A TALE OF TAILS: MODEL COLLAPSE AS A CHANGE OF SCALING LAWS

Yunzhen Feng[*†‖] **Elvis Dohmatob**[*‡] **Pu Yang**[§] **Francois Charton**[‡] **Julia Kempe**[†‡¶]
[*]Equal Contribution
[†]Center for Data Science, New York University
[‡]Meta FAIR
[§]School of Mathematical Sciences, Peking University
[¶]Courant Institue of Mathematical Sciences, New York University
[‖]yf2231@nyu.edu

ABSTRACT

As AI model size grows, neural *scaling laws* have become a crucial tool to predict the improvements of large models when increasing capacity and the size of original (human or natural) training data. Yet, the widespread use of popular models means that the ecosystem of online data and text will co-evolve to progressively contain increased amounts of synthesized data. In this paper we ask: *How will the scaling laws change in the inevitable regime where synthetic data makes its way into the training corpus?* Will future models, still improve, or be doomed to degenerate up to total *(model) collapse*? We develop a theoretical framework of model collapse through the lens of scaling laws. We discover a wide range of decay phenomena, analyzing loss of scaling, shifted scaling with number of generations, "un-learning" of skills, and grokking when mixing human and synthesized data. Our theory is validated by large-scale experiments with a transformer on an arithmetic task and text generation using the large language model `Llama2`.

1 INTRODUCTION

Groundbreaking advances in generative AI algorithms for text, images and code are ushering in the "synthetic data age": increasingly we consume data generated by large scale models like GPT4 (Achiam et al., 2023), Stable Diffusion (Rombach et al., 2022) and their successors. At the same time a key driver behind the current success of large models is their consumption of massive amount of web-scale data for training. The improvements of larger models are governed by scaling laws in which error falls off as a power in the size of training data; and the emergence of new skills seems tightly linked to covering increased scales of training data. Our understanding of what the future holds in a world where models are trained on other models (or their own) synthesized data is only at its beginning, but some works indicate the possibility of complete collapse of learning, so called model collapse[1].

Scaling Laws. In many domains of machine learning including speech, translation, vision, video, and mathematical problem solving, empirically observed neural scaling laws (Hestness et al., 2017; Rosenfeld et al., 2020; Kaplan et al., 2020; Hoffmann et al., 2022; Gordon et al., 2021; Henighan et al., 2021; Aghajanyan et al., 2023) demonstrate that test error often falls off as a power law with the amount of training data, model size, and compute. Theoretically, scaling laws have been derived in a variety of settings (e.g. (Hutter, 2021; Cabannes et al., 2023) for "LLM-like" models).

Scaling laws are intimately related to the emergence of abilities (Wei et al., 2022) in larger models, that are not present in smaller ones; and to skills that appear with decreasing loss (Gordon et al., 2021; Arora & Goyal, 2023). This bolsters the now common narrative "scaling is all you need".

Model Collapse. Current LLMs (Devlin et al., 2018; Liu et al., 2019; Brown et al., 2020; Touvron et al., 2023), including GPT-4 (Achiam et al., 2023), were trained on predominantly human-generated text; similarly, diffusion models like DALL-E (Ramesh et al., 2021), Stable Diffusion (Rombach et al., 2022), Midjourney (Midjourney, 2023) are trained on web-scale image datasets.

[1]Not to be confused with neural collapse which refers to clustering of last-layer features at the end of training (Papyan et al., 2020)

These training corpora already potentially exhaust all the available clean data on the internet. A growing number of synthetic data generated with these increasingly popular models starts to populate the web, often indistinguishable from "real" data. We have thus already raced into the future where our training corpora are irreversibly mixed with synthetic data and this situation stands to get worse. Recent works call attention to the potential dramatic deterioration in the resulting models, an effect referred to as *"model collapse"* (Shumailov et al., 2023). Facets of this phenomenon have been demonstrated *empirically* in various settings (Hataya et al., 2023; Martínez et al., 2023a;b; Bohacek & Farid, 2023; Briesch et al., 2023; Guo et al., 2023). Theoretically, a few works are emerging to analyze the effect of iterative training on self-generated (or mixed) data (see Related Work): Shumailov et al. (2023) coin the term *"model collapse"* to characterize complete reversion to the mean, Alemohammad et al. (2023) analyze *"self-consuming loops"* and Bertrand et al. (2023) show that iterative synthetic training leads to a *"clueless generator"*.

With these first warning signs in place, we thus ask:

> *How is the current scaling paradigm affected by synthetic data in the training corpus?*

To this end, we carefully zoom into the scaling behavior of LLM-style models. Theoretical derivations of scaling laws always assume a heavy-tailed distribution (power-law, aka Zipf) on the input features ("heavy tail in, power scaling law out"). This distribution is of the form

$$p_i \propto i^{-\beta}, \quad i = 1, 2, \ldots \tag{1}$$

Such distributions are ubiquitous in natural datasets, from Zipf's law (Zipf, 1935) in distribution of word frequencies, to biological data, earthquake magnitudes, financial data etc. - this is the data being consumed by large models at scale. But what distribution do AI-models generate when trained on such data? Figure 2 provides an empirical answer for a large scale LLM (Llama2-7B) and a transformer model trained on an arithmetic task. Regenerating heavy-tailed data affects the distribution in two possible ways: (1) "Cutting off" the tail of the distribution and/or (2) "Narrowing" the tail (see Figure 1 for a cartoon illustration). The mechanisms leading to this, apart from finite sampling bias (as already proposed in (Shumailov et al., 2023) - see Section 2 for a derivation in the Zipf-setting), stem from deliberate choices in the generation algorithms of the models: in LLMs via truncated next token prediction at inference (e.g. selecting more likely tokens via *top-p* or *top-k* truncation, concentrating the probability distribution by lowering the temperature); in vision models like GANs via truncation or in diffusion models through guidance.

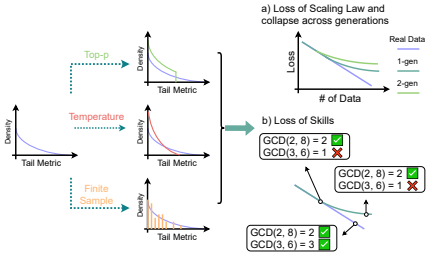

Figure 1: (Cartoon). Top-p (nucleus) sampling, temperature scaling of LLM generation, and finite sample bias lead to truncated or narrowed "tails" (left side), causing loss of scaling laws (top right) and loss of skills (bottom right). Here we visualize calculating the greatest common divisor (GCD) with answer 2 and 3 as two skills.

Summary of Main Contributions. We present a high-level summary of our main theoretical contributions, some of which are highlighted in Figure 3. We empirically verify these theoretical predictions (see Figure 4): (1) in large-scale experiments on an LLM, fine-tuning Llama2-7B (Touvron et al., 2023) on an approximately $2M$ sample dataset from Wikitext-103 and (2) for transformer models trained to predict the greatest common divisor (Charton, 2023).

Assuming a true distribution as in Equation 1, consider training a model on T AI data-generated data. The synthesized data amounts to a version of the true data distribution with the tail cut at some finite rank k or the tail narrowed to a smaller exponent. Our main findings are as follows.

(1) A Double Scaling Law. We establish new scaling laws that explain model collapse in simplified (non-autoregressive) LM (Hutter, 2021) and toy bigram LLMs (refer to Theorems 2.1 and 4.2)[2]

$$E_{test} \asymp T^{-c} + k^{-c'}. \tag{2}$$

or equivalently (refer to Corollary 2.2), for finite-sample induced cut-off $k = k(T_0)$ when the generating model is trained on T_0 amount of data, $E_{test} \asymp T^{-c} + T_0^{-c''}$, where the exponents c, c', c'' only depend on the tail behavior of the true distribution. This result is illustrated in Figure 3.

[2]The notation $f(T) \lesssim g(T)$ means that $f(T) \leq Cg(T)$ for sufficiently large T and an absolute constant C, while $f(T) \asymp g(T)$ means $f(T) \lesssim g(T) \lesssim f(T)$.

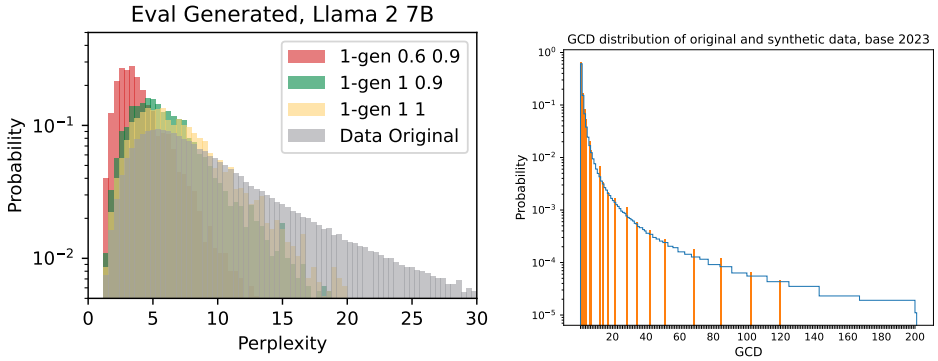

Figure 2: **Tails of AI-generated data: Left.** Perplexity diagram of the Wikitext-103 test set, measured with `Llama2-7B` as the anchor model. We query the Wikitext-finetuned `Llama2-7B` to generate AI data, which is compared to the original set. Perplexity is calculated solely for the generated positions in AI and original datasets. AI data is generated for various settings of (p, T). **Right.** Distribution of greatest common divisors (GCD) of pairs of random integers (original data (blue) scaling as $p(GCD = k) \propto k^{-2}$). A transformer is trained on $300M$ samples and used as a generator on a test set of randomly sampled integer pairs, giving the truncated GCD distribution.

For AI-"tail-narrowing", when data remains heavy-tailed, with a smaller exponent $\beta' \in (1, \beta)$, the downstream Hutter LLM will scale as (Corollary 2.3)

$$E_{test} \asymp T^{-(\beta-1)/\beta'}. \tag{3}$$

(2) Model Collapse over Multiple Generations. For n-fold recursion of AI data-generation equation 9, where each generation of the model consumes data produced by the previous generation, we establish a universality principle of the form

$$E_{test} = E_{test}^{clean} + n \times \text{ new scaling terms}, \tag{4}$$

where E_{test}^{clean} is the usual test error of the model trained on clean data (not AI-generated). This means that in Equations equation 2 for example, the $k^{-c'}$ is replaced by $nk^{-c'}$. One possible interpretation of this multiplicative degradation is that, over time (i.e as the number of generations becomes large), the effect of large language models (like ChatGPT) in the wild will be a pollution of the web to the extend that learning will be impossible. This will likely increase the value and cost of clean / non-AI-generated data.

(3) Mitigation Strategies. In Theorem 3.2 we show that mixing AI-generated data with even a small amount of clean data mitigates model collapse by introducing a grokking phenomenon. The length of the plateau is of order k^{β}/π, where π is the proportion of training data which is from the true distribution (i.e clean data). When $\pi = 0$ (i.e only AI-generated data available), this plateau goes on forever (as in Equation 2). When $\pi > 0$, however small, the plateau finally halts, and the error continues to decrease à la T^{-c}. This grokking phenomenon holds in the setting of *deterministic* ground truth labels (like in the models of (Hutter, 2021; Cabannes et al., 2023)). For transformer models, such deterministic settings are found for instance in arithmetic tasks, and we demonstrate it empirically in our GCD experiments. The grokking effect becomes attenuated in probabilistic settings, where it can lead to an S-shaped learning curve (see Figure 19). We also identify regimes where adding AI data is beneficial and discuss ways to curate "tail" data to mitigate AI-data effects.

Related Work. Theoretically, scaling laws have been derived in various settings: for non-parametric models (Schmidt-Hieber, 2017; Suzuki, 2019; Bordelon et al., 2020), in the kernel regime under Gaussian design (Spigler et al., 2020; Cui et al., 2021; 2022; 2023; Maloney et al., 2022), or in memorization-like settings with discrete data (Hutter, 2021; Debowski, 2023; Michaud et al., 2023). Taking finite model capacity and optimization into account, Cabannes et al. (2023) recently proved scaling laws in constraint-capacity associative memories. In Appendix E, we extend our proofs to this model.

Less than a handful of works begin to provide theoretical explanations for the behavior of models in the "synthetic data age". Shumailov et al. (2023) attribute model collapse to two mechanisms: a

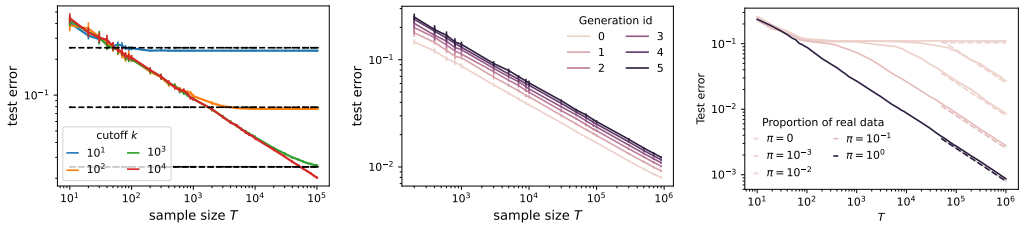

Figure 3: **Illustration of Our Main Results for Simplified LLMs. Left plot.** Empirical confirmation of the double scaling law. The true distribution of the data is Zipf with exponent $\beta = 3/2$. Broken lines correspond to $k^{-(\beta-1)}$, for varying T and different values of k. **Middle plot.** Model collapse over multiple generations. Again $\beta = 3/2$, $T_0 = T$ across all generations with no additional tail-cutting, regeneration for 5 times. **Right plot.** Notice the grokking behavior, as perfectly predicted by the Theorem 3.2. For any given value π for the proportion of real data, the broken lines are asymptotes $E_{test} \asymp (\pi T)^{-c}$ and each plateau has length of order k^β/π, both predicted by the theorem. See Figure 10 for similar results with other values of k.

finite sampling bias cutting off low-probability "tails", thus leading to more and more peaked distributions and function approximation errors; they theoretically analyze the (single) Gaussian case and provide empirical evidence for VAEs, Gaussian mixtures and the OPT language model (125M parameters). In the context of vision models, Alemohammad et al. (2023) analyze *"self-consuming loops"* by introducing a sampling bias that narrows the variance of the data at each generation, and, in addition to empirical demonstration on GANs and denoising diffusion probabilistic models, provide theoretical analysis for the Gaussian model. Finally, let us mention the study of Bertrand et al. (2023) which sheds light on the critical role of data composition in the stability and effectiveness in generative models, applicable to VAEs Kingma & Welling (2014), diffusion models and normalizing flows. They explore scenarios involving a mix of clean data, representative of the true distribution, and synthesized data from previous iterations of the generator. Their analysis reveals that if the data mix consists exclusively of synthesized data, the generative process is likely to degenerate over time (*"clueless generator"*). Using fixed-point analysis across iterations, they find that when the proportion of clean data in the mix is sufficiently high, the generator, under certain technical conditions, retains the capability to learn. A recent paper Fan et al. (2023) empirically observe deteriorated scaling laws when training on synthetic data for text-to-image models. To our knowledge, our work is the first to theoretically and empirically analyze model collapse in the context of scaling laws and emergent abilities to provide a rich new landscape of AI-data induced phenomena.

2 A Deterministic Infinite Memory Model

Here, we present the core of our theory for the simplest case of (i) *infinite memory* and (ii) *a deterministic ground truth* labeling function $i \mapsto y_i$, studied by Hutter (2021) (the *"Hutter LLM"*). Both restrictions will be lifted in later sections, where we also analyze an *probabilistic autoregressive* version (Section 4) and *limited memory* models (Appendix E). Token i is drawn according to the Zipf law in Equation 1, which e.g. models distribution of various metrics in language. Another interpretation of the appearance of a power-law is offered by the "Quantization Hypothesis" paper of Michaud et al. (2023): one may think of each i as some discrete skill, needed to solve a problem for example; thus, the skills occur at different rates p_i. The shape parameter $\beta > 1$ controls the length of the tail of this distribution: bigger values of β correspond to longer tails.

2.1 What Causes Model Collapse ?

Tail Cutting. As mentioned, deliberate choices in the AI generation algorithm (like top-p or top-k next token prediction) immediately lead to a chopped tail at k. When viewed as skills, we can say that only the kth most frequent outcomes ("skills") are considered. But even when no tails are cut deliberately, the finite size T_0 of the training set (sampling bias) induces an effective tail-cutting. This can be seen as follows: Sample an iid dataset of size T_0, and estimate the histogram p_{AI}; this new distribution plays the role of an AI data-generator. An integer i appears in the support of p_{AI} a number of times which is $T_0 p_i$ on average. Roughly speaking[3], this means that the support of p_{AI} is $\{i \mid p_i \leq C/T_0\} = \{i \mid i \leq k\}$, where

$$k = k(T_0) \asymp T_0^{1/\beta}. \tag{5}$$

[3]This can be made rigorous via standard concentration arguments.

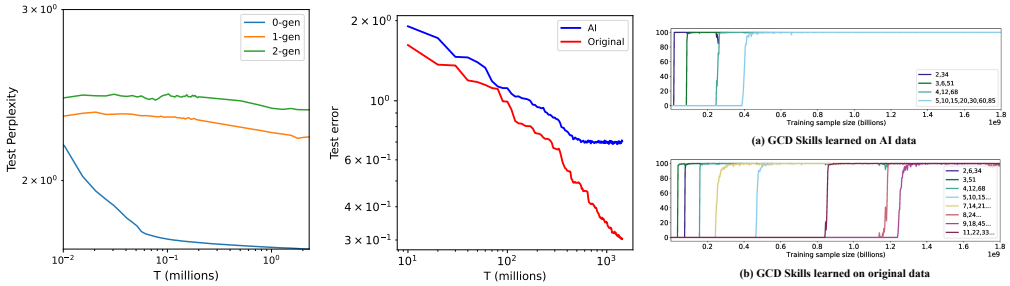

Figure 4: **Experimental Results (Details in Section 5). Left plot.** The scaling law for finetuning `Llama2-7B` on the Wikitext-103 dataset. '0-gen' utilizes the original data, while subsequent generations use data generated by the previous model. **Middle plot.** Scaling law of the transformer model trained to predict GCD of two integers. Data is synthesized from a 0th generation model trained on $300K$ samples. Note the tapered-off scaling of the model trained on synthesized data, as predicted by our theory. **Right plot.** "Skills" (bursts of new GCDs) learned by the GCD-transformer on original (bottom) and AI data (top). We see how the disappearance of scaling leads to the disappearance of abilities, mastered by the model trained on clean data.

Therefore, the transformation $p \to p_{\text{AI}}$ amounts to chopping off the tail of p at rank k.

Tail Narrowing. Figure 2 (for Llama2) shows that in addition to tail cutting, tail narrowing effects happen during AI-generation. One mechanism for this is lowered temperature during next-token prediction. Assume a softmax distribution on the logits z_i for the ith token: $p_i = e^{z_i}/\sum_j e^{z_j}$. Define $q_i^T = e^{z_i/T}/\sum_j e^{z_j/T}$ for general temperature T. Then $p_i \asymp i^{-\beta}$ morphs into $q_i^T \asymp i^{-\beta/T}$ (to first order). Temperature scaling directly causes narrowing of tail for $T > 1$. Other mechanisms can come to play: for instance, for autoregressive models with perplexity, token-wise tail cutting can result in tail narrowing for sequence-perplexity (see Figure 35 and discussion in Appendix J).

2.2 A New Scaling Law in the Hutter LLM

For a deterministic ground-truth labelling function $i \mapsto j_i$, consider a Hutter "LLM" Hutter (2021)

$$\widehat{f}(i) := \begin{cases} j_i, & \text{if } (i, j_i) \in \mathcal{D}_T, \\ \perp, & \text{otherwise,} \end{cases} \tag{6}$$

constructed on a sample $\mathcal{D}_T := \{(i_t, j_t) \mid t \in [T]\}$ of size T from unmitigated Zipf distribution p Equation 1, the test error obeys the following scaling law Hutter (2021)

$$E_{test} \asymp T^{-(1-1/\beta)}. \tag{7}$$

Now, let q be a k-tail-cutting version of p, i.e $q_i \propto p_i$ if $i \leq k$ and $q_i = 0$ otherwise. When constructed ("trained") on $\mathcal{D}_T$ of size T, now from q, the test error (w.r.t to the true data distribution p) of this model is $E_{test} := \mathbb{P}_{i\sim p}(\widehat{f}(i) \neq j_i) = \sum_{i\geq 1} p_i \mathbb{P}(\widehat{f}(i) \neq j_i)$. That is, we train on data from the AI distribution q and test on original distribution p. We prove the following scaling law for tail cutting (all proofs are relegated to Appendix C):

Theorem 2.1. *Consider long-tail real-world data with exponent $\beta > 1$, and let the cutoff for AI-generated data be k. Then, for large k and T samples from the AI, the test error of the downstream "LLM" scales like so $E_{test} \asymp T^{-(\beta-1)/\beta} + k^{-(\beta-1)} \asymp \min(T, k^\beta)^{-(\beta-1)/\beta}$.*

Thus, as soon as $T \gtrsim k^\beta$, the AI-generated sample size T ceases to be a "scalable" resource: collecting more AI-generated samples will not improve the performance of the downstream model, i.e performance plateaus and we lose scaling. The result is illustrated empirically in Figure 3, left and Figure 8 (Appendix B). When we assume that the AI-generator was trained on T_0 samples, we get a similar loss of scaling stemming from the tail cutting from finite sampling bias (Equation 5):

Corollary 2.2 ("Finite Initial Sample Size")**.** *With $c = 1 - 1/\beta$, it holds that*

$$E_{test} \asymp T^{-c} + T_0^{-c}. \tag{8}$$

These theoretical are empirically confirmed in the Figure 9. In the case of tail narrowing, the scaling behavior changes; instead of a plateau, we obtain a slower decay rate:

Corollary 2.3 ("Tail Narrowing")**.** *In the setting of Theorem 2.1, consider AI-generated data to also be long-tail data, albeit with smaller exponent $\beta' \in (1, \beta)$. Then, the downstream Hutter LLM trained on AI-generated data will scale as $E_{test} \asymp T^{-(\beta-1)/\beta'}$.*

2.3 Collapse Over Multiple Generations of AI Data

We now examine the cumulative impact of prior loss of scaling across multiple generations. Consider n-fold recursive AI data-generation, i.e

$$p \to p_{\text{AI}(1)} \to p_{\text{AI}(2)} \to \ldots \to p_{\text{AI}(n)}. \tag{9}$$

Each arrow corresponds to drawing a sample of size T_0. If we iterate n times the argument leading to Equation 8, we get the following scaling for the test error $E_{test}^{(n)} = E_{test}^{(n)}(T)$ for learning on T samples from the nth generation and testing on the true data distribution,

$$E_{test}^{(n)} \asymp T^{-c} + \underbrace{T_0^{-c} + \ldots + T_0^{-c}}_{n \text{ times}} = T^{-c} + C_n T_0^{-c} = T^{-c}\left(n(T/T_0)^c + 1\right), \tag{10}$$

where $c := 1 - 1/\beta$. We deduce the following result.

Theorem 2.4 (Informal)**.** *Model collapse (as spoken of in the literature) occurs iff $n \gg (T_0/T)^c$.*

For example, if $T_0 \gg T$ (e.g $T_0 \geq CT \log T$) and n is constant (e.g $n = 25$), then model collapse will not occur if we learn on the nth generation of AI data. On the other hand, if $T_0 \lesssim T$, then model collapse will eventually occur.

In particular, taking $T_0 \asymp T$, we get

$$E_{test}^{(n)} \asymp C_n T^{-c} \asymp nT^{-c}. \tag{11}$$

Note how the loss scales linearly with the number of generations. Figure 3, middle, illustrates how an increased number of generations moves the loss scaling curve progressively to the right. This leads to eventual model collapse.

3 Mitigating Model Collapse via Data Mixing

Here we explore the possibility of alleviating model collapse via the acquisition of even a tiny amount of data from the true data distribution, to complement AI polluted data. We study two phenomena: (1) In the case of mixing π-fraction of the original data with a $(1 - \pi)$ fraction of AI-generated data we exhibit a startling "grokking" phenomenon where test loss plateaus with increasing training data to finally decrease again according to the scaling law of the original model, and (2) in the scenario where we would like to compensate for missing "tail", we acquire some data from the tail of the original distribution to show that this needs to be done with caution: getting data from "too deep" in the tail is worthless while data closer to the precise "missing" tail can be beneficial. All proofs can be found in Appendix C.

3.1 Acquiring Missing Tail

To counter the effect of tail cutting and the resulting plateau in scaling, we might resort to adding curated data that emphasize the tail. Theorem 3.1 studies this effect; it shows, in particular that if we "overshoot" and only curate tail that is too deep, our efforts will be worthless. Rather, there is a fine line around the chopped tail k (within a factor of $(1 + o(1))$ of k), where we need to place our data curation efforts to achieve the desired effect, a return of the scaling beyond the plateau.

Suppose we "buy" a chunk of the tail of the real data distribution corresponding to $i = N, N+1, \ldots$; let the distribution be π (thus, supported on $\{N, N+1, \ldots\}$). Now, let k, N, and T tend to infinity such that $N/k \to C$, with $C \in [1, \infty]$. We have the following sharp phase-transition.

Theorem 3.1. *(A) If $C = 1$, e.g if $N = k + \sqrt{k}$, then $E_{test} \asymp T^{-c}$. That is, we perfectly anneal the tail-chopping effect of AI-generated data.*

(B) If $C > 1$, then $E_{test} \asymp T^{-c} + k^{-\alpha}$ (which recovers the result of Theorem 2.1), and so "buying" the Nth tail of the real data distribution is worthless.

3.2 A Grokking Phenomenon

Here we show how even small amounts of original data can mitigate the above "scaling law collapse" by introducing a grokking phenomenon where test error plateaus and eventually continues to decline.

Theorem 3.2 (Grokking with Tail Cutting)**.** *Consider a sample of size T of which a proportion π comes from the true distribution p and the remainder comes from a version p' of p with its tail chopped off at rank k. We have the following scaling laws for the Hutter LLM defined in Equation 6.*

(A) ***Early-Stage Dynamics.*** *For $T \ll k^\beta/\pi$, it holds that $E_{test} \asymp T^{-(1-1/\beta)} + k^{-(\beta-1)}$. Thus, during this stage, the money spent on acquiring some clean data is not amortized!*

(B) ***Later-Stage Dynamics.*** *As soon as $T \geq Ck^\beta/\pi$ (where C is an absolute constant), it holds that $E_{test} \asymp (\pi T)^{-(1-1/\beta)}$. Thus, during this stage, we recover the unpolluted sample-size law scaling $T^{-(1-1/\beta)}$, up to within a multiplicative constant $\pi^{-(1-1/\beta)}$ (which can be seen as an increase in the price of data). For fixed T and tunable π, this error rate scales like $\pi^{-(1-1/\beta)}$, which is yet another scaling law.*

Effectively, the above theorem predicts that for any fixed $\pi \in (0, 1)$ –no matter how small– the test error grokks w.r.t sample size T. The result is empirically confirmed in Figure 3, right (see Figure 10 for another illustration). We experimentally confirm this new phenomenon for transformer models trained to calculate the GCD (see Appendix H), which indicates its applicability for a wider class of LLMs with underlying deterministic ground truth, like for arithmetic tasks. In Appendix C.4 we state and prove a similar theorem in the case of *tail narrowing* of synthetic data.

Benefits of Mixing with AI Data. The above machinery allows us to analyze a particular regime where AI-data can help improve performance. Taking $T = T_{real} + T_{AI}$ and $\pi = T_{real}/T$, we have the following important corollary of Theorem 3.2.

Corollary 3.3. *For $T_{real} \ll k^\beta$, it holds that $E_{test} \asymp (T_{real} + T_{AI})^{-(1-1/\beta)} + k^{-(\beta-1)}$.*

Figure 5 illustrates how AI data can boost performance, up to a certain point, when its benefits plateau. This result might contribute to our understanding of why, sometimes, adding AI-generated data might lead to better models, especially when generated by a stronger model (e.g. He et al. (2023); Shipard et al. (2023); Bansal & Grover (2023); Lin et al. (2023)). More in Appendix A.

4 A Tailed Bigram Model

We will now proceed to a more complex model, bringing us closer to capturing the *probabilistic* and *autoregressive* nature of LLMs (next token prediction). In this Section we will define the data generating process, define the new model (Hutter++), and establish that the original scaling law (with clean data) still holds. We then proceed to show similar loss of scaling for AI-data.

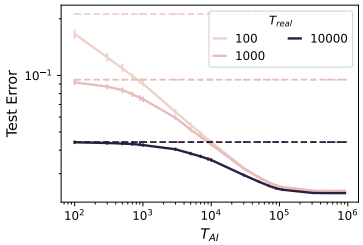

Figure 5: Mixing T_{real} real data with T_{AI} AI data. The dotted lines depict test errors of real data alone. $k = 1,000, \beta = 3/2$.

A first step is to consider *probabilistic* ground truth labels to replace the deterministic Hutter prediction $i \mapsto y_i$ with a probability distribution $p(j|i)$ on $\mathbb{N}_*$ with power law decay (as in Equation 1). To account for the fact that the most frequent next token j depends on the preceding token i we model

$$p(j \mid i) \propto \pi_i(j)^{-\beta}, \tag{12}$$

(instead of $j^{-\beta}$), where π_i is a permutation associated to every i providing the order of outputs. To summarize, we think of the data as pairs (i, j), where the distribution of i is governed by some $p(i)$ as in the deterministic Hutter setting, and $p(j|i)$ is given by Equation 12.

This setting can be made *autoregressive* by generating sequences step by step, using the preceding output as the next input. We can think of each successive pair of tokens as of the pairs (i, j) above, with the only difference that the marginal distribution $p(i)$ changes. We thus make no assumptions on $p(i)$ in what follows (except for a mild technical condition). Proofs can be found in Appendix D.

A Scaling Law for Hutter++. Consider non-deterministic outputs as in Equation 12, where $\pi_1, \pi_2, \ldots$ are functions from $\mathbb{N}_*$ to $\mathbb{N}_*$.

Theorem 4.1. *Suppose $\beta \in (1, \infty) \setminus \{2\}$ and set $c := \min(1 - 1/\beta, 1/2)$. If $\sum_i p_i^{1-c} < \infty$, then $E_{test} \lesssim T^{-c}$. Moreover, if $\beta \in (1, 2)$ and the mappings $\pi_1, \pi_2, \ldots$ are permutations, then $E_{test} \asymp T^{-c}$.*

Our Hutter++ algorithm induces exactly the same scaling law as the classical setup Hutter (2021) !

We now return to our main problem, understanding model collapse in the probabilistic setting and consider the Hutter++ presented above. Thus, suppose the learner only has access to at most a dataset of size T containing the kth head of the conditional distribution $p(\cdot \mid i)$. That is, sampled from: $i \sim p$, $j \sim p(j \mid i)1[j \leq k]$ (normalized appropriately), where $p(\cdot \mid i)$ is as in Equation 12.

Theorem 4.2. *(A) If $\beta \in (1, \infty) \setminus \{2\}$ and $\sum_i p_i^{1-c} < \infty$ where $c := \min(1 - 1/\beta, 1/2)$ as before, then $E_{test} \lesssim T^{-c} + k^{-\beta c}$. (B) Furthermore, if the mappings $\pi_1, \pi_2, \ldots$ are permutations and $\sum_i p_i^{1-c} < \infty$, then $E_{test} \asymp T^{-c} + k^{-\beta c}$.*

Autoregressive Bigrams. Similarly these results hold for autoregressive bigram model, where $p(i_1, i_2, \ldots, i_L) = p(i_1) \prod_{\ell=1}^{L-1} p(i_{\ell+1} \mid i_\ell)$, and each $p(j \mid i)$ is as in Equation 12. The result is empirically confirmed in Figure 11 in Appendix B.

Multiple Generations. The mechanics of the proof of Theorem 2.4 apply in this setting. See Figure 12 in Appendix B illustrating that Equation 11 keeps holding for probabilistic data.

Grokking for Mixtures. Technically speaking, this grokking phenomenon only holds for models with deterministic ground truth labels, like the Hutter LLM and the limited capacity associative memory model. For the *probabilistic* setting of bigrams (or text LLMs) the theorem cannot hold in its pure form, because if we train on a mixture of two distributions (clean and synthetic) but test only on the clean distribution, the distance between these two distributions will always be a lower bound on the test error. However, we can see that remnants of a "smoothed" grokking-law persist in the form of an S-shaped scaling (see Figure 19 in Appendix B).

5 Experiments

In this section we present our experimental results to demonstrate evidence of various predictions we have made theoretically. We showcase four scenarios of increasing level of complexity: an empirical Hutter++ model, autoregressive bigram models with *perplexity loss*, an arithmetic transformer to predict the GCD of two integers Charton (2023) and a large-scale LLM, Llama2-7B Touvron et al. (2023), trained on a large data corpus (`Wikidata-103`).

In our theoretical analysis, motivated by empirical observations (see Figure 2) or by the effect of finite data sampling bias on heavy-tailed data, we have assumed that generated data follows patterns of either a tail cutoff or tail narrowing. In our subsequent experiments, we depart from theoretical assumptions on tail-cutting/narrowing to allow the widely deployed top-p selection or temperature scaling mechanisms to give possibly intermingled effects on the generated data distribution.

Empirical Hutter++ Model. In Figure 6, we use an initial model that is trained on $T_0 = 100,000$ samples from the original distribution. For the Gen 1 line, the data are all generated from this initial model. From Gen 2 onwards, models are iteratively trained on data produced by the most performant model of the preceding generation, effectively eliminating the possibility that model collapse results from inadequate sampling. For Gen 1, a notable degradation in data scalability is observed, alongside a rapid decline in model performance across generations. These observations not only validate our theoretical result but also reaffirm our assumptions. A similar pattern is evident with temperature scaling, as shown in Figure 16.

Autoregressive Bigram Models with Perplexity Loss. We move one step further towards LLMs to investigate autoregressive bigram models. The dataset now comprises sequentially generated integers, as in Equation 12, with the model trained on all tokens. We use the averaged perplexity score of the test set as the test error metric. Our study encompasses a range of effects—such as top-p inference, temperature scaling, limited real data, and training on progressively larger AI datasets. Consistent with the findings in Section 4, we observe the same patterns of scaling loss and progressive model collapse across generations. Relevant figures are provided in Appendix G.

Transformers Learning the GCD. Our first illustration of our theory "in the wild" is for sequence-to-sequence transformer models for an arithmetic task: predicting the greatest common divisor (GCD) of two integers, encoded as sequences of digits in some base B following Charton (2023). This setup is a perfect intermediate step between our toy models and large scale LLMs; it uses the transformer architecture and training algorithms on sizeable models, while the underlying data has a deterministic nature. Over the course of training the model progressively learns new GCDs and with

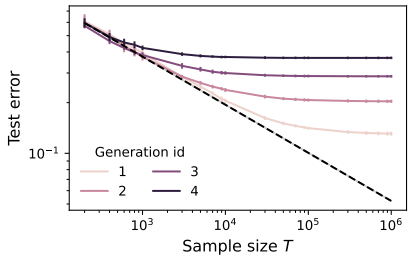

Figure 6: **Hutter++ on Bigram with limited data and top-p.** The initial model is trained on $T_0 = 100,000$ samples. It generates T samples for Gen 1. Starting from Gen 2 models are trained on data generated by the most powerful model from the previous generation. Top-p-0.95 cutting and $\beta = 3/2$.

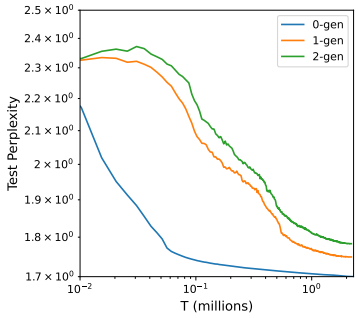

Figure 7: **Mixing Llama Generated Data with Original Data** Based on Figure 4 left, we further mix the generated data with original data, with ratio 98 to 2. Adding original data significantly mitigates the model collapse. Look how the mixing curve validates our predicted curve of the grokking phenomenon as in Figure 3

them also their products with already learned GCDs. We can thus view each such learned group, usually learned in "bursts", as a new skill. For the purpose of this experiment, we use this model after $300M$ samples as the generator of AI-data. In Figure 4 we validate the predicted scaling law for a single generation and observe 'un-learning' of skills when training exclusively with generated data, as well as a grokking effect when training with mixtures. See Appendix H for a full description and more details and Figures.

Experiments on LLMs. We finetune Llama2 with LoRA, generating synthetic AI data for the next finetuning iteration. Inspired by the setup in Shumailov et al. (2023), we use `Wikidata-103`, partitioned into approximately 2.2 million sequences of 128 tokens. AI data is generated through prompt completion, using the first 96 tokens from the original sequences as prompts. The model is trained only on the last 32 tokens to preclude information leakage, i.e. the model being trained on the ground truth of the same 32 tokens. The evaluations are conducted exclusively on the same 32 tokens. We use top-p 0.9 and temperature 0.9 across all generation. The results, depicted in Figure 4 (left), illustrate a scaling law decay over several generations. The first generated dataset still contain useful but limited information and the utility of the second generation's data markedly diminishes. These phenomena corroborate the anticipated loss of scaling law and model collapse, further indicating that model collapse is even more pronounced here, highlighting the challenges in training next generation LLMs. More details and results in Appendix I.

Moreover, we conduct experiments to investigate mixing a proportion of real data with AI-generated data. Figure 7 demonstrates the effect of blending a random 2% of original data with AI data across all fine-tuning phases. It significantly mitigates model collapse, with the emergence of a grokking curve as predicted in Theorem 3.2.

6 Conclusion

In the advent of the "synthetic data age", our work signals the end of current neural scaling laws and opens the door to a puzzling array of new phenomena in a world where the training corpora are enriched with AI generated data. We demonstrate that scaling laws cease to persist; test error tapers off due to altered, less heavy tailed, data distributions. As noted already in prior work, in a fully synthetic data world learning will stop and models will degenerate - their scaling will halt and revert completely. Yet, new opportunities arise from careful mixture and data curation, as we have shown, with interesting effects at the interplay of clean and synthesized data. We must recognize new learning plateaus and, for instance, adjust to changed learning curves from blending clean and synthetic data to avoid unintended early stopping. A notable feature of our work is that our theory is *effective* - we observe the predicted phenomena for relevant large models in two different settings.

Taken together, our contributions call for a more responsible, or "collapse-aware", proliferation of synthesized data. Scale is *not* all you need: more work on effective watermarking for synthetic data is needed, to make it more distinguishable from the original human data. Thus, clean / real data will become an even more valuable resource, as we are ushering in the "beyond scaling" era.

7 IMPACT STATEMENT

This study contributes to the discourse on AI scaling laws by tackling the emergent challenge presented by synthetic data within training datasets. It holds particular relevance for fields that depend heavily on data accuracy, such as natural language processing and generative AI model development. We investigate the concept of "model collapse" as a cautionary tale, highlighting the potential risks to sustainable model performance in the face of an increasing reliance on synthetic data. Through a combination of theoretical analysis and empirical evidence, we advocate for the strategic integration of clean data to mitigate these risks and enhance the robustness of AI systems. Our findings carry significant implications for industries where decision-making is progressively data-driven, suggesting a need for a shift in data management strategies to emphasize the preservation of model integrity and longevity. Additionally, this research highlights a looming issue: the potential scarcity of clean data. It underscores the urgency of focusing on the cultivation and preservation of high-quality datasets for AI training, to safeguard the future of AI development.

8 ACKNOWLEDGEMENTS

YF and JK acknowledge support through NSF NRT training grant award 1922658. YF and PY would like to thank Di He for discussions and suggestions. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

REFERENCES

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

Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models, 2023.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-consuming generative models go mad. *arXiv preprint arxiv:2307.01850*, 2023.

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

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic data from diffusion models improves imagenet classification. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, 2023.

Daniel Berend and Aryeh Kontorovich. On the convergence of the empirical distribution. *ArXiv Preprint*, 2012.

Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. On the stability of iterative retraining of generative models on their own data. *arXiv preprint arxiv:2310.00429*, 2023.

Matyas Bohacek and Hany Farid. Nepotistically trained generative-ai models collapse, 2023.

Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1024–1034. PMLR, 2020.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. Large language models suffer from their own output: An analysis of the self-consuming training loop, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Max F Burg, Florian Wenzel, Dominik Zietlow, Max Horn, Osama Makansi, Francesco Locatello, and Chris Russell. Image retrieval outperforms diffusion models on data augmentation. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Vivien Cabannes, Elvis Dohmatob, and Alberto Bietti. Scaling laws for associative memories, 2023.

Andrea Caponnetto and Ernesto de Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7:331–368, 2007.

François Charton. Can transformers learn the greatest common divisor?, 2023.

Mayee F Chen, Nicholas Roberts, Kush Bhatia, Jue WANG, Ce Zhang, Frederic Sala, and Christopher Re. Skill-it! a data-driven skills framework for understanding and training language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Hugo Cui, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborova. Generalization error rates in kernel regression: The crossover from the noiseless to noisy regime. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Hugo Cui, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová. Generalization error rates in kernel regression: the crossover from the noiseless to noisy regime. *Journal of Statistical Mechanics: Theory and Experiment*, 2022(11):114004, nov 2022.

Hugo Cui, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová. Error scaling laws for kernel classification under source and capacity conditions. *Machine Learning: Science and Technology*, 4(3):035033, August 2023. ISSN 2632-2153. doi: 10.1088/2632-2153/acf041.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. Auggpt: Leveraging chatgpt for text data augmentation, 2023.

Lukasz Debowski. A simplistic model of neural scaling laws: Multiperiodic Santa Fe processes, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training... for now. *arXiv preprint arXiv:2312.04567*, 2023.

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 (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5915–5922, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.478.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. The curious decline of linguistic diversity: Training language models on synthetic text, 2023.

Ryuichiro Hataya, Han Bao, and Hiromi Arai. Will large-scale generative models corrupt future datasets? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20555–20565, October 2023.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and XIAOJUAN QI. Is synthetic data from generative models ready for image recognition? In *The Eleventh International Conference on Learning Representations*, 2023.

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 arXiv:2103.05847*, 2021.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

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 arXiv:1712.00409*, 2017.

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, 2022.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve, 2022.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Shaobo Lin, Kun Wang, Xingyu Zeng, and Rui Zhao. Explore the power of synthetic data on few-shot object detection. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 638–647, 2023. doi: 10.1109/CVPRW59228.2023.00071.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Alexander Maloney, Daniel A. Roberts, and James Sully. A solvable model of neural scaling laws, 2022.

Gonzalo Martínez, Lauren Watson, Pedro Reviriego, José Alberto Hernández, Marc Juarez, and Rik Sarkar. Combining generative artificial intelligence (ai) and the internet: Heading towards evolution or degradation? *arXiv preprint arxiv: 2303.01255*, 2023a.

Gonzalo Martínez, Lauren Watson, Pedro Reviriego, José Alberto Hernández, Marc Juarez, and Rik Sarkar. Towards understanding the interplay of generative artificial intelligence and the internet. *arXiv preprint arxiv: 2306.06130*, 2023b.

Ian R. McKenzie, Alexander Lyzhov, Michael Martin Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Xudong Shen, Joe Cavanagh, Andrew George Gritsevskiy, Derik Kauffman, Aaron T. Kirtland, Zhengping Zhou, Yuhui Zhang, Sicong Huang, Daniel Wurgaft, Max Weiss, Alexis Ross, Gabriel Recchia, Alisa Liu, Jiacheng Liu, Tom Tseng, Tomasz Korbak, Najoung Kim, Samuel R. Bowman, and Ethan Perez. Inverse scaling: When bigger isn't better. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Eric J Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Midjourney. Midjourney ai, 2023. URL `https://www.midjourney.com/`.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in Hilbert space. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3351–3361. Curran Associates, Inc., 2020.

Atsushi Nitanda and Taiji Suzuki. Optimal rates for averaged stochastic gradient descent under neural tangent kernel regime. In *International Conference on Learning Representations*, 2021.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 18–24 Jul 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations*, 2020.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *Annals of Statistics*, 48, 08 2017. doi: 10.1214/19-AOS1875.

Jordan Shipard, Arnold Wiliem, Kien Nguyen Thanh, Wei Xiang, and Clinton Fookes. Diversity is definitely needed: Improving model-agnostic zero-shot classification via stable diffusion, 2023.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arxiv:2305.17493*, 2023.

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, December 2020. ISSN 1742-5468. doi: 10.1088/1742-5468/abc61d.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *ArXiv*, abs/1906.02243, 2019.

Taiji Suzuki. Adaptivity of deep reLU network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2019.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754.

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. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. Survey Certification.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

GK Zipf. The psycho-biology of language: an introduction to dynamic philology. 1935.

A PRIOR WORK

Model Collapse: The phenomenon appeared in the *recent* literature in the context of language and image generation. Several recent works demonstrate facets of this phenomenon *empirically* in various settings Hataya et al. (2023); Martínez et al. (2023a;b); Bohacek & Farid (2023); Briesch et al. (2023); Guo et al. (2023); Fan et al. (2023). Only few recent works also provide some accompanying theoretical analysis Shumailov et al. (2023); Alemohammad et al. (2023); Bertrand et al. (2023) which we outline now.

Shumailov et al. (2023) define model collapse and attribute it to two mechanisms: finite sampling when training a model (leading to cut off of low-probability data) and function approximation errors (the model is not sufficiently expressive to model the true distribution). They observe (and, for a single Gaussian, prove) that upon iteratively resampling finite "training data" the generated distribution becomes more and more peaked. Other models studied empirically are mixtures of (two) Gaussians and VAEs on MNIST. To study language models, Shumailov et al. (2023) iteratively fine tune Meta's OPT-125M model on `wikidata2`. For generation of new text they use a 5-way beam search, which, by its nature, (approximatively) generates only low-perplexity data.

Alemohammad et al. (2023) conduct an empirical and analytical analysis on generative image models of what they term the "self-consuming" or "autophaguous" loop. They conclude that without enough fresh real data at each generation, future models necessarily will have their precision or recall decrease. They model the influence of each new AI-generation via a generic *sampling bias* $0 \leq \lambda \leq 1$. In the case of image generation this refers to feature parameters at generation that favor quality over diversity (suitably quantified). More precisely, $\lambda = 1$ corresponds to unbiased sampling and $\lambda = 0$ corresponds to sampling from the modes of the generative distribution. λ models biased sampling methods commonly used in generative modeling practice, such as truncation in BigGAN and StyleGAN or guidance in diffusion models. In the case of Gaussian distributions, λ is the shrinking factor of the variance of the next generation. Their empirical work studies GANs and denoising diffusion probabilistic models for image generation on FFHQ and MNIST and single Gaussians for both theoretical and empirical observations. As in Shumailov et al. (2023) they observe (and prove for the case of a single Gaussian) that estimation error alone leads to vanishing variance with number of iterations. Alemohammad et al. (2023) also empirically observe an initial boost in performance in a regime where modest amounts of synthetic data are mixed with the original data before larger amounts of synthetic data lead to ultimate degradation. This might mimick larger-scale results that demonstrate how synthetic data mixed with true data improves performance in some scenarios (see *Benefits of synthesized data* below). Indeed, in its simplest form, data augmentation (rotations, cropping etc.), a widespread highly beneficial practice in ML training, can be viewed as the simplest form of data generation.

Let us mention the study of Bertrand et al. (2023) in the context of image generation, which sheds light on the critical role of data composition in the stability and effectiveness in generative models. They explore scenarios involving a mix of clean data, representative of the true distribution, and synthesized data from previous iterations of the generator. Their analysis reveals that if the data mix consists exclusively of synthesized data, the generative process is likely to degenerate over time, leading to what they describe as a 'clueless generator'. Thus, the generator collapses: it progressively loses its ability to capture the essence of the data distribution it was intended to model. Conversely, they found that when the proportion of clean data in the mix is sufficiently high, the generator, under certain technical conditions, retains the capability to learn and accurately reflect the true data distribution. This work sheds light on the critical role of data composition in the stability and effectiveness of generative models.

Several empirical studies confirm the deleterious effect of training on self-generated data: In the context of image generation, Martínez et al. (2023a;b) report degradation of models trained on AI-generated data. Specifically, they use a Denoising Diffusion Implicit Model and a few (relatively small) datasets (e.g. Orchids, MNIST) to demonstrate visual degradation when training in successive generations of AI-generated data. Hataya et al. (2023) *"conclude that generated images negatively affect downstream performance, while the significance depends on tasks and the amount of generated images"*, Bohacek & Farid (2023) reports that the popular StableDiffusion model collapses when iteratively retrained on self-generated faces, even with as little as 3% synthetic data mixed

into the original training set. For text, Briesch et al. (2023) use *nanoGPT*[4] on a curated 10K logical-expression dataset to demonstrate the iterative collapse of self-consuming loops - the model and dataset are sufficiently small to allow training from scratch. Guo et al. (2023) observe a decline in linguistic diversity metrics across iteratively fine-tuned LLMs.

Mitigation: To our knowledge, rigorous theory (or even empirical demonstrations) on mitigation strategies against model collapse are yet to come, with one notable exception in Bertrand et al. (2023) (see below). Several works discuss the need for *detection* of AI-generated images or text (to avoid retraining on them), for example motivating research into watermarking strategies. Bertrand et al. (2023) analyze iterative retraining on a mixture of synthesized and original data under several technical assumptions and find that there are fixed points governing the stability of iterative retraining.

Benefits of Synthesized Data There is a range of results showing benefits of AI-synthesized data in training better models, though mostly these results pertain to image data, specifically in the context of diffusion models Azizi et al. (2023); He et al. (2023); Shipard et al. (2023); Bansal & Grover (2023); Lin et al. (2023), though not only (see Dai et al. (2023); Xu et al. (2023); Huang et al. (2022); Wang et al. (2023) for chat-related examples). One might argue that they either throw model-collapse caution to the winds or, possibly, settle in the protected corner where mild amounts of synthetic data (or larger amounts of "mildly synthetic" data, like in the case of data augmentation) helps. In particular, often benefits of synthetic data are observed when the synthetic data is generated by a model trained for a different use case than the downstream task (like images synthesized from diffusion models helping classification models) or generated by a stronger model He et al. (2023); Shipard et al. (2023); Bansal & Grover (2023); Lin et al. (2023). However, other works critically analyze the purported benefit of generated data. Burg et al. (2023) find that while synthesized data from a diffusion model helps improving downstream tasks, such as classification, using the *pre-training data* of the diffusion model alone gives even stronger performance (which we can interpret as evidence of mild first-generation model collapse). All in all it is fair to say that the impact of data augmentation using generative models is still not fully understood.

Scaling Laws: Neural scaling laws have been ubiquitously observed in vision, language and speech. Early large scale empirical studies are performed in Hestness et al. (2017); Rosenfeld et al. (2020), demonstrating power law scaling across a range of learning scenarios. This is followed by well-known large-scale studies from OpenAI Kaplan et al. (2020) and DeepMind Hoffmann et al. (2022), which empirically demonstrate power-law scaling in LLMs across a wide set of scales. Essentially, this empirically establishes that

$$L(N, D) \sim N_C \cdot N^{-\alpha_N} + D_C \cdot D^{-\alpha_D},$$

where L is the per-token cross entropy loss (in nats), N, D are the number of (non-embedding) parameters and data, respectively, and N_C, D_C and α_N, α_D are constants determined by the data distribution and the model specifications.

This study was extended to demonstrate many more power law relations in various scenarios (vision transformer, video modeling, multimodal models, and mathematical problem solving) Henighan et al. (2021). In the machine translation (MT) setting, (Gordon et al., 2021) quantify scaling laws for standard benchmarks like BLEU and explain them via cross-entropy power-law scaling, thus positing a first universality of scaling laws across metrics. Hernandez et al. (2021) demonstrate similar empirical power-law scaling for transfer learning and Aghajanyan et al. (2023) provide a vast experimental body of evidence for scaling laws in mixed-modal language models.

However, a few results have nuanced the view of scaling as a panacea to improved loss. For instance, McKenzie et al. (2023) present evidence for "inverse sclaing" where flaws in the training objective or the data lead to U-shaped scaling.

Theoretical Models for Scaling Laws: From a theoretical angle, scaling laws have been shown analytically even before the emergence of large foundation models. For instance, Caponnetto & de Vito (2007) characterize the power-law generalization error of regularized least-squares kernel

[4] https://github.com/karpathy/nanoGPT

algorithms. The role of optimization can also be taken into account in this setting (Nitanda & Suzuki (2021)). In the nonparametric literature, for example Schmidt-Hieber (2017) and Suzuki (2019) derived the test error scaling of deep neural network in fitting certain target functions and Bordelon et al. (2020) analyze spectral dependence.

More recently, scaling laws have been shown for kernel models under the Gaussian design, e.g. in Spigler et al. (2020); Cui et al. (2021; 2022) for regression and Cui et al. (2023) for classification. Maloney et al. (2022) study scaling laws for the random feature model in the context of regression. In the context of memorization for heavy-tailed data scaling laws have been shown in the infinite-memory setting Hutter (2021), for "quantized" skills Michaud et al. (2023) and for certain random data-generation processes Debowski (2023). When taking model capacity and optimization into account, Cabannes et al. (2023) recently proved scaling laws in constraint-capacity associative memories.

To our knowledge, however, very few papers deal with the decay of scaling in the case of self-consuming loops. A notable example is Mobahi et al. (2020) which studies iterated retraining in the context of self-(knowledge-)distillation in the kernel setting. However, this analysis is very distinct from our work, not only because it places itself in the kernel setting with Gaussian design, but also because it assumes the distillation setting, where the "generation" stage is carefully optimized for the next stage training. In the case of synthesized data in the wild, this assumption can of course not be made.

Emergence of "Skills" and Scaling Laws: Scaling laws give us an insight on bang-for-the-buck style trade-off for model training. However, cross-entropy loss is not a goal in and of itself: we want to train models that are endowed with a larger and larger skill set as we scale them up. For instance, Gordon et al. (2021) provide intuition and empirics for the scaling of BLEU score for MT with cross-entropy loss as

$$BLEU(L) \approx Ce^{-kL},$$

demonstrating "emergence" of good BLEU performance with scale. This type of "emergence" has been massively confirmed in Wei et al. (2022), where a working definition of "emerging" is "not present in smaller models, but appears in larger models". In this sense, Wei et al. (2022) demonstrate empirically a large number of "skills" appearing with scale, like Multi-Task NLU, Modular arithmetic, word unscrambling and transliteration.

A theoretical model, providing an underpinning of the necessity of scaling laws for the emergence of skill has recently been given by Arora & Goyal (2023). They analyse "emergence" with the scaling laws as a departure point in a model that links cross-entropy loss in LLMs to basic skills to show that scaling laws enable the model to learn (and generalize) efficiently.

Strengthening the tie between scaling laws and emergent skill, albeit in the *opposite* direction, Michaud et al. (2023) posit that skills that emerge in "quanta" imply a scaling law of the loss. Related, Chen et al. (2023) assume a hierarchy of skills to derive data curation mechanisms to precipitate the emergence of skills, though they do not allude to scaling laws directly.

B Complimentary Figures for Sections 2, 3 and 4

Hutter LLM. Figures 8, 9 and 10 further illustrate our theory for simple Hutter LLM.

Hutter++. We now provide complementary illustrations of predictions made from the theory we have developed for the generalized Hutter models as in Equation equation 12 in Section 4, without departing from our theoretical assumptions. We also show how theory from the infinite memory model in Section 2 continues to hold in this bigram setting. Figure 11 confirms the scaling law of Theorem 4.2.

In Figure 3 (middle) we have seen an illustration of the translated scaling curves under n-fold synthesized data in the Hutter LLM. Figure 12 illustrates this phenomenon for the slightly more complex tailed bigram model.

Both Figures 3 (middle) and 12 illustrate the setting where each model consumes as much training data as its predecessor ($T_0 = T$). We now relax the assumption that each successive model has

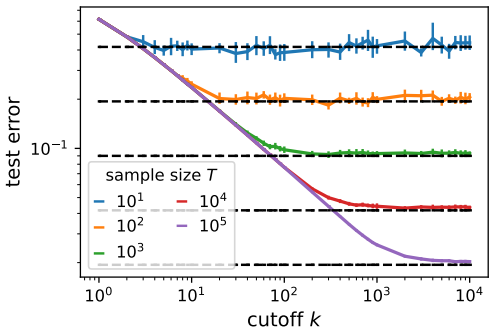

Figure 8: **Scaling on Hutter LLM for Varying T.** Empirical confirmation of Theorem 2.1. Here, $\beta = 3/2$ and error bars correspond to 10 iid runs of sampling AI-generated data (i.e the distribution q). Broken lines correspond to the Hutter rate $T^{-(\beta-1)/\beta}$, for varying k and different values of T. Figure 3, left, illustrates the same for varying T and several settings of k. Note the perfect match with the theorem.

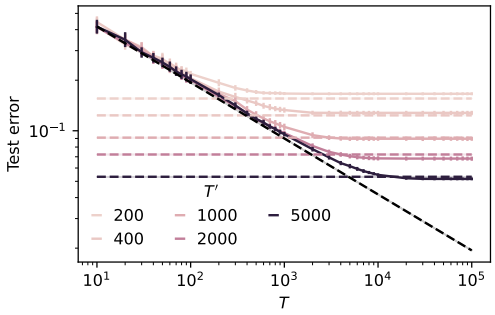

Figure 9: **Scaling on Hutter LLM for Varying k.** A sample of size T_0 is used to approximate the true distribution p via p_{AI}. Then, a Hutter-type model is learned on a sample of size T from p_{AI}, and evaluated on the true data distribution p. Each horizontal line corresponds to the asymptote $k^{-\beta c} \asymp T_0^{-c}$, for different values of T_0. The diagonal line corresponds to T^{-c}.

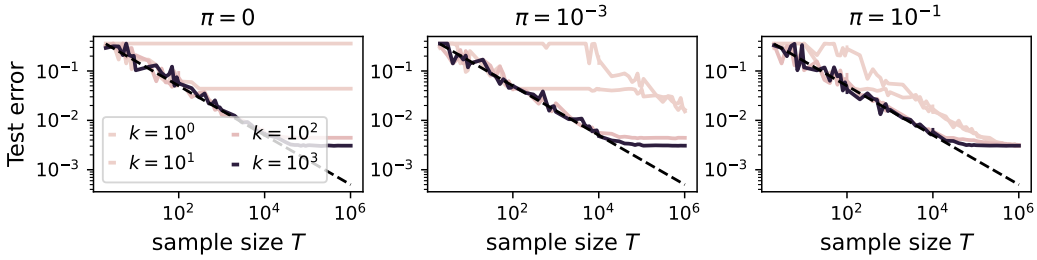

Figure 10: Empirical Validation of Theorem 3.2. The broken line corresponds to the $T^{-(1-1/\beta)}$ scaling law that would hold throughout in the absence of pollution. Notice the grokking behavior predicted by the theorem. For this experiment, the Zipf exponent of the true data distribution p is $\beta = 2$.

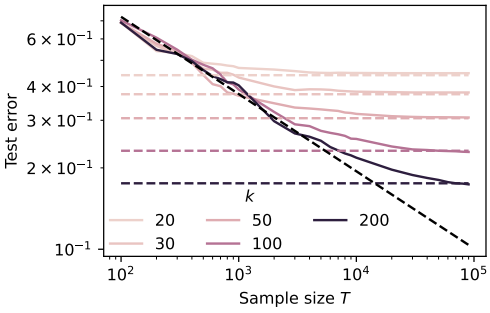

Figure 11: **Model Collapse for Hutter++.** Empirical confirmation of Theorem 4.2. Here $p(j \mid i)$ is as in equation 12, with $\beta = 7/5$. The horizontal broken lines correspond to $k^{-\beta c}$ for different values of k, where $c := \min(1 - 1/\beta, 1/2)$. The diagonal broken line corresponds to T^{-c} (classical error rate without cutoff).

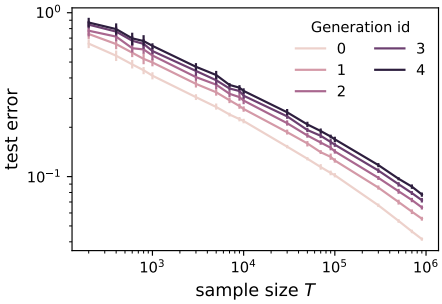

Figure 12: **Hutter++ Model on Paired Bigram Data.** Empirical confirmation of Theorem 2.4 for probabilistic paired bigram data with $\beta = 3/2$, $T_0 = T$ across all generations with no additional tail-cutting, regeneration for 9 times. The result verifies the model collapse across generation.

strictly the same amount of training data as its predecessor. We assume that the generation 0 model is trained on T_0 (here, $T_0 = 100,000$) amount of original data to generate AI data for generation 1. All future generations, starting from generation 2, are trained on data generated by the most powerful model from the previous generation ($T = 1,000,000$ data in this case). Figure 13 (for Hutter LLM) and 14 (for Hutter++ on paired bigram data) show the resulting scaling behavior. We take this setting even further by adding a top-p tail cutting mechanism and a temperature scaling mechanism for each synthetic data generation. Figure 6 cuts at $p = 0.95$ and Figure 16 at temperature 0.9.

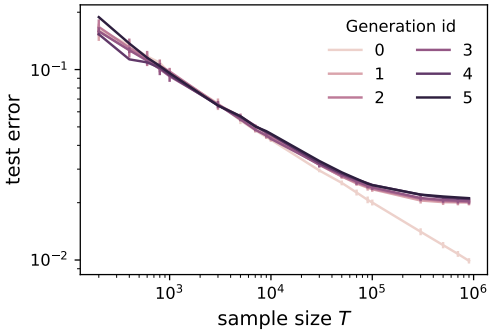

Figure 13: **Empirical Hutter LLM.** Bigram model with deterministic labeling function. Initial model trained on $T_0 = 100,000$ samples. It generates T samples for Gen 1. Starting from Gen 2 models are trained on data generated by the most powerful model from the previous generation. $\beta = 3/2$. In this setting, there is mild model collapse coming from the finite sample bias.

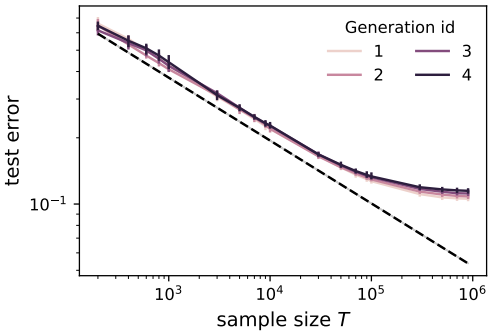

Figure 14: **Empirical Hutter++ Model.** Same setting as in Figure 6. Initial model trained on $T_0 = 100,000$ samples. No top-p inference or temperature scaling is used. $\beta = 3/2$. In this setting, there is mild model collapse coming from the finite sample bias as well.

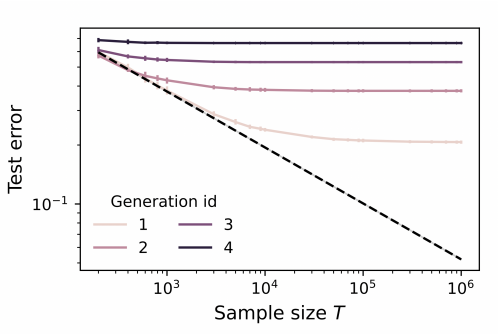

Figure 15: **Empirical Hutter++ Model.** Same setting as in Figure 14 with top $p = 0.90$ synthesizing. No temperature scaling is used. $\beta = 3/2$. Top-p selection significantly deteriorate the model collapse.

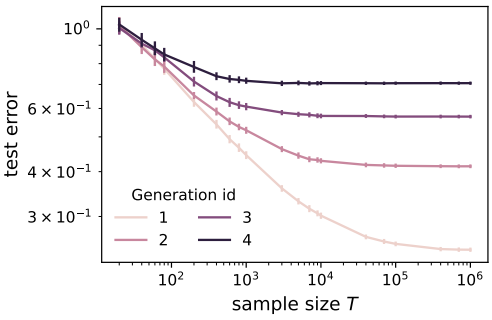

Figure 16: **Empirical Hutter++ Model.** Same setting as in Figure 14 with temperature 0.9 synthesizing. No top-p selection is used. $\beta = 3/2$. Compared with Figure 14, temperature also create strong model collapse across multiple generation.

We now study mixing of clean and synthesized data in the bigram setting. Figures 17 and 18 add top-p tail-cutting when synthesizing, and start with $T_0 = 10,000$ original data samples, which are successively blended with synthesized data from the largest model. Note that in this setting we observe a reversion of scaling laws with increased AI data. This needs to be compared with the orange curve in Figure 20 in the deterministic Hutter setting. The probabilistic nature of the bigram models leads to a new effect here.

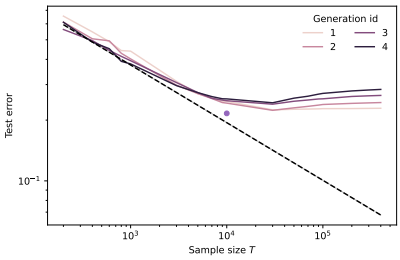

Figure 17: **Empirical Hutter++ Model with Mixing.** The initial "clean" dataset comprises $T_0 = 10,000$ samples. For future generations, the largest model is used to synthesize data. For $T \leq 20,000$, training data is an equal mix of clean and generated data, for $T > 20,000$ all clean data is used; the remaining training data is synthetic (so the ratio of clean data diminishes). Top-p 0.9, no temperature scaling, $\beta = 3/2$.

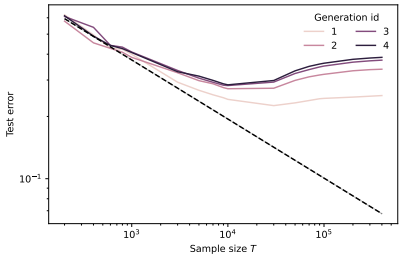

Figure 18: **Empirical Hutter++ Model with Mixing.** Same setting as in Figure 17 with top p 90, no temperature scaling, and $\beta = 3/2$.

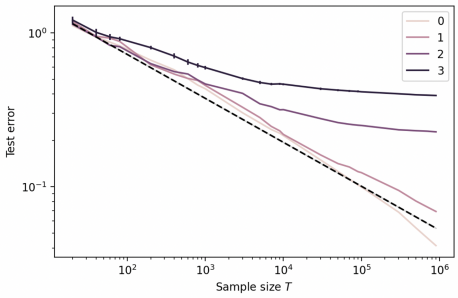

Figure 19: **S-shape "Smoothed Grokking".** Bigram data with Hutter++ model, mixing clean data with AI generated data with ratio 50 to 50. The grokking line is smoothed in the probabilistic setting. Line 1, 2, 3 are generated by using 10,000, 1,000, and 100 data to train the generating model. Compared to Figure 17, we do not limit the number of accessible real data now. $\beta = 3/2$.

C Proofs for the infinite memory (Hutter) Model (Sections 2 and 3)

C.1 Proof of Theorem 2.1

Observe that the model $\widehat{f}$ makes an error on i if and only if the ith "skill" never occurred in the training dataset $\mathcal{D}_T$, i.e either (1) $i \geq k+1$, or (2) $1 \leq i \leq k$ and $i_t \neq i$ for all $t \in [T]$. We deduce that

$$E_{test} = \mathbb{P}_{i\sim p}(\widehat{f}(i) \neq y_i) = \sum_{i \geq k+1} p_i + \sum_{1 \leq i \leq k} p_i(1-p_i)^T$$
$$\asymp k^{-(\beta-1)} + \sum_{1 \leq i \leq k} p_i e^{-p_i T},$$

where $c := 1 - 1/\beta \in (0,1)$, and we have used the elementary fact that $\sum_{i \geq k+1} i^{-\beta} \asymp k^{-(\beta-1)}$ for large k. For the second sum, we will need the following lemma.

Lemma C.1. *The following identity holds*

$$T^c \sum_{i=1}^{k} p_i e^{-Tp_i} \asymp \Gamma(c, Tk^{-\beta}) - \Gamma(c, T) = O(1), \tag{13}$$

where $\Gamma(s,x) := \int_x^\infty u^{s-1}e^{-u}\mathrm{d}u$ *defines the incomplete gamma function. In particular, for* $k=\infty$ *and large* T*, it holds that* $\sum_{i=1}^\infty p_i e^{-Tp_i} \asymp T^{-c}$.

Proof. Consider the function $h(z) := ze^{-Tz}$ for $z \in (0,1)$. Its derivative is $h'(z) = e^{-Tz}(1-Tz)$. Thus, h is increasing on $(0,1/T)$ and decreasing on $(1/T,\infty)$. Furthermore, note that $p_i \le 1/T$ iff $i \ge T^{1/\beta}$. We deduce that

$$\sum_{i=1}^k p_i e^{-Tp_i} \asymp \int_1^k x^{-\beta}e^{-Tx^{-\beta}}\mathrm{d}x.$$

Under the change of variable $u = u(x) := Tx^{-\beta}$, we have $x = x(u) = (u/T)^{-1/\beta}$ and so $\mathrm{d}x = -(T^{1/\beta}u^{-1-1/\beta}/\beta)\mathrm{d}u$. Also $u(1) = T$ and $u(k) = Tk^{-\beta}$. We deduce that

$$\begin{aligned}\sum_{i=1}^k p_i e^{-Tp_i} &\asymp \int_1^k x^{-\beta}e^{-Tx^{-\beta}}\mathrm{d}x = \int_{Tk^{-\beta}}^T (u/T)e^{-u}(T^{1/\beta}u^{-1-1/\beta}/\beta)\mathrm{d}u \\ &\asymp T^{-(1-1/\beta)}\int_{Tk^{-\beta}}^T u^{-1/\beta}e^{-u}\mathrm{d}u \\ &\asymp T^{-(1-1/\beta)}\left(\Gamma(1-1/\beta, Tk^{-\beta}) - \Gamma(1-1/\beta, T)\right) \\ &= T^{-c}\left(\Gamma(c,Tk^{-\beta}) - \Gamma(c,T)\right),\end{aligned}$$

and we are done for the first part.

For the second part, note that $\Gamma(c,T) = o(1)$ for large T so that

$$(\Gamma(c,Tk^{-\beta}) - \Gamma(c,T))|_{k=\infty} = \Gamma(c,0) - \Gamma(c,T) = \Theta(1) - o(1) = \Theta(1),$$

from which the result follows. □

We now consider two separate cases for the relative scaling of k and T.

– Case 1: $T \gtrsim k^\beta$**.** Here, we have thanks to Lemma C.1

$$E_{test} \asymp k^{-(\beta-1)} + O(T^{-c}) \asymp k^{-(\beta-1)}, \tag{14}$$

since $k^{-(\beta-1)} \gtrsim T^{-(\beta-1)/\beta} = T^{-c}$.

– Case 2: $1 \ll T \lesssim k^\beta$**.** Here, thanks to Lemma C.1 we have $\Gamma(c,T) = o(1)$ and $\Gamma(c,Tk^{-\beta}) = \Theta(1)$. We deduce that

$$E_{test} \asymp k^{-(\beta-1)} + T^{-c}\left(\Gamma(c,Tk^{-\beta}) - \Gamma(c,T)\right) \asymp k^{-(\beta-1)} + T^{-c} \asymp T^{-c}, \tag{15}$$

since $k^{-(\beta-1)} \lesssim T^{-(\beta-1)/\beta} = T^{-c}$. Putting things together then gives the claimed result. □

C.2 Proof of Corollary 2.3

Indeed, let $p_i \propto i^{-\beta}$ and $(p_{AI})_i = q_i \propto i^{-\beta'}$. Then,

$$E_{test} \asymp \sum_i p_i(1-q_i)^T \asymp \sum_i p_i e^{-q_iT} \asymp \int_1^\infty x^{-\beta}e^{-x^{-\beta'}T}\mathrm{d}x. \tag{16}$$

Setting $u = x^{-\beta'}T$ gives $x = T^{1/\beta'}u^{-1/\beta'}$, and so $\mathrm{d}x = -(T^{1/\beta'}/\beta')u^{-(1+1/\beta')}\mathrm{d}u$. We deduce that

$$\begin{aligned}E_{test} &\asymp T^{-(\beta-1)/\beta'}\int_1^T u^{\beta/\beta'}u^{-(1+1/\beta')}e^{-u}\mathrm{d}u = T^{-(\beta-1)/\beta'}\int_1^T u^{(\beta-1)/\beta'-1}e^{-u}\mathrm{d}u \\ &\asymp T^{-c}\Gamma(c,T) = T^{-c}(1+o(1)), \text{ with } c := (\beta-1)/\beta'.\end{aligned}$$

That is, $E_{test} \asymp T^{-c}$ as claimed. □

C.3 Proof of Theorem 3.2 and Corollary 3.3

Suppose that of T samples available for training our model, πT are samples from the true distribution $p = Zipf(\beta)$ and $(1-\pi)T$ are from AI data distribution p' which is a version of p with its tail chopped off at rank k, i.e such that $p'_i \propto p_i 1[i \leq k]$. Thus the dataset is drawn from the distribution given by $q_i = \pi p_i + (1-\pi)p'_i$. Test error of a Hutter LLM then writes

$$\begin{aligned} E_{test} &= \sum_{i\geq 1} p_i(1-q_i)^T = \sum_{1\leq i\leq k} p_i(1-p_i)^T + \sum_{i\geq k+1} p_i(1-\pi p_i)^T \\ &\asymp \sum_{1\leq i\leq k} p_i e^{-p_i T} + \sum_{i\geq k+1} p_i e^{-\pi p_i T}. \end{aligned} \tag{17}$$

Now, thanks to Lemma C.1, it is clear that for any integers $1 \leq r < R \leq \infty$ and large z, one has

$$\sum_{r\leq i\leq R} p_i e^{-p_i z} \asymp z^{-c}\left(\Gamma(c, zR^{-\beta}) - \Gamma(c, zr^{-\beta})\right), \tag{18}$$

where $c = 1 - 1/\beta \in (0,1)$ and Γ is the (upper) incomplete gamma function. Applying equation 18 with $(r,k,z) = (1,k,T)$ gives

$$T^c \sum_{1\leq i\leq k} p_i e^{-p_i T} \asymp \Gamma(c, Tk^{-\beta}) - \Gamma(c,T) = \begin{cases} \Theta(1) - o(1) = \Theta(1), & \text{if } 1 \ll T \lesssim k^\beta, \\ o(1) - o(1) = o(1), & \text{if } T \gtrsim k^\beta \gg 1. \end{cases} \tag{19}$$

On the other hand, applying equation 18 with $(r,k,z) = (k+1, \infty, \pi T)$ and assuming $\pi = \Theta(1)$ gives

$$\sum_{i\geq k+1} p_i e^{-\pi p_i T} \asymp (\pi T)^{-c}\gamma(c, \pi T(k+1)^{-\beta}) \asymp \begin{cases} (\pi T)^{-c}, & \text{if } \pi T \gtrsim k^\beta \gg 1, \\ (k+1)^{-\beta c} \asymp k^{-\beta c}, & \text{if } k^\beta \gg \pi T. \end{cases} \tag{20}$$

Putting things together gives the result. □

Recall that Bertrand et al. (2023) also formally study such mixtures for iterative retraining. In their setting, they show the existence of fixed points in the mixture proportion that delineates the region of model collapse. These results are complimentary and not contradictory to ours: they combine mixing, large number of iteration, and data-decay, thus studying a combination of effects (under different theoretical conditions, not focusing on scaling laws) that our preceding theorems address separately.

C.4 Grokking for Tail Narrowing

Theorem C.2 (Grokking with Tail Narrowing)**.** *Consider a sample of size T of which a proportion π comes from the true distribution $p = Zip(\beta)$ and the remainder comes from a version $p' = Zip(\beta')$. We have the following scaling law for the Hutter LLM,*

$$E_{test} \asymp (\pi T)^{-c} + ((1-\pi)T^{-c'}), \tag{21}$$

where $c := (\beta-1)/\beta$ and $c' := (\beta-1)/\beta'$.

Define $\overline{T} := (\pi/(1-\pi))^{-a}$, where $a := s/(1-s)$, and $s := \beta/\beta'$. Then,

*(A) **Early-Stage Dynamics.** For $T \lesssim \overline{T}$, it holds that $E_{test} \asymp ((1-\pi)T)^{-c'}$. Thus, if $\beta' > \beta$, the money spent on acquiring some clean data is not amortized!*

*(B) **Later-Stage Dynamics.** As soon as $T \gtrsim \overline{T}$, it holds that $E_{test} \asymp (\pi T)^{-c}$. Similarly, we recover the unpolluted sample-size law scaling T^{-c}. For fixed T and tunable π, this error rate scales like π^{-c}.*

Proof. Let q be the mixture of p and p'. We prove the result for $\beta' \geq \beta$; the case $\beta' \leq \beta$ is analogous. So, one may write

$$E_{test} = \sum_{i\geq 1} p_i(1-q_i)^T \asymp \sum_{i\geq 1} p_i e^{-\pi i^{-\beta} + (1-\pi)i^{-\beta'}} \asymp \sum_{1\leq i\leq \overline{T}^{1/\beta}} p_i e^{-\pi i^{-\beta}} + \sum_{i\geq \overline{T}^{1/\beta}} p_i e^{-(1-\pi)i^{-\beta'}}, \tag{22}$$

where we have used the fact that $(1-\pi)i^{-\beta'} \geq \pi i^{-\beta}$ iff $i \leq (\pi/(1-\pi))^{-1/(\beta'-\beta)} = \overline{T}^{1/\beta}$. The result then follows from equation 18. □

Remark C.3. Let us conclude by saying that clean data always helps, since E_{test} is decreasing function of π. Indeed, from equation 17, the derivative w.r.t π is $E'_{test}(\pi) = -T\sum_{i\geq k+1} p_i^2(1-\pi p_i)^{T-1} \leq 0$.

C.5 An interesting detour: Grokking for Fixed-size AI Dataset.

Now consider the scenario where the AI synthesized dataset has fixed size T_{AI} (e.g a frozen chunk of the web), while the clean dataset size is a scalable parameter T_{real}. Taking $T = T_{real} + T_{AI}$ and $\pi = T_{real}/T$, we have the following corollary of Theorem 3.2, which includes Corrolary 3.3.

Corollary C.4. *We have the following.*

(A) Early-Stage Dynamics. *For $T_{real} \ll k^{\beta}$, it holds that*

$$E_{test} \asymp (T_{real} + T_{AI})^{-(1-1/\beta)} + k^{-(\beta-1)} \tag{23}$$

(B) Later-Stage Dynamics. *As soon as $T_{real} \geq Ck^{\beta}$ (where C is an absolute constant), it holds that*

$$E_{test} \asymp T_{real}^{-(1-1/\beta)}. \tag{24}$$

As mentioned in Section 3, AI synthesized data is helpful in the regime where real data is scarce. Once more of real data becomes available the model grokks for a while and then forgets the AI synthesized data to recover the normal scaling law w.r.t T_{real}. Figure 20 gives an illustration of this phenomenon in various settings.

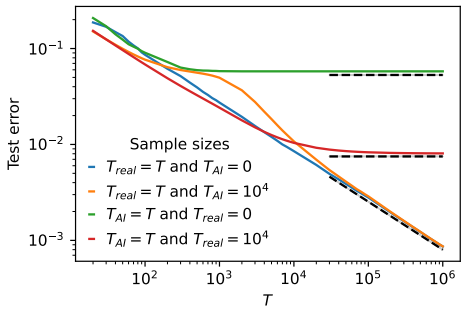

Figure 20: **Hutter LLM.** true distribution of the data is Zipf with exponent $\beta = 2$. Here, the scalable resource is either clean data or AI data-generated data, corresponding to a version of real data with its tail cut at rank k (here we use $k = 10$). We either mix with a fixed amount (here $T' = 10^4$ samples) of the other resource, or we don't mix at all. Then we scale up the scalable resource by cranking up T. As predicted by Corollary 3.3, the orange curve always grokks: AI synthesized data is helpful in the regime where real data is scarce; once more of real data becomes available the model grokks for a while and then forgets the AI synthesized data. Note that the green curve (only AI data) and red curve (AI + real data) don't grokk because the optional resource (real data) is not being scaled; if it is also scaled, then green and red will provably grokk (as in Figure 3). The diagonal broken line corresponds to the standard Hutter scaling law $E_{test} \asymp T^{-c}$, where $c := 1 - 1/\beta$. The horizontal broken lines correspond to $E_{test} \asymp k^{-(\beta-1)}$ and $E_{test} \asymp T'^{-c}$, both predicted by Theorem 2.1.

C.6 Proof of Theorem 3.1

Note that explicitly,

$$\pi_i \asymp \begin{cases} N^{\alpha} p_i, & \text{if } i \geq N, \\ 0, & \text{else,} \end{cases} \tag{25}$$

where $\alpha := \beta - 1$. This is because the normalization constant is $\sum_{i \geq N} p_i = \sum_{i \geq N} i^{-\beta} \asymp N^{-\alpha}$. Now, mix this distribution with q with equal weights $1/2$, to obtain a new distribution

$$\begin{aligned} q'_i = q_i/2 + \pi_i/2 &= \begin{cases} q_i/2, & \text{if } i \leq k, \\ \pi_i/2, & \text{if } k \geq N, \\ 0, & \text{otherwise} \end{cases} \\ &\asymp \begin{cases} p_i, & \text{if } i \leq k, \\ N^{\alpha} p_i, & \text{if } k \geq N, \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{26}$$

For simplicity, assume $N \geq k + 1$ (otherwise, we have all of p). Build a "Hutter" LLM from an iid sample of size T from this distribution (this is equivalent to mixing T samples from q and T samples from π. Then, it is easy to see that the test error is given by

$$\begin{aligned} E_{test} = \sum_{i \geq 1} p_i (1 - q'_i)^T \asymp \sum_{1 \leq i \leq k} p_i (1 - p_i)^T + \\ \sum_{k+1 \leq i \leq N-1} p_i + \sum_{i \geq N} p_i (1 - N^{\alpha} p_i)^T. \end{aligned} \tag{27}$$

Thanks to previous computations, we know that for large k, N, and T

- The first sum is of order $T^{-c} \left(\Gamma(c, Tk^{-\beta}) - \Gamma(c, T) \right) = O(T^{-c})$.
- The third sum is of order $T^{-c} \left(\Gamma(c, 0) - \Gamma(c, TN^{\alpha} N^{-\beta}) \right) = T^{-c} \left(\Gamma(c, 0) - \Gamma(c, TN) \right) \asymp T^{-c}$.
- The second sum is of order $k^{-\alpha} - N^{-\alpha} = ((\frac{N}{k})^{\alpha} - 1) N^{-\alpha}$, where $\alpha := \beta - 1$.

We deduce that

$$E_{test} \asymp T^{-c} + \left(\left(\frac{N}{k} \right)^{\alpha} - 1 \right) N^{-\alpha}, \text{ for large } k, N, T, \tag{28}$$

and the result follows. □

D Proofs for the Tailed Bigram Model (Section 4)

D.1 The Hutter++ Algorithm

We now present an extension of the Hutter model equation 6 which is adapted to bigrams. Let $n_T(i) = \sum_{t=1}^{T} 1[i_t = i]$ be the number times the context i_t appears in the dataset $\mathcal{D}_T$ and $n_T(i, j) = \sum_{t=1}^{T} 1[(i_t, j_t) = (i, j)]$ be the number of times the pair (i, j) appears in the dataset. Note that $n_T(i) \sim Bin(T, p_i)$. As soon as $n_T(i) \geq 1$, define

$$q_T(j \mid i) := n_T(i, j)/n_T(i).$$

This is an empirical version of $p(\cdot \mid i)$ based on an iid sample of size $n_T(i)$. For a theoretical analysis, we shall consider the following test error metric based on total-variation (TV) $E_{test} := \sum_i p_i \, \mathbb{E}\left[TV(q_T(\cdot \mid i), p(\cdot \mid i))\right]$, where $TV(a, b) := \sum_j |a_j - b_j|$ is the total-variation distance and the expectation is over the randomness in q_T. An asset here is that Berend & Kontorovich (2012) can be used to control the quantities $\mathbb{E}\left[TV(q_T(\cdot \mid i), p(\cdot \mid i))\right]$. Note that TV is upper-bounded by the square-root of KL-divergence, thanks to *Pinker's inequality.* This gives indication that our results could also apply in the setting of autoregressive models with perplexity loss, like LLMs.

D.2 Warm-up: Revisiting the Classical Hutter Setup

As a sanity check, with the Total Variation Loss, let us momentarily consider the non-autoregressive setup where $p(\cdot \mid i) = \delta_{y_i}$ for all i, as in classical Hutter. Then, an easy computation shows that

$$TV(q_T(\cdot \mid i), p(\cdot \mid i)) = 1 - q_T(y_i \mid i) + \sum_{j \neq y_i} q_T(j \mid i) = 2(1 - q_T(y_i \mid i)).$$

Now, by construction, $q_T(y_i \mid i) = 1[i \in \mathcal{D}_T]$. Thus,

$$\mathbb{E}\left[1 - q_T(y_i \mid i)\right] = \mathbb{P}(i \notin \mathcal{D}_T) = (1-p_i)^T.$$

We deduce that

$$\mathbb{E}\left[TV(q_T(\cdot \mid i), p(\cdot \mid i))\right] = 2(1-p_i)^T.$$

Therefore,

$$\begin{aligned} E_{test} &= \sum_i p_i \mathbb{E}\left[TV(q_T(\cdot \mid i), p(\cdot \mid i))\right] \\ &= 2\sum_i p_i (1-p_i)^T \asymp T^{-(1-1/\beta)}, \end{aligned} \tag{29}$$

and we recover the classical Hutter result! Thus, our test metric is pointing in the right direction, conceptually.

D.3 Proof of Theorem 4.1

The proof will be based on the results of Berend & Kontorovich (2012).

Upper-Bound. Observe that for any choice of mappings $\pi_1, \pi_2, \ldots$, we have

$$\begin{aligned} a_T(i) &:= \sum_{j \mid p(j|i) \le 1/n_T(i)} p(j \mid i) \asymp \sum_{j \mid \pi_i(j) \ge n_T(i)^{1/\beta}} \pi_i(j)^{-\beta} \le \sum_{k \mid k \ge n_T(i)^{1/\beta}} k^{-\beta} \asymp n_T(i)^{-(1-1/\beta)} \\ b_T(i) &:= n_T(i)^{-1/2} \sum_{j \mid p(j|i) \ge 1/n_T(i)} \sqrt{p(j \mid i)} \asymp n_T(i)^{-1/2} \sum_{j \mid \pi_i(j) \le n_T(i)^{1/\beta}} \pi_i(j)^{-\beta/2} \\ &\lesssim n_T(i)^{-1/2} \sum_{k \mid k \le n_T(i)^{1/\beta}} k^{-\beta/2} \asymp n_T(i)^{-c}. \end{aligned}$$

We deduce that $c_T(i) := a_T(i) + b_T(i) \lesssim n_T(i)^{-c}$ for any i. Importantly, the hidden constants don't depend on i. Therefore, thanks to [Lemma 9] Berend & Kontorovich (2012), we have

$$\begin{aligned} E_{test} &\le \sum_i p_i \mathbb{E}\left[c_T(i)\right] \lesssim \sum_i p_i \mathbb{E}\left[n_T(i)^{-c}\right] \overset{(*)}{\le} \sum_i p_i (\mathbb{E}\left[n_T(i)\right])^{-c} = \sum_i p_i (T p_i)^{-c} = T^{-c} \sum_i p_i^{1-c} \\ &\lesssim T^{-c}, \end{aligned} \tag{30}$$

where we have used Jensen's inequality in (*), since the function $x \mapsto x^{-c}$ is concave.

Lower-Bound. WLOG[5] consider the following specific choice of permutations defined by $\pi_i(j) = j$ (i.e doesn't depend on i). Then,

$$\begin{aligned} a_T(i) &= \sum_{j \ge n_T(i)^{1/\beta}} j^{-\beta} \asymp n_T(i)^{-(1-1/\beta)}, \\ b_T(i) &= n_T(i)^{-1/2} \sum_{j \le n_T(i)^{1/\beta}} j^{-\beta} \asymp n_T(i)^{-c}. \end{aligned}$$

Thanks to the definition of E_{test} and [Proposition 5] Berend & Kontorovich (2012), we deduce that if $\beta \in (1,2)$, then

$$E_{test} \ge \sum_i p_i \mathbb{E}\left[(a_T(i) + b_T(i) - n_T(i)^{-1/2})\right] \asymp \sum_i p_i \mathbb{E}\left[n_T(i)^{-c} - n_T(i)^{-1/2})\right] \asymp \sum_i p_i \mathbb{E}\left[n_T(i)^{-c}\right], \tag{31}$$

i.e $E_{test} \gtrsim \sum_i p_i \mathbb{E}\left[n_T(i)^{-c}\right]$. Now, since $n_T(i) \sim Bin(T, p_i)$, standard Binomial concentration arguments tell us that $n_T(i) \le 1.5 T p_i$ w.p $1 - e^{-C p_i T}$, where C is an absolute constant. We deduce that

$$E_{test} \gtrsim \sum_i p_i (1.5 T p_i)^{-c} (1 - e^{-C p_i T}) \asymp T^{-c} \sum_i p_i^{1-c} - T^{-c} \underbrace{\sum_i p_i^{1-c} e^{-C p_i T}}_{o(1)} \asymp T^{-c},$$

which completes the proof. □

[5] A summable series of nonnegative numbers (like in $a_T(i)$ and $b_T(i)$) can be reordered without changing the value.

D.4 PROOF OF THEOREM 4.2

It suffices to replace $n_T(i)$ in equation 30 and equation 31 of the proof of Theorem 4.1 with $n_T(i) \wedge k^\beta$, and use the elementary fact that $(n_T(i) \wedge k^\beta)^{-c} = n_T(i)^{-c} \vee k^{-\beta c} \asymp n_T(i)^{-c} + k^{-\beta c}$. The rest of the proof proceeds as that of Theorem 4.1.

D.5 EXTENSIONS

Note that the above setup can be extended to the following

$$p(j \mid i) = \rho(\pi_i(j)),$$

where ρ is a distribution on $\mathbb{N}_*$. In particular, taking $\rho(z) \propto z^{-\beta}$, recovers the setup considered above. It is clear that mechanics of the proof of Theorem 4.1 should be applicable here, leading to scaling laws which depend explicitly on ρ.

E CAPACITY-LIMITED MEMORY MODELS: A TRIPLET SCALING LAW

We now consider study model collapse phenomenon in the context of the following simple associative memory model studied in Cabannes et al. (2023)

$$\begin{aligned} f_T(i) &:= \arg\max_y H_T(i,y), \text{ where} \\ H_T(i,y) &:= e_i^\top M_T u_y, \\ M_T &:= \sum_i q_T(i) e_i u_{f_\star(i)}^\top \in \mathbb{R}^{d\times d}. \end{aligned} \tag{32}$$

This is a transformer-like finite-memory extension of the infinite-memory model in Hutter (2021). The integer $d \geq 1$ then plays the role of the "capacity" of the resulting model. Here, $f_\star : [N] \to [m]$ is an unknown function, for example, reduction modulo m, i.e $f_\star(i) := ((i-1) \bmod m) + 1$; $q_T = q(\mathcal{D}_T)$ is probability distribution on $[N]$ which encodes an arbitrary learner, estimated using and iid sample $\mathcal{D}_t = \{(i_t, y_t) \mid t \in [T]\}$ of size T collected from a probability distribution on $[N] \times [m]$, of the form

$$i \sim p = \text{Zipf}(\beta), \quad y = f_\star(i). \tag{33}$$

The embedding vectors $e_1, e_2, \ldots e_N$ and $u_1, u_2, \ldots, u_m$ are a system of unit-vectors in $\mathbb{R}^d$, constructed so that the matrix $\mathbb{R}^{d\times d}$ remembers the input/output pairs (i,j) it has seen, i.e $e_i^\top M u_{f_\star(i)} \approx q_T(i)$ if $(i, f_\star(i)) \in \mathcal{D}_T$. The weights $q_T(i)$ ensure that different memories are memorized faster than others.

Cabannes et al. (2023) proposed that iid random embeddings from the uniform distribution on the unit-sphere in $\mathbb{R}^d$ be used. In this setting, for different choices of q, the following general scaling law was established

$$E_{test} \asymp T^{-(1-1/\beta)} + d^{-c_q}, \tag{34}$$

where the exponent $c_q \in (0,\infty)$ depends on β and the algorithm q. For example, when q encodes the counting measure $q_T(i) := n_T(i)/T$ (reminiscent of SGD), it was shown that $c_q = (1-1/\beta)/2 \in (0, 1/2)$. Another algorithm $q_T(i) := 1[n_T(i) \geq 1]/\sum_\ell 1[n_T(\ell) \geq 1]$ (remniscent of ADAM) was proposed which attains a optimal error rate (over all algorithms based on random embeddings) with $c_q = \beta - 1$.

In the context of model collapse which is the main focus of this manuscript, we have the following.

Theorem E.1 (Triplet Scaling Law)**.** *For all the algorithms q considered in Cabannes et al. (2023), one has the following triplet scaling law w.r.t sample size T, embedding dimension d, and frequency cutoff k,*

$$E_{test} \asymp T^{-(1-1/\beta)} + d^{-c_q} + k^{-(\beta-1)}. \tag{35}$$

This result is empirically confirmed in Figure 21 and proved in Appendix F.

F PROOF AND ILLUSTRATION OF TRIPLET SCALING LAW (THEOREM E.1)

For any i, on average it takes $1/p_i$ iid samples from p to see the context i at least once. The effect of tail-cutting at rank k is effectively to replace the sample size T by $\min(T, T_k)$, where $T_k = \max\{1/p_i \mid i \in [k]\}$. In the case where $p = Zipf(\beta)$, we have $T_k = 1/p_k \asymp k^\beta$. On other hand the model equation 32 proposed in Cabannes et al. (2023) on Zipf data, the test error writes

$$E_{test} \asymp T^{-c} + d^{-c_q}, \tag{36}$$

where $c := 1 - 1/\beta \in (0,1)$ and the exponent $c_q \in (0,\infty)$ depends on β and the algorithm q used to update the embeddings in the memory matrix M_T in equation 32. We deduce that tail-cutting at rank k changes the test error to

$$E_{test} \asymp \min(T, T_k)^{-c} + d^{-c_q} \asymp T^{-c} + k^{-\beta c} + d^{-c_q},$$

as claimed. □

Figure 21 confirms the Triplet Scaling Law.

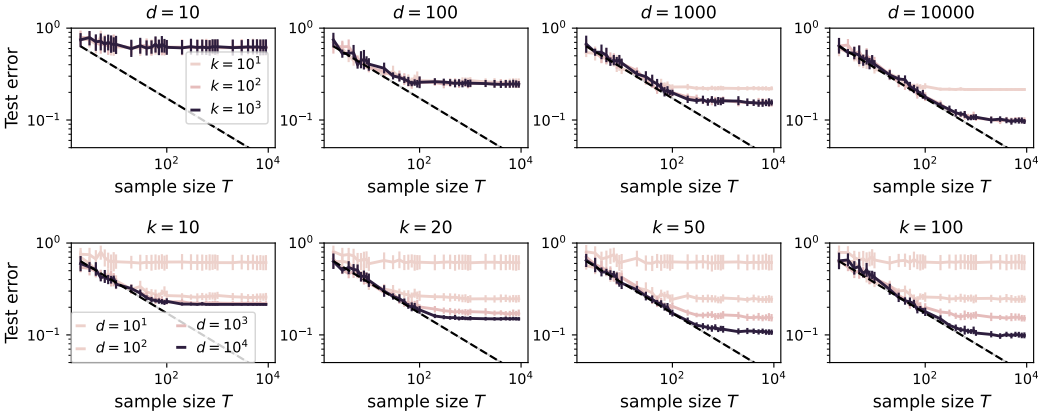

Figure 21: **Capacity-Limited Memory Models.** Empirical confirmation of the Triplet Scaling Law established in Theorem E.1

G Details and Results from the Autoregressive Bigram model with Perplexity

We showcase experiments in the autoregressive bigram model with perplexity loss. We generate sequences of length 100. Figures 22, Figure 23 and 24 aim to reproduce the "paired bigram" Figure 12 in this setting, adding a top p mechanism and a temperature mechanism. Figure 26, Figure 26 and Figure 27 regenerates the setting of Figure 6 with the same top p and temperature.

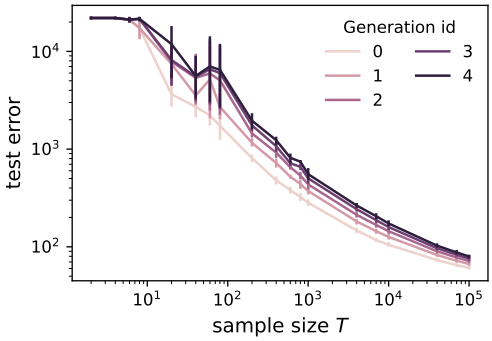

Figure 22: **Autoregressive Bigram Model with Perplexity Loss.** Empirical confirmation of Theorem 2.4 for autoregressive data with top-p 1, Temperature 1. Each sequence data have length 100. Same setting as Figure 12. $\beta = 3/2$.

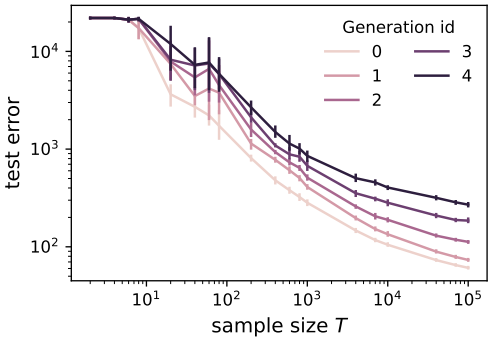

Figure 23: **Autoregressive Bigram Model with Perplexity Loss.** Empirical confirmation of Theorem 2.4 for autoregressive data with top-p 0.9, Temperature 1. Each sequence data have length 100. $\beta = 3/2$.

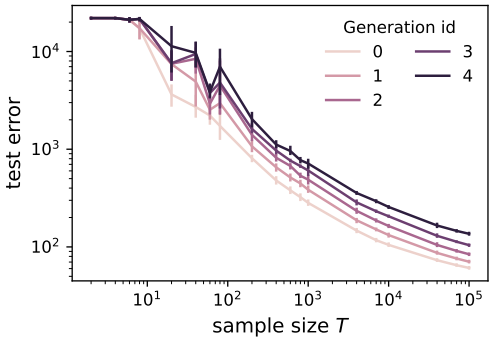

Figure 24: **Autoregressive Bigram Model with Perplexity Loss.** Empirical confirmation of Theorem 2.4 for autoregressive data with top-p 1, Temperature 0.9. Each sequence data have length 100. Same setting as Figure 12. $\beta = 3/2$.

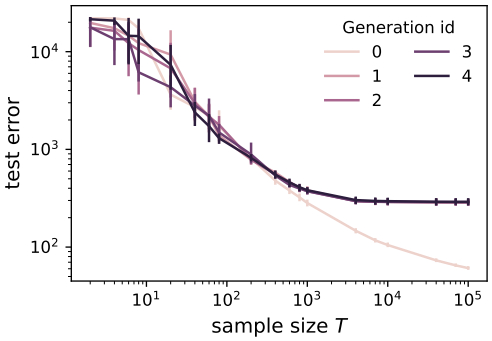

Figure 25: **Autoregressive Bigram Model with Perplexity Loss.** Each sequence data have length 100. Initial model trained on $T_0 = 10,000$ samples. It generates T samples for Gen 1. Starting from Gen 2 models are trained on data generated by the most powerful model from the previous generation. Top-p 1, temperature 1, $\beta = 3/2$.

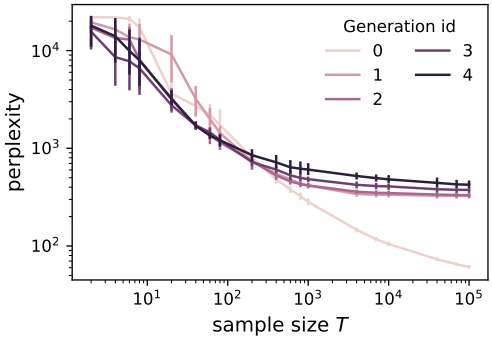

Figure 26: **Autoregressive Bigram Model with Perplexity Loss.** Each sequence data have length 100. Same setting as Figure 25. Top-p 0.9, temperature 1, $\beta = 3/2$.

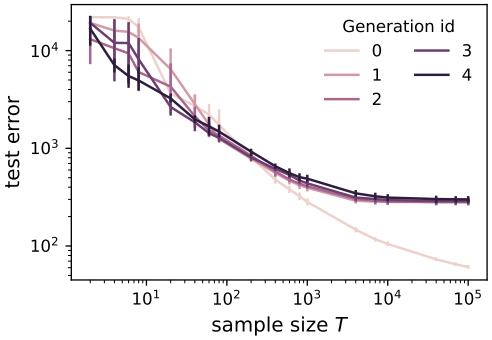

Figure 27: **Autoregressive Bigram Model with Perplexity Loss.** Each sequence data have length 100. Same setting as Figure 25. Top-p 1, temperature 0.9, $\beta = 3/2$.

H DETAILS AND RESULTS ON TRANSFORMER ARITHMETIC EXPERIMENTS

Charton (2023) trains sequence-to-sequence transformers to predict the greatest common divisor (GCD) of two positive integers, encoded as sequences of digits in some base B. He observes that model predictions are deterministic: for any pair (a, b) with GCD k, the model predicts a single value $f(k)$. Predictions are correct (i.e. $f(k) = k$) when the GCD is a product of divisors of the base, or of small primes. In all other case, the model prediction is the largest correct prediction (i.e. l such that $f(l) = l$) that divides k. The list of correct predictions $\mathcal{L}$ varies with the encoding base B. For instance, for $B = 10$, after 300 million examples, the model correctly predicts $\mathcal{L} = \{1, 2, 4, 5, 8, 10, 16, 20, 25, 40, 50, 80, 100...\}$, the GCD of 20 and 30 will be correctly predicted as 10, but the GCD of 210 and 140 will be incorrectly predicted as 10 (instead of 70).

We use these models to generate "dirty" training data $\mathcal{D}(B)$: uniformly sampled pairs of integers (a, b) and their (sometimes incorrect) pseudo-GCD, as generated by a trained transformer using base B. Note: this dataset can be as large as we want. We also create a correct training dataset $\mathcal{C}(B)$, by sampling pairs (a, b) and their correct GCD.

In these experiments, we train models on $\mathcal{D}(B)$ and $\mathcal{C}(B)$, for different values of B. Our goal is to determine whether extensive training on "dirty" data impacts model accuracy.

We focus on 6 bases: $B = 10, 420, 1000, 2017, 2023$ and 4913, after training transformers (on correct GCD) over about 300 millions pairs of integers between one and one million, we achieve the performances listed in Table 1. There, accuracy stands for the proportion of random uniform pairs (a, b) that the model can predict correctly, correct GCD is the number of GCD under 100 that the model correctly predicts (i.e. k such that $f(k) = k$), and correct model predictions are the products of numbers in the associated sets. These models are used to generate $\mathcal{D}(B)$.

In these experiments, all models have four layers, 512 dimensions and 8 attention heads. We consider two architectures: an encoder-only model (17.2M parameters), and an encoder-decoder model (38.7M parameters). The encoder-only model has 2.25 times less parameters, trains twice as fast, and incurs no performance penalty.

Table 1: **Initial performances.** 4-layer transformers trained to predict GCD, on 300 million examples. Our *test* set only contains GCD up to 100, and accuracy is computed on a reweighted test with equal occurance of each GCD. Thus, the Correct GCD lists all those that can be formed from the correct predictions by forming products across the sets (within the first 100 GCD). We freeze the 0th generation model at this stage and use its prediction to generate synthetic data. For each GCD outside the set of its correct predictions, the model will predict the largest GCD it has learned that divides the ground truth.

Base	Accuracy	Correct GCD	Correct predictions
10	85	13	{1,2,4,8,16} {1,5,25}
420	97	38	{1,2,4,8,16}{1,3,9}{1,5,25}{1,7}
1000	94	22	{1,2,4,8,16} {1,5,25}{1,3}
2017	85	4	{1,2}{1,3}
2023	91	16	{1,2,4}{1,3}{1,7}{1,17}
4913	93	17	{1,2,4}{1,3}{1,5}{1,17}

We then train new models (with the same architecture) to predict GCD, from AI data (generated by the above model), and compare to training with correct data – from correct computation of the GCD. When trained on small number of examples (less than 100 million), models learning from AI data achieve better accuracy (Table 2). We believe this is due to the fact that AI data smoothes away all the hard case, therefore presenting the model with a cleaner signal in the initial stages.

This pattern changes after extensive training. Table 3 compares performance of models trained on $300M$ and 1 billion examples. For all bases B, models trained on $\mathcal{C}(B)$ learn new GCD as training proceeds, whereas models learned on $\mathcal{D}(B)$ never learn beyond their original performance.

Figures 28 and 29 show that we get the picture predicted by theory: the dirty model learns (until about 300M examples) and then stops learning (while the clean model continues) - its scaling law tapers off as predicted in Theorem 2.1. All the skills the clean model learns after this point are skills the model trained on synthesized data cannot learn (see Figure 30 showing when new learned groups

Table 2: **Correctly predicted GCD after 30, 60 and 90 million examples.** Dirty and correct datasets.

	30M examples		60M examples		90M examples	
Base	AI	Correct	AI	Correct	AI	Correct
10	13	13	13	13	13	13
420	34	34	38	34	38	35
1000	17	13	22	13	22	14
2017	4	2	4	2	4	4
2023	6	6	11	6	11	6
4913	6	4	7	7	7	7

Table 3: **Correctly Predicted GCD after** $300M$ **and** 1 **Billion Examples.** AI and correct datasets.

	300M examples		1B examples	
Base	AI	Correct	AI	Correct
10	13	14	13	31
420	38	38	38	40
1000	22	25	22	33
2017	4	6	4	9
2023	16	16	16	32
4913	17	16	17	31

of GCD emerge, and Figure 31 for the learning curve of two models, one trained on the original data, the other on AI data).

Mixing and Grokking We now proceed to train our model on randomly mixed clean and synthesized data for various mixture rates. We train with mixtures of clean and dirty data for mixture fractions of $9\%, 27\%, 50\%$ and 73% of AI-generated data, for bases 1000, 2023 and 4913, to see the grokking effect. Figure 32 illustrates the results. We can see that even for the average curves over the 10 seeds one can discern a grokking-like delayed learning for the mixtures with relatively small amounts of AI data. This effect can be studied

The models used to generate the data were trained on about 300M examples, and correctly predict 22, 16 and 17 GCD below 100 for bases 1000, 2023 and 4913 respectively. We know (Table 3) that more training on AI-data data only will not improve those performances. On the other hand, we know that models trained on clean data will achieve larger performance. Specifically, out of 10 models trained on clean data, for base 1000, all 10 predict 23 GCD or more after 1.4B examples. The median number of examples needed for the models to predict 23 GCD or more is 465M. For base 2023, 7 models out of 10 predict 17 GCD or more after 2.1B examples. The median number of training samples after which the model bests a model trained on dirty data only is 530M. Finally, for base 4913, 9 clean models out of 10 predict more than 18 GCD after 1.7B examples. The median number of samples is 1.1B.

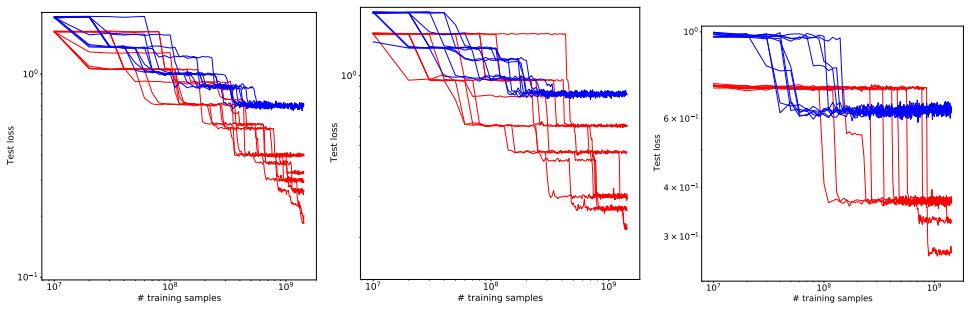

Figure 28: **Test loss for GCD learning.** Test loss of 10 models trained on clean and generated data. From left to right: base 4913, 2023, 1000. Models trained on clean data (red) continune to learn (decreasing test loss) while models trained on AI generated data (blue) stops learning.

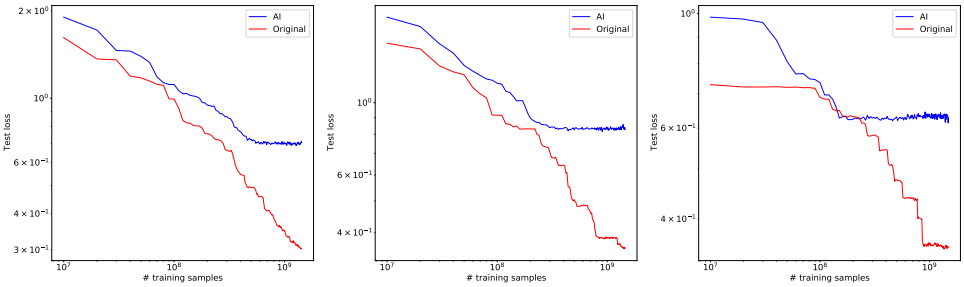

Figure 29: **Average test loss for GCD learning.** Averaged over 10 models trained on clean and generated data. From left to right: base 4913, 2023, 1000.

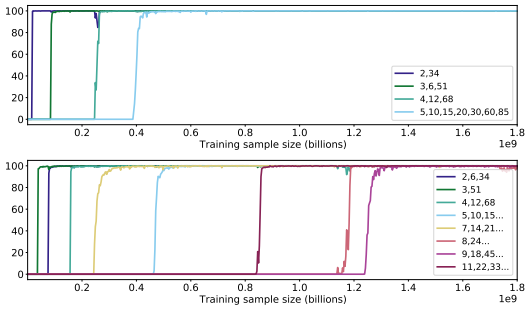

Figure 30: **Emergence of skills (groups of GCDs learned together).** Original (bottom) and AI-synthesized data (top). Base 4913. 1 model for clean/AI data, respectively.

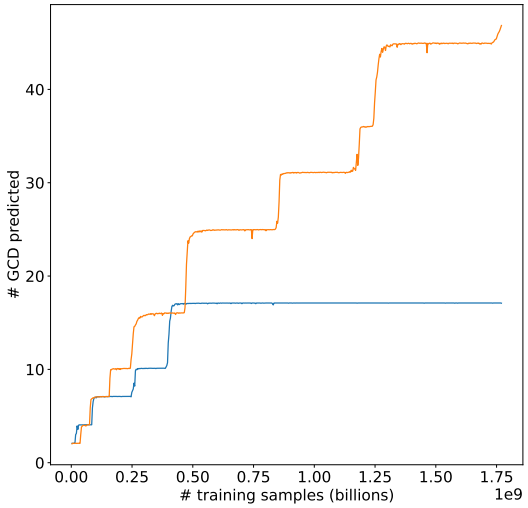

Figure 31: **Learning the GCD.** Learning curve, base 4319. Orange: training on correct GCD. Blue: training on AI generated data.

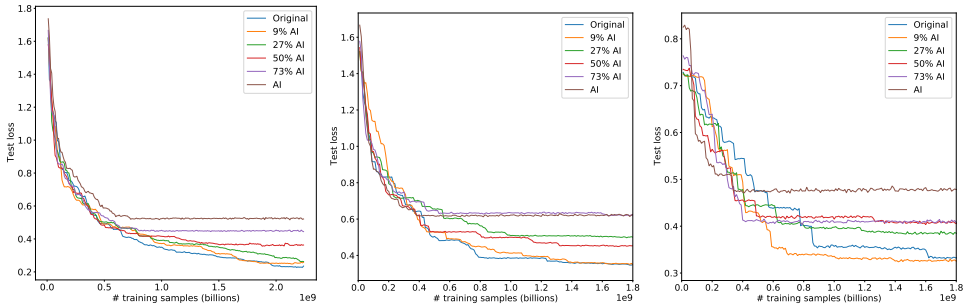

Figure 32: **Grokking in GCD Learning on mixed data.** Error losses of models trained on mixtures of clean and AI generated GCD data. 10 models. From left to right: base 4913, 2023 and 1000.

When zooming in to when the mixture models learn to predict GCD that are "unlearnable" with an AI-trained model, the grokking effect becomes more apparent.

Table 4 summarizes by listing the time (# of samples) when the mixture models finally learn a GCD that a purely AI-trained model cannot learn, and the delay (in millions samples) since the previous GCD was learned (see also Figure 30 to illustrate the comparison between the clean and the AI-trained model):

Table 4: **Samples until Mixture Models Learn a GCD that AI-trained Models Cannot Learn.** * small number of experiments

	Base 1000			Base 2023			Base 4913		
mixture rate	successes	samples (M)	delay	successes	sample (M)	delay	successes	samples (M)	delay
0% (clean)	10/10	465	243	7/10	530	567	10/10	1180	520
9%	8/10	560	320	8/10	715	530	9/10	910	340
27%	5/10	790	560	7/10	790	1220	10/10	1390	680
50%	2/10*	1310*	190*	7/10	1140	1220	8/10	1280	1180
73%	0	-	-	0	-	-	0	-	-

The delay period increases with increasing fraction of AI data in the mix. Thus, Table 4 clearly demonstrates the grokking effect of increasing plateau length with fraction of AI data, as predicted by our theory[6].

I Details of Experiments with Llama2

In the realm of large language models (LLMs), the prevailing approach involves a pretraining and finetuning paradigm. For instance, GPT-3 undergoes pretraining on approximately 45TB of text data from diverse sources. This extensive pretraining endows it with a robust capability for a variety of downstream tasks, employing methods such as zero-shot learning, few-shot learning, or finetuning. Our study evaluates the phenomenon of model collapse in scenarios close to the contemporary 'synthetic data age.'

Utilizing one of the most advanced open-source models, `Llama-2 7B`, our research investigates the effects on LLMs when they undergo finetuning[7] with data generated by other LLMs. To ensure

[6]We were constrained to stop the experiments at after about 3B samples for most, due to heavy use of compute resources. This probably explains why for the larger AI-mixtures only a few experiments could successfully find new GCDs - the other experiments where still in the pre-grokking phase when they were stopped.

[7]Quoting Shumailov et al. (2023), we state that one can, in principle, replicate an experiment described here with training an LLM from scratch to demonstrate scaling law decay. Given that training a single moderately large model produces twice the American lifetime worth of CO2 Strubell et al. (2019), we opted to not run such an experiment and instead focus on a more feasible finetuning setting. Note that just the language experiments described in the paper took weeks to run.

the generation of high-quality data and to provide a relevant but not trivial downstream task, we employ the Wikitext-103 dataset. We segment this dataset into chunks of 128 tokens, between each with a stride of 64 tokens, resulting in approximately 2.2 million chunks. Denote this dataset as $\mathcal{D}_0$. The task for generation involves producing the final 32 tokens given the initial 96 tokens from each chunk in the original dataset. In the initial generation (0-th generation), we use the `Llama-2 7B FT` model, which has been finetuned on $\mathcal{D}_0$, applying a generation loss that focuses solely on the cross-entropy loss of the final 32 tokens. We denote this initial model as $\mathcal{M}_0$, which demonstrates enhanced capacity for the generation task compared to the standard Llama-2 7B model. By querying $\mathcal{M}_0$ with the original 96 tokens from $\mathcal{D}_0$, we generate the dataset $\mathcal{D}_1$ and subsequently finetune `Llama-2 7B` on this dataset to obtain $\mathcal{M}_1$. This process is sequentially repeated to generate $\mathcal{D}_i$ from $\mathcal{M}_{i-1}$ and obtain $\mathcal{M}_i$ through finetuning. By comparing the performance of various $\mathcal{M}$ models on the test set derived from Wikitext-103, also segmented into 128-token chunks, we aim to investigate the model collapse in LLMs.

To prevent information leakage across chunks, we restrict the training to only include the loss on the final 32 tokens for all generations. Consequently, the models are never trained on the first 96 tokens coming from the original corpus. The size of the 2.2 million chunks can provide sufficient data for finetuning while avoiding overfitting, given the capacity of `Llama-2 7B`. Throughout the finetuning process, we maintain consistent settings using learning rate $5e^{-5}$ for LoRA, using Adam optimizer, dropout rate 0.1, trainable parameter fraction 0.062%. To eliminate the possibility of model collapse due to insufficient sampling and to gain insights into scenarios where more AI-generated data is produced than the model has been trained (or finetuned) on, we consistently utilize a model trained on half the dataset for generating subsequent datasets.

For completeness, we include Figure 33 with loss on the full chunks and Figure 34 that mix the generated data with original data. The mixing curve also aligns well with the grokking phenomenon predicted by theory.

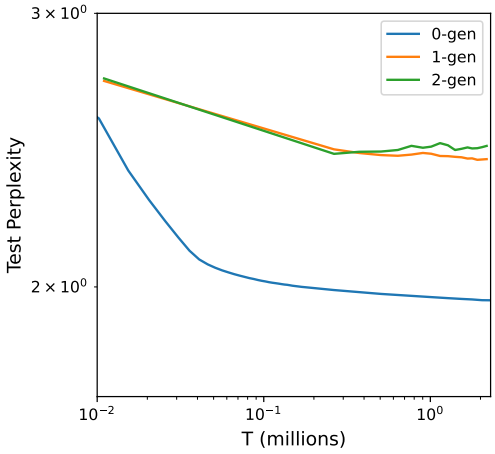

Figure 33: **Llama Generated Data.** Llama2 finetuning when the loss for training and evaluation is the cross-entropy for all tokens in the chunks, including the prompt.

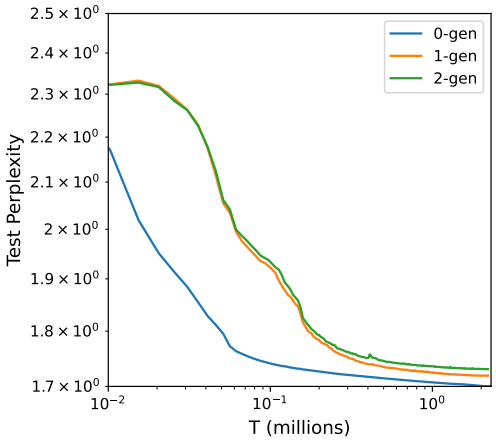

Figure 34: **Mixing Llama Generated Data with Original Data.** Similar setting as Figure 4 left. Starting from gen 1, we mix the generated data with the original one with a ratio of 90 to 10. Top p 0.9 and temperature 0.9.

J More Studies on Tail Cutting and Tail Narrowing Effects

Here, we illustrate how tail cutting in the next-token distribution can lead to tail-narrowing for metrics that take the entire sequence into account, like perplexity. Figure 35 illustrates this for the autoregressive bigram model. This effect is likely due to the combinatorial factors we obtain when considering an additive (or multiplicative) measure like perplexity.

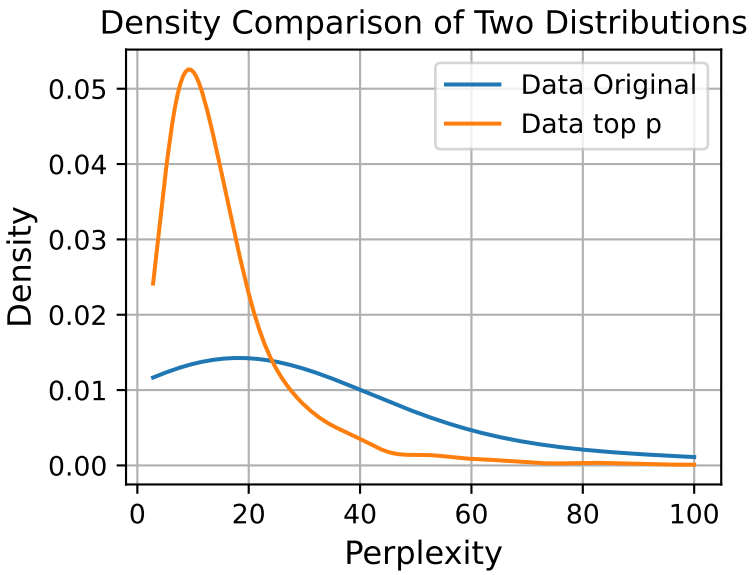

Figure 35: Sequential bigram data: top p = 0.95 leads to similar effect as tail narrowing. 1000 data with sequence length 100.