag{209}$$

As $(S-r_k)^2 = \Theta(S^2)$, we can let

$$\mathscr{L}_2 + \mathscr{M} = \Theta\left(S^2\sum_{k>(\eta ST)^{1/(\alpha+1)}}\mathcal{P}_s(k)\right) \tag{210}$$

and using Proposition 5 gives the desired estimate $\mathscr{L}_2+\mathscr{M} = \Theta(S^2(\eta ST)^{-\alpha/(\alpha+1)})$. For $\mathscr{E}_1$, estimating sum of $\mathcal{P}_s(k)^2$ using Lemma 6 gives

$$\mathscr{E}_1 = O\left(\eta S^3 T \sum_{k>(\eta ST)^{1/(\alpha+1)}} k^{-2(\alpha+1)}\right) = O\left(S^2(\eta ST)^{-\alpha/(\alpha+1)}\right). \tag{211}$$

For $\mathscr{E}_2$ we always have

$$\mathscr{E}_2 \le S^2 \sum_{k\le(\eta ST)^{1/(\alpha+1)}} \mathcal{P}_s(k)\min\left(1, \frac{S^2}{r^2}e^{-4\eta ST\mathcal{P}_s(k)}\right) \tag{212}$$

regardless of the size of N, so it suffices to bound this sum. If we denote $l = (\eta ST)^{1/(\alpha+1)}$ and define

$$F_2(z) = \min\left(1, \frac{S^2}{r^2}e^{-4\eta STz}\right), \tag{213}$$

it suffices to show the bound

$$\sum_{k\le l}\mathcal{P}_s(k)F_2(\mathcal{P}_s(k)) = O\left(\left(\log(S/r)\right)^{\alpha/(\alpha+1)}(\eta ST)^{-\alpha/(\alpha+1)}\right). \tag{214}$$

We will approximate this sum as

$$\sum_{k\le l}\mathcal{P}_s(k)F_2(\mathcal{P}_s(k)) = \sum_{k\le l}(\mathcal{P}_s(k)-\mathcal{P}_s(k+1))\frac{\mathcal{P}_s(k)}{\mathcal{P}_s(k+1)-\mathcal{P}_s(k)}F_2(\mathcal{P}_s(k)) \tag{215}$$

$$= \sum_{k\le l}(\mathcal{P}_s(k)-\mathcal{P}_s(k+1))\frac{k^{-\alpha-1}}{(\alpha+1)k^{-\alpha-2}(1+O(k^{-1}))}F_2(\mathcal{P}_s(k)) \tag{216}$$

$$= O\left(\sum_{k\le l}(\mathcal{P}_s(k)-\mathcal{P}_s(k+1))\mathcal{P}_s(k)^{-1/(\alpha+1)}F_2(\mathcal{P}_s(k))\right). \tag{217}$$

to obtain the form of Riemann sum approximation for the integral of

$$\int_{z=\mathcal{P}_s(l)}^{\infty} z^{-1/(\alpha+1)}F_2(z)dz \tag{218}$$

at $\mathcal{P}_s(l) < \mathcal{P}_s(l-1) < \cdots < \mathcal{P}_s(1)$. As $F_2(z)$ is decreasing function, this Riemann sum is always less than the integral, so we obtain

$$\sum_{k\le l}\mathcal{P}_s(k)F_2(\mathcal{P}_s(k)) = O\left(\int_{z=\mathcal{P}_s(l)}^{\infty} z^{-1/(\alpha+1)}F_2(z)dz\right). \tag{219}$$

We note that $\mathcal{P}_s(l) = (\zeta(\alpha+1)\eta ST)^{-1}$. The threshold for $F_2(z)$ to become 1 is given at

$$\frac{S^2}{r^2}e^{-4\eta STz} = 1 \quad \Leftrightarrow \quad z = \frac{1}{2\eta ST}\log\frac{S}{r}. \tag{220}$$

As $r < \sqrt{S}$, this value is always greater than $\mathcal{P}_s(l)$. Thus we can divide our integral as

$$\int_{(\zeta(\alpha+1)\eta ST)^{-1}}^{\infty} z^{-1/(\alpha+1)}F_2(z)dz \tag{221}$$

$$= \int_{(\zeta(\alpha+1)\eta ST)^{-1}}^{(2\eta ST)^{-1}\log(S/r)} z^{-1/(\alpha+1)}dz + \int_{(2\eta ST)^{-1}\log(S/r)}^{\infty} z^{-1/(\alpha+1)}\frac{S^2}{r^2}e^{-4\eta STz}dz. \tag{222}$$

The first part is bounded by

$$\int_{(\zeta(\alpha+1)\eta ST)^{-1}}^{(2\eta ST)^{-1}\log(S/r)} z^{-1/(\alpha+1)}dz = O\left(\left((2\eta ST)^{-1}\log(S/r)\right)^{\alpha/(\alpha+1)}\right) \tag{223}$$

which can be shown to be $O\left((\log(S/r))^{\alpha/(\alpha+1)}(\eta ST)^{-\alpha/(\alpha+1)}\right)$. For the second part, we apply substitution of $w = 4\eta STz$ to show

$$\int_{(2\eta ST)^{-1}\log(S/r)}^{\infty} z^{-1/(\alpha+1)}\frac{S^2}{r^2}e^{-4\eta STz}dz = \frac{S^2}{r^2}(4\eta ST)^{-\alpha/(\alpha+1)}\int_{2\log(S/r)}^{\infty} w^{-1/(\alpha+1)}e^{-w}dw \tag{224}$$

$$= \frac{S^2}{r^2}(4\eta ST)^{-\alpha/(\alpha+1)}\Gamma\left(\frac{\alpha}{\alpha+1}, 2\log\frac{S}{r}\right) \tag{225}$$

and applying the asymptotic $\Gamma(s,x) = O(x^{s-1}e^{-x})$ suggests that this is bounded by

$$\ll \frac{S^2}{r^2}(4\eta ST)^{-\alpha/(\alpha+1)}\left(\log\frac{S}{r}\right)^{-1/(\alpha+1)} e^{-2\log(S/r)} = O\left((\eta ST)^{-\alpha/(\alpha+1)}\right). \tag{226}$$

■

Theorem 1. *(Parameter scaling law) Assume the following conditions: $n_s > N$ with $\lim(N/n_s) = \gamma < 1$ (γ can be zero), and there exists $0 < r < \sqrt{S}$ such that $r < \mathcal{R}_k(0) < S/2$ for all k. If $N, T \to \infty$ while satisfying $N^{\alpha+1} = o(T)$, the expected loss $\mathbf{E}_{\mathcal{D}}[\mathcal{L}]$ for all datasets $\mathcal{D}$ of size D satisfies*

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \frac{S^2(1-\gamma^\alpha)}{2\alpha\zeta(\alpha+1)}N^{-\alpha} + O\left(S^2N^{-\min(\alpha+1,2\alpha)} + S^2\left(\log(S/r)\right)^{\alpha/(\alpha+1)}(\eta ST)^{-\alpha/(\alpha+1)} + S^2D^{-1/2}f_\alpha(N) + S^4\eta^2T^2D^{-1}\right) \tag{227}$$

where

$$f_\alpha(N) = \begin{cases} 1 & \text{if } \alpha > 1 \\ \log N & \text{if } \alpha = 1 \\ N^{(1-\alpha)/2} & \text{if } \alpha < 1. \end{cases} \tag{228}$$

The constant on the O term only depends on α. When $D \gg T^3$, then all the error terms involving D are negligible.

Proof In the situation $n_s = \infty$ and $D = \infty$, Proposition 6 shows that

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \frac{S^2}{2\alpha\zeta(\alpha+1)}N^{-\alpha} + O\left(S^2N^{-(\alpha+1)} + S^2\left(\log(S/r)\right)^{\alpha/(\alpha+1)}(\eta ST)^{-\alpha/(\alpha+1)}\right). \tag{229}$$

We consider the effect of n_s first. As $\mathscr{L}_1$ becomes error term in this estimation, letting n_s as finite value has no effect on overall estimation. The term $\mathscr{L}_2$ accounts for the main term, and letting n_s as finite value changes it to

$$\frac{N^{-\alpha} - n_s^{-\alpha}}{\alpha\zeta(\alpha+1)}\frac{S^2}{2} + O(N^{-\min(\alpha+1,2\alpha)}S^2). \tag{230}$$

This accounts for the factor $(1-\gamma^\alpha)$ on the main term and $O(N^{-\min(\alpha+1,2\alpha)}S^2)$ added to the error term. The effect of D is exactly described in Proposition 4, contributing the error term of $O\left(S^2D^{-1/2}f_\alpha(N) + S^4\eta^2T^2D^{-1}\right)$. Regarding the sufficient condition for D, if $D \gg T^3$ then we have

$$S^4\eta T^2D^{-1} \ll T^{-\alpha/(\alpha+1)}, \quad S^2D^{-1/2}f_\alpha(N) \ll T^{-3/2}N^{1/2} \ll T^{-1} \tag{231}$$

so all error terms involving D are less than $O(T^{-\alpha/(\alpha+1)})$. ■

For the situation $T = O(N^{\alpha+1})$ however, the error terms $\mathscr{E}_1$ and $\mathscr{E}_2$ are of same size, so we can only say that the main term is of $O(S^2(\eta ST)^{-\alpha/(\alpha+1)})$.

Theorem 2. *(Upper bound for time scaling law) Assume the following conditions: $n_s > N$, and there exists there exists $0 < r < \sqrt{S}$ such that $r < \mathcal{R}_k(0) < S/2$ for all k. If $N, T \to \infty$ while satisfying $\eta ST = O(N^{\alpha+1})$, the expected loss $\mathbf{E}_{\mathcal{D}}[\mathcal{L}]$ is*

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = O\left(S^2\left(\log(S/r)\right)^{\alpha/(\alpha+1)}(\eta ST)^{-\alpha/(\alpha+1)} + S^2D^{-1/2}f_\alpha(N) + S^4\eta^2T^2D^{-1}\right) \tag{232}$$

with constant on O only depending on α and $\limsup((\eta ST)^{1/(\alpha+1)}/N)$, with f_α defined as in Theorem 1. If $D \gg NT^2$ and $D \gg T^3$, then all the error terms involving D are negligible.

Proof The error term regarding D can be obtained in the same way as Theorem 1, so we will let $D = \infty$ for the rest of the proof. Also we can let $n_s = \infty$, as we observed that it contributes at most to the constant factor of the upper bound and does not change the scaling.

In the decomposition of Proposition 6, we always have

$$\mathscr{E}_2 = O\left(S^2 \left(\log(S/r)\right)^{\alpha/(\alpha+1)} (\eta ST)^{-\alpha/(\alpha+1)}\right) \tag{233}$$

and

$$\mathscr{E}_1 = O\left(S^2 (\eta ST)^{-\alpha/(\alpha+1)}\right) \tag{234}$$

holding regardless of N, so it only remains to consider $\mathscr{L}_2 + \mathscr{M}$. If $(\eta ST)^{1/(\alpha+1)} < N$, then $\mathscr{L}_2 + \mathscr{M}$ is of size $O\left(S^2(\eta ST)^{-\alpha/(\alpha+1)}\right)$. If $(\eta ST)^{1/(\alpha+1)} \geq N$, then N and $(\eta ST)^{1/(\alpha+1)}$ has same order, so $\mathscr{L}_2 + \mathscr{M} = \mathscr{L}_2 = \Theta(S^2 N^{-\alpha})$ is $O\left(S^2(\eta ST)^{-\alpha/(\alpha+1)}\right)$. Thus in either cases we have the desired bound. ■

Theorem 3. *(Lower bound for time scaling law) Assume the following conditions: $n_s > N$ and $0 < \mathcal{R}_k(0) < S/2$. If $N, T \to \infty$ while satisfying $(8\zeta(\alpha+1)^{-1}\eta ST)^{1/(\alpha+1)} < N$, the expected loss $\mathbf{E}_{\mathcal{D}}[\mathcal{L}]$ is*

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] \geq \kappa S^2 (\eta ST)^{-\alpha/(\alpha+1)} + O\left(\eta^{-1} ST^{-1} + S^2 D^{-1/2} f_\alpha(N) + S^4 \eta^2 T^2 D^{-1}\right) \tag{235}$$

for κ and constant on O only depending on α, with f_α defined as in Theorem 1. If $D \gg NT^2$ and $D \gg T^3$, then all the error terms involving D are negligible.

Proof The error term regarding D can be obtained in the same way as Theorem 1, so we will let $D = \infty$ for the rest of the proof. We only show the lower bound for $\mathscr{L}_1$, holding regardless of N and n_s. In Lemma 7 (Eq. (188)) we have

$$F(z) \geq \frac{(S - r_k)^2}{2} - \frac{8\eta S^3 T}{27} z \geq \frac{S^2}{8} - \frac{8\eta S^3 T}{27} z \tag{236}$$

for $z \geq 0$, so if $z \leq (4\eta ST)^{-1}$ then $F(z) \geq S^2/8 - 2S^2/27 > S^2/20$. The condition $\mathcal{P}_s(k) \leq (4\eta ST)^{-1}$ is equivalent to that $k \geq (4\zeta(\alpha+1)^{-1}\eta ST)^{1/(\alpha+1)}$. In evaluating $\mathscr{L}_1 = \sum_{k=1}^N \mathcal{P}_s(k) F(\mathcal{P}_s(k))$, we will only add over k in range of

$$(4\zeta(\alpha+1)^{-1}\eta ST)^{1/(\alpha+1)} < k < (8\zeta(\alpha+1)^{-1}\eta ST)^{1/(\alpha+1)}. \tag{237}$$

From the assumption this interval sits inside $1 < k < N$. For such k we use use upper bound of $F(\mathcal{P}_s(k)) > S^2/20$. Then by using Proposition 5 we can obtain

$$\mathscr{L}_1 \geq \frac{S^2}{20} \sum_{(4\zeta(\alpha+1)^{-1}\eta ST)^{1/(\alpha+1)} < k < (8\zeta(\alpha+1)^{-1}\eta ST)^{1/(\alpha+1)}} \mathcal{P}_s(k) \tag{238}$$

$$= \frac{S^2}{20}\left(\frac{(\zeta(\alpha+1)^{-1}\eta ST)^{-\alpha/(\alpha+1)}}{\alpha\zeta(\alpha+1)}(4^{-\alpha/(\alpha+1)} - 8^{-\alpha/(\alpha+1)}) + O\left((\eta ST)^{-1}\right)\right). \tag{239}$$

The possible effect of n_s on the main term is to multiply both the main term by and T by $(1 + n_s^{-\alpha})$, so it increases the bound. ■

The condition $(8\zeta(\alpha+1)^{-1}\eta ST)^{1/(\alpha+1)} < N$ is not absolutely necessary for lower bound. The condition $(\eta ST)^{1/(\alpha+1)} = \Theta(N)$ and $n_s \geq 2N$ would suffice and one can formulate similar theorem, although the constant of lower bound might be much smaller if $(\eta ST)^{1/(\alpha+1)}/N$ is small.

Lastly, we provide a simpler version of those results combined and discuss the special case where the optimal compute $C = NT$, or the given engineering budget, is specified.

Corollary 3. *(Summary of large data estimation) Assuming $D \gg NT^2, T^3$ and $n_s \gg N^{1+\epsilon}$ such that effects of n_s and D are negligible, then for $N, T \to \infty$ we have*

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \Theta_{\eta,S,r}\left(\max(N^{-\alpha}, T^{-\alpha/(\alpha+1)})\right), \tag{240}$$

where $\Theta_{\eta,S,r}$ denotes that the implied constant depends on η, S, α and $r = \min \mathcal{R}_k(0) > 0$. In particular, we have

$$N^{\alpha+1} = O(T) \quad \Rightarrow \quad \mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \Theta_{\eta,S,r}(N^{-\alpha}) \tag{241}$$

and

$$T = O(N^{\alpha+1}) \quad \Rightarrow \quad \mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \Theta_{\eta,S,r}(T^{-\alpha/(\alpha+1)}). \tag{242}$$

Proof Apply Theorem 1 if $N^{\alpha+1} = o(T)$ and Theorem 2 and Theorem 3 if $N^{\alpha+1} \gg T$. ■

Corollary 4. *(The 'computationally optimal' case) Denote $C = NT$ and assume the conditions in Corollary 3. Then we have*

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] \gg C^{-\alpha/(\alpha+2)}. \tag{243}$$

When $N = \Theta(C^{1/(\alpha+2)})$ and $T = \Theta(C^{(\alpha+1)/(\alpha+2)})$, we achieve $\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \Theta(C^{-\alpha/(\alpha+2)})$. (Its implied constant may depend on implied constant for growth of N and T.)

Proof The first part follows from

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] \gg \max(N^{-\alpha}, T^{-\alpha/(\alpha+1)}) \tag{244}$$

and

$$\max(N^{-\alpha}, T^{-\alpha/(\alpha+1)}) \geq (N^{-\alpha})^{1/(\alpha+2)}(T^{-\alpha/(\alpha+1)})^{(\alpha+1)/(\alpha+2)} = (NT)^{-\alpha/(\alpha+2)}. \tag{245}$$

The second part can be checked by substituting $(N, T) = (C^{1/(\alpha+2)}, C^{(\alpha+1)/(\alpha+2)})$ (or their constant multiples) to Corollary 3. ■

J.5 Computing the constant for time scaling law

While we have found the time scaling law $\mathbf{E}[\mathcal{L}] = O(T^{-\alpha/(\alpha+1)})$ holding for $T = O(N^{\alpha+1})$, bounds in Theorem 2 and Theorem 3 were chosen rather lazily and do not depict the correct picture. We will find the constant using more refined estimation, but we require additional assumptions on parameters. We will focus on the setting where D and n_s are large enough to be negligible, $\mathcal{R}_k(0) = r$ is fixed, and $T = O(N^{\alpha+1})$ with fixed constant such that time scaling law holds.

Theorem 4. *(Constant for time scaling law) Denote $\mathcal{L}^\infty$ as the loss when $D, n_s \to \infty$ so that their effect is negligible:*

$$\mathcal{L}^\infty = \mathcal{L}^\infty(T, N) = \sum_{k=1}^{N} \mathcal{P}_s(k) \frac{S^2}{2\left(1 + \left(\frac{S}{r} - 1\right)^{-1} e^{2\eta \mathcal{P}_s(k) S T}\right)^2} + \frac{S^2 N^{-\alpha}}{\alpha\zeta(\alpha+1)}. \tag{246}$$

When $T, N \to \infty$ and $\lim N/(\eta S T)^{1/(\alpha+1)} = \lambda$ for a fixed constant $\lambda \in (0, \infty]$, the following limit exists:

$$\mathcal{A}(\lambda) = \lim_{T,N\to\infty} (\eta S T)^{\alpha/(\alpha+1)} \mathcal{L}^\infty(T, N). \tag{247}$$

The prefactor constant $\mathcal{A}$ as the a function of λ (when $\lambda = \infty$ then let $\lambda^{-\alpha} = \lambda^{-(\alpha+1)} = 0$) is

$$\mathcal{A}(\lambda) = \frac{\zeta(\alpha+1)^{-1/(\alpha+1)}}{\alpha} \int_{\lambda^{-(\alpha+1)}/\zeta(\alpha+1)}^{\infty} u^{-1/(\alpha+1)} \Phi_{S,r}(u) du + \frac{S^2}{2\alpha\zeta(\alpha+1)} \lambda^{-\alpha}, \tag{248}$$

where

$$\Phi_{S,r}(u) = \frac{S^2}{2\left(1 + \left(\frac{S}{r} - 1\right)^{-1} e^{2u}\right)^2}. \tag{249}$$

Proof We first observe

$$\mathcal{L}^{\infty} = \sum_{k=1}^{N} \mathcal{P}_s(k)\Phi_{S,r}(\eta ST\mathcal{P}_s(k)) + \frac{S^2N^{-\alpha}}{\alpha\zeta(\alpha+1)}. \tag{250}$$

We will seek to convert it into Riemann sum form of certain integral. We start from noting that

$$\mathcal{P}_s(k) = (\mathcal{P}_s(k) - \mathcal{P}_s(k+1))\frac{k}{\alpha}(1+O(k)) \tag{251}$$

$$= \frac{\zeta(\alpha+1)^{-1/(\alpha+1)}}{\alpha}(\mathcal{P}_s(k) - \mathcal{P}_s(k+1))\mathcal{P}_s(k)^{-1/(\alpha+1)}(1+O(k)) \tag{252}$$

Denote $u_k = \eta ST\mathcal{P}_s(k)$, then the sum can be approximated to

$$\sum_k \mathcal{P}_s(k)\Phi_{S,r}(\eta ST\mathcal{P}_s(k)) \tag{253}$$

$$\approx \sum_k (\mathcal{P}_s(k) - \mathcal{P}_s(k+1))\mathcal{P}_s(k)^{-1/(\alpha+1)}\Phi_{S,r}(\eta ST\mathcal{P}_s(k)) \tag{254}$$

$$= (\eta ST)^{-\alpha/(\alpha+1)} \sum_k (u_k - u_{k+1})u_k^{-1/(\alpha+1)}\Phi_{S,r}(u_k) \tag{255}$$

if we ignore small k. As $\Phi_{S,r}$ is decreasing, this corresponds to Riemann sum taking minimum in the interval $[u_{k+1}, u_k]$. So integral provides an upper bound for this sum. Similarly we can approximate it with Riemann sum taking maximum in $[u_k, u_{k-1}]$ if we use

$$\mathcal{P}_s(k) = \frac{\zeta(\alpha+1)^{-1/(\alpha+1)}}{\alpha}(\mathcal{P}_s(k-1) - \mathcal{P}_s(k))\mathcal{P}_s(k-1)^{-1/(\alpha+1)}(1+O(k)) \tag{256}$$

instead. As $\Phi_{S,r}$ shows exponential decay, we can ignore values at small k, so this shows

$$(\eta ST)^{-\alpha/(\alpha+1)} \sum_k (u_k - u_{k+1})u_k^{-1/(\alpha+1)}\Phi_{S,r}(u_k) \approx \int_{u_N}^{\infty} u^{-1/(\alpha+1)}\Phi_{S,r}(u)du \tag{257}$$

and from that

$$u_N = \eta ST N^{-(\alpha+1)}\zeta(\alpha+1)^{-1} = \lambda^{-(\alpha+1)}\zeta(\alpha+1)^{-1} \tag{258}$$

we obtain our desired result. ■

Theorem 4 basically tells that for $N = \lambda(\eta ST)^{1/(\alpha+1)}$ and D, n_s large enough, we have

$$\mathcal{L} \sim \mathcal{A}(\lambda)(\eta ST)^{-\alpha/(\alpha+1)} \tag{259}$$

with $\mathcal{A}(\lambda)$ given as Eq. (248), thus specifying the constant for time scaling law. For finite λ, this theorem covers the computationally optimal case of $(N, T) = (\lambda_1 C^{1/(\alpha+2)}, \lambda_2 C^{(\alpha+1)/(\alpha+2)})$ for some nonzero constant λ_1, λ_2. For $\lambda = \infty$, it describes the case $T = o(N^{\alpha+1})$ where effect of N is negligible.

Corollary 5. *Denote $\mathcal{L}^{\infty}$ as $\mathcal{L}^{\infty}$ as the loss when $D, n_s \to \infty$ same as Eq. (246). Denote $C = NT$ and suppose that*

$$(N, \eta ST) = (\lambda(\eta SC)^{1/(\alpha+2)}, \lambda^{-1}(\eta SC)^{(\alpha+1)/(\alpha+2)}) \tag{260}$$

for a fixed constant $0 < \lambda < \infty$. Then as $C \to \infty$, we have

$$\mathcal{L}^{\infty} = \mathcal{A}\left(\lambda^{(\alpha+2)/(\alpha+1)}\right)\lambda^{\alpha/(\alpha+1)}(\eta SC)^{-\alpha/(\alpha+2)}(1+o(1)) \tag{261}$$

where $\mathcal{A}$ is given as Eq. (248) of Theorem 4.

Proof As $\lim N/(\eta ST)^{1/(\alpha+1)} = \lambda^{(\alpha+2)/(\alpha+1)}$ under above conditions, we can apply Theorem 4 and substituting Eq. (260) into Eq. (259) gives the desired result. ■

Technically we can optimize $\mathcal{L}^\infty$ for a given fixed value of $C = NT$ by letting λ as argument of minimum of $\mathcal{A}\left(\lambda^{(\alpha+2)/(\alpha+1)}\right)\lambda^{-\alpha/(\alpha+1)}$, although it seems almost impossible to obtain any form of formula for such λ.

Lastly, we provide the following estimate for the time scale constant $(\mathcal{A}(\lambda))$ when r is small, especially the first term in Eq. (248).

Proposition 7. *As $r \to 0$, we have ($\Lambda > 0$ fixed)*

$$\int_\Lambda^\infty u^{-1/(\alpha+1)}\Phi_{S,r}(u)du \approx \left(2\log\frac{S-r}{r}\right)^{\alpha/(\alpha+1)}\frac{S^2(\alpha+1)}{4\alpha}. \tag{262}$$

Proof Denote $M = (\frac{S}{r} - 1)$, and replace u by $(\log M)v$. Then we have

$$\int_\Lambda^\infty u^{-1/(\alpha+1)}\Phi_{S,r}(u)du = (\log M)^{\alpha/(\alpha+1)}\frac{S^2}{2}\int_{\Lambda/\log M}^\infty \frac{v^{-1/(\alpha+1)}dv}{\left(1+M^{2v-1}\right)^2} \tag{263}$$

$$= (\log M)^{\alpha/(\alpha+1)}\frac{S^2}{2}\int_0^\infty 1_{v\geq\Lambda/\log M}\frac{v^{-1/(\alpha+1)}dv}{\left(1+M^{2v-1}\right)^2}. \tag{264}$$

As $M \to \infty$, the integrand converges to

$$\lim_{M\to\infty} 1_{v\geq\Lambda/\log M}\frac{v^{-1/(\alpha+1)}dv}{\left(1+M^{2v-1}\right)^2} = \begin{cases} v^{-1/(\alpha+1)} & \text{if } v \leq 1/2 \\ 0 & \text{if } v > 1/2. \end{cases} \tag{265}$$

The integrand is bounded by $v^{-1/(\alpha+1)}$ if $v \leq 1/2$ and $v^{-1/(\alpha+1)}e^{-2(2v-1)}$ if $v > 1/2$, those of which are all integrable. So we can apply Lebesgue's dominated convergence theorem to show

$$\lim_{M\to\infty}\int_{\Lambda/\log M}^\infty \frac{v^{-1/(\alpha+1)}dv}{\left(1+M^{2v-1}\right)^2} = \int_0^\infty\left(\lim_{M\to\infty} 1_{v\geq\Lambda/\log M}\frac{v^{-1/(\alpha+1)}dv}{\left(1+M^{2v-1}\right)^2}\right) = \int_0^{1/2} v^{-1/(\alpha+1)}dv. \tag{266}$$

Thus we have

$$\lim_{r\to 0}\left(\log\frac{S-r}{r}\right)^{-\alpha/(\alpha+1)}\int_\Lambda^\infty u^{-1/(\alpha+1)}\Phi_{S,r}(u)du = \frac{S^2}{2}\int_0^{1/2} v^{-1/(\alpha+1)}dv = \frac{2^{1/(\alpha+1)}S^2(\alpha+1)}{4\alpha} \tag{267}$$

which can be observed to be equivalent to desired expression of Eq. (262). ■

J.6 Estimates for large T and threshold between data/parameter scaling

The estimates for small D requires different techniques from estimates for large D. We will consider the situation T grows much faster than D and N, and discuss when data scaling law of $\mathcal{L} = \Theta(D^{-\alpha/(\alpha+1)})$ happens. We will consider more simpler setting of '$n_s = \infty$' or equivalently that effects of n_s is negligible ($n_s = \omega(N)$ seems to suffice) and $\mathcal{R}_k(0) = r < S$ is fixed, although it won't be impossible to discuss their subtle effects.

First we single out effect of T by comparing $\mathcal{L}(T)$ and $\mathcal{L}(\infty)$. We remind

$$\mathcal{L}_k(T) = \frac{S^2}{2\left(1+\left(\frac{S}{r}-1\right)^{-1}e^{2\eta d_k ST/D}\right)^2} \tag{50}$$

and its limit when $T \to \infty$ is given as

$$\mathcal{L}_k(\infty) = \lim_{T\to\infty}\mathcal{L}_k(T) = \begin{cases} \frac{(S-r)^2}{2} & \text{if } d_k = 0 \\ 0 & \text{if } d_k > 0. \end{cases} \tag{268}$$

Proposition 8. *Suppose that $\mathcal{R}_k(0) = r < S$ is fixed. For large T, we have*

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}(T)] - \mathbf{E}_{\mathcal{D}}[\mathcal{L}(\infty)] = O\left(S^4 r^{-2} D e^{-4\eta ST/D}\right). \tag{269}$$

Proof As $\mathcal{L}_k(T)$ is decreasing in T, we always have $\mathcal{L}_k(T) \geq \mathcal{L}_k(\infty)$ so therefore

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}(T)] - \mathbf{E}_{\mathcal{D}}[\mathcal{L}(\infty)] \geq 0. \tag{270}$$

So we only need to establish an upper bound for $\mathcal{L}_k(T) - \mathcal{L}_k(\infty)$. We note that $\mathcal{L}_k(T) - \mathcal{L}_k(\infty)$ when $d_k = 0$, so one can write

$$\mathcal{L}_k(T) - \mathcal{L}_k(\infty) = 1_{d_k>0}\mathcal{L}_k(T) \tag{271}$$

where $1_{d_k>0}$ denotes the characteristic function

$$1_{d_k>0} = \begin{cases} 1 & \text{if } d_k > 0 \\ 0 & \text{if } d_k = 0. \end{cases} \tag{272}$$

We use simple bound of

$$\mathcal{L}_k(T) < \frac{S^2}{2\left(\left(\frac{S}{r}-1\right)^{-1} e^{2\eta d_k ST/D}\right)^2} < \frac{S^4}{2} r^{-2} e^{-4\eta d_k ST/D}. \tag{273}$$

As d_k follows binomial distribution $B(D, \mathcal{P}_s(k))$, considering its moment generating function gives

$$\mathbf{E}_{d_k}[e^{-4\eta d_k ST/D}] = \left(1 - \mathcal{P}_s(k) + \mathcal{P}_s(k) e^{-4\eta ST/D}\right)^D \tag{274}$$

so thus

$$\mathbf{E}_{d_k}[1_{d_k>0} e^{-4\eta d_k ST/D}] = \left(1 - \mathcal{P}_s(k) + \mathcal{P}_s(k) e^{-4\eta ST/D}\right)^D - (1 - \mathcal{P}_s(k))^D. \tag{275}$$

Meanwhile, for $0 \leq u, v \leq 1$ real numbers, we have

$$|u^D - v^D| = |u - v||u^{D-1} + u^{D-2}v + \cdots + v^{D-1}| \leq D|u - v| \tag{276}$$

so applying this inequality to above gives

$$\mathbf{E}_{d_k}[1_{d_k>0} e^{-4\eta d_k ST/D}] \leq D\mathcal{P}_s(k) e^{-4\eta ST/D}. \tag{277}$$

Thus we can deduce

$$\begin{aligned} \mathbf{E}_{d_k}[\mathcal{L}_k(T)] - \mathbf{E}_{d_k}[\mathcal{L}_k(\infty)] &= \mathbf{E}_{d_k}[1_{d_k>0}\mathcal{L}_k(T)] && (278) \\ &< \frac{S^4 r^{-2}}{2} \mathbf{E}_{d_k}[1_{d_k>0} e^{-4\eta d_k ST/D}] && (279) \\ &\leq \frac{S^4 r^{-2}}{2} D e^{-4\eta ST/D} \mathcal{P}_s(k) && (280) \end{aligned}$$

and thus

$$0 \leq \mathbf{E}_{\mathcal{D}}[\mathcal{L}(T)] - \mathbf{E}_{\mathcal{D}}[\mathcal{L}(\infty)] < \frac{S^4 r^{-2}}{2} D e^{-4\eta ST/D} \sum_{k=1}^{\infty} \mathcal{P}_s(k)^2 = O\left(S^4 r^{-2} D e^{-4\eta ST/D}\right). \tag{281}$$

■

This provides an almost complete account for the effect of very large T. We will let $T = \infty$ from this point. We have

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}(\infty)] = \frac{(S-r)^2}{2} \sum_{k=1}^{N} \mathcal{P}_s(k)(1 - \mathcal{P}_s(k))^D + \frac{S^2}{2} \sum_{k=N+1}^{\infty} \mathcal{P}_s(k). \tag{282}$$

Applying Lemma 6 gives

$$\sum_{k=N+1}^{\infty} \mathcal{P}_s(k) = \frac{N^{-\alpha}}{\alpha\zeta(\alpha+1)} + O(N^{-\alpha-1}) \tag{283}$$

so it suffices to focus on the first sum. We will divide the range of k into two $1 \le k \le M$ and $M < k \le N$. For the sum over $1 \le k \le M$, we will apply the following simple bound (in the last part we used $1 - x \le e^{-x}$)

$$0 \le \sum_{k=1}^{M} \mathcal{P}_s(k)(1-\mathcal{P}_s(k))^D \le (1-\mathcal{P}_s(M))^D \le e^{-\mathcal{P}_s(M)D}. \tag{284}$$

For the sum over $M < k \le N$, we will approximate the sum into some integral, which happens to be incomplete gamma function.

Proposition 9. *For $2 < M < N$ integers, we have*

$$\sum_{k=M+1}^{N} \mathcal{P}_s(k)(1-\mathcal{P}_s(k))^D \tag{285}$$

$$= D^{-\alpha/(\alpha+1)} \frac{\zeta(\alpha+1)^{-1/(\alpha+1)}}{\alpha} \left(\Gamma\left(\frac{\alpha}{\alpha+1}, D\mathcal{P}_s(N)\right) - \Gamma\left(\frac{\alpha}{\alpha+1}, D\mathcal{P}_s(M)\right)\right) \tag{286}$$

$$+ O\left(D^{-(2\alpha+1)/(\alpha+1)} + D^{-\alpha/(\alpha+1)}M^{-1}\right). \tag{287}$$

Here Γ denotes the incomplete gamma function

$$\Gamma(s,x) = \int_x^{\infty} y^{s-1}e^{-y}dy. \tag{288}$$

Proof Consider the interval $[\mathcal{P}_s(N), \mathcal{P}_s(M)]$ and its partition $\mathscr{P} = \{\mathcal{P}_s(N) < \mathcal{P}_s(N-1) < \cdots < \mathcal{P}_s(M)\}$. For a function $f(x) = x^{-1/(\alpha+1)}(1-x)^D$, we will consider its upper and lower Darboux sums with respect to $\mathcal{P}$. As f is decreasing in $(0,1]$, its upper and lower Darboux sums are given respectively as

$$U(f,\mathscr{P}) = \sum_{k=M}^{N-1} (\mathcal{P}_s(k) - \mathcal{P}_s(k+1))\mathcal{P}_s(k+1)^{-1/(\alpha+1)}(1-\mathcal{P}_s(k+1))^D \tag{289}$$

$$L(f,\mathscr{P}) = \sum_{k=M}^{N-1} (\mathcal{P}_s(k) - \mathcal{P}_s(k+1))\mathcal{P}_s(k)^{-1/(\alpha+1)}(1-\mathcal{P}_s(k))^D. \tag{290}$$

and those give bound of the integral of f as

$$L(f,\mathscr{P}) \le \int_{\mathcal{P}_s(N)}^{\mathcal{P}_s(M)} f(x)dx \le U(f,\mathscr{P}). \tag{291}$$

Meanwhile, by noting that

$$\mathcal{P}_s(k) = \frac{\zeta(\alpha+1)^{-1/(\alpha+1)}}{\alpha}(\mathcal{P}_s(k) - \mathcal{P}_s(k+1))\mathcal{P}_s(k)^{-1/(\alpha+1)}(1+O(k^{-1})) \tag{292}$$

one can show

$$\sum_{k=M}^{N} \mathcal{P}_s(k)(1-\mathcal{P}_s(k))^D \tag{293}$$

$$= \frac{\zeta(\alpha+1)^{-1/(\alpha+1)}}{\alpha} \left(\sum_{k=M}^{N-1} (\mathcal{P}_s(k) - \mathcal{P}_s(k+1))\mathcal{P}_s(k)^{-1/(\alpha+1)}(1-\mathcal{P}_s(k))^D \right)(1+O(M^{-1})) \tag{294}$$

$$= \frac{\zeta(\alpha+1)^{-1/(\alpha+1)}}{\alpha} L(f,\mathscr{P})(1+O(M^{-1})). \tag{295}$$

Applying similar argument for upper Darboux sum gives

$$\sum_{k=M}^{N} \mathcal{P}_s(k)(1-\mathcal{P}_s(k))^D = \frac{\zeta(\alpha+1)^{-1/(\alpha+1)}}{\alpha} U(f,\mathscr{P})(1+O(M^{-1})) \tag{296}$$

and from Eq. (291) it follows

$$\sum_{k=M}^{N} \mathcal{P}_s(k)(1-\mathcal{P}_s(k))^D = \frac{\zeta(\alpha+1)^{-1/(\alpha+1)}}{\alpha} \left(\int_{\mathcal{P}_s(N)}^{\mathcal{P}_s(M)} x^{-1/(\alpha+1)}(1-x)^D dx \right) (1+O(M^{-1})). \tag{297}$$

From now we will estimate the integral

$$\int_{\mathcal{P}_s(N)}^{\mathcal{P}_s(M)} x^{-1/(\alpha+1)}(1-x)^D dx. \tag{298}$$

We replace $x = y/D$ in the integral inside, then it becomes

$$\int_{\mathcal{P}_s(N)}^{\mathcal{P}_s(M)} x^{-1/(\alpha+1)}(1-x)^D dx = D^{-\alpha/(\alpha+1)} \int_{D\mathcal{P}_s(N)}^{D\mathcal{P}_s(M)} y^{-1/(\alpha+1)} \left(1-\frac{y}{D}\right)^D dy. \tag{299}$$

We want to approximate $\left(1-\frac{y}{D}\right)^D$ by e^{-y}, so we will estimate difference between them. We have

$$D\log(1-y/D) = -y - \sum_{k=2}^{\infty} \frac{y^k}{kD^{k-1}} \tag{300}$$

so if $D > 2y$ then

$$-y > D\log(1-y/D) = -y - \frac{1}{D}\sum_{k=2}^{\infty} \frac{y^k}{kD^{k-2}} > -y - \frac{1}{D}\sum_{k=2}^{\infty} \frac{y^k}{2(2y)^{k-2}} = -y - \frac{y^2}{D} \tag{301}$$

so

$$e^{-y}\left(1-\frac{y^2}{D}\right) < e^{-y}e^{-y^2/D} < \left(1-\frac{y}{D}\right)^D < e^{-y}, \tag{302}$$

where we used the inequality $1-x \le e^{-x}$. As $\mathcal{P}_s(M) < 1/2$ if $M > 2$ (obvious from $\mathcal{P}_s(M) < (\mathcal{P}_s(1)+\mathcal{P}_s(2))/2 < 1/2$), any y in the interval $[D\mathcal{P}_s(N), D\mathcal{P}_s(M)]$ satisfies $D > 2y$. So we can apply this approximation in every y. It follows that

$$\int_{D\mathcal{P}_s(N)}^{D\mathcal{P}_s(M)} y^{-1/(\alpha+1)} \left(1-\frac{y}{D}\right)^D dy \tag{303}$$

$$= \int_{D\mathcal{P}_s(N)}^{D\mathcal{P}_s(M)} y^{-1/(\alpha+1)} e^{-y} dy + O\left(\int_{D\mathcal{P}_s(N)}^{D\mathcal{P}_s(M)} y^{-1/(\alpha+1)} e^{-y} \frac{y^2}{D} dy \right) \tag{304}$$

$$= \int_{D\mathcal{P}_s(N)}^{D\mathcal{P}_s(M)} y^{-1/(\alpha+1)} e^{-y} dy + O\left(D^{-1} \int_0^{\infty} y^{-1/(\alpha+1)} e^{-y} y^2 dy \right) \tag{305}$$

$$= \Gamma\left(\frac{\alpha}{\alpha+1}, D\mathcal{P}_s(N)\right) - \Gamma\left(\frac{\alpha}{\alpha+1}, D\mathcal{P}_s(M)\right) + O(D^{-1}). \tag{306}$$

Combining this with Eq. (297) and Eq. (299) gives the desired result. ■

We combine Proposition 8 and Proposition 9 together to obtain this final estimation result.

Theorem 5. *(Scaling laws for large time estimation) Suppose that $N, D \to \infty$ and $n_s \gg N^{1+\epsilon}$ for some $\epsilon > 0$ so that effect of n_s is negligible. Suppose that $\mathcal{R}_k(0) = r$ for all $1 \le k \le N$.*

1. *(Parameter scaling law) If $N = o(D^{1/(\alpha+1)})$, then we have*

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \frac{S^2}{2\alpha\zeta(\alpha+1)} N^{-\alpha} + O\left(S^2 D^{-\alpha/(\alpha+1)} + S^2 N^{-\alpha-1} + S^4 r^{-2} D e^{-4\eta S T/D} \right). \tag{307}$$

2. *(Data scaling law) If $D = O(N^{\alpha+1})$ and $\mu = \lim(D/N^{\alpha+1})$ exists (it can be zero), then*

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = D^{-\alpha/(\alpha+1)} \left(\frac{(S-r)^2 \zeta(\alpha+1)^{-1/(\alpha+1)}}{2\alpha} \Gamma\left(\frac{\alpha}{\alpha+1}, \frac{D}{N^{\alpha+1}\zeta(\alpha+1)} \right) + \frac{S^2 (D/N^{\alpha+1})^{\alpha/(\alpha+1)}}{2\alpha\zeta(\alpha+1)} \right) + O\left(S^2 D^{-(2\alpha+1)/(2\alpha+2)} + S^4 r^{-2} D e^{-4\eta S T/D} \right) \tag{308}$$

Here Γ denotes the incomplete gamma function

$$\Gamma(s,x) = \int_x^\infty y^{s-1} e^{-y} dy. \tag{309}$$

In particular, if $D = o(N^{\alpha+1})$ such that $\mu = 0$, we have

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = D^{-\alpha/(\alpha+1)} \frac{(S-r)^2 \zeta(\alpha+1)^{-1/(\alpha+1)}}{2\alpha} \Gamma\left(\frac{\alpha}{\alpha+1} \right) (1 + o(1)) + O\left(S^4 r^{-2} D e^{-4\eta S T/D} \right). \tag{310}$$

In either cases, $T \gg D(\log D)^{1+\epsilon}$ for some $\epsilon > 0$ implies that error terms involving T are negligible.

Proof Proposition 8 states

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}(T)] - \mathbf{E}_{\mathcal{D}}[\mathcal{L}(\infty)] = O\left(S^4 r^{-2} D e^{-4\eta S T/D} \right) \tag{269}$$

and we showed

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}(\infty)] = \frac{(S-r)^2}{2} \sum_{k=1}^{N} \mathcal{P}_s(k)(1 - \mathcal{P}_s(k))^D + \frac{S^2}{2} \sum_{k=N+1}^{\infty} \mathcal{P}_s(k) \tag{282}$$

and

$$\sum_{k=N+1}^{\infty} \mathcal{P}_s(k) = \frac{N^{-\alpha}}{\alpha\zeta(\alpha+1)} + O(N^{-\alpha-1}). \tag{283}$$

For the sum of $\mathcal{P}_s(k)(1 - \mathcal{P}_s(k))^D$ over $1 \le k \le N$, we use the estimate (see Eq. (284)) of

$$\sum_{k=1}^{M} \mathcal{P}_s(k)(1 - \mathcal{P}_s(k))^D = O\left(e^{-\mathcal{P}_s(M) D} \right) \tag{311}$$

and the estimate of Proposition 9. Combining all those gives

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] \tag{312}$$

$$= \frac{S^2 N^{-\alpha}}{2\alpha\zeta(\alpha+1)} \tag{313}$$

$$+ D^{-\alpha/(\alpha+1)} \frac{(S-r)^2 \zeta(\alpha+1)^{-1/(\alpha+1)}}{2\alpha} \left(\Gamma\left(\frac{\alpha}{\alpha+1}, D\mathcal{P}_s(N) \right) - \Gamma\left(\frac{\alpha}{\alpha+1}, D\mathcal{P}_s(M) \right) \right) \tag{314}$$

$$+ O\left(S^2 (D^{-(2\alpha+1)/(\alpha+1)} + D^{-\alpha/(\alpha+1)} M^{-1} + N^{-\alpha-1} + e^{-\mathcal{P}_s(M) D}) + S^4 r^{-2} e^{-4\eta S T/D} \right). \tag{315}$$

We will prove our main statement by choosing appropriate M depending on size comparison between D and N.

1. If $N = o(D^{1/(\alpha+1)})$, then we let $M = 3$, and also regard all incomplete gamma function values as $O(1)$. Then it follows

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \frac{S^2 N^{-\alpha}}{2\alpha\zeta(\alpha+1)} + O\left(S^2 D^{-\alpha/(\alpha+1)} + S^2 N^{-\alpha-1} + S^4 r^{-2} e^{-4\eta S T/D} \right) \tag{316}$$

and thus obtaining the parameter scaling law.

2. Suppose $D = O(N^{\alpha+1})$ and $\mu = \lim(D/N^{\alpha+1})$ exists. We want

$$D^{-\alpha/(\alpha+1)} \frac{(S-r)^2 \zeta(\alpha+1)^{-1/(\alpha+1)}}{2\alpha} \Gamma\left(\frac{\alpha}{\alpha+1}, D\mathcal{P}_s(N)\right) + \frac{S^2 N^{-\alpha}}{2\alpha\zeta(\alpha+1)} \tag{317}$$

to be our main term, and set $M < N$ such that the term

$$S^2 D^{-\alpha/(\alpha+1)} \Gamma\left(\frac{\alpha}{\alpha+1}, D\mathcal{P}_s(M)\right) \tag{318}$$

and error terms not depending on T given as

$$O\left(S^2(D^{-(2\alpha+1)/(\alpha+1)} + D^{-\alpha/(\alpha+1)}M^{-1} + N^{-\alpha-1} + e^{-\mathcal{P}_s(M)D})\right) \tag{319}$$

are all bounded by $O(D^{-(2\alpha+1)/(2\alpha+2)})$. Set $M = D^{1/(2\alpha+2)}$. Then $\mathcal{P}_s(M) = D^{-1/2}/\zeta(\alpha+1)$, so applying the asymptotic $\Gamma(s,x) = O(x^{s-1}e^{-x})$ gives

$$\Gamma\left(\frac{\alpha}{\alpha+1}, D\mathcal{P}_s(M)\right) = O\left(D^{-1/2(\alpha+1)} e^{-\sqrt{D}/\zeta(\alpha+1)}\right). \tag{320}$$

This term and $e^{-\mathcal{P}_s(M)D} = e^{-\sqrt{D}/\zeta(\alpha+1)}$ are less than $D^{-\alpha/(\alpha+1)}M^{-1} = O(D^{-(2\alpha+1)/(2\alpha+2)})$, and obviously $D^{-(2\alpha+1)/(\alpha+1)}$ is less than $D^{-(2\alpha+1)/(2\alpha+2)}$. Thus it follows that

$$\begin{aligned}\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = D^{-\alpha/(\alpha+1)} \frac{(S-r)^2 \zeta(\alpha+1)^{-1/(\alpha+1)}}{2\alpha} \Gamma\left(\frac{\alpha}{\alpha+1}, \frac{D}{N^{\alpha+1}\zeta(\alpha+1)}\right) + \frac{S^2 N^{-\alpha}}{2\alpha\zeta(\alpha+1)} \\ + O\left(S^2 D^{-(2\alpha+1)/(2\alpha+2)} + S^4 r^{-2} D e^{-4\eta S T/D}\right).\end{aligned} \tag{321}$$

Regarding the final statement regarding sufficient condition for large T, $T \gg D(\log D)^{1+\epsilon}$ implies

$$De^{-4\eta ST/D} < De^{-4\eta S(\log D)^{1+\epsilon}} < D \cdot D^{-4\eta S(\log D)^{\epsilon}} \ll D^{-K} \tag{322}$$

for any $K > 0$, showing that the error term $O\left(S^4 r^{-2} D e^{-4\eta ST/D}\right)$ is negligible compared to all other error terms of Eq. (307) and Eq. (308). ■

We also provide a summary of all large time estimation results.

Corollary 6. *(Summary of large time estimation) Assuming $T \gg D(\log D)^{1+\epsilon}$ and $n_s \gg N^{1+\epsilon}$ such that effects of n_s and T are negligible, and $\mathcal{R}_k(0) = r$ for all $1 \le k \le N$. Then for $D, N \to \infty$, we have*

$$\mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \Theta_{\eta,S,r}\left(\max(N^{-\alpha}, D^{-\alpha/(\alpha+1)})\right), \tag{323}$$

where $\Theta_{\eta,S,r}$ denotes that the implied constant depends on η, S, r and α. In particular, we have

$$N^{\alpha+1} = O(D) \quad \Rightarrow \quad \mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \Theta_{\eta,S,r}(N^{-\alpha}) \tag{324}$$

and

$$D = O(N^{\alpha+1}) \quad \Rightarrow \quad \mathbf{E}_{\mathcal{D}}[\mathcal{L}] = \Theta_{\eta,S,r}(D^{-\alpha/(\alpha+1)}). \tag{325}$$

Proof Just summarize the results of Theorem 5. ■