An Information Theory of Compute-Optimal Size Scaling, Emergence, and Plateaus in Language Models

Anuj K. Nayak **Lav R. Varshney**
University of Illinois at Urbana-Champaign
{anujk4, varshney}@illinois.edu

Abstract

Recent empirical studies show three phenomena with increasing size of language models: *compute-optimal size scaling*, *emergent capabilities*, and *performance plateauing*. We present a simple unified mathematical framework to explain all of these language model scaling phenomena, building on recent skill-text bipartite graph frameworks for semantic learning. Modeling the learning of concepts from texts as an iterative process yields an analogy to iterative decoding of low-density parity check (LDPC) codes in information theory. Thence, drawing on finite-size scaling characterizations of LDPC decoding, we derive the compute-optimal size scaling (Chinchilla rule) for language models. Further, using tools from random network theory, we provide a simple explanation for both emergence of complex skills and plateauing of performance as the size of language models scale. We see multiple plateaus.

1 Introduction

To optimally use computational resources when training language models, several recent studies have empirically investigated how model size and dataset size should scale with compute budget [12, 9], finding a certain *allometric rule* much like in mathematical biology [24, 8]. As the sizes of language models continue to increase, large improvements in performance have been observed in certain complex tasks with only a small improvement in the model's loss [25] (but see [21]). The larger language models are therefore said to exhibit *emergent capabilities* on complex tasks. More recently, there has been prevalent discourse in the AI community that further increases in language model size lead to *plateauing* of performance [6, 20]. Although, there have been attempts to explain one or two of these empirical phenomena, a unified mathematical framework that explains all three of these empirically observed phenomena is lacking.

To provide simple and insightful explanations of empirical phenomena, several abstract frameworks have been proposed [2, 15, 17], all based on a skill-text bipartite graph that operates at a semantic level and captures key real-world properties [26]. Arora and Goyal [2] explain emergent phenomena by assuming a compute-optimal size scaling rule (Chinchilla allometry rule) [9]. Liao et al. [15] also assume compute-optimal (Chinchilla) size scaling to explain emergence. Michaud et al. [17] assume power-law scaling and that each text piece contains only one skill, which may be very different than real-world scenarios. These existing frameworks explain neither the Chinchilla rule nor the plateau phenomenon. These three frameworks abstract the gradient dynamics of language model training [2]; an alternate mathematical framework considers dynamics to explain the Chinchilla rule and loss function plateaus but does not consider emergence [5].

Here we take an approach that builds on information and coding theory [16] that does so, and also predicts multiple plateaus. In particular, we draw on mathematical ideas around low-density parity check (LDPC) codes (which achieve Shannon optimality) [23, 19] and random graph theory [3].

Machine Learning and Compression Workshop at the 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Though statistical language modeling and information theory were introduced in the same paper [22], modern connections between the two are still fairly limited, cf. [4].

Our information-theoretic approach is inspired by skill-text bipartite graph frameworks of [2, 15, 17] and is closest to [15]. We make a small modification by separating notions of concepts and skills, as in well-established human cognitive architectures [18] that have simple hierarchies [14, 1, 13]. The key difference in our work is to have much more detailed and expressive analysis using non-asymptotic techniques rather than asymptotic ones [7]. Indeed, such finitary analysis is necessary to even consider size scaling. Recall that [2, 15] assume Chinchilla scaling, whereas we derive it without it being built into our framework. Further, with the help of random network theory, we provide a simple explanation for emergent abilities of language models in complex tasks when their sizes exceed a certain threshold. We show that plateauing of performance with size-scaling is just a consequence of diversity of skills required for a task. Moreover, plateauing indicates the possibility of multiple emergences as language models continue to scale further. Our work is a step in the direction of grounding empirical phenomena observed due to scaling of language models on a rigorous mathematical footing. Since our work provides a mathematical explanation for scaling laws and emergent abilities, it also helps in policy making by providing insight into the relationship between capabilities and resources such as data and compute [10].

2 Graph-based framework

Our framework is based on the notion of learning as two levels. First, a set of concepts are learnt from a set of texts with each text involving one or more skills. Second, learning concepts enables the language model to acquire skills, and after encountering a sufficient number of texts with co-occurring pairs of skills, it eventually acquires compositional abilities resulting in emergent phenomena in various complex tasks. The framework naturally leads to information-theoretic analysis in Section 3.

2.1 Texts, concepts, and skills

A set of tokens constitute a text piece from which a language model can learn a wide variety of concepts. This is modeled as a concept-text bipartite graph similar to the skill-text bipartite graph in [15]. In a given training session (single epoch training), a language model chooses to learn only a subset of concepts from a text piece. The total number of skills a model can learn depends on its size. Here we consider a hierarchy of skills: basic skills in the first layer and multiple layers of advanced skills. Basic skills are easily acquired from concepts, whereas acquiring advanced skills additionally requires certain prerequisite skills. We formalize the above notions in the subsequent sections.

2.2 Notation

Let $\mathcal{T}$ be a subset of text pieces from a set $\mathfrak{T}$, and let $\mathcal{R}$ be a subset of concepts from a set $\mathfrak{R}$. Let the model size N (number of parameters) be proportional to the number of concepts $R = |\mathcal{R}|$, i.e., $N = \varsigma R$, for some $\varsigma > 0$.[1] Similarly, let τ be the number of tokens in a text piece $t \in \mathcal{T}$ with $T = |\mathcal{T}|$, implying that the dataset size $D = \tau T$. For a given compute budget C,[2] a language model of size N can be trained using a dataset of size D so the constraint $6ND \leq C$ is satisfied (see [9]).

Correspondingly, for a given compute budget, $G_1^{(C)} = (\mathcal{T} \cup \mathcal{R}, E_{\mathcal{TR}})$ denotes a concept-text bipartite graph, where an edge $e_{tr} \in E_{\mathcal{TR}}$ indicates that the language model can learn concept r from text t. Let the degrees of text pieces (number of skills required to understand a text) be binomially distributed with a fixed mean degree d_t, i.e., $P_R = \text{Binomial}(n, p) = \text{Binomial}(R, d_t/R)$. The corresponding generating function is $P_R(x) = \sum_i P_i x^i$. Let the degree distribution of concepts be $L_T = \text{Binomial}(T, d_r/T)$, where $d_r = d_t T/R$. Note that $d_t/R = d_r/T =: p$. There is an alternate point of view: If we assume that there exists an edge between a text piece and a concept with probability d_t/T, then a typical graph will have text and concept degree distributions close to P_R and L_T, respectively. It is generally useful to view degree distribution from an edge-perspective, which is $\lambda_T(x) = L'_T(x)/L'_T(1)$ and $\rho_R(x) = P'_R(x)/P'_R(1)$ [19].

[1] Here, a *concept* is similar to a *skill quantum* in [17].

[2] Compute budget is measured in number of floating point operations or FLOPs [9].

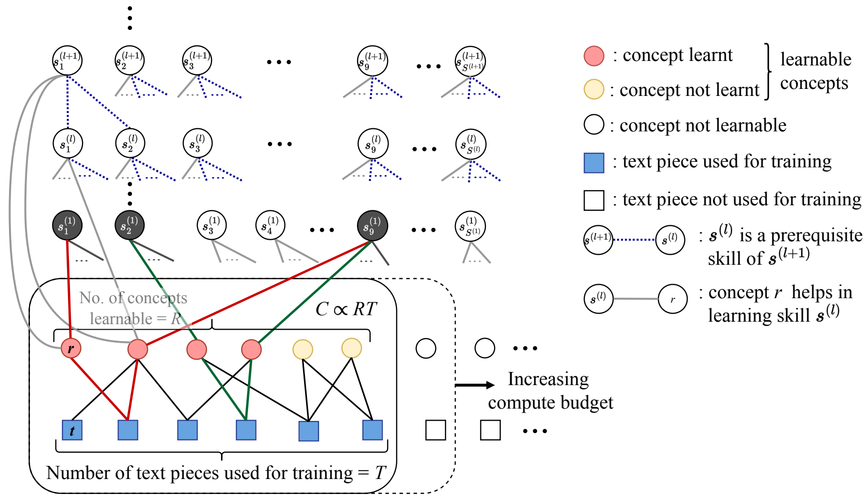

Figure 1: A unified graph-based framework of learning concepts and skills by language models.

Let $G_2 = (\mathfrak{R} \cup \mathcal{S}, E_{\mathfrak{R}\mathcal{S}})$ be a skill-concept graph, where $\mathcal{S} = \cup_l \mathcal{S}^{(l)}$ denotes a set of hierarchical skills, with finite number $S^{(l)}$ of skills in each level l. Each concept is connected to a unique skill at every level l, i.e., each concept enables learning of one skill at each level, and each skill $s^{(l)}$ is connected to σ_l prerequisite skills at level $l-1$. Our unified framework is represented by the graph $G^{(C)} = G_1^{(C)} \cup G_2$ as shown in Figure 1.

2.3 Learning concepts from text pieces

Following the approach described in [15], we assume that a language model learns concepts from text pieces as an iterative peeling process. Let $\mathcal{R}_+^{(u)}$ denote the set of concepts learnt, and $\mathcal{R}_-^{(u)}$ denote the set of concepts not learnt in peeling iteration u. Initially, all the concepts are unlearned, i.e., $\mathcal{R}_-^{(0)} = \mathcal{R}$ and $\mathcal{R}_-^{(0)} = \emptyset$. Next, a language model learns a concept $r \in \mathcal{R}_-^{(0)}$ if a text piece $t \in \mathcal{T}$ is uniquely connected to r yielding $\mathcal{R}_+^{(1)} = \{r\}$ and $\mathcal{R}_-^{(1)} = \mathcal{R}_-^{(0)} \setminus \{r\}$. Before the next iteration, the edge e_{tr} and concept node r from the graph are removed. The next iteration starts by finding another text piece uniquely connected to a concept in $\mathcal{R}_-^{(1)}$, and the process continues until there is either no more text piece/s connected to a unique concept in $\mathcal{R}_-$ or all the concepts are learnt, i.e., $\mathcal{R}_+ = \mathcal{R}$.

2.4 Acquisition of skills and composition of skills

A skill $s^{(l+1)}$ at level $l+1$ is considered acquired when two conditions hold: 1) all the σ_{l+1} prerequisite skills at the lower level l are learnt, and 2) at least one concept associated with $s^{(l+1)}$ is learnt. A pair of concepts (r_1, r_2) is considered connected (denoted by $r_1 - r_2$) if there is a path $r_1 - t - r_2$ through at least one text $t \in \mathcal{T}$. Then, for a fixed level l, a skill-graph $G_2^{(l)} = (\mathcal{S}^{(l)}, E_{\mathcal{S}^{(l)} \times \mathcal{S}^{(l)}})$ is constructed as follows: A pair of skills s_1 and s_2 in $\mathcal{S}^{(l)}$ has a direct link (i.e., $e_{s_1 s_2} \in E_{\mathcal{S}^{(l)} \times \mathcal{S}^{(l)}}$) if there are at least η_l distinct paths $s_1^{(l)} - r_1 - r_2 - s_2^{(l)}$ (with at least η_l distinct pairs of concepts (r_1, r_2)), and all the $2\sigma_l$ prerequisite skills required for both skills are acquired. The intuition behind this construction is that a pair of skills is connected (and therefore can be composed) if they co-occur sufficiently many times through distinct pairs of concepts in the training data, and all prerequisite skills of both skills are already acquired. Further, since more advanced skills are generally hard to learn, skills at higher levels (larger values of l) need larger values of η_l.

2.5 Defining emergence

As the model size increases there is a sharp increase in performance (e.g. accuracy) of the language model on certain complex tasks which the model was not trained on known as emergent phenomena in language models [25]. In this context, there are several definitions of skill emergence in the literature [2, 15, 21, 17]. In our framework, advanced skills (larger l) are acquired from concepts and more basic skills, rather than directly from text pieces. To describe the composition of skills not seen in training, we begin by asserting transitivity of skill composition for a fixed skill level l: if the training data contains enough text pieces with composition of both pairs $(s_1^{(l)}, s_2^{(l)})$ and $(s_2^{(l)}, s_3^{(l)})$, then a language model is capable of composing skill $s_1^{(l)}$ and $s_3^{(l)}$. Consequently, a language model successfully performs a sub-task requiring a composition of a set of skills $\mathcal{S}_\theta^{(l)} \subseteq \mathcal{S}^{(l)}$ if there is a path between every pair of skills belonging to $\mathcal{S}_\theta^{(l)}$ in graph $G_2^{(l)}$. For small compute budgets, dataset size corresponding to compute-optimal performance is small, in which case the training data contains composition of only a small number of skill pairs. As compute budget increases, the size of the training data increases, and therefore the number of composed skill pairs seen by the language model during training increases. Beyond a certain compute-budget threshold and due to skill composition transitivity, the ability of the language model to compose most skill pairs emerges, appearing as a phase transition around this compute-budget threshold, which we call as skill emergence. As we will see in Section 3.3, this phase transition is related to the appearance of a giant connected component (GCC) in random graphs with increasing edge probability. Our definition of emergence exhibits phase transition as empirically observed in language models, and our finitary analysis helps in conforming to the definition of emergence in [25].

3 Explaining all three phenomena

Using the framework in Section 2, we aim to explain the compute-optimal (Chinchilla) scaling rule by applying non-asymptotic information-theoretic tools to the bipartite graph $G_1^{(C)}$, and explain emergence and plateauing phenomena based on the density of connections in the skill-graphs $\{G_2^{(l)}\}_l$.

3.1 Compute-optimal scaling rule

Let $\mathcal{R}_+ \subseteq \mathcal{R}$ denote the set of concepts learnt after the peeling process terminates. The goal of the language model is to maximize the number of concepts learnt under the compute budget constraint C, which yields the following constrained optimization problem.

$$\begin{aligned} \underset{R,T}{\text{maximize}}\ & \mathbb{E}_{G_1^{(C)} \sim (\lambda_T, \rho_R)}[R_+] \\ \text{s.t. } & RT \leq C', \end{aligned} \tag{1}$$

where the number of model parameters $N = \varsigma R$, number of tokens in a text piece is τ, $C' = \frac{C}{6\,\varsigma\,\tau}$, and (R^*, T^*) is the maximizer of the objective function in (1). For a bipartite graph sampled from a degree distribution pair (λ_T, ρ_R), computing the exact number of learned concepts is computationally expensive. Fortunately, the observation that the peeling process is equivalent to iterative decoding of LDPC codes when the codeword symbols are corrupted by erasure, allows us to sidestep this difficulty. Before providing an expression for the objective function, some notations are as follows: let $f(x, \epsilon) = \epsilon \lambda_T(1 - \widetilde{\rho}_R(1 - x))$, then the decoding threshold $\epsilon^* = \inf\{\epsilon \in [0, 1] : x = f(x, \epsilon)$ has a solution in $x \in (0, 1]\}$, x^* be a critical point satisfying $x^* = f(x^*, \epsilon^*)$, $\nu^* = \epsilon^*\ L_T(1 - \widetilde{\rho}_{\mathcal{R}}(1 - x^*))$. The objective function in (1) is given by (see Appendix C.2 for more details):

$$\mathbb{E}_{G_1^{(C)} \sim (\lambda_T, \rho_R)}[R_+] = R\left(1 - \frac{P_{b,\lambda_T,\widetilde{\rho}_R}}{\epsilon}\right) \approx R\left(1 - \frac{\nu^*}{\epsilon} Q\left(\sqrt{\frac{R}{\epsilon}}\frac{(\epsilon^* - \epsilon)}{\alpha}\right)\right), \tag{2}$$

where α depends on $(\lambda_T, \widetilde{\rho}_R)$, and $Q(\cdot)$ is the complementary Gaussian cumulative distribution function.

Compute optimal size scaling of model size and dataset size with increasing compute budget obtained by numerically solving (1) is shown in Figure 2(a) (also see Appendix A for more insights). The curves being parallel in logarithmic scale indicates that N and D must scale equally with C. In this figure, we set $\varsigma = 2 \times 10^5$, $\tau = 8 \times 10^5$, and $d_t = 6$. Our finitary analysis also allows us to prove the optimality of the Chinchilla rule (see Appendix B).

3.2 Scaling of excess entropy

Under finitary analysis, for every compute budget C, there is an associated error rate $P_{b,\lambda_T,\widetilde{\rho}_R}/\epsilon$ which indicates a fraction of concepts not learnt even after the peeling process is complete. Similar

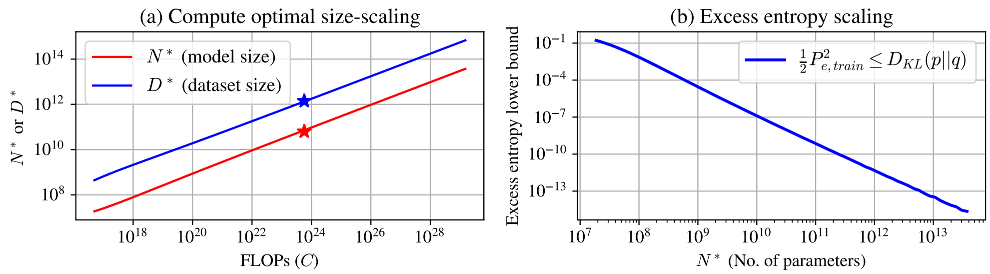

Figure 2: (a) Model and dataset size pair (N^*, D^*) as a function of compute budget C. The markers correspond to the Chinchilla model [9] with a compute budget of 5.76×10^{23} FLOPs; (b) Scaling of the lower bound of excess entropy with model size N^*.

to [2], we consider the cloze questions associated with text pieces connected to unlearnt concepts are incorrectly answered. Therefore, the training error is equivalent to the probability that a check node (text piece) is connected to the stopping set (unlearnt concepts) at least twice. Refer to [19] on stopping sets. The training error corresponding to (N, D) given a compute budget C is (see Appendix D for the calculation):

$$P_{e,train} = 1 - \left(1 - \frac{d_t P_b}{R}\right)^{R-1} - d_t P_b \left(1 - \frac{d_t P_b}{R}\right)^{R-1} \approx 4 d_t^2 \epsilon^{-2} P_{b,\lambda_T,\widetilde{\rho}_R}^2. \tag{3}$$

Using Pinsker's inequality $D_{KL}(P||Q) \ge \frac{1}{2}||P - Q||_1^2$, and the equivalence between total variation distance and error rate on cloze questions [2], we obtain the following lower bound on excess entropy (also shown in Figure 2(b)):

$$\text{Excess entropy} \ge \frac{1}{2} P_{e,train}^2 \approx 2 d_t^4 \epsilon^{-4} P_{b,\lambda_T,\widetilde{\rho}_R}^4. \tag{4}$$

3.3 Emergence and plateauing

We aim to provide a simple explanation to these empirical phenomena using random graph theory. Let p_l denote the probability there is a direct link between any two pairs of skills at level l. For a fixed (R, T), p_l evaluates as (see Appendix E for the derivation):

$$p_l \ge \begin{cases} \left(1 - g(R, p_{rr}, \eta_l)\right) \gamma_{l-1}^{2\sigma_l} & \text{if } \eta_l \le \binom{R}{2} p_{rr} \\ \frac{1}{\sqrt{8\eta_l\left(1 - \eta_l/\binom{R}{2}\right)}} g(R, p_{rr}, \eta_l) \gamma_{l-1}^{2\sigma_l} & \text{otherwise,} \end{cases} \tag{5}$$

where $g(R, p_{rr}, \eta_l) = \exp\left(-\binom{R}{2} D_{KL}\left(\frac{\eta_l}{\binom{R}{2}} || p_{rr}\right)\right)$, p_{rr} is the probability that a pair of concepts occur in at least one text piece, and γ_{l-1} is the probability that a skill belongs to GCC of $G_2^{(l)}$ (which we show next). Note that the skill graph $G_2^{(l)}$ is equivalent to an Erdös-Rényi (ER) random graph with $S^{(l)}$ nodes and edge probability p_l. A pair of skills in level l can be composed if there is a path between them in $G_2^{(l)}$, and both skills being in GCC of $G_2^{(l)}$ is a sufficient condition. Suppose γ_l is ratio of the size of GCC in $G_2^{(l)}$ to $S^{(l)}$, and is equivalent to the probability that a skill at level l is in GCC. For an ER graph with edge probability p_l, solution to $\gamma_l = 1 - \exp\left(-p_l S^{(l)} \gamma_l\right)$ yields γ_l [3]:

$$\gamma_l = 1 + \frac{1}{p_l S^{(l)}} W_0\left(-p_l S^{(l)} \exp\left(-p_l S^{(l)}\right)\right), \tag{6}$$

where $W_0(\cdot)$ is the upper branch of the Lambert W function. The ratio γ_l has a phase transition at $p_l = 1/S^{(l)}$. To see this, note that $W_0(xe^x) = x$ for $x < -1$. Therefore, whenever $p_l < 1/S^{(l)}$, γ_l is identically zero. As p_l increases beyond $1/S^{(l)}$, $|W_0(\cdot)|$ starts decreasing and γ_l increases.

For a particular skill level l, γ_l and p_l can be computed recursively using (6) and (5), with the following initial conditions: $\gamma_0 = 1$ and $\sigma_l = 0$ (no prerequisite skill is required to learn basic skills,

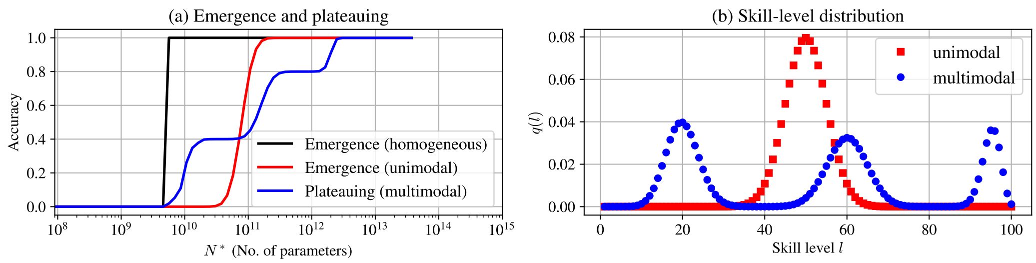

Figure 3: (a) Emergence and performance plateauing for different types of tasks; (b) Skill-level distribution $q(l)$ for unimodal and multimodal heterogeneous tasks.

i.e., skills at $l = 1$). Consider a complex task consisting of subtasks requiring m skills at level l with probability $q(l, m)$. The model performs the subtask successfully only if there is a path between every pair of those skills in $G_2^{(l)}$. The accuracy of the task is:

$$\text{Accuracy} \geq \sum_{l,m} q(l, m)\gamma_l^m. \tag{7}$$

Next, we demonstrate numerically fast emergence (similar to phase transition), slow emergence, and plateauing (multiple emergences) are consequences of tasks with different choices of $q(m, l)$. For illustration, let $q(m, l) = q(m)q(l)$, and $q(m) = 1/6$ for all $m \in \{2, \dots, 7\}$, number of skill levels $L = 100$, $S^{(l)} = 10^3$, $\eta_l = \exp(7l/L)$, $\sigma_l = \log_2(l)$ for all $l \in \{1, \dots, L\}$. Consider a homogeneous task requiring skills at only one level, say $l = 10$, then the accuracy (according to (7)) exhibits a step phase transition with increasing model size (black curve in Figure 3(a)). However, empirically observed accuracy curves exhibit smoother phase transitions [25]. To demonstrate this, consider a heterogeneous task with binomial distribution over the skill levels, i.e., $q(l) = \binom{L}{l}(\frac{1}{2})^L$ (red curve in Figure 3(b)). The corresponding accuracy is shown by the red curve in Figure 3(a). In general, a smooth single phase transition can be obtained by a unimodal distribution over skill levels with a sufficiently large variance. Finally, consider a heterogeneous task with diverse tasks characterized by a mixture of binomial distributions over the skill levels, i.e., $q(l) = \sum_i w_i \text{Binomial}(L, \pi_i)$, with $(w_i)_i \in (2/5, 2/5, 1/5)$ and $(\pi_i)_i = (0.2, 0.6, 0.95)$ (blue curve in Figure 3(b)). The blue curve in Figure 3(a) shows the corresponding accuracy. In general, a multimodal distribution over skill levels results in emergence at multiple scales and plateaus between them. Our framework yields an interesting trend associated with the plateauing of performance: plateauing indicates the possibility of one (or more) upcoming emergent phenomenon (phenomena), which one would encounter with further scaling.

4 Conclusion

We presented a simple unified framework to explain all three empirical phenomena observed with size scaling of language models. Existing frameworks assume compute-optimal scaling rule to explain emergent phenomena. We use non-asymptotic information theory to explain both compute-optimal size scaling and emergent abilities of language models. Moreover, we explain more recent empirical phenomenon of plateauing of performance using random network theory, and also predict that plateauing implies the possibility of multiple emergent phenomena with further size scaling.

There are some open questions and considerations worth exploring. Since, we do not consider training time in our framework, we do not explain other empirical phenomena such as double descent or grokking [11]. Perhaps future work can either extend our framework or propose a different framework to explain them. Even though the sequential learning of concepts through peeling yields a certain ordering to concepts, there is no inherent ordering and we do not consider concept hierarchies [27, 28]. One can explore the advantages of doing so. Evidently, the degree distribution of texts is related to the model's architecture. Therefore, optimizing the degree distribution enables a language model to learn more concepts from text pieces. Further, the quality of the training data is related to text-to-concept edge deletions in sequential concept learning, which can be incorporated into our framework. This is a line of future work that has natural analogues in optimization of communication systems and fault-tolerant computation.

Acknowledgment

We appreciate discussion with and valuable suggestions from Razan Baltaji, Akhil Bhimaraju, and Moulik Choraria.

This work was supported in part by National Science Foundation grant PHY-2112890 and by DARPA grant "Modeling and Measuring Scientific Creativity".

References

[1] John R. Anderson. *Rules of the Mind*. Lawrence Erlbaum Associates, Inc., 1993.

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

[3] Albert-László Barabási. *Network Science*. Cambridge University Press, 2016.

[4] Sourya Basu, Moulik Choraria, and Lav R. Varshney. Transformers are universal predictors. In *Neural Compression Workshop (ICML 2023)*, July 2023.

[5] Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. A dynamical model of neural scaling laws. In *Proceedings of the 41st International Conference on Machine Learning*, pages 4345–4382, July 2024.

[6] Steven Byrnes. AI doom from an LLM-plateau-ist perspective, 2023.

[7] Changyan Di, David Proietti, I. Emre Telatar, Thomas J. Richardson, and Rüdiger L. Urbanke. Finite-length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Transactions on Information Theory*, 48(6):1570–1579, 2002.

[8] J. B. S. Haldane. On being the right size. *Harper's Magazine*, pages 425–427, March 1926.

[9] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. arXiv:2203.15556, March 2022.

[10] Sara Hooker. On the limitations of compute thresholds as a governance strategy. arXiv:2407.05694 [cs.AI], July 2024.

[11] Yufei Huang, Shengding Hu, Xu Han, Zhiyuan Liu, and Maosong Sun. Unified view of grokking, double descent and emergent abilities: A comprehensive study on algorithm task. In *Proceedings of the Conference on Language Modeling*, October 2024.

[12] 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:2001.08361, January 2020.

[13] Davis E. Kieras and Davis E. Meyer. An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human–Computer Interaction*, 12(4):391–438, 1997.

[14] John E. Laird, Allen Newell, and Paul S. Rosenbloom. SOAR: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64, September 1987.

[15] Kuo-Yu Liao, Cheng-Shang Chang, and Y.-W. Peter Hong. A mathematical theory for learning semantic languages by abstract learners. arXiv:2404.07009, April 2024.

[16] Robert J. McEliece. *The Theory of Information and Coding*. Cambridge University Press, 2nd edition, 2002.

[17] Eric Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 28699–28722. Curran Associates, Inc., 2023.

[18] Allen Newell. *Unified Theories of Cognition*. Harvard University Press, 1990.

[19] Tom Richardson and Rüdiger Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008.

[20] Gordon Ritter and Wendy Lu. The first wave of AI innovation is over. here's what comes next. *Fast Company*, July 2024.

[21] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 55565–55581. Curran Associates, Inc., 2023.

[22] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3/4):379–423/623–656, July/October 1948.

[23] Nicolas Sourlas. Spin-glass models as error-correcting codes. *Nature*, 339:693–695, June 1989.

[24] D'Arcy Wentworth Thompson. *On Growth and Form*. Cambridge University Press, 1917.

[25] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. arXiv:2206.07682, June 2022.

[26] 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:2310.17567, October 2023.

[27] Haizi Yu, James A. Evans, and Lav R. Varshney. Information lattice learning. *Journal of Artificial Intelligence Research*, 77:971–1019, 2023.

[28] Haizi Yu, Igor Mineyev, and Lav R. Varshney. A group-theoretic approach to computational abstraction: Symmetry-driven hierarchical clustering. *Journal of Machine Learning Research*, 24(47):1–61, 2023.

5 Appendix

A IsoFLOP curves

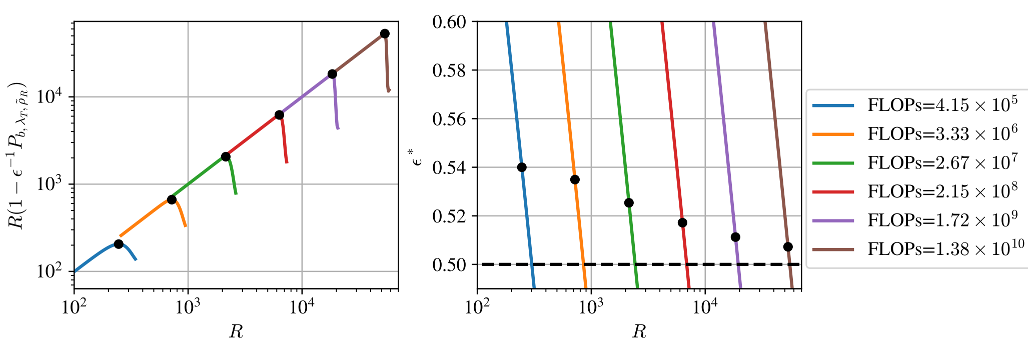

Figure 4: **IsoFLOP curves**: (**left**) Number of concepts learnt as a function of R for different compute budget; (**right**) Block erasure threshold as a function of the number of concepts R for different compute budget. In both subfigures, solid black markers indicate the points corresponding to R^*.

In Figure 4, the objective function in (1) is plotted against the number of concepts R for multiple compute budgets. In the left subfigure, each curve corresponds to a fixed compute budget. Note that smaller values of R correspond to smaller language model sizes, in which case the dataset size (number of texts T) is more than necessary for the model to learn all the skills. Contrarily, for large model sizes, the smaller dataset size is insufficient to learn the concepts well. There is an optimum model size and dataset size pair (equivalently R and T) such that the number of concepts learnt is maximized, as indicated by a solid black marker for each compute budget C. This figure is analogous to isoFLOP curves in [9, Figure 2], where training loss is plotted against model size for different compute budgets.

B Optimality of Chinchilla scaling rule

Proposition 1. **Compute-optimal scaling rule**: For compute-optimal performance of a language model, the dataset size (D) and model size (N) must scale equally with the increasing compute budget C (or FLOPs).

Proof. The approach is to prove that neither $T/R = o(1)$ nor $R/T = o(1)$ maximizes the objective function in (1). This implies that R/T must be a constant, i.e., R and T must scale equally with compute budget C.

Denote ϵ^* be the decoding threshold corresponding to the degree distribution pair $(\lambda_T, \widetilde{\rho}_R)$. From the matching condition [19], we have

$$\epsilon^* \leq \frac{\int \widetilde{\rho}_R}{\int \lambda_T} =: \epsilon^*_{ub}$$

(a) If $\frac{T}{R} = o(1)$ (i.e., $\frac{T}{R}$ decays as $C \to \infty$), then

$$\epsilon^*_{ub} - \epsilon \leq \epsilon \left(\left(1 - e^{-d/\epsilon} + \frac{d^2}{\epsilon R} \right) \left(\frac{1}{d} + \frac{T}{R} \right) - 1 \right) \xrightarrow{C \to \infty} \epsilon \left(\frac{(1 - e^{-d/\epsilon})}{d} - 1 \right) < 0,$$

which implies that $P_{b,\lambda_T,\widetilde{\rho}_R} \to 1$. Therefore, number of skills learnt vanishes for large C.

(b) Consider $\frac{R}{T} = o(1)$. From the fixed point characterization of decoding threshold of LDPC codes, we have

$$\begin{aligned} f(x, \epsilon^*) &= \epsilon^* \lambda_T (1 - \widetilde{\rho}_R(1 - x)), \\ &= \epsilon^* (1 - (1 - xp)^{\frac{R}{\epsilon} - 1} p)^{T-1}, \end{aligned} \tag{8}$$

where $p = d_t/R$. Since $R/T = o(1)$, the number of text pieces T grows strictly faster than R with respect to compute budget C, implying that the second term in (8), i.e., $(1 - (1 - xp)^{\frac{R}{\epsilon}-1}p)^{T-1} \to 0$ for large C. Therefore, for a non-trivial solution, i.e., $x = f(x, \epsilon^*) \in (0, 1]$, the decoding threshold ϵ^* must be very large. As a result, the post-decoding bit erasure rate $P_{b,\lambda_T,\widetilde{\rho}_R}$ vanishes for large C.

Suppose, (R^*_C, T^*_C) such that $R^*_C/T^*_C = o(1)$ minimizes (1). Now, consider $\hat{R}_C = R^*_C(1 + \delta)$ and $\hat{T}_C = T^*_C/(1 + \delta)$. Note that $\hat{R}_C/\hat{T}_C = (1 + \delta)^2 R^*_C/T^*_C = o(1)$. Therefore, for any $\delta' \in (0, \delta)$, there exists C_0 such that for all $C \geq C_0$ the bit erasure rate $\epsilon^{-1} P_{b,\lambda_{\hat{T}_C},\widetilde{\rho}_{\hat{R}_C}} \leq \delta'/(1 + \delta')$. Now consider the ratio of number of concepts learnt:

$$\frac{\hat{R}_C(1 - \epsilon^{-1} P_{b,\lambda_{\hat{T}_C},\widetilde{\rho}_{\hat{R}_C}})}{R^*_C(1 - \epsilon^{-1} P_{b,\lambda_{T^*_C},\widetilde{\rho}_{R^*_C}})} \geq \frac{R^*_C(1 + \delta) \left(1 - \frac{\delta'}{1+\delta'}\right)}{R^*_C} = \frac{1 + \delta}{1 - \delta'} > 1, \tag{9}$$

where the first inequality is by substitution and using the fact that $\epsilon^{-1} P_{b,\lambda_{T^*_C},\widetilde{\rho}_{R^*_C}}$ is non-negative, and the second inequality is because $\delta' < \delta$. Therefore, (R^*_C, T^*_C) is not a maximizer, which is a contradiction. Therefore, R/T cannot be $o(1)$.

Therefore, R/T must asymptotically be a constant. In other words, the model size N and dataset size D must scale equally with compute budget C.

$\square$

C Solving (1): Maximizing concept learning under compute budget constraint

C.1 A brief summary of belief propagation decoding of LDPC codes under erasure

Low-density parity check (LDPC) codes are a family of error-correction codes, whose noisy codewords can be decoded in a computationally efficient manner using belief propagation. Before getting into deriving the probability that a concept is learnt from text pieces, we provide a very short summary of belief propagation decoding of LDPC codes when codeword symbols are corrupted by erasure. An LDPC code can be graphically represented by a Tanner graph, which is a bipartite graph with a set of variable nodes (codeword symbols) and check nodes (parity checks). Each codeword satisfies all the parity checks. Given a degree distribution pair (for variable and check nodes), there is a channel noise threshold ϵ^* above which the decoder fails to decode the transmitted codeword. Consider a noisy version of a transmitted codeword with $\epsilon < \epsilon^*$ fraction of the symbols are erased. Belief propagation decoding starts by finding a check node where all except one symbol are recieved correctly (not erased). Then the erased symbol is determined as the one satisfying the parity. The next iteration starts by finding another check node with only one erased codeword symbol. This process continues until either all the codeword symbols are decoded or the decoder gets stuck with no parity checks containing only one erased symbol. The latter is declared as a decoding failure.

C.2 Computing $\mathbb{E}_{G_1^{(C)} \sim (\lambda_T, \rho_R)}[R_+]$

The objective function in (1) can be rewritten as:

$$\mathbb{E}_{G_1^{(C)} \sim (\lambda_T, \rho_R)}[R_+] = R(1 - \Pr\{r \notin \mathcal{R}_+ | R, T\}). \tag{10}$$

where $\Pr\{r \notin \mathcal{R}_+ | R, T\})$ is the probability that a concept r is remains unlearnt after peeling. Learning concepts from texts by the peeling process described in Section 2.3 is identical to belief propagation decoding of an LDPC code when the channel noise is erasure. To see this, treat R concepts as erased codeword symbols (subset of variable nodes), and T text pieces as parity checks. To obtain one-to-one correspondence, we need un-erased symbols (the remaining subset of variable nodes). Therefore, we choose (arbitrarily) a channel noise parameter $\epsilon \in (0, 1)$, add $\frac{1-\epsilon}{\epsilon} R$ nodes (dummy nodes) to the set of variable nodes, and treat them as un-erased symbols. Next, add edges between every pair of dummy variable node and a parity check node with probability $p = \frac{d_t}{R}$. Consequently, the degree distribution of the parity check nodes (text pieces) is modified, i.e., its degree distribution is binomial with parameters R/ϵ (instead of R) and d_t/R, but the degree distribution of variable nodes remains unchanged. Let us call the resulting parent graph $\widetilde{G}_1$[3] (see Figure 5) with the following text and concept degree distributions,

$$\widetilde{P}_R = \text{Binomial}(R/\epsilon, p), \text{ and} \tag{11}$$

$$\widetilde{L}_T = L_T = \text{Binomial}(T, p), \tag{12}$$

respectively. Here, for a compute budget C, we set $T = \frac{C}{6\varsigma\tau R}$.

[3] In this section, we omit superscript (C) in $\widetilde{G}_1^{(C)}$ for brevity.

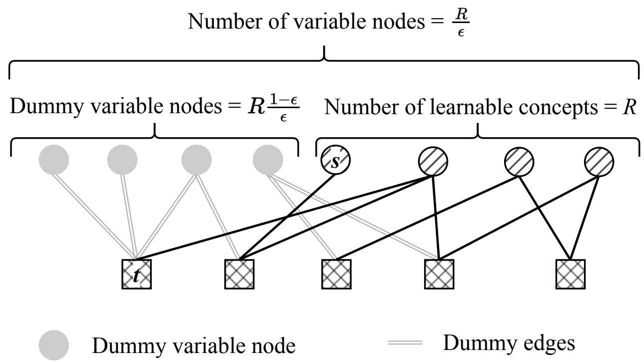

Figure 5: Bipartite graph $\widetilde{G}_1$.

In belief propagation decoding (peeling) of a codeword affected by erasures, the post-decoding bit erasure rate depends only on the residual graph consisting only variable nodes corresponding to erased symbols, parity checks connecting those variable nodes, and edges between them. Therefore, the post-decoding bit erasure rate is invariant to the choice of ϵ.[4] Therefore, we can make the following equivalence between concept learning and bit erasure rate:

$$\Pr\{r \notin \mathcal{R}_+ | R, T\} = \frac{P_{b,\lambda_T,\widetilde{\rho}_R}}{\epsilon}, \tag{13}$$

where $P_{b,\lambda_T,\widetilde{\rho}_R}$ is the post-decoding bit erasure rate, and $\lambda_T(x) = \frac{L'_T(x)}{L'_T(1)}$ and $\widetilde{\rho}_R(x) = \frac{\widetilde{P}'_R(x)}{\widetilde{P}'_R(1)}$ are variable and check node degree distributions from edge perspective, respectively. To compute $P_{b,\lambda_T,\widetilde{\rho}_R}$ we need the following ingredients: degree distributions λ_T and $\widetilde{\rho}_R$, decoding threshold ϵ^*, and scaling factors ν^* and α which depend on degree distributions. Degree distribution of text pieces from the node perspective is

$$P_R(x) = \sum_i \binom{R}{i} p^i (1-p)^{R-i} x^i, \tag{14}$$

$$\widetilde{P}_R(x) = \sum_i \binom{R/\epsilon}{i} p^i (1-p)^{(R/\epsilon)-i} x^i, \tag{15}$$

which gives the following text degree distribution from the edge perspective:

$$\widetilde{\rho}_R(x) = \frac{\widetilde{P}'_R(x)}{\widetilde{P}'_R(1)} = \frac{\sum_i i\binom{R/\epsilon}{i} p^i (1-p)^{(R/\epsilon)-i} x^{i-1}}{\sum_i i\binom{R/\epsilon}{i} p^i (1-p)^{(R/\epsilon)-i}}. \tag{16}$$

Noting that $i\binom{R/\epsilon}{i} = R\binom{R/\epsilon-1}{i-1}$ we obtain the degree distribution of text pieces from edge perspective:

$$\begin{aligned} \widetilde{\rho}_R(x) &= \frac{\sum_{j=0}^{(R/\epsilon)-1} \frac{R}{\epsilon} p \binom{R/\epsilon-1}{j} p^{i-1} (1-p)^{(R/\epsilon)-i} x^{i-1}}{\frac{R}{\epsilon} p} & (17) \\ &= (px + (1-p))^{\frac{R}{\epsilon}-1}. & (18) \end{aligned}$$

Similarly, the degree distribution of concepts (remains unchanged for a fixed R, T) from the edge perspective is

$$\lambda_T(x) = (px + (1-p))^{T-1}. \tag{19}$$

Next the belief propagation decoding threshold ϵ^* is obtained from its fixed point characterization [19, Section 3.12]:

$$\epsilon^* = \inf\{\epsilon \in [0,1] : x = f(x, \epsilon) \text{ has a solution in } x \in (0,1]\}, \tag{20}$$

[4]Here we choose $\epsilon = 0.5$ (instead of close to 0 or 1) for numerical convenience.

where $f(x,\epsilon) = \epsilon\lambda_T(1-\widetilde{\rho}_R(1-x))$, and the critical point x^* satisfies $x^* = f(x^*, \epsilon^*)$.

From finite-length scaling law of error rates in belief propagation decoding [19, Section 3.23], we have the following (approximate) closed-form expression for post-decoding bit erasure rate:

$$P_{b,\lambda_T,\widetilde{\rho}_R} \approx \nu^* Q\left(\sqrt{\frac{R}{\epsilon}}\frac{(\epsilon^*-\epsilon)}{\alpha}\right), \tag{21}$$

where $\nu^* = \epsilon^*\, L_T(1-\widetilde{\rho}_{\mathcal{R}}(1-x^*))$, $Q(\cdot)$ is the complementary standard Gaussian cumulative distribution function, and the scaling parameter α is given by [19, Section 3.23]

$$\alpha = \left(\frac{\rho(\bar{x}^*)^2 - \rho((\bar{x}^*)^2) + \rho'(\bar{x}^*)(1-2x^*\rho(\bar{x}^*)) - (\bar{x}^*)^2\rho'((\bar{x}^*)^2)}{L_T'(1)\lambda_T(y^*)^2\rho'(\bar{x}^*)^2} + \right. \tag{22}$$

$$\left.\frac{(\epsilon^*)^2\lambda(y^*)^2 - (\epsilon^*)^2\lambda_T((y^*)^2) - (y^*)^2(\epsilon^*)^2\lambda_T'((y^*)^2)}{L_T'(1)\lambda(y^*)^2}\right)^{1/2}, \tag{23}$$

where x^* is the unique critical point, $\bar{x}^* = 1-x^*$, and $y^* = 1-\widetilde{\rho}_R(1-x^*)$.

D Calculation of $P_{e,train}$

Recall that the training error is equivalent to finding the probability that a text piece is connected to an unlearnt concept, i.e.,

$$P_{e,train} = \Pr\left(|\{e_{tr} \in G_1^{(C)}\}_{r\in\mathcal{R}_-}| \geq 2\right), \text{ for any } t \in \mathcal{T}, \tag{24}$$

$$= \sum_{k\geq 2}^{R} \Pr\left(\text{degree}(t) = k, \{|\{e_{tr} \in G_1^{(C)}\}_{r\in\mathcal{R}_-}| \leq 1\}^c\right), \tag{25}$$

$$= \sum_{k\geq 2}^{R} \binom{R}{k} p^k(1-p)^{R-k}\left(1-(1-P_b)^k - kR(1-P_b)^{k-1}\right), \tag{26}$$

where the edge probability $p = d_t/R$ and $P_b = \epsilon^{-1}P_{b,\lambda_T,\widetilde{\rho}_R}$. The last equation simplifies to:

$$P_{e,train} = 1 - \left(1-\frac{d_tP_b}{R}\right)^{R-1} - d_tP_b\left(1-\frac{d_tP_b}{R}\right)^{R-1}, \tag{27}$$

which is obtained by computing the expectation of each of the three terms within the summation in (26) and substituting $p = d_t/R$. Further using the approximations $(1-x)^n \approx 1-nx$ and $R-1 \approx R$ for large R, the training error is approximately $P_{e,train} \approx 4d_t^2P_b^2$.

E Calculation of p_l

Recall that p_l is the probability that the composition of a pair of skills in level l is seen at least η_l times in the training data. For a fixed pair of skills (s_1, s_2), the probability there is a path between the pair of skills through some pair of concepts (r_1, r_2) is

$$\begin{aligned}\Pr(s_1-r_1-r_2-s_2) &= \Pr(s_1-r_1, r_1-r_2, r_2-s_2),\\ &= \Pr(s_1-r_1)\Pr(r_1-r_2)\Pr(r_2-s_2),\\ &= \frac{1}{S^{(l)}}\left(1-\left(1-\frac{d_t^2}{R^2}\right)^T\right)\frac{1}{S^{(l)}} =: p_{rr},\end{aligned}$$

where the second inequality is due to independence of s_1-r_1, r_1-r_2 and r_2-s_2. Let X be a random variable indicating the number of distinct paths $s_1-r_1-r_2-s_2$ between s_1 and s_2. Now, $\Pr\left(\text{composition of}(s_1,s_2) \text{ in training data}\right) =: p_l$ is

$$\begin{aligned}p_l &= \Pr(X \geq \eta_l, \text{all prerequisite skills of } s_1 \text{ and } s_2 \text{ are acquired}),\\ &\geq \Pr(X\geq\eta_l)\Pr(\text{all prerequisite skills of } s_1 \text{ and } s_2 \text{ are acquired}).\end{aligned}$$

Note that the total number of distinct paths between s_1 and s_2 equals the total number of concept pairs (r_1, r_2) which is $\binom{R}{2}$, each with probability p_{rr}. Therefore, X follows a binomial distribution, i.e., Binomial $\left(\binom{R}{2},\ p_{rr}\right)$. From Chernoff's bound for binomial distribution, we obtain the following lower bounds:

$$\Pr(X \geq \eta_l) \geq \begin{cases} \left(1 - \exp\left(-\binom{R}{2} D_{KL}\left(\frac{\eta_l}{\binom{R}{2}} || p_{rr}\right)\right)\right) & \text{if } \eta_l \leq \binom{R}{2} p_{rr} \\ \frac{1}{\sqrt{8\eta_l\left(1 - \frac{\eta_l}{\binom{R}{2}}\right)}} \exp\left(-\binom{R}{2} D_{KL}\left(\frac{\eta_l}{\binom{R}{2}} || p_{rr}\right)\right) & \text{otherwise.} \end{cases} \tag{28}$$

The probability of acquiring prerequisite skills of both skills s_1 and s_2 is (assuming $R \gg \sigma_l$),

$$\begin{aligned} \Pr(\text{all prerequisite skills of } s_1 \text{ and } s_2 \text{ are acquired}) &\geq \Pr(\text{all } \sigma_l \text{ prerequisites} \in \text{GCC})^2, \\ &= \gamma_{l-1}^{2\sigma_l}. \end{aligned}$$