CAN SYNTHETIC DATA TRAINING UNBLOCK DATA BOTTLENECKS? AN ANALYSIS OF KEY CONSIDERATIONS

Iyngkarran Kumar

January 2024

Executive Summary

I review key considerations to the question of whether synthetic data training can meaningfully unblock the text data bottleneck. I first look at compute-based considerations for the success of synthetic data training (i.e., How much synthetic data needs to be produced to train powerful models? When will this be possible?). I then look at other factors; primarily ones that concern the quality of synthetic data relative to real data (i.e., whether this leads to any fundamental limitations to the usefulness of synthetic data training).

Compute Bottleneck Considerations

Recent work suggests that we may need 10^{18} high-quality text tokens to automate R&D (transformative AI)—though there is significant uncertainty around this estimate. If this is true however, human-produced text stocks would have to grow at their current rate for over 150 years to reach this figure.

I do a back-of-the-envelope calculation (BOTEC) to estimate when frontier AI labs will be able to produce this many tokens, extrapolating lab R&D budgets, the efficiency of computer hardware and the parameter count of models over the coming years, and find that we may only be able to produce $\sim 5 \times 10^{16}$ synthetic data tokens in this period. At face value, this suggests that in the next few years synthetic data will be unable to scale to training transformative AI, even when making the significant assumption that it is of identical quality to human-produced data.

However, interpretation of the result above should take into account the uncertainty around the training requirements for transformative AI, as well as the fact that the modelling doesn't account for the impacts of AI in accelerating AI research and the economy more broadly.

Data Quality Considerations

The quality of model-produced text can differ from human-produced text in clearly noticeable ways, so in assessing the usefulness of synthetic data training one must account for this quality difference. I discuss an intuitive way in which this can be done which has previously been used to measure the contribution of model fine-tuning—namely, through making "effective data" comparisons.

I outline the results of a recent paper, which allows effective data comparisons to be made between real and synthetic images. The paper finds that when averaged over all tasks, synthetic image data training is quantitatively, but not qualitatively different to real data training. However, there are some individual tasks for which synthetic data training cannot substitute for real image training, due to models fundamentally being unable to produce some 'visual concepts'.

I speculate about the implications of text-based models being fundamentally unable to produce 'text concepts'; specifically, what these text concepts are and how this may lead to performance-degrading phenomena such as model collapse. On the former point, I think that abstract NLP capabilities are useful ways to think about 'text concepts', but note that not much work exists in enumerating and benchmarking models on these underlying NLP capabilities.

1 Introduction

This piece is a result of an investigation over a few weeks into the ability for synthetic data. Here, synthetic data refers to data produced directly by ML models, in contrast to that produced by augmenting existing human-produced data. I also primarily focus on text data stocks due to their importance in driving AI progress over the past few years.

In the first section I overview some key concepts which feature in subsequent discussions, such as neural scaling laws and, of course, data bottlenecks themselves—this can be skipped if you're already familiar. I then look at compute-based bottlenecks to producing synthetic datasets at the scale required to train transformative[1] AI models (Section 3), after which I focus on how the relative quality of synthetic datasets will influence its utility relative to real data (Section 4), leaving aside questions of compute feasibility.

It's worth noting that data stocks could increase in many other ways as opposed to just scaling synthetic data. For example, improvements in sample efficiency may mean that we need far less data to train models to a given level of capability, or there may be synergies between training on different modalities that effectively expand text datasets by orders of magnitude[2]. Another pathway may be through changes in data collection practices—for example, if governments made it mandatory to record speech in all public places this would also contribute to a huge increase in recorded speech and thus text data stocks. However, in this piece I focus on synthetic data contributions only. Why? One reason is due to tractability, but I also think that if synthetic data production and training 'works', this could lead to extremely quick gains in capabilities. A limiting case of this would be something like AlphaGo Zero, where in just one day models can generate (and train) on volumes of data that would have taken humans hundreds of years to produce.

As always, comments, feedback, and directions to relevant work are welcome.

2 Background

2.1 What are Neural Scaling Laws?

Neural scaling laws are empirically determined mathematical laws that allow one to predict the loss of a model as a function of the model size (number of parameters, N) and the size of the dataset that it is trained on (D). Here, 'loss' refers to upstream loss—scaling laws have been found for downstream tasks but are usually more involved, depending on model architecture and hyperparameters (in addition to N and D) [Villalobos, 2023]. The functional form of upstream scaling laws is given below.

$$L = \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} + E \tag{1}$$

Following the terminology in Hoffmann et al. [2022], the loss term that is dependent on N can be seen as a 'functional approximation error'—where the model has insufficient capacity to express the function it is trying to learn—whereas loss term dependent on D is a 'convergence error', appearing because model only has access to a finite number of data points to learn a continuous function. E is the loss that an infinitely large model with infinite data points would achieve (i.e., the best possible performance that can be achieved on the task). The A and B correspond to the functional/convergence error a model will achieve in the limit of very low model size and dataset size, while α and β capture the importance of scaling these two terms respectively. Whilst this simple functional form holds over orders of magnitude of parameter sizes, dataset size and compute [Kaplan et al., 2020, Hoffmann et al., 2022], previous work has found that more complex functional forms are needed to model loss over larger domains of N, D and C[3]—however, to keep things simple, I focus on the form given in Equation (1) for this piece.

The implications of neural scaling laws holding over many more orders of magnitude appears are huge. Simply by making models bigger and training them on larger datasets can we achieve models with qualitatively new and useful capabilities, in contrast to the view that the path to powerful AI systems is through a handful of genius algorithmic insights[4]. Thus, the task of building transformative models becomes more one of economic and engineering feasibility.

2.2 The Contributions of Model Size and Data

An implication of the neural scaling law form given in Equation (1) is that, generally, increases to N and D do not contribute equally to improving loss. As pointed out in nostalgebraist [2022], with a model like Gopher [Rae et al.,

[1] Transformative AI is becoming an increasingly loaded term—here I'm envisioning something that could automate all knowledge work, particularly R&D.

[2] This paper gives a theoretical approach to this idea.

[3] See Alabdulmohsin et al. [2022] or Caballero et al. [2023]. The latter essentially argues that A, B, alpha and beta change almost discontinuously at certain N, D and compute thresholds.

[4] That's not to say that these won't contribute!

2022] there are very few gains to be made from increasing model size relative to increasing dataset size. To quote directly:

> "In terms of the impact on LM loss, Gopher's parameter count might as well be infinity. There's a little more to gain on that front, but not much.
>
> Scale the model up to 500B params, or 1T params, or 100T params, or 3 $\uparrow\uparrow\uparrow$ 3 params ... and the most this can ever do for you is an 0.052 reduction in loss"

On the other hand, Gopher's dataset size would need to be increased by a factor of just 2.3 to achieve the same loss reduction of 0.052.

More generally, when given an increase in training compute budget there is a best way to allocate that between increasing parameter count and dataset size[5]. The Hoffmann scaling laws [Hoffmann et al., 2022] that were determined in 2022 placed a greater importance on dataset scaling than the previous Kaplan scaling laws [Kaplan et al., 2020] did, suggesting that datasets and parameter count be scaled equally[6]—but the broader idea is that to obtain the best performance, model size and training dataset size must grow together rather than independently. Which puts significant importance on the following question...

2.3 When Will We Run Out of Data?

Epoch investigated this question in November 2022 [Villalobos et al., 2022b], comparing growth rates in training dataset sizes (for the image and language domains) to growth rates in the respective dataset stocks. They employed two methods to project growth in training dataset size; the first looks at historical growth rates in training dataset size and extrapolates this over the next few decades. The second method assumes that we'll be doing compute optimal scaling, and thus uses future compute extrapolations to determine the dataset size[7] that will be used in these training runs.

What do their results say? Their compute projections predict that data stocks of high-quality text, low-quality text and image data will be exhausted by median dates of 2024, 2040 and 2038 respectively, though with a number of caveats. They make clear that their investigation assumes no further improvements in data efficiency; it also doesn't model contributions from synthetic data stocks and improved transfer learning when training on multimodal datasets. Other things assumed are that the Hoffmann scaling laws will remain correct at larger model scales, and that there are no changes in data collection practices that would significantly grow data stocks.

The rest of this post aims to evaluate the synthetic data consideration for reasons given above, with some brief discussion also on sample efficiency.

3 Compute-Bottlenecks to Scaling Synthetic Data

Let's start by making the simplifying assumption that synthetic (text) data is of the same quality as real data; that is, we can't distinguish between a model-produced text corpus and a human-produced one. This is already a large assumption—in many domains, such as extended reasoning problems, current models differ noticeably from human-generated responses[8] and this is likely to remain the case in some domains over the next few years. Nevertheless, let's grant the assumption and see what other barriers remain for synthetic data to solve data bottlenecks.

The main issue is the amount of synthetic data that may be required to train transformative models. A recent analysis by Epoch provides a tentative median estimate of 10^{35} FLOPs to train a model that can emulate scientific reasonin [Barnett and Besiroglu, 2023][9] There are considerations that push this compute estimate higher or lower but let's use

[5] Training compute = $6 \times N \times D$

[6] It's worth being aware that the optimal scaling strategy may change as larger N and D domains are explored—in fact, this seems likely to be the case if the idea in Caballero et al. [2023] is correct. If once past a certain (N,D) threshold there is a phase change at which the optimal scaling method becomes lopsidedly focused on parameter scaling, then data bottlenecks become much less of an issue. But the reverse scenario could also happen.

[7] For example, if compute budgets in 5 years are 100× what they are today, then datasets will be $\sqrt{100} = 10\times$ the size of those at present.

[8] See the second half of this post for a bunch of examples in which ChatGPT struggles to solve fairly basic reasoning problems. See also this.

[9] In essence, the analysis maps model loss (which can be predicted from training compute via scaling laws) onto a quantity known as the 'k-performance'—the number of tokens that a model can produce before the output becomes noticeably different from human-generated text. If a model can produce text that is roughly the length of a scientific manuscript and indistinguishable from human-produced output, then this may be a decent proxy for a model that can automate significant parts of the research science work. This is where 10^{35} FLOPs comes from.

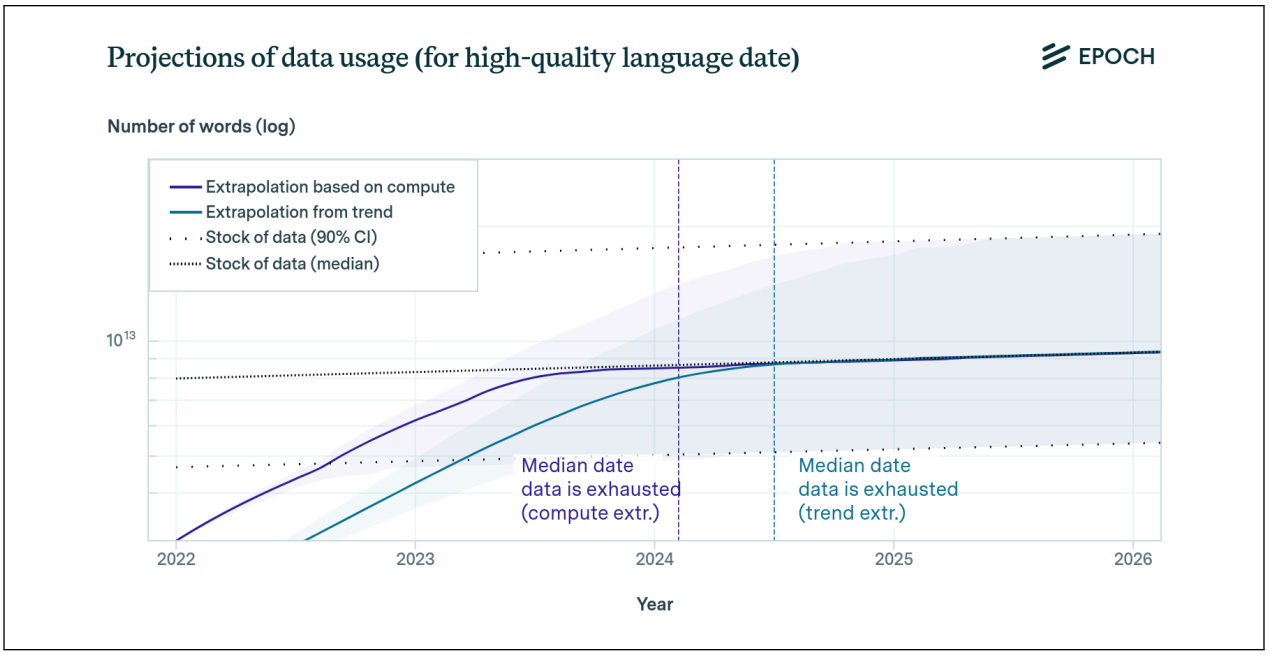

Figure 1: Extrapolating the future of high-quality text data training datasets. Their results suggest a doubling time of roughly 10 years for language text stocks, whereas training datasets have been doubling approximately once per year. Source: [Villalobos et al., 2022b]

10^{35} FLOPs for now. That means we'd need to scale datasets by a further five orders of magnitude[10], and assuming that it's high-quality text data that will be needed means that a dataset size of 10^{18} tokens will be required. This means we'd need human-produced data stocks to continue growing at their current rate for roughly 165 years[11]. The underlying point here is that scaling synthetic datasets to the point of being able to train powerful AI does not appear easy, even making the simplifying assumption that the synthetic dataset is the same quality as existing human-generated ones.

As a result, in the following section, I've put together a simple model that estimates how many synthetic data tokens we'll be able to generate at various points during the next 30 years. I'm hesitant to make predictions past 30 years because that seems to be the characteristic time for AI paradigms to rise and fall[12].

3.1 A Simple Model for Producing Tokens

Let's assume the following:

In a given year t, a lab has a training budget \$B, some of which is used to produce synthetic data tokens (and the rest of which is then used to train a model on these tokens). In that year, the quality of hardware gives c FLOPs/\$, and we're using a model of size N to produce these tokens. How many tokens N_t can a lab generate?

The importance of these factors is hopefully clear—larger budgets (B) means that more tokens can be generated, and advances in hardware efficiency (c) have similar implications. However, increasing the size of the model that generates the tokens means that token generation gets more expensive. The total number of FLOPs that a lab will have to spend

[10] $\sim 10^{25}$ FLOPs were needed for GPT-4, which gives 10 orders of magnitude (OOMs) to make up. Splitting that equally between parameter and dataset scaling as explained in the previous section gives the 5 OOMs of dataset scaling.

[11] I'm using the median estimated growth rate of 7.15% from page 5 of Villalobos et al. [2022b]. Even using the upper estimate of a $\sim$17.5% growth rate gives us 70 years until we hit 10^{18} tokens.

[12] For example, there was 25–30 years between the Dartmouth workshop and proliferation of expert systems [Wikipedia, 2024], and $\sim$25 years between the rise of expert systems and the deep learning revolution circa 2012.

is $B \cdot c$, but only 25% of that will be spent on generating the dataset[13]—meaning the FLOP budget for generating synthetic tokens is $\frac{1}{4} \cdot B \cdot c$. Divide this through by $2N$[14], which is the FLOP cost for generating a token and we get:

$$N_t = \frac{B \cdot c}{8N} \tag{2}$$

We're interested in how many tokens can be generated in a given year t, and recall that 10^{18} tokens may be the number required to train an AI that can significantly accelerate scientific progress. Before we dive into what this model predicts, here are its limitations:

- I don't factor in increases in data efficiency, primarily because I couldn't find good sources on this. Factoring that in would lead to the ability to generate 10^{18} tokens arriving quicker than expected. However, I think it's pretty easy to integrate this into the existing model—if we expect gains in sample efficiency to reduce data requirements by two orders of magnitude (OOMs), then we can just look at the predictions for when cumulatively 10^{16} tokens will be generated.
- Remember that for now we're ignoring the relative quality of synthetic data to real data, and factoring this in would push the date to the 10^{18} threshold back. I discuss how token quality can be measured quantitatively in a later section.

See this spreadsheet for the full details. The parameter values can be varied (see the Parameters sheet), allowing you to explore how different inputs change token production abilities.

3.2 Results and Comments

Main scenario: Using the assumptions discussed directly below, the model predicts that in approximately 30 years we'll only be able to generate 5×10^{16} tokens—still 20 times too small for the 10^{18} threshold. Taking this result at face value says that current compute constraints will limit the ability to train transformative models[15] on synthetic data for at least three more decades due to data constraints. Relative to recent levels of progress this reads as surprising to me—looking at the growth in capabilities of the GPT-N series and related models, 30 years feels like a long time.

One thing that helps resolve this discrepancy is seeing that the compute overhang may be soon exhausted, which would move AI progress into a slower growth mode. As I discuss below, training budgets have been growing at roughly 3.1× per year[16] but it seems unlikely that this can continue much longer; indeed, in the model used to get these results, budget growth switches to a slower growth mode of 1.2×[17] around 2028. It's worth noting the magnitude of the difference between 1.2× and 3.1× annual growth—the former takes 25 years to grow by a factor of 100, whereas the latter takes just 4 years. If we extend the 3.1× budget growth rate by five more years (at which point training budgets would be $2.5 trillion), then the ability to produce 5×10^{16} tokens arrives a whole 10 years earlier. The main takeaway is that we should be watching closely how much longer growth of this magnitude can be maintained.

A final brief note—this prediction also doesn't square with very short timelines (on the order of 5 years) as discussed here, which seems to be down to orders of magnitude lower compute requirements for transformative AI, and significant acceleration of both software progress (sample efficiency) and hardware progress (FLOPs/$) upon transformative AI arriving.

Ok, now more on how budgets, inference efficiency and parameter counts are being modelled. I have highlighted the parameters that can be changed in the spreadsheet.

3.3 Modelling Assumptions

3.3.1 Budgets

To model training budgets I'm essentially assuming that training budgets will continue to grow at their current rate of roughly 3.1× per year [Cottier, 2023] until hitting a ceiling set by a lab's annual R&D expenditure, at which point

[13] That's because we'd need $\sim 2ND$ FLOPs to generate a dataset of size D with a model of size N, and then approximately 6ND FLOPs to *actually train* a model on that generated dataset, meaning that we'd need to split a FLOP budget in the ratio of 1:3. See Sevilla et al. [2022] for more on where the multiplicative factors come from. I am implicitly assuming that the model used to *generate* the tokens and the model being *trained* on these tokens will be the same size, which technically won't be the case. But I think they'll be sufficiently close in size to justify the simplification.

[14] See footnote above

[15] By transformative models I have in mind something like the systems that automate scientific R&D as described in Karnofsky [2021]

[16] That's a doubling time of 7 months!

[17] See the section directly below for a justification of the 1.2× figure

budgets will slow to growing at the R&D expenditure growth rate—which sits at roughly 1.2× per year[18]. The training budget for GPT-4 was roughly $\$10^8$ whereas the annual R&D expenditure of Google and Microsoft is two orders of magnitude larger at $\sim\$10^{10}$—so I'm assuming that we see 3.1× growth rates until this $\$10^{10}$ threshold is hit, after which we see 1.2× growth rates. The model estimates that this happens around 2028.

The parameters that can be changed in the spreadsheet are:

- Budget threshold (the budget after which the slower growth mode starts)—Currently $\$10^{10}$
- Pre-threshold growth rate—Currently 3.1×
- Post-threshold growth rate—Currently 1.2×

3.3.2 Inference Efficiency

I opted for a doubling rate of 2.1 years for FLOP/$ (annual growth rate of 1.45×), inline with the results found here. Some sources suggest that progress is far more rapid, arguing that inference costs had dropped by 10× in 16 months[Patel, 2023]—but I think there are pretty strong reasons to suggest that this was a one-off acceleration rather than a permanent trend[19] and we'll see progress revert to something more like Moore's Law. Unlike the budget modelling above, I didn't include any step-changes in the growth rate of inference efficiency. This parameter can also be edited in the spreadsheet.

3.3.3 Model Size

Model size has been growing at approximately 2.8× per year[Villalobos et al., 2022a]—but two things could change that in the near future:

The Data Bottleneck: Scaling laws tell us that the growth rate of training dataset size determines the growth rate of model size (and vice versa). It seems that the 2.8× growth rate of model size in previous years has been facilitated by language training dataset sizes growing at similar rates [Epoch AI, 2023]. However, if synthetic data becomes the new way of generating training datasets, then dataset growth becomes compute-bound[20]. The maximum growth rate of synthetic training dataset stocks will be equal to the growth rate of compute-budgets, but Hoffmann scaling laws say that this compute-budget growth should be split equally between increasing the model size and dataset size. Long story short, if synthetic data becomes the main source of data for training (which is assumed here) we should naively expect the growth rate of model size growth to converge to the square root of the compute growth rate. This in turn can be determined from the budget size and inference efficiency growth rates discussed above.

Sparse Mixture of Experts (SMoEs): When producing synthetic data, what matters is the growth rate of inference costs. But instead, we've just discussed the growth rate of the total number of parameters in a network being trained. For dense models, these two quantities are pretty much equal.

This isn't the case for the SMoE architecture[Adams, 2024] however, where an increase in total parameter count can be divided in two ways: it can be used to increase the size of expert networks, or it can be used to increase the number of expert networks. In the limit, SMoE architectures would mean that any problem can be solved purely by increasing the number of experts, holding the expert network size (and thus inference costs) fixed[21].

Thus, if SMoE continues to remain useful, the implications of this on modeling the parameter count (used for inference) are as follows:

- GPT-4 seems to be SMoE, with an estimated 16 expert models, each with 110×10^9 parameters.[22]
- I haven't been able to find figures on the growth rate of expert network size compared to growth rate of the number of expert networks, so for now I'll assume that when increasing the total number of network parameters, we allocate equally between the two. That is, the ratio of the growth in expert network size to growth in expert number is 1:1.

The parameters that can be changed in the spreadsheet are:

- Initial parameter count—Currently 280×10^9

[18] Estimated from Google's data here

[19] Specifically, the 10× increase may have come from the specialisation of chips for LLM inference, and there may not be much room left at the bottom for further gains.

[20] Rather than bound by the rate at which humans produce data

[21] Note: I don't expect this to be the case, but it's a useful intuition builder.

[22] See Adams [2024]. Two experts are used for inference, with 55×10^9 shared parameters, so I set the initial parameter count used for inference to $\sim 280 \times 10^9$.

- Ratio of expert size to expert number growth—Currently 1:1

4 Incorporating Differences in Quality Between Synthetic and Real Datasets

In the previous section we made the huge simplifying assumption that synthetic data text will be of similar quality to human-generated language datasets, and saw that even in this case there are non-trivial compute-based hurdles to overcome. Now that assumption is relaxed, to see how we can think about the difference in quality between two datasets and implications of this on the ability for synthetic data to unblock data bottlenecks.

4.1 Effective Dataset Contributions

An intuitive method to compare the quality between two datasets (let's call them dataset A and dataset B, with D_A and D_B tokens respectively) is by assuming that dataset B contributes some 'effective' number of dataset A tokens. We can work out the number of tokens that dataset B contributes by just finding the number of tokens that are required to match the loss achieved when training on dataset A. The figure below captures the main idea.

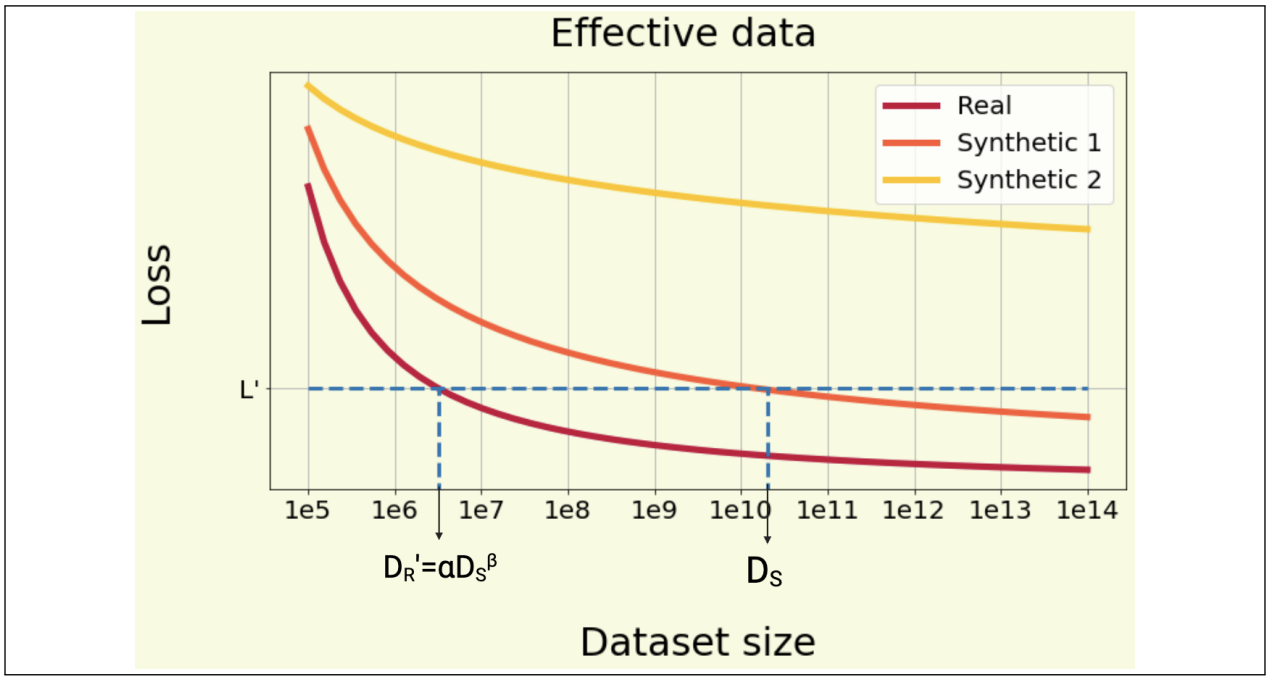

Figure 2: Effective data. Note that the dataset sizes on x-axis are for illustration only.

Here, the 'Real' line is the scaling law for real data, and two scaling laws are shown for synthetic datasets. We can clearly see that training on D_S tokens from synthetic dataset 1 is equivalent in loss terms to training on D'_R tokens from the real dataset. The synthetic 2 line shows that for some datasets, there is no realistically achievable number of tokens that will match the real dataset in performance.

In order to determine how many real dataset tokens a synthetic dataset contributes, assume that training on either leads to a power law in model loss. Then we get[23]:

$$D_{\text{effective}} = \alpha \cdot D_{\text{synthetic}}^{\beta} \tag{3}$$

Making some reasonable assumptions about the parameters of each scaling law gives β as less than one[24], meaning that there are diminishing marginal returns to effective dataset size as synthetic data is scaled—the magnitude of β determines how quickly these diminishing returns set in. α, on the other hand, determines the usefulness of synthetic data in the low data limit—i.e., when there isn't much of it.

[23] If we assume that dataset A gives us a power law of $\log(L) = k_A \cdot \log(D_A) + c_A$ and dataset B gives us $\log(L) = k_B \cdot \log(D_B) + c_B$, setting these two equal and rearranging for D_A in terms of D_B gives the effective data.

[24] Assuming that the gradient $k_B < k_A$ (i.e., Dataset B is generally a lower quality dataset) gives beta less than one.

4.2 Scaling Laws for Synthetic Images

The key requirement for making effective data comparisons is to have scaling laws for both real and synthetic data training, and a recent work does this for the image domain,[Fan et al., 2023] so here I briefly discuss their results. To obtain the scaling laws, they train a classifier on the real ImageNet dataset as well as multiple synthetically generated ImageNet datasets, determine classification loss, and compare the gradient of a $\log(L)$–$\log(D)$ plot. The synthetically generated ImageNets were created by prompting popular text-to-image models (such as Stable Diffusion and Imagen). I focus on two of the main results that are relevant to this post, but there's more to the paper than just what is discussed below.

4.2.1 Key Result 1

With some caveats, Figure 3 showed that synthetic data training performed quantitatively worse[25] but not significantly qualitatively different from real data. That is, for most of the loss values that real data training achieves, there appears to be a realistically attainable synthetic dataset scale that could achieve the given loss. For example, extending the best-performing synthetic dataset (orange line) to the 64M image scale would have covered most of the loss range of the red line, but the authors didn't do this due to insufficient model capacity, rather than due to fundamental issues with the synthetic data represented by the orange line.

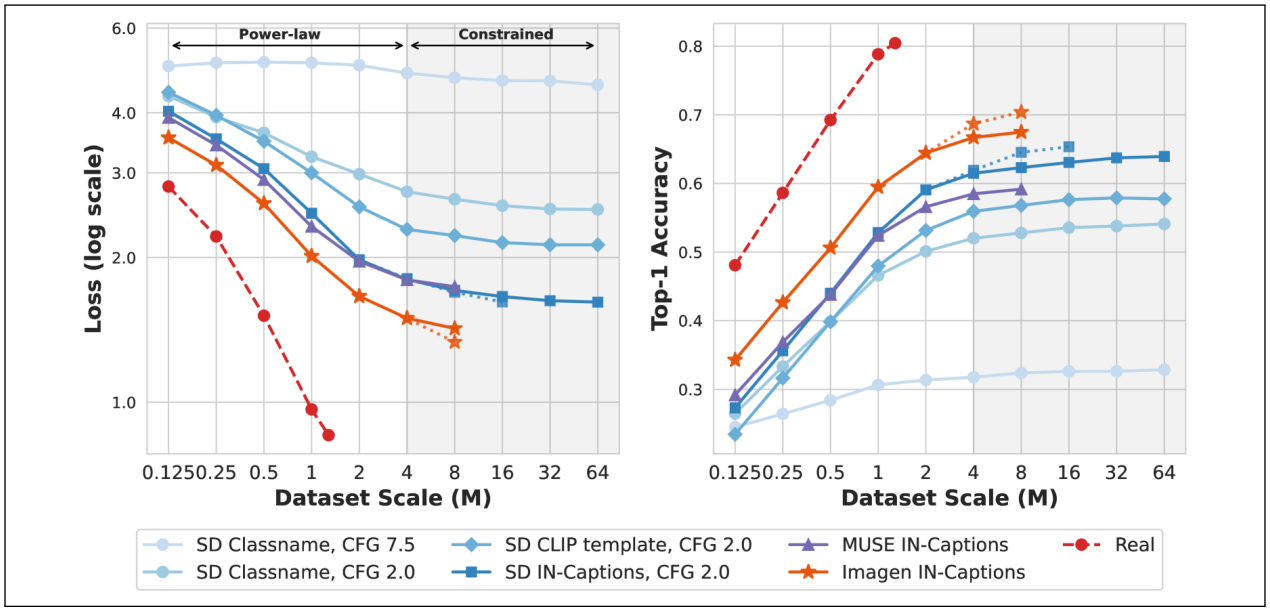

Figure 3: Scaling laws for synthetic image training for supervised learning, taken directly from Figure 3 of Azizi et al. [2023]. SD, MUSE and Imagen are different models used to produce the datasets, 'Classname' and 'IN-captions' are different prompting methods, and the guidance scale (CFG) is also indicated. The constrained region indicates where further loss gains are limited by insufficient model size.

The caveats:

- Significant tuning of inference parameters such as text prompts and the model's guidance scale[26] are needed to obtain the optimal scaling behaviour. Notice the difference in scaling behaviour between Stable Diffusion (SD) generated images with a guidance scale of 7.5 and using the Classname prompting method, compared to using a guidance scale of 2.0 and using IN-Captions prompting method. The first of these is like the Synthetic 2 dataset above—it seems that no amount of scaling will be able to match the real Imagenet dataset.
- These results were determined at a relatively small dataset scale—they generated synthetic data scaling at a scale up to 64 million images. For reference, Stable Diffusion was trained on approximately 300× as many images [80.lv, 2023].

[25] And this was just for evaluating on the classic ImageNet dataset. There was very little difference between synthetic and real training on out-of-distribution validation sets such as ImageNet-R and ImageNet-Sketch and in some of these cases synthetic data training *outperformed* real data training.

[26] See here for a guidance scale explainer

- The loss curve of the figure above was obtained when averaging loss over all classes. See the next subsection for further details on the paper's per-class analysis.

4.2.2 Key Result 2

The paper also found noticeable differences in scaling behaviour across classes. See Figure 4 for an example, in which synthetic image training scales similarly to real image training for classifying images of Ox, but displays no useful scaling behaviour for classifying tigers instead. Interestingly, when looking at the synthetic tiger images, they don't appear significantly lower quality than those for the Ox class (see Figure 8 of the paper).

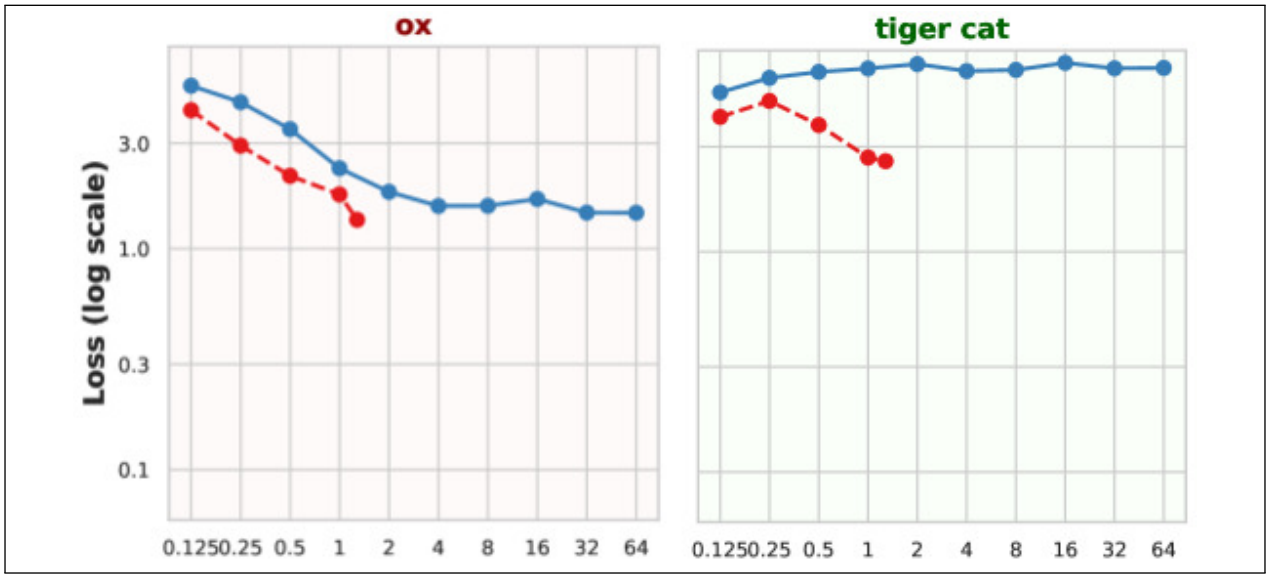

Figure 4: Synthetic scaling laws for two ImageNet classes, adapted from Figure 7 of Azizi et al. [2023]. The blue lines represent the synthetic datasets whilst the red is real ImageNet training.

Note that scaling up the number of tiger images to any scale wouldn't meaningfully improve performance, suggesting that these models are fundamentally unable to represent some visual concepts that are required to identify a tiger.

4.3 Speculating About Broader Implications of These Results

4.3.1 Text Concepts and Model Collapse

What are the broader implications of these results, especially in the important text domain? I find the second result particularly interesting—perhaps text models may fundamentally be unable to reproduce some 'text concepts' that are present in high-quality, human-generated NLP datasets. Section 3 argued that in the next decade or two we could see text-producing models effectively create an internet-sized body of text each year, but if these corpora are noticeably lower quality to human-produced text then this would limit their use in training further models.

What would these text concepts be? Abstract NLP capabilities that underlie the completion of a wide range of tasks are likely candidates, such as those found in a recent paper [Michaud et al., 2023]. Examples from that paper of abstract NLP capabilities (or 'quanta') are:

- The ability to increment numerical sequences (Figure 1)
- The ability to predict the correct noun (Figure 14)
- The ability identify that 'time' comes after 'Once upon a' (Figure 14)

Figures 1, 13 and 14 from the referenced paper give further examples.

These specific NLP capabilities seem to be pretty easy—most current models can them all very well. But to hypothesise some tasks that even future models may struggle to do:

- Predict the correct noun, given that the noun has been mentioned less than three times in the past 100,000 tokens (more generally, reasoning over long context-windows)
- Propose explanatory theories that exceed some threshold Kolmogorov complexity.

To my knowledge, there aren't any great enumerations of abstract NLP capabilities that are needed to complete a wide range of NLP tasks[27]—most current benchmarks instead focus on tracking model performance on clearly identifiable domains of human knowledge. However, I would be interested in seeing more work that identifies underlying NLP capabilities such as those above, as it is along these dimensions that I expect model-produced text to differ from human generated text. It's also worth noting that fine-tuning[28] may help overcome the inability for models to produce given text-concepts as shown below; concretely, if it's found that the formal analytical rigour present in many areas of pure math is consistently absent from model-produced text, some fine-tuning on papers from a high-quality mathematics journal could do the job.

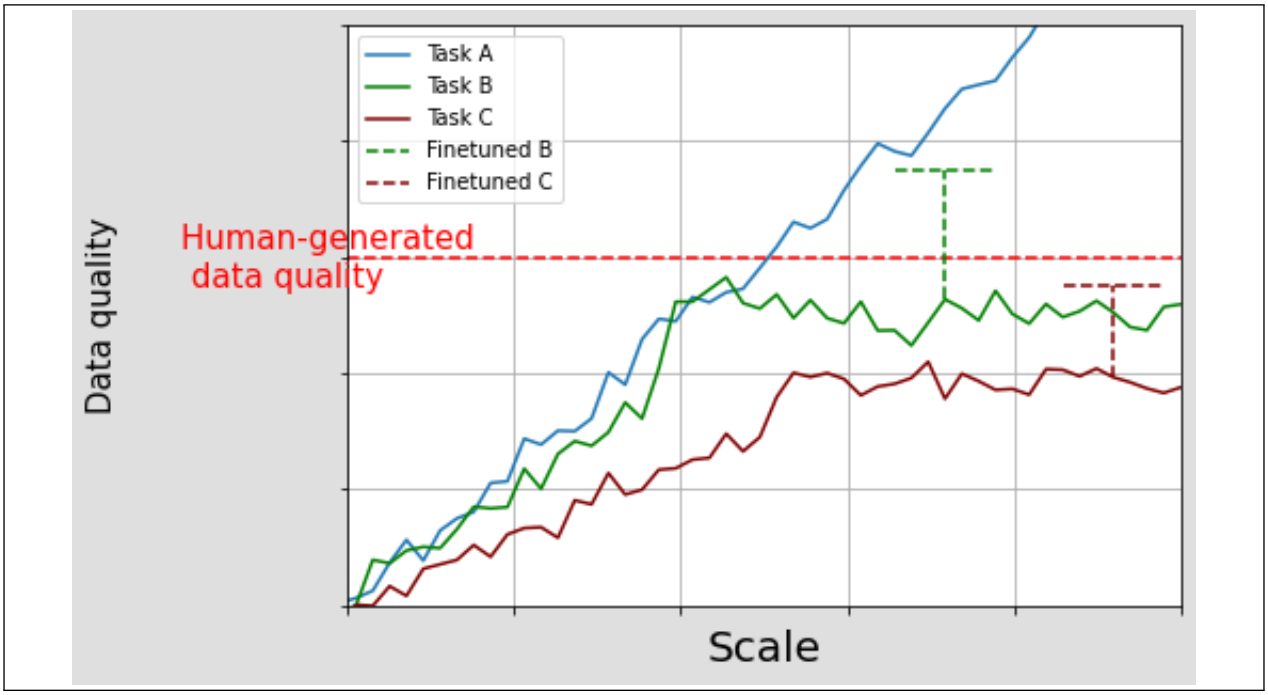

Figure 5: Will model-produced text be able to reproduce all 'text concepts' present in human-produced text? It's tasks of type C that would limit the usefulness of synthetic data. For concreteness, imagine that task A was 'number of tokens that a model is able to reliably reason over', B may be 'complexity of theories that the model can produce, (as measured by some complexity measure)' and C may be 'Number of raw facts that a model can recall in the subject area of biology'. In this case, the model-produced text, even after fine-tuning, would be unable to substitute completely for human produced text in the area of biology.

What happens if we train on text that is missing some key text concepts? At a high-level, we'll get a training curve akin to the Synthetic 2 dataset in the figure above. In more detail, we'll see a phenomenon like model collapse [Shumailov et al., 2023]—in which successive training on a text dataset that is an approximation of the original text distribution leads to significantly degraded performance. The paper above shows that after a few iterations of synthetic data training, the text output of models reduces to an incoherent ramble (section 2.1).

They do also show that model collapse can be mitigated by incorporating some of the real (human generated) data into training, and so the ratio of original to synthetic data needed to avoid model collapse is another factor influencing the usefulness of synthetic data training. A 1 to 1 ratio will mean real data remains a strong data bottleneck, whereas a 1 to 1000 split will be far less of an issue.

4.3.2 Self-Play

A final consideration for the usefulness of synthetic data is related to how easily we can get models to self-prompt absent human intervention. A limiting case of this is something like AlphaGo Zero, which simulated over 500 years of playing Go in just 3 days of real time.[29] For the language model case, we could imagine prompting 100 models

[27] E.g., MMLU, which "covers 57 subjects across STEM, the humanities, the social sciences, and more."

[28] And other post pre-training enhancements

[29] AlphaGo Zero training details say that the model played 4.9 million games in 72 hours [He, 2017], taking 1 Go game as lasting 1 hour, we get ∼500 years of play.

with ‘you are Isaac Newton, interact as you think he would’, ‘you are Aristotle, interact as you think he would’ etc., letting these models interact for a week, and then coming back to a thousand internets worth of insightful scientific, philosophical and political analysis. If instead AlphaGo Zero required a human course correction after making a few hundred Go moves, this would have significantly bottlenecked the volume of synthetic data that was produced. We should keep our eyes on future generations of LLM multi-agent interactions such as ChatDev to see how well models can produce high-quality output without human prompting.

References

80.lv. Exploring the images used to train stable diffusion's ai, 2023. URL https://80.lv/articles/exploring-the-images-used-to-train-stable-diffusion-s-ai/. 80 Level.

Richard Adams. The next era of ai: Inside the breakthrough gpt-4 model, 2024. URL https://hackernoon.com/the-next-era-of-ai-inside-the-breakthrough-gpt-4-model. HackerNoon.

Ibrahim Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision, 2022. URL https://arxiv.org/abs/2209.06640.

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Scaling up visual and vision-language representation learning with noisy text supervision. *arXiv preprint arXiv:2312.04567*, 2023. URL https://arxiv.org/abs/2312.04567. Note: This is a placeholder for the synthetic image scaling paper referenced in the text.

Matthew Barnett and Tamay Besiroglu. The direct approach, 2023. URL https://epoch.ai/blog/the-direct-approach. Accessed: 2025-12-16.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws, 2023. URL https://arxiv.org/abs/2210.14891.

Ben Cottier. Trends in the dollar training cost of machine learning systems, 2023. URL https://epoch.ai/blog/trends-in-the-dollar-training-cost-of-machine-learning-systems. Accessed: 2025-12-16.

Epoch AI. Key trends and figures in machine learning, 2023. URL https://epoch.ai/trends. Accessed: 2025-12-16.

Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training ... for now, 2023. URL https://arxiv.org/abs/2312.04567.

Yuze He. Alphago zero cost analysis, 2017. URL https://www.yuzeh.com/data/agz-cost.html. Personal Website.

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

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. URL https://arxiv.org/abs/2001.08361.

Holden Karnofsky. Transformative ai timelines part 1 of 4: What kind of ai?, 2021. URL https://www.cold-takes.com/transformative-ai-timelines-part-1-of-4-what-kind-of-ai/. Cold Takes Blog.

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. URL https://openreview.net/forum?id=3tbTw2ga8K.

nostalgebraist. Chinchilla's wild implications. https://www.lesswrong.com/posts/6Fpvch8RR29qLEWNH/chinchilla-s-wild-implications, 2022. LessWrong.

Rohan Patel. Why databricks bought mosaic, 2023. URL https://unsupervisedlearning.substack.com/p/why-databricks-bought-mosaic-and. Unsupervised Learning Substack.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis insights from training gopher, 2022. URL https://arxiv.org/abs/2112.11446.

Jaime Sevilla, Lennart Heim, Marius Hobbhahn, Tamay Besiroglu, Anson Ho, and Pablo Villalobos. Estimating training compute of deep learning models, 2022. URL https://epoch.ai/blog/estimating-training-compute. Accessed: 2025-12-16.

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. URL https://arxiv.org/abs/2305.17493.

Pablo Villalobos. Scaling laws literature review, 2023. URL https://epoch.ai/blog/scaling-laws-literature-review. Accessed: 2025-12-16.

Pablo Villalobos, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, Anson Ho, and Marius Hobbhahn. Machine learning model sizes and the parameter gap, 2022a.

Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. https://epochai.org/blog/will-we-run-out-of-ml-data-evidence-from-projecting-dataset, 2022b. Epoch AI.

Wikipedia. History of artificial intelligence, 2024. URL https://en.wikipedia.org/wiki/History_of_artificial_intelligence. Wikipedia.