# What if synthetic data is all you need

**Igor Molybog** *
University of Hawai'i at Manoa
igormolybog@gmail.com

## Abstract

This paper presents a theoretical prediction of the optimal scaling regime for large language models that utilize synthetically generated human language data during pre-training. Our findings suggest that the most effective strategy is to release models successively, using the previous model to generate the next one while increasing the amount of training compute by a factor of about 5 for each new model. The training on synthetic data roughly doubles the computational cost compared to training on natural data. If the exponential growth of AI compute continues, our results indicate that the industry should be prepared to swiftly adopt new models on an approximately yearly basis.

## 1 Introduction

The progress in the development of Artificial Intelligence (AI) systems hinges on the availability of large, diverse, and high-quality datasets used for their training. Increasing the data intensity of machine-learning training leads to AI systems with a more compute-efficient life cycle [25]. Acquiring these datasets can be a significant challenge posed by the cost of data collection [16] due to data scarcity [3] and privacy concerns [1]. The work focuses on using a large language model (LLM) to generate synthetic text that is further applied for self-supervised training of a language model of a higher quality. We are interested in this framework as a potential solution to the data bottleneck problem recently identified for textual and other data modalities [32].

We are prioritizing natural language data modality in this paper for several reasons. First, the training runs of large language models are already consuming an amount of data comparable to the total amount of digitalized text. The year 2050 could see the utilization of training datasets of a size comparable to the total corpus of human-generated public text data if scaling continues at the same pace as the last decade, as predicted by Villalobos et al. [32]. Second, human language is one of the data modalities that cannot be scaled by usual capital-intensive economic means, such as producing and deploying an increasingly larger number of sensors or spending vast computational resources on simulation. Until recently, manual labor was the only way to generate high-quality natural language data. Third, the utilization of AI-generated human language data in the design of the next-generation AI systems is a possible solution to the challenge of super-human alignment [7]. Finally, the scaling properties of language models are well-studied relative to generative models in other modalities, and there is a tested predictive theory about their properties [21, 19, 4].

Synthetic human language, math, and coding data have already helped improve LLMs' reasoning, planning, multi-language, and multi-modal abilities and increase their alignment to human values [22]. Transferring this success to other aspects of LLM, specifically to the self-supervised pre-training stage of a foundational model, is currently facing challenges as models trained on synthetic data may fail to generalize to real-world scenarios [28, 17] due to the issues with factuality and fidelity [35, 18]. Straightforward utilization of synthesized data in pre-training faces the issue of model collapse [26]. Researchers have linked the absence of rare samples and corner cases from the synthetic distribution

*The author thanks Epoch AI for their financial support of this project, and Pablo Villalobos for his helpful review of this article.

| | $\bar{L}$ | $A$ | $B$ | $\alpha$ | $\beta$ |
|---|---|---|---|---|---|
| [4] | 1.82 | 514.0 | 2115.2 | 0.35 | 0.37 |
| [19] | 1.69 | 406.4 | 410.7 | 0.34 | 0.28 |

Table 1: Estimation of scaling law parameters

to the underlying mechanism of model collapse and overall quality degradation of synthetic data [11]. They suggest that mixing natural data into the synthetic training set could offset this issue [15].

Prior research analyzed the techniques to increase the quality of machine-generated data by increasing the compute intensity of its inference [31]. The techniques relevant to LLM include Chain-of-Thought and related techniques [34, 33, 36, 5], language model cascade schemes [10] such as scratchpading [23], learned verifiers [8], selection-inference [9] and bootstrapping [37]. Assuming that a language model can generate higher-quality data than its original training set, we analytically derive the compute-efficient regime of LLM scaling after reaching the data bottleneck. In the absence of the universal notion of data quality, we focus on the inverse average perplexity metric of the generative model as measured on the natural data distribution as a proxy for the quality of the synthetic data. We consider the scenario of training a chain of generative language models of increasing quality, where every next model is trained from scratch on a mix of the natural data and synthetic data generated by the predecessor model. The main conclusions of our analysis are the following:

- If the target model is of relatively high quality, then training a chain of intermediate synthesizers is the rational strategy for synthetic data generation, as opposed to training the target model right away.
- It is optimal to let every consecutive model in the chain utilize approximately five times more training compute than the previous one.
- The multistage generation of synthetic datasets doubles the computational resources needed for training compared to utilizing natural data.

Our results allow us to develop a rule-of-thumb for making architectural decisions for synthetic data generation procedures. We conduct a sensitivity analysis of our results and show that they are robust to the choice of assumptions.

## 2 Background

Our work relies on the dense transformer scaling laws [21, 19], which are empirical formulas that tie the value of cross-entropy loss of the model to the number of model parameters and the number of training data points. The laws take the form

$$L(M, D) = \bar{L} + \frac{A}{M^{\alpha}} + \frac{B}{D^{\beta}} \tag{1}$$

where $L$ is the value of cross-entropy loss of the model with $M$ non-embedding parameters trained on $D$ natural language tokens. The exact values of the parameters $\bar{L}, A, B, \alpha, \beta$ in the scaling law depend on the dataset and the model architecture scaling scheme. However, the form of the law is very robust for a wide range of natural language datasets and for the standard strategies of scaling transformer architecture [30]. For the widely used dense transformer architecture [29] scaled according to the usual recipe [21], these parameters are currently estimated with values presented in Table 1 and were shown to vary depending on the data quality [6]. We primarily use the latest estimation for the generic dataset provided by [4] and conduct sensitivity analysis (Section 5.3) using the other sets of values, showing the robustness of the key results.

The amount of training data $D$ and the size of the model $M$ can be related [21] to the total amount of non-embedding training compute $T$ as

$$T \approx 6MD$$

For a fixed amount of training compute $T$, Hoffmann et al. [19] calculated the number of training data tokens $D(T)$ and the size of the model $M(T)$ that would result in a model of the highest quality. Introducing $\kappa = \frac{\alpha A}{\beta B}$, these can be written explicitly

$$D(T) = T^{\frac{\alpha}{\alpha+\beta}}\delta, \text{ where } \delta = 6^{-\frac{\alpha}{\alpha+\beta}}\kappa^{\frac{-1}{\alpha+\beta}}$$
$$M(T) = T^{\frac{\beta}{\alpha+\beta}}\mu/2, \text{ where } \mu = 2 \cdot 6^{-\frac{\beta}{\alpha+\beta}}\kappa^{\frac{1}{\alpha+\beta}}$$

We refer to this as the training compute optimal (TCO) regime. Substituting the scaling law parameters from [4] into these expressions closely recovers the rule of thumb established by [19]

$$\frac{D(T)}{M(T)} \approx 20$$

The total amount of inference compute $I$ for a model with a relatively short context window can be estimated [21] as

$$I \approx 2MN,$$

where $N$ is the total number of tokens the model generates at the inference phase of its life cycle. The problem of optimizing the total amount of compute over the entire life cycle has been studied by [25] and considered in Section C. The main conclusion is that the models should be "overtrained" relative to the TCO regime ($D > D(T)$ while $M < M(T)$) to achieve optimal quality subject to a fixed combined budget of train and inference compute. The demand of the model dictates the total number of tokens $N$ generated by the model, which determines the degree of overtraining. The total demand from human users over the life span of a model is difficult to predict as it depends on multiple factors besides the quality of the foundational model, such as the quality of alignment, the time of its introduction to the market, the pricing and marketing strategy, the time to deprecation, inference throughput and latency, etc. The demand for synthetic data generation is more comprehensible and has the potential to surpass the demand from human users after the data bottleneck is reached. Thus, in this work, we focus on synthetic data generation as the primary driver for the demand and interpret $N$ as the total number of synthetic tokens generated.

## 3 Assumptions

Following the findings of [27, 20], we measure the quality of a large language model as the inverse of the reducible cross-entropy loss expressed as $Q = (L - \bar{L})^{-1}$. Given $T$ FLOP of the training compute, the best model quality that can be achieved in the TCO regime according to the scaling law [19] is

$$Q(T) = T^{\frac{\alpha\beta}{\alpha+\beta}}/\lambda, \text{ where } \lambda = 6^{\frac{\alpha\beta}{\alpha+\beta}}\left[A\kappa^{\frac{-\alpha}{\alpha+\beta}} + B\kappa^{\frac{\beta}{\alpha+\beta}}\right]$$

The inverse function

$$T(Q) = (\lambda Q)^{\frac{\alpha+\beta}{\alpha\beta}}$$

determines the minimum amount of training compute that needs to be spent to achieve the model of quality $Q$. Thus, the quality of a model can be equivalently represented through the inverse of the reducible cross-entropy loss or with the amount of training compute that would be spent in a TCO regime to achieve the same cross-entropy loss value.

We measure the quality of a synthetic dataset through the quality of the model that generated it. A model of quality $Q$ is assumed to generate the data of quality $Q$ if no data quality-improving modifications are applied. The synthetic data of quality $Q$ is considered sufficiently good for training a model with a quality of up to $Q$. In other words, the model can only get as good as its training data is. Natural data is assumed to be of infinite quality.

We assume that we can boost the quality of the data synthesized by a model at the inference time at the cost of spending additional inference compute. This assumption comes from the results of the previous studies [31, 14] aimed at understanding the properties of the techniques that improve the quality of generated data [10]. The most promising technique to achieve this remains unclear at this point. Still,

the overall analysis of a wide range of techniques in different domains of AI [31] has concluded that multiplicative trade-offs in compute between the training and inference phases are achievable. More formally, if the original model of quality $Q$ required $T$ floating-point operations (FLOP) of compute during training and $\iota$ FLOP/token of compute during inference, then for a value of $\gamma \in [1, \bar{\gamma}]$, there is a combination of techniques that allows spending $T/\gamma$ FLOP in training and $\gamma\iota$ FLOP/token in inference to obtain an AI system that generates data of the same quality $Q$ as the original one. Here $\bar{\gamma}$ is the span of the achievable trade-off. To carry this reasoning over to the case of improved generation quality, we assume that a modified system that uses $\gamma\iota$ FLOP/token of inference compute delivers the data quality of $Q(f(\gamma)T)$ if the original system that used $\iota$ FLOP/token of inference compute delivered the generation of quality $Q(T)$. Here $f$ is a monotone function representing the exact form of the training-inference compute trade-off. In Section A.2 we show that the results of [31] imply that $f(\gamma) = \gamma^{\frac{\alpha+\beta}{\alpha+2\beta}}$. For our theoretic derivations, we assume $f(\gamma)$ to take the parametric form $f(\gamma) = \gamma^h$ and later find that the main results only weakly depend on the value of $h$.

We assume the synthesizer models are trained according to the TCO ratio between the model size and the number of training tokens. This assumption simplifies the theoretical derivation significantly, while our numerical analysis in Section D shows that the TCO regime delivers a similar value of the total computational cost to the one obtained without this assumption. The generality of the training-inference trade-off also places overtraining and undertraining into the arsenal of tools available for balancing the training and inference cost terms.

In Section 5, we analyze how much our results depend on the exact formulation of the assumptions above, assuring that our conclusions will hold in a wide range of scenarios that are possible in the future.

## 4 Main results

We consider the process of training $n$ synthesizer LLMs before training the target model. We index the models in the order they are being trained, with the base model having the index $i = 0$ and the target model having the index $i = n$. All of the models are trained in the TCO regime. The base model is trained using exclusively natural data, while every consecutive model is trained using a mix of natural data and the data generated according to the synthetic data quality assumption. $Q_i$ denotes the quality of the model $i$ before applying any quality-improving modifications, with the base model having the quality $Q_0$ and the target model having the quality $Q_n$. Our key result is the simple rule of thumb that allows us to identify the optimal number $n^\star$ of models in the chain of synthesizers from the relative quality of the target and the base models $Q_n/Q_0$ and derive the relative quality of the successive models $Q_{i+1}/Q_i$ which turns out to be a universal quantity. The ratio of the training compute costs $T_{i+1}/T_i$, where $T_i = T(Q_i)$, can equivalently measure the relative quality of the successive models. We establish that the optimal values of the ratios $T_{i+1}/T_i$ and $Q_{i+1}/Q_i$ are independent of the model index $i$ and can be found as

$$\frac{T_{i+1}}{T_i} = \gamma \quad \text{or} \quad \frac{Q_{i+1}}{Q_i} = \gamma^{\frac{\alpha\beta}{\alpha+\beta}}$$

where $\gamma$ is the solution to the equation

$$(1 + h(c-1))\gamma^{h(1+c)} - (1+ch)\gamma^{ch} - 3h\gamma^{h-1} = 0$$

where $c = \frac{\alpha}{\alpha+\beta}$ and $h = \frac{\alpha+\beta}{\alpha+2\beta}$. Solving this equation for the scaling law parameters derived by Besiroglu et al. [4] results in

$$\gamma = T_{i+1}/T_i \approx 5.28 \text{ and } Q_{i+1}/Q_i \approx 1.35,$$

implying that the amount of training compute $T_i$ spent on model $i+1$ should be approximately $5.28$ times higher than the amount of training compute $T_i$ spent on model $i$. A prominent corollary of this result derived in Section A.6 is that the multistage synthesis consumes approximately double ($2.19\times$) the required computational resources compared to training on natural data. This factor is relatively small when considering the current pace of compute scaling. It does not increase with the scale of the models, affirming the feasibility of the multistage synthesis approach.

The existence of $\gamma$ and its value are direct corollaries of the scaling law and do not depend on the qualities of the base or target models. This result can inform the optimal immediate scaling decisions

if the target model quality is not clearly defined. This ambiguity is often the case today when planning the model training several generations in advance. However, the optimal number of training stages can then be computed as

$$n^\star(Q_n/Q_0) \approx \left\lfloor \frac{(\alpha+\beta)\log Q_n/Q_0}{\alpha\beta\log\gamma} \right\rceil = \left\lfloor \frac{\log T_n/T_0}{\log\gamma} \right\rceil$$

if the target model quality is clearly defined, where $\lfloor\cdot\rceil$ denotes rounding the value to the nearest integer. The quality of this approximation is illustrated in Figure 1.

### 4.1 Multistage data synthesis problem

Based on the assumptions, we can formulate the problem of optimal $n$-stage data synthesis. We first introduce the notation. The synthesizer model sizes are denoted as $M_0$, $M_1, \ldots, M_{n-1}$, and the target model size denoted as $M_n$. The amount of synthetic data generated by model $i$ is denoted as $N_i$ with $N_n = 0$. The amount of its training data is $D_i = D_0 + N_{i-1}$. The amount of training compute of a model $i$ is $T_i = 6M_iD_i$ and the amount of inference compute spent by model $i$ is $2\gamma_i M_i N_i$, where $\gamma_i$ is the multiplier that accounts for the trade-off between the amount of inference compute per token generated and the quality of the data generated by model $i$. The problem of minimizing the total amount of computing resources spent to train the target model subject to the model and data quality constraints takes the form derived in Section A

$$\begin{aligned} Z(n) = \min_{M_i,N_i,T_i,\gamma_i\in[1,\bar\gamma]} \quad & 2\textstyle\sum_{i=0}^{n-1}\gamma_i M_i N_i + \sum_{i=1}^{n} T_i \\ \text{s.t.} \quad & M_i = M(T_i) \quad i \in \{0\ldots n\} \\ & N_i = D(T_{i+1}) - D_0 \quad i \in \{0\ldots n-1\} \\ & Q(T_n) = Q_n \\ & Q(T_0) = Q_0 \\ & Q(T_i) = Q(f(\gamma_{i-1})T_{i-1}) \quad i \in \{1\ldots n\} \end{aligned} \tag{2}$$

The problem formulation and solution details are available in Section A. The solution is robust to the choice of the constraint parameter $\bar\gamma$ representing the span of the inference-training trade-off. For the values $\bar\gamma > 10$, the box constraints are not binding and can be omitted.

The behavior of $Z(n)$ defined by (2) for different values of $\frac{Q_n}{Q_0}$ is illustrated in Figure 5. The graph of $Z(n)$ has a distinct minimum value over $n$ that depends on $\frac{Q_n}{Q_0}$. We call this value $n^\star(\frac{Q_n}{Q_0})$. Its dependency on $\frac{Q_n}{Q_0}$ is depicted in Figure 1. The value of $n^\star$ is a piecewise constant function of the logarithm of the relative model quality $\log\frac{Q_n}{Q_0}$ with steps of approximately equal size.

Let us denote the optimal value of $\gamma_i$ of the $n$-stage problem as $\gamma_i(n)$. We plot $\gamma_i(n^\star)$ obtained for $f(\gamma) = \gamma^{\frac{\alpha+\beta}{\alpha+2\beta}}$ in Figure 2a and empirically observe that $\gamma_0(n^\star) \approx \ldots \approx \gamma_{n^\star-1}(n^\star)$. We use this fact in Section A.5 to theoretically derive that

$$n^\star(Q_n/Q_0) \approx \left\lfloor \frac{(\alpha+\beta)\log Q_n/Q_0}{\alpha\beta\log\gamma} \right\rceil$$

and that

$$\gamma_0(n^\star) \approx \ldots \approx \gamma_{n^\star-1}(n^\star) \approx \gamma = 5.3$$

Thus, we show that the constant scaling factor is optimal in the synthetic data generation regime, estimate its value, and develop a simple recipe for the optimal scaling behavior. Remarkably, the scaling factor does not depend on the values of $Q_n$ or $Q_0$, which makes the initial assumptions on the quality of the models and the exact number of stages unnecessary. Dropping those assumptions and taking the value of $n$ to infinity, the optimal scaling regime would still be multiplying the amount of training compute by $\gamma \approx 5.3$ for every next model in the chain.

## 5 Sensitivity analysis

In this Section, we explore the resilience of our primary findings to alterations in our initial assumptions and provide additional justifications for them. Given our focus on predictive rather than

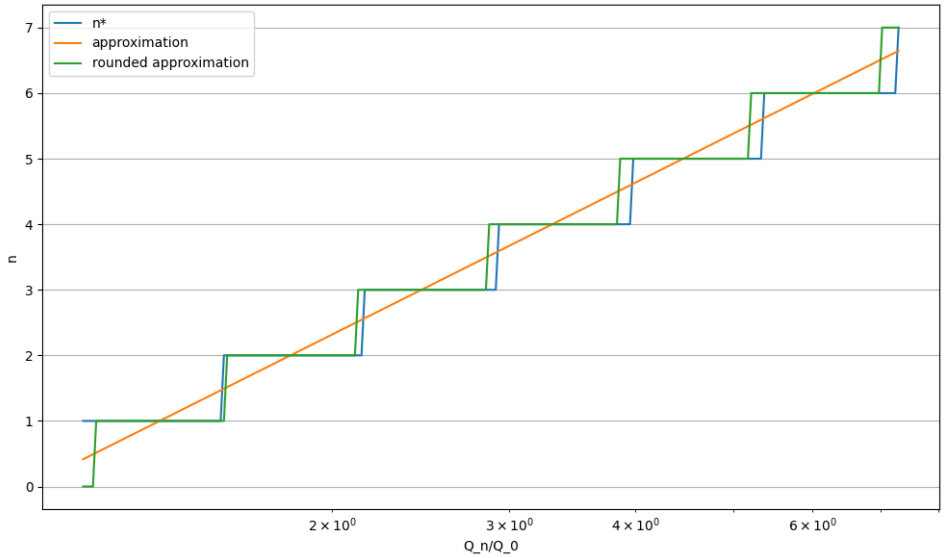



Figure 1: The optimal number $n^\star$ of training stages as a function of the number relative model quality $Q_n/Q_0$, its continuous approximation $\frac{(\alpha+\beta)\log Q_n/Q_0}{\alpha\beta\log\gamma}$ and its piece-wise constant approximation $\left\lfloor \frac{(\alpha+\beta)\log Q_n/Q_0}{\alpha\beta\log\gamma} \right\rceil$.

descriptive theory, these assumptions carry inherent uncertainties. Therefore, it is crucial to identify which ones may not hold in the future and understand the potential implications on our results should they fail.

### 5.1 Sufficient data quality

Our main results are derived under strong assumptions on the data quality, implying that all of the data generated in the previous synthesizing stages has to be discarded. This is captured mathematically by the equation

$$N_i = D(T_{i+1}) - D_0$$

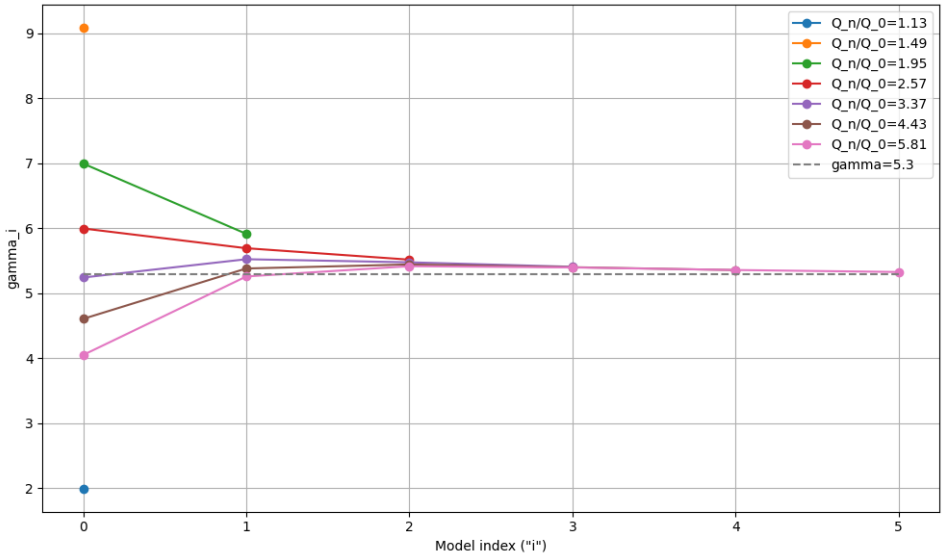
in the constraints of the problem (2). In fact, some of the previously generated synthetic data might get reused by curriculum learning or after filtering. To account for that possibility, we consider the other extreme case: when none of the synthetically generated data is discarded. This case can be captured mathematically by the equation

$$N_i = D(T_{i+1}) - D(T_i)$$

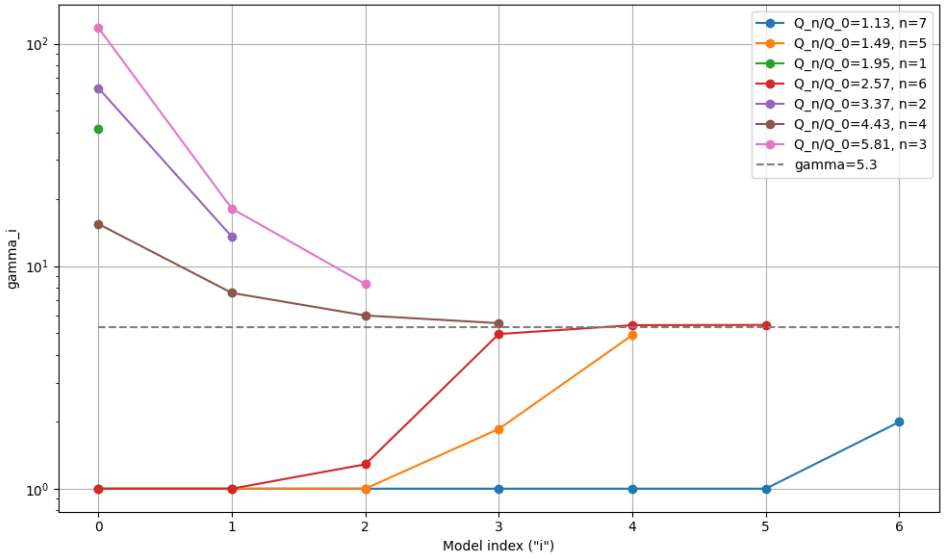
which can enter (2) by replacing $N_i = D(T_{i+1}) - D_0$. The results of this modification are briefly discussed in Section A.1. The outcome of the numerical experiment solving the modified version of (2) is almost indistinguishable from the results reported in Figures 2a and 3a, testifying that the exact policy of data handling would not impact the reported findings.

### 5.2 Inference-training trade-off

The training-inference trade-off is fundamental in our synthetic data generation setup. There is no guarantee that it will not change in the future with the introduction of new data quality improvement techniques. Hence, we considered several possible variants of the trade-off function $f(\gamma)$. The theory in Section A is derived for a function in the form $f(\gamma) = \gamma^h$, and it is natural to study the dependency of the numerical results on the value of $h$. We believe that only monotonically increasing functions with $f(1) = 1$ can represent the training-inference trade-off. Thus, we also look at other functions of this sort, including $f(\gamma) = e^{\gamma-1}$, $f(\gamma) = 1 + \log\gamma$, and linear functions of form $f(\gamma) = g\gamma - g + 1$ with different values of $g$. The results of this investigation are collected in Figure 4. It is evident that the convergence of $\gamma_i$ to a value of around 5 as $i$ approaches $n^\star$ holds in all of the considered cases. However, the manner of this convergence can be split into two different categories. In the majority




(a) The behavior of $\gamma_i(n)$ with increasing $i$ for $n = n^\star(\frac{Q_n}{Q_0})$

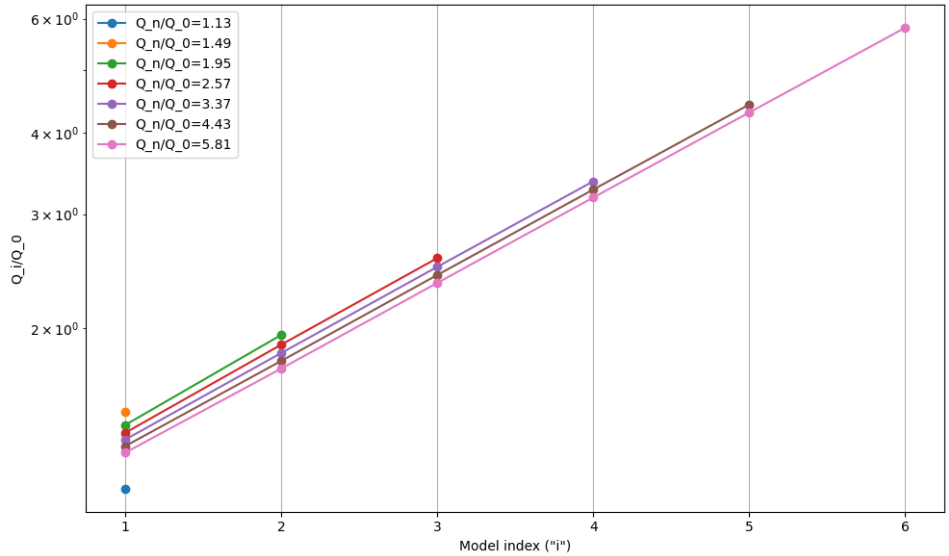



(b) The behavior of $\gamma_i(n)$ with increasing $i$ for arbitrarily chosen values of $n$.

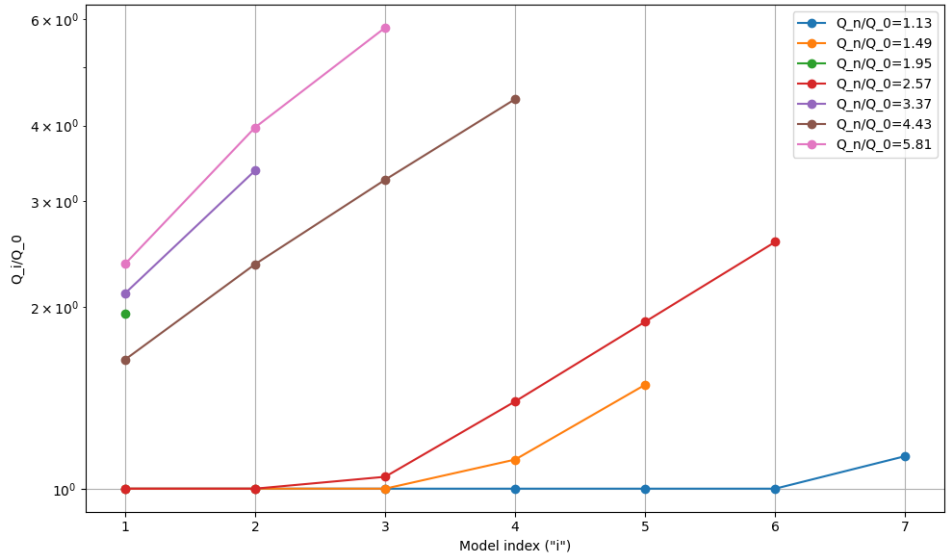
Figure 2: The values $\gamma_i(n)$ for $i$ from 0 to $n-1$ that correspond to the optimal solution of (2) and (3) for different values of the relative quality of the base and the target models $Q_n/Q_0$

of cases, including $f(\gamma) = \gamma^h$ for all $h > 0$, the linear functions with $g > 1$ and $f(\gamma) = 1 + \log \gamma$, the convergence is uniform over $\frac{Q_n}{Q_0}$. In other words, given model index $i$, if for a smaller value of $\frac{Q_n}{Q_0}$ the difference between $\gamma_i(n^\star(\frac{Q_n}{Q_0})) - \gamma_{n^\star}(n^\star(\frac{Q_n}{Q_0}))$ is small, then for a larger value of $\frac{Q_n}{Q_0}$ the difference $\gamma_i(n^\star(\frac{Q_n}{Q_0})) - \gamma_{n^\star}(n^\star(\frac{Q_n}{Q_0}))$ is small as well. This does not hold for the linear function with $g = 0.3$ and $f(\gamma) = e^{\gamma-1}$. For these functions, the value of $\gamma_i$ remains around 1 and increases only upon approaching $i = n^\star$. This behavior makes it impossible to take the ratio $\frac{Q_n}{Q_0}$ between the target and base model qualities to infinity since, in this case, the predicted behavior would be to train an infinite number of identical models without producing any synthetic data at all.

Although the uniform convergence holds for $f(\gamma) = \gamma^h$ for all $h$, the convergence rate is slower for the smaller values of $h$. We conclude that the final results are very robust to the nature of the




(a) The behavior of $\frac{Q_i}{Q_0}$ with increasing $i$ for $n = n^\star(\frac{Q_n}{Q_0})$




(b) The behavior of $\frac{Q_i}{Q_0}$ with increasing $i$ for arbitrarily chosen values of $n$.

Figure 3: The relative quality $Q_i/Q_0$ of the $i$-th synthesizer and the base model depending on $i$ from 1 to $n$ that correspond to the optimal solutions illustrated in Figure 2 for different values of the relative quality of the base and the target models $Q_n/Q_0$

training-inference trade-off but have their limits of applicability. This caveat highlights that the main conclusions of this work, although intuitive, are not trivial or self-evident and are based on our current understanding of the nature of machine learning scaling.

### 5.3 Scaling law

We have considered the robustness of our conclusions with respect to the coefficients in the scaling law (1), which are likely to change in the future due to the advances in machine learning training algorithms and models. After reproducing the experiments with the coefficients from [19] indicated in Table 1, we observe that almost all of the numerical results coincide with a high degree of accuracy with the results reported using coefficients estimated by [4]. The only exception is the result about

the balance between the training and inference compute reported in Section A.4, where we provide the results using both sets of scaling law coefficients. We do not duplicate the rest of the results for the new scaling law coefficients as they are indistinguishable from the ones reported in Figures 1, 2 and 3.

### 5.4 Training regime

We examine the assumption that the synthesizer models are trained according to the training compute optimal (TCO) regime presented by [19]. While not strictly necessary, this assumption greatly simplifies both numerical experiments and theoretical derivations, yielding well-interpretable results. However, the realism of this assumption is not self-evident, especially in view of the results by [25]. In this Section, we summarize the results that motivate this assumption and then mention additional efforts we have made to verify it.

In Section C, we quantified the optimal model size in the simple training-inference setup of [25], which neglects the fact that the data generated during the inference phase will be used in training of the future model. The resulting model size was smaller than the corresponding TCO model size and could be approximated as (12):

$$M(Q)\left(1-\frac{1}{3(\alpha+\beta)}\left(\frac{Q_N}{Q}\right)^{\frac{1}{\beta}}\right),$$

where $Q$ is the quality of the model used for inference and $Q_N$ is the best quality of a model that could be trained on all of the $N$ tokens of the generated data.

In Section B, we quantified the optimal model size in the one-stage synthetic data generation setup, which neglects the fact that the target model will be used to generate new synthetic data. The resulting model size was larger than the corresponding TCO model size with the approximation (8):

$$M(Q_1)\left(1+\frac{1}{3\,(\beta+\alpha)}\left(\frac{Q_1}{Q_0}\right)^{\frac{1}{\beta}}\right),$$

where $Q_0$ is the quality of the base model and $Q_1$ is the quality of the target model. Here, through $M(Q)=M(T(Q))$, we denote the TCO model size corresponding to the model quality $Q$.

Let us take into account simultaneously that the target model will be used to generate new synthetic data and that the data generated will be used in training the future model. The target model quality equals $Q_1=Q$ in equations (12) and (8), respectively. The base model quality would be $Q_0$, and the future model quality is almost equal to $Q_N$ in the realistic case that the amount of generated data far exceeds the amount of natural data ($D \ll N$). We can expect that $\frac{Q_1}{Q_0} \approx \frac{Q_N}{Q}$ since the values of $\gamma_i$ are close to each other in a variety of considered settings. Thus, the size of the target model in the first-order approximation should be close to the TCO regime $M(Q_1)$ as the average between (12) and (8).

In Section D, we describe an experiment that allowed us to check that the assumption of the TCO training regime does not lead to a dramatic increase in the compute costs, implying that the TCO regime is close to optimal. Additional evidence for this behavior comes from the result described in Section A.3 that the training compute exceeds the inference compute on average by a factor of over $40$ in long chains of synthesizer models. However, this result appears to be highly dependent on the exact parameters of the scaling law as noted in Section 5.3.

## 6 Conclusion

We presented the main result, which is that the practice of releasing state-of-the-art AI models one generation after the other will continue after the data bottleneck has been reached. Based on the rationality of computing resources, we predict that every new version of LLM will consume approximately five times more computing power than the previous one. Thus, a new version is expected to be released roughly every year [12]. We conducted an extensive sensitivity analysis of our results to find their limits of applicability and identify the most risky assumptions. Our study indicated that some alternative forms of the inference-training trade-off would result in conceptually

different results. We identified the assumption on the training regime of the synthesizer models as the most uncertain one and reduced its risks by eliminating it from core numerical experiments. We conclude that the predicted yearly scheduling of model releases, although intuitive, is not a trivial result and is grounded in our current understanding of the nature of machine learning scaling.

## References


[1] N. C. Abay, Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and L. Sweeney. Privacy preserving synthetic data release using deep learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 510–526. Springer, 2019.

[2] S. Altman. OpenAI now generates about 100 billion words per day. `https://twitter.com/sama/status/1756089361609981993`, 2024. Accessed: 2024-06-10.

[3] R. Babbar and B. Schölkopf. Data scarcity, robustness and extreme multi-label classification. *Machine Learning*, 108(8):1329–1351, 2019.

[4] T. Besiroglu, E. Erdil, M. Barnett, and J. You. Chinchilla scaling: A replication attempt. *arXiv preprint arXiv:2404.10102*, 2024.

[5] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.

[6] X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[7] C. Burns, P. Izmailov, J. H. Kirchner, B. Baker, L. Gao, L. Aschenbrenner, Y. Chen, A. Ecoffet, M. Joglekar, J. Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *ArXiv preprint*, abs/2312.09390, 2023. URL `https://arxiv.org/abs/2312.09390`.

[8] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168, 2021. URL `https://arxiv.org/abs/2110.14168`.

[9] A. Creswell, M. Shanahan, and I. Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. *ArXiv preprint*, abs/2205.09712, 2022. URL `https://arxiv.org/abs/2205.09712`.

[10] D. Dohan, W. Xu, A. Lewkowycz, J. Austin, D. Bieber, R. G. Lopes, Y. Wu, H. Michalewski, R. A. Saurous, J. Sohl-Dickstein, et al. Language model cascades. *arXiv preprint arXiv:2207.10342*, 2022.

[11] E. Dohmatob, Y. Feng, P. Yang, F. Charton, and J. Kempe. A tale of tails: Model collapse as a change of scaling laws. *arXiv preprint arXiv:2402.07043*, 2024.

[12] Epoch AI. Key trends and figures in machine learning, 2023. URL `https://epochai.org/trends`. Accessed: 2023-06-10. The training compute of notable ML models has been growing at 4.1x per year since 2010.

[13] Epoch AI. Parameter, compute and data trends in machine learning, 2024. URL `https://epochai.org/data/epochdb/visualization`. Accessed: 2024-06-10.

[14] E. Erdil. Optimally allocating compute between inference and training, 2024. URL `https://epochai.org/blog/optimally-allocating-compute-between-inference-and-training`.

[15] M. Gerstgrasser, R. Schaeffer, A. Dey, R. Rafailov, H. Sleight, J. Hughes, T. Korbak, R. Agrawal, D. Pai, A. Gromov, et al. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*, 2024.

[16] F. Gilardi, M. Alizadeh, and M. Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023.

[17] L. Guarnera, O. Giudice, and S. Battiato. Deepfake detection by analyzing convolutional traces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 666–667, 2020.

[18] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

[19] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. Training compute-optimal large language models. *ArXiv preprint*, abs/2203.15556, 2022. URL https://arxiv.org/abs/2203.15556.

[20] Y. Huang, J. Zhang, Z. Shan, and J. He. Compression represents intelligence linearly. *arXiv preprint arXiv:2404.09937*, 2024.

[21] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *ArXiv preprint*, abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361.

[22] R. Liu, J. Wei, F. Liu, C. Si, Y. Zhang, J. Rao, S. Zheng, D. Peng, D. Yang, D. Zhou, et al. Best practices and lessons learned on synthetic data for language models. *arXiv preprint arXiv:2404.07503*, 2024.

[23] M. Nye, A. J. Andreassen, G. Gur-Ari, H. Michalewski, J. Austin, D. Bieber, D. Dohan, A. Lewkowycz, M. Bosma, D. Luan, C. Sutton, and A. Odena. Show your work: Scratchpads for intermediate computation with language models. *ArXiv preprint*, abs/2112.00114, 2021. URL https://arxiv.org/abs/2112.00114.

[24] OpenAI. Gpt-4o and more tools to chatgpt free. https://openai.com/index/gpt-4o-and-more-tools-to-chatgpt-free/, 2023. Accessed: 2023-06-10.

[25] N. Sardana and J. Frankle. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. *arXiv preprint arXiv:2401.00448*, 2023.

[26] I. Shumailov, Z. Shumaylov, Y. Zhao, Y. Gal, N. Papernot, and R. Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023.

[27] Y. Tay, M. Dehghani, S. Abnar, H. W. Chung, W. Fedus, J. Rao, S. Narang, V. Q. Tran, D. Yogatama, and D. Metzler. Scaling laws vs model architectures: How does inductive bias influence scaling? *arXiv preprint arXiv:2207.10551*, 2022.

[28] B. Van Breugel, Z. Qian, and M. Van Der Schaar. Synthetic data, real errors: how (not) to publish and use synthetic data. In *International Conference on Machine Learning*, pages 34793–34808. PMLR, 2023.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[30] P. Villalobos. Scaling laws literature review, 2023. URL https://epochai.org/blog/scaling-laws-literature-review.

[31] P. Villalobos and D. Atkinson. Trading off compute in training and inference, 2023. URL https://epochai.org/blog/trading-off-compute-in-training-and-inference.

[32] P. Villalobos, J. Sevilla, L. Heim, T. Besiroglu, M. Hobbhahn, and A. Ho. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *ArXiv preprint*, abs/2211.04325, 2022. URL https://arxiv.org/abs/2211.04325.

[33] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. 2022. URL https://arxiv.org/abs/2203.11171.

[34] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[35] E. Wood, T. Baltrusaitis, C. Hewitt, S. Dziadzio, T. J. Cashman, and J. Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 3661–3671. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00366. URL https://doi.org/10.1109/ICCV48922.2021.00366.

[36] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023. *URL https://arxiv.org/pdf/2305.10601. pdf*, 2023.

[37] E. Zelikman, Y. Wu, J. Mu, and N. Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

# A Multistage data synthesis appendix

In this Section, we formulate the problem (2) of optimal compute allocation across multistage process of model training and synthetic data generation. The problem is manipulated into a tractable form for a generic training-inference trade-off function. Based on the current understanding of the training-inference trade-off, we derive $f(\gamma) = \gamma^{\frac{\alpha+\beta}{\alpha+2\beta}}$ as the most likely form of the function. For $f(\gamma) = \gamma^h$, we further simplify the problem and propose a simple method of its approximate solution.

## A.1 Formulating the multistage synthesis problem

The objective function of the total amount of computing resources spent to train the target model, which can be approximated as

$$2\gamma_0 M_0 N_0 + T_1 + \ldots + 2\gamma_{n-1} M_{n-1} N_{n-1} + T_n = 2\sum_{i=0}^{n-1} \gamma_i M_i N_i + \sum_{i=1}^{n} T_i$$

The condition of equality between the training data quality and the quality of the model being trained can be expressed as

$$Q(T_i) = Q(f(\gamma_{i-1})T_{i-1})$$

Due to the simple form of the function $Q(T)$, it is equivalent to

$$T_i = f(\gamma_{i-1})T_{i-1}$$

which allows to express $T_i$ through $T_n$ as

$$T_i = f(\gamma_i)^{-1} T_{i+1} = T_n \prod_{j=i}^{n-1} f(\gamma_j)^{-1} = T_n \Gamma_i$$

where $\Gamma_i = \prod_{j=i}^{n-1} f(\gamma_j)^{-1} = f(\gamma_{i-1})\Gamma_{i-1}$. Both $T_0 = (\lambda Q_0)^{\frac{\alpha+\beta}{\alpha\beta}}$ and $T_n = (\lambda Q_n)^{\frac{\alpha+\beta}{\alpha\beta}}$ are fixed by the problem formulation. Thus, a key constraint on $\gamma_i$ is that

$$(\lambda Q_0)^{\frac{\alpha+\beta}{\alpha\beta}} = T_0 = T_n \prod_{j=0}^{n-1} f(\gamma_j)^{-1} = (\lambda Q_n)^{\frac{\alpha+\beta}{\alpha\beta}} \prod_{j=0}^{n-1} f(\gamma_j)^{-1}.$$

The amount of synthetic data generated can be expressed as a function of training compute:

$$N_i = D(T_{i+1}) - D(T_0) = \left[T_{i+1}^{\frac{\alpha}{\alpha+\beta}} - T_0^{\frac{\alpha}{\alpha+\beta}}\right] \delta,$$

which is guaranteed to be non-negative due to $\gamma_i \geq 1$. Substituting the expressions for $N_i$ and $M_i$ through the training compute, we get a new form for the objective:

$$\sum_{i=0}^{n-1} \gamma_i \left[T_{i+1}^{\frac{\alpha}{\alpha+\beta}} - T_0^{\frac{\alpha}{\alpha+\beta}}\right] T_i^{\frac{\beta}{\alpha+\beta}} \mu\delta + \sum_{i=1}^{n} T_i$$

Notice that $\mu\delta = 1/3$. An equivalent objective function would be

$$\sum_{i=0}^{n-1} \gamma_i \left[T_{i+1}^{\frac{\alpha}{\alpha+\beta}} - T_0^{\frac{\alpha}{\alpha+\beta}}\right] T_i^{\frac{\beta}{\alpha+\beta}} + 3\sum_{i=1}^{n} T_i$$

or

$$\gamma_0 \left[ T_1^{\frac{\alpha}{\alpha+\beta}} - T_0^{\frac{\alpha}{\alpha+\beta}} \right] T_0^{\frac{\beta}{\alpha+\beta}} + 3T_n + \sum_{i=1}^{n-1} \gamma_i \left[ T_{i+1}^{\frac{\alpha}{\alpha+\beta}} - T_0^{\frac{\alpha}{\alpha+\beta}} \right] T_i^{\frac{\beta}{\alpha+\beta}} + 3T_i$$

The constant term $3T_n$ can be omitted in the objective function. Let us substitute $T_i = \Gamma_i T_n$

$$T_n \left( \gamma_0 \left[ \Gamma_1^{\frac{\alpha}{\alpha+\beta}} - \Gamma_0^{\frac{\alpha}{\alpha+\beta}} \right] \Gamma_0^{\frac{\beta}{\alpha+\beta}} + \sum_{i=1}^{n-1} \gamma_i \left[ \Gamma_{i+1}^{\frac{\alpha}{\alpha+\beta}} - \Gamma_0^{\frac{\alpha}{\alpha+\beta}} \right] \Gamma_i^{\frac{\beta}{\alpha+\beta}} + 3\Gamma_i \right)$$

Dividing by positive constant $T_n$ we simplify the objective into

$$\gamma_0 \left[ \Gamma_1^{\frac{\alpha}{\alpha+\beta}} - \Gamma_0^{\frac{\alpha}{\alpha+\beta}} \right] \Gamma_0^{\frac{\beta}{\alpha+\beta}} + \sum_{i=1}^{n-1} \gamma_i \left[ \Gamma_{i+1}^{\frac{\alpha}{\alpha+\beta}} - \Gamma_0^{\frac{\alpha}{\alpha+\beta}} \right] \Gamma_i^{\frac{\beta}{\alpha+\beta}} + 3\Gamma_i$$

Using the property that $\Gamma_{i+1} = f(\gamma_i)\Gamma_i = \Gamma_0 \prod_{j=0}^{i} f(\gamma_j)$ brings the objective to

$$\gamma_0 \left[ f(\gamma_0)^{\frac{\alpha}{\alpha+\beta}} \Gamma_0^{\frac{\alpha}{\alpha+\beta}} - \Gamma_0^{\frac{\alpha}{\alpha+\beta}} \right] \Gamma_0^{\frac{\beta}{\alpha+\beta}} +$$
$$+ \sum_{i=1}^{n-1} \gamma_i \left[ \Gamma_0^{\frac{\alpha}{\alpha+\beta}} \prod_{j=0}^{i} f(\gamma_j)^{\frac{\alpha}{\alpha+\beta}} - \Gamma_0^{\frac{\alpha}{\alpha+\beta}} \right] \Gamma_0^{\frac{\beta}{\alpha+\beta}} \prod_{j=0}^{i-1} f(\gamma_j)^{\frac{\beta}{\alpha+\beta}} + 3\Gamma_0 \prod_{j=0}^{i-1} f(\gamma_j)$$

dividing the objective by $\Gamma_0$ one gets

$$\gamma_0 \left[ f(\gamma_0)^{\frac{\alpha}{\alpha+\beta}} - 1 \right] + \sum_{i=1}^{n-1} \gamma_i \left[ \prod_{j=0}^{i} f(\gamma_j)^{\frac{\alpha}{\alpha+\beta}} - 1 \right] \prod_{j=0}^{i-1} f(\gamma_j)^{\frac{\beta}{\alpha+\beta}} + 3 \prod_{j=0}^{i-1} f(\gamma_j)$$

We get the following nonlinear problem with box constraints:

$$z(n) = \min_{\gamma_j \in [1, \bar{\gamma}]} \quad \gamma_0 \left[ f(\gamma_0)^{\frac{\alpha}{\alpha+\beta}} - 1 \right] + \sum_{i=1}^{n-1} \gamma_i \left[ \prod_{j=0}^{i} f(\gamma_j)^{\frac{\alpha}{\alpha+\beta}} - 1 \right] \prod_{j=0}^{i-1} f(\gamma_j)^{\frac{\beta}{\alpha+\beta}} + 3 \prod_{j=0}^{i-1} f(\gamma_j) \tag{3}$$

$$s.t. \qquad \left( \frac{Q_n}{Q_0} \right)^{\frac{\alpha+\beta}{\alpha\beta}} = \prod_{j=0}^{n-1} \gamma_j \tag{4}$$

Note that if $N_i = D(T_{i+1}) - D(T_i)$ instead of the original assumption $N_i = D(T_{i+1}) - D_0$ then the objective (3) turns into

$$\gamma_0 \left[ f(\gamma_0)^{\frac{\alpha}{\alpha+\beta}} - 1 \right] + \sum_{i=1}^{n-1} \gamma_i \left[ f(\gamma_i)^{\frac{\alpha}{\alpha+\beta}} - 1 \right] \prod_{j=0}^{i-1} f(\gamma_j) + 3 \prod_{j=0}^{i-1} f(\gamma_j)$$

The relationship between $z(n)$ from (3) and $Z(n)$ from (2) is straightforward

$$\frac{Z(n)}{T_0} = \frac{z(n)}{3} + \left( \frac{Q_n}{Q_0} \right)^{\frac{\alpha+\beta}{\alpha\beta}}$$

although the optimization problem (3) defining $z(n)$ is much easier to solve then (2).

### A.2 The most likely form of $f$

The previous research [31, 14] of the trade-off between training and inference compute of large language models has been focused on the ways of reducing the amount of training compute at the expense of inference compute without sacrificing the quality. We reformulate their main equi-quality findings in a way that is more suitable for our setup of improving the data quality at the expense of the inference compute. Let us introduce the two-variable model quality function

$$R(T, \iota)$$

where $T$ is the amount of training compute put into a model in the TCO regime and $\iota$ is the amount of inference compute used per synthesized token. If no data quality improvement techniques are applied then $\iota = I/N = 2M(T)$ and

$$R(T, T^{\frac{\beta}{\alpha+\beta}}\mu) = Q(T)$$

We are interested in understanding the value of $R(T, \gamma 2M(T))$. Let $\mathcal{T}$ be such that $R(T, \gamma T^{\frac{\beta}{\alpha+\beta}}\mu) = R(\mathcal{T}, \mathcal{T}^{\frac{\beta}{\alpha+\beta}}\mu) = Q(\mathcal{T})$. The equi-quality training-inference trade-off is formulated [31] as

$$R(T, \iota) = R(T/\xi, \xi\iota)$$

for any value $\xi$ such that $0 < \underline{\xi} < \xi < \bar{\xi}$. Thus, we have that

$$\mathcal{T} = T/\xi \text{ and } \mathcal{T}^{\frac{\beta}{\alpha+\beta}}\mu = \xi\gamma\mu T^{\frac{\beta}{\alpha+\beta}}, \text{ so } T/\xi = (\xi\gamma)^{\frac{\alpha+\beta}{\beta}} T.$$

Solving this yields $\xi = \gamma^{-\frac{\alpha+\beta}{\alpha+2\beta}}$ and thus

$$R(T, \gamma\mu T^{\frac{\beta}{\alpha+\beta}}) = Q(\gamma^{\frac{\alpha+\beta}{\alpha+2\beta}} T)$$

in other words, a modified system that uses $\gamma I$ FLOP of inference compute delivers the data quality of $Q(\gamma^{\frac{\alpha+\beta}{\alpha+2\beta}} T)$ if the original system that used $I$ FLOP of inference compute delivered the generation of quality $Q(T)$.

Note that for both sets of the scaling law parameters listed in Table 1

$$\frac{\alpha+\beta}{\alpha+2\beta} \approx \frac{2}{3}$$

### A.3 Solving multistage synthesis

The solution to (3) turns out to be robust to the choice of the constraint parameter $\bar{\gamma}$, as soon as the choice is realistic. For the values $\bar{\gamma} > 10$, and $f(\gamma) = \gamma^{\frac{\alpha+\beta}{\alpha+2\beta}}$ the box constraints are not binding, and can be omitted. We plot the normalized value $z(n) \left(\frac{Q_0}{Q_n}\right)^{\frac{\alpha+\beta}{\alpha\beta}}$ as a function of $n$ in Figure 5. Note that $\left(\frac{Q_0}{Q_n}\right)$ does not depend on $n$ here and determines the curve on the plot. The functions $z(n)$ depicted in Figure 5 achieve their distinct optimal values at a point that depends on $\frac{Q_n}{Q_0}$. We name this this value $n^\star(\frac{Q_n}{Q_0}) = \arg\min_n z(n)$. The behavior of $n^\star(\frac{Q_n}{Q_0})$ as a function of $\frac{Q_n}{Q_0}$ is depicted in Figure 1 where it takes the form of a piece-wise constant function with every step having an approximately equal size if x-axis represents the logarithm of the relative model quality between target and base.

We denote the optimal solution as $\gamma_0(n^\star(\frac{Q_n}{Q_0})), \ldots, \gamma_{n^\star(\frac{Q_n}{Q_0})-1}(n^\star(\frac{Q_n}{Q_0}))$, which are the values of the optimization variables that bring the objective (3) its optimal value $z(n^\star(\frac{Q_n}{Q_0}))$. We plot these values for different ratios $\frac{Q_n}{Q_0}$ in Figure 2a. As $i$ grows, the values quickly converge to a constant $\gamma$ independent of $\frac{Q_n}{Q_0}$. This behavior corresponds to the exponential growth of $\frac{Q_i}{Q_0}$ with $i$ observed in Figure 3a. For suboptimal values of $n \neq n^\star(\frac{Q_n}{Q_0})$ the convergence of $\gamma_i(n)$ toward $\gamma$ with $i$ is disrupted as shown in Figure 2b. This disruption causes the irregular scaling pattern depicted in Figure 3b. Other relevant values, such as the amount of synthetic data generated, the model size, and the amount of computational resources spent on training and inference, all follow similar scaling patterns to the one depicted in 3a as long as the number of synthesis steps remains optimal $n = n^\star(\frac{Q_n}{Q_0})$.

### A.4 Ratio between training and inference compute

The distribution of the total compute over training and inference phases of the life cycles of the $n^\star(\frac{Q_n}{Q_0})$ models is demonstrated in Figure 6a. The ratio between the two quantities levels out approaching the value of approximately $47$, assuming all of the inference compute budget is being used for synthetic data generation. This result supports the claim of Section 5.4 that the training-compute-optimal regime is a good approximation of the regime that is close to being optimal for the synthesizer model training within our framework.

Figure 6a utilized the solutions of the problem (2) obtained assuming the scaling law parameters found by [4]. For comparison, we also plot a similar figure 6b using the parameters from [19]. The corresponding curve still converges to a constant, although its limiting value is close to $3.35$. The obtained results align with the claim by [14] that the ratio between the inference and training compute will not explode and will not vanish over time as the availability of compute continues to grow.

### A.5 Deriving the behavior of $n^\star(\frac{Q_n}{Q_0})$ for $f(\gamma) = \gamma^h$

Let us denote $r = \frac{T_n}{T_0} = \left(\frac{Q_n}{Q_0}\right)^{\frac{\alpha+\beta}{\alpha\beta}}$. If $\gamma_0 = \gamma_1 = \ldots \gamma_{n-1} = \gamma$ then $\gamma = r^{1/n}$ by (4) and the function $z(n)$ can be written as

$$z(n) = \gamma \left[\gamma^{h\frac{\alpha}{\alpha+\beta}} - 1\right] + \sum_{i=1}^{n-1} \gamma \left[\prod_{j=0}^{i} \gamma^{h\frac{\alpha}{\alpha+\beta}} - 1\right] \prod_{j=0}^{i-1} \gamma^{h\frac{\beta}{\alpha+\beta}} + 3\prod_{j=0}^{i-1} \gamma^h$$

We apply a series of simplifying algebraic modifications that we record here for convenience. First, we substitute $\gamma = r^{1/n}$

$$z(n) = r^{1/n} \left[r^{\frac{h}{n}\frac{\alpha}{\alpha+\beta}} - 1\right] + \sum_{i=1}^{n-1} r^{1/n} \left[\prod_{j=0}^{i} r^{\frac{h}{n}\frac{\alpha}{\alpha+\beta}} - 1\right] \prod_{j=0}^{i-1} r^{\frac{h}{n}\frac{\beta}{\alpha+\beta}} + 3\prod_{j=0}^{i-1} r^{\frac{h}{n}},$$

simplify the product of identical terms

$$z(n) = r^{1/n} \left[r^{\frac{h\alpha/n}{\alpha+\beta}} - 1\right] + \sum_{i=1}^{n-1} r^{1/n} \left[r^{\frac{i+1}{n}\frac{h\alpha}{\alpha+\beta}} - 1\right] r^{\frac{i}{n}\frac{h\beta}{\alpha+\beta}} + 3r^{hi/n},$$

and take $r^{1/n}$ as a common factor

$$z(n) = r^{1/n} \left(\left[r^{\frac{h\alpha/n}{\alpha+\beta}} - 1\right] + \sum_{i=1}^{n-1} \left[r^{\frac{i+1}{n}\frac{h\alpha}{\alpha+\beta}} - 1\right] r^{\frac{i}{n}\frac{h\beta}{\alpha+\beta}} + 3r^{\frac{hi-1}{n}}\right).$$

We neglect the "$-1$" term in the brackets

$$z(n) \approx r^{1/n} \left(r^{\frac{h\alpha/n}{\alpha+\beta}} + \sum_{i=1}^{n-1} r^{\frac{hi(\alpha+\beta)+h\alpha}{n(\alpha+\beta)}} + 3r^{\frac{hi-1}{n}}\right)$$

and take $r^{\frac{h\alpha}{n(\alpha+\beta)}}$ out of the summation

$$z(n) \approx r^{1/n} \left(r^{\frac{h\alpha/n}{\alpha+\beta}} + r^{\frac{h\alpha}{n(\alpha+\beta)}} \sum_{i=1}^{n-1} r^{\frac{hi}{n}} + 3\sum_{i=1}^{n-1} r^{\frac{hi-1}{n}}\right)$$

Using the formula for the sum of the geometric series $\sum_{i=1}^{n-1} r^{\frac{hi}{n}} = r^{\frac{h}{n}} \frac{r^{h\frac{n-1}{n}} - 1}{r^{\frac{h}{n}} - 1}$ we obtain

$$z(n) \approx r^{1/n} \left( r^{\frac{h\alpha}{n(\alpha+\beta)}} + r^{\frac{h\alpha}{n(\alpha+\beta)}} r^{\frac{h}{n}} \frac{r^{h\frac{n-1}{n}} - 1}{r^{\frac{h}{n}} - 1} + 3r^{\frac{h-1}{n}} \frac{r^{h\frac{n-1}{n}} - 1}{r^{\frac{h}{n}} - 1} \right)$$

For $n >= 2$ we will neglect the first term in the sum.

$$z(n) \approx r^{1/n} \left( r^{\frac{h\alpha}{n(\alpha+\beta)}} r^{\frac{h}{n}} + 3r^{\frac{h-1}{n}} \right) \frac{r^{h\frac{n-1}{n}} - 1}{r^{\frac{h}{n}} - 1}$$

We neglect the "$-1$" in the numerator

$$z(n) \approx r^{h\frac{n-1}{n}+\frac{1}{n}} \left( r^{\frac{h\alpha}{n(\alpha+\beta)}} r^{\frac{h}{n}} + 3r^{\frac{h-1}{n}} \right) \frac{1}{r^{\frac{h}{n}} - 1}$$

which is equivalent to

$$z(n) \approx r^{h-\frac{h-1}{n}} \left( r^{\frac{h\alpha}{n(\alpha+\beta)}} r^{\frac{h}{n}} + 3r^{\frac{h-1}{n}} \right) \frac{1}{r^{\frac{h}{n}} - 1}$$

and, by taking $r^{\frac{h-1}{n}}$ out of the parentheses,

$$z(n) \approx r^{h} \left( r^{\frac{h\alpha}{n(\alpha+\beta)}} r^{\frac{1}{n}} + 3 \right) \frac{1}{r^{\frac{h}{n}} - 1}.$$

The positive constant $r^h$ does not affect the solution, so divide by it.

$$z(n) \approx \left( r^{\frac{h\alpha+\alpha+\beta}{n(\alpha+\beta)}} + 3 \right) \frac{1}{r^{\frac{h}{n}} - 1} = z'(n) \tag{5}$$

Coming back to the notation of $\gamma = r^{1/n}$, the function is

$$z_\gamma(\gamma) = \frac{\gamma^{1+\frac{h\alpha}{\alpha+\beta}} + 3}{\gamma^h - 1}$$

And its derivative is

$$\frac{dz_\gamma}{d\gamma}(\gamma) = \frac{(1+\frac{\alpha h}{\alpha+\beta})\gamma^{\frac{\alpha h}{\alpha+\beta}}(\gamma^h - 1) - h\gamma^{h-1}(3 + \gamma^{1+\frac{\alpha h}{\alpha+\beta}})}{(\gamma^h - 1)^2}$$

The firs-order necessary condition of local optimality demands the numerator of the derivative to be equal to zero at the optimal solution:

$$(1+\frac{\alpha h}{\alpha+\beta})\gamma^{\frac{\alpha h}{\alpha+\beta}}(\gamma^h - 1) = h\gamma^{h-1}(3 + \gamma^{1+\frac{\alpha h}{\alpha+\beta}}).$$

We open the parentheses

$$(1+\frac{\alpha h}{\alpha+\beta})\gamma^{h(1+\frac{\alpha}{\alpha+\beta})} - (1+\frac{\alpha h}{\alpha+\beta})\gamma^{\frac{\alpha h}{\alpha+\beta}} = 3h\gamma^{h-1} + h\gamma^{h(1+\frac{\alpha}{\alpha+\beta})}$$

and rearrange the terms

$$(1+h(\frac{\alpha}{\alpha+\beta} - 1))\gamma^{h(1+\frac{\alpha}{\alpha+\beta})} = 3h\gamma^{h-1} + (1+\frac{\alpha h}{\alpha+\beta})\gamma^{\frac{\alpha h}{\alpha+\beta}}$$

Substituting $\frac{\alpha}{\alpha+\beta} = c$ we get a compact condition of optimality which does not have an analytical solution but can be easily solved numerically

$$(1+h(c-1))\gamma^{h(1+c)} - (1+ch)\gamma^{ch} - 3h\gamma^{h-1} = 0.$$

After the optimal value of $\gamma$ has been found it could be substituted into $r = \frac{T_n}{T_0} = \left(\frac{Q_n}{Q_0}\right)^{\frac{\alpha+\beta}{\alpha\beta}} = \gamma^n$ results in the approximation

$$n^\star(Q_n/Q_0) \approx \left\lfloor \frac{(\alpha+\beta)\log Q_n/Q_0}{\alpha\beta\log\gamma} \right\rceil = \left\lfloor \frac{\log T_n/T_0}{\log\gamma} \right\rceil$$

which demonstrates a remarkably high quality as captured by Figure 1 where the approximation is plotted against the exact solution for $n^\star(Q_n/Q_0)$. For Figure 1, we find the solution of the equation for the specific values of $h = \frac{\alpha+\beta}{\alpha+2\beta}$, $\alpha = 0.35$ and $\beta = 0.37$, obtaining the unique root $\gamma \approx 5.3$. Solving $r^{\frac{1}{n}} = 5.3$ results in $n^\star \approx 0.6 \ln r$. For $h = 1$, the root is $\gamma \approx 5.7$ which corresponds to $n^\star \approx 0.574 \ln r$.

### A.6 Comparing computational cost to training on natural data

Training compute optimal (TCO) regime of training of a single model $i$ with $T_i$ FLOP of compute requires $D(T_i)$ of prepared tokens. Each of the tokens requires $\gamma 2M(T_i/\gamma)$ FLOP for generation due to the relation $T_{i-1} = T_i/\gamma$. This amounts to $2\gamma M(T_i/\gamma)D(T_i)$ FLOP of compute which after simplification is reduced to $\gamma^{1-\frac{\beta}{\alpha+\beta}}\mu\delta T_i$. Noticing that $\mu\delta = 1/3$ and substituting $\gamma = 5.28$ results in data generation cost at step $i$ of around $0.78T_i$. Taking $i$ to infinity and summing up the cost across stages, we arrive at the total training cost

$$\text{training cost} = \sum_{j=0}^{\infty} T_i\gamma^{-j} = \frac{\gamma}{1-\gamma}T_i \approx 1.23T_i$$

and total data generation cost

$$\text{generation cost} \approx 0.78\frac{\gamma}{1-\gamma}T_i \approx 0.96T_i$$

which implies

$$\text{total computational cost} \approx 2.19T_i$$

## B One-stage synthetic data generation setup

Let us consider the problem of training a target model of quality $Q_1$ by synthesizing data using a base model of quality $Q_0 = Q(T_0)$. Thus, the base model consists of $M_0 = M(T_0)$ parameters and the size of its training dataset is $D_0 = D(T_0)$. The cost of generating synthetic data is $2\gamma M_0 N_0$ where $\gamma$ is fixed and such that $Q(f(\gamma)T_0) = Q_1$, while the cost of training the target model is $6M_1(D_0+N_0)$ where $M_1$ is the size of the target model to be trained. We consider the training-inference trade-off function $f(\gamma) = \gamma$ due to its simplicity and the resilience to its exact form observed in Section 5.2. Our task is to determine the optimal regime of training, which consists of the size $M_1$ of the target model and the amount of data generated by the base model $N_0$ that are associated with the lowest amount of total compute spent. Thus, the decision variables here are $M_1$ and $N_0$, and the optimization problem can be written as

$$\begin{array}{rl} \min_{M_1,N_0} & 6M_1(D_0+N_0) + 2\gamma M_0 N_0 \\ \text{s.t.} & \frac{A}{M_1^\alpha} + \frac{B}{(D_0+N_0)^\beta} = \frac{1}{Q_1} \end{array}$$

The gradient of the objective function can be written as

$$\nabla C(M_1, N_0) = \begin{pmatrix} 6(D_0+N_0) \\ 6M_1 + 2\gamma M_0 \end{pmatrix}$$

While the gradient of the constraint is

$$\nabla L(M_1, N_0) = \begin{pmatrix} -\alpha A M_1^{-\alpha-1} \\ -\beta B(D_0+N_0)^{-\beta-1} \end{pmatrix}$$

Leveraging the first-order Karush-Kunh-Tucker necessary conditions of local optimality, we search for the values of the primal variables $M$ and $N$ and the dual variable $\xi$ such that the following system is satisfied

$$\begin{cases} \nabla C(M_1, N_0) - \xi \nabla L(M_1, N_0) = 0 \\ \frac{A}{M_1^\alpha} + \frac{B}{(D_0+N_0)^\beta} = L^\star \end{cases}$$

The first equation gets expanded as $6(D_0 + N_0) = -\xi \alpha A M_1^{-\alpha-1}$ and $6M_1 + 2\gamma M_0 = -\xi \beta B (D_0 + N_0)^{-\beta-1}$. The dual variable can be eliminated, yielding

$$6(D_0 + N_0) M_1^{\alpha+1} = \frac{\alpha A}{\beta B}(6M_1 + 2\gamma M_0)(D_0 + N_0)^{\beta+1}$$

or

$$B(D_0 + N_0)^{-\beta} = \frac{\alpha A}{3\beta} \frac{3M_1 + \gamma M_0}{M_1^{\alpha+1}}$$

Substituting this expression into the constraint gives the following equation on $M_1$

$$\gamma M_0 M_1^{-\alpha-1} + 3\left(\frac{\beta}{\alpha} + 1\right) M^{-\alpha} - \frac{3\beta}{\alpha A} L^\star = 0 \tag{6}$$

The size of the base model can be expressed through the base model quality as

$$M_0 = Q_0^{\frac{1}{\alpha}} \lambda^{\frac{1}{\alpha}} \mu / 2$$

and condition of data quality $Q(\gamma T_0) = Q_1$, implies

$$\gamma = \left(\frac{Q_1}{Q_0}\right)^{\frac{\alpha+\beta}{\alpha\beta}}.$$

Let us look for the target model size in the following form:

$$M_1 = \rho Q_1^{\frac{1}{\alpha}} \lambda^{\frac{1}{\alpha}} \mu / 2 = \rho M(Q_1)$$

where $\rho$ is a generic function of $Q_0$ and $Q_1$ with positive values.

The optimality condition can be rewritten as

$$\left(\frac{Q_1}{Q_0}\right)^{\frac{\alpha+\beta}{\alpha\beta}} Q_0^{\frac{1}{\alpha}} \lambda^{\frac{1}{\alpha}} \mu 2^{-1} (\rho Q_1^{\frac{1}{\alpha}} \lambda^{\frac{1}{\alpha}} \mu 2^{-1})^{-\alpha-1} + 3\left(\frac{\beta}{\alpha} + 1\right)(\rho Q_1^{\frac{1}{\alpha}} \lambda^{\frac{1}{\alpha}} \mu 2^{-1})^{-\alpha} - \frac{3\beta}{\alpha A} Q_1^{-1} = 0$$

After simplifying we get

$$\left(\frac{Q_1}{Q_0}\right)^{\frac{1}{\beta}} Q_1^{-1} \lambda^{-1} \mu^{-\alpha} 2^\alpha \rho^{-\alpha-1} + 3\left(\frac{\beta}{\alpha} + 1\right) Q_1^{-1} \lambda^{-1} \mu^{-\alpha} 2^\alpha \rho^{-\alpha} - \frac{3\beta}{\alpha A} Q_1^{-1} = 0$$

Multiplying by $Q_1$ brings

$$\left(\frac{Q_1}{Q_0}\right)^{\frac{1}{\beta}} \lambda^{-1} \mu^{-\alpha} 2^\alpha \rho^{-\alpha-1} + 3\left(\frac{\beta}{\alpha} + 1\right) \lambda^{-1} \mu^{-\alpha} 2^\alpha \rho^{-\alpha} - \frac{3\beta}{\alpha A} = 0$$

Multiplying by $\lambda \mu^\alpha$ results in

$$\left(\frac{Q_1}{Q_0}\right)^{\frac{1}{\beta}} \rho^{-\alpha-1} + 3\left(\frac{\beta}{\alpha} + 1\right) \rho^{-\alpha} - \frac{3\beta}{2^\alpha \alpha A} \lambda \mu^\alpha = 0$$

After multiplying by $\rho^{\alpha+1}$ we get

$$\left(\frac{Q_1}{Q_0}\right)^{\frac{1}{\beta}} + 3\left(\frac{\beta}{\alpha}+1\right)\rho - \frac{3\beta}{2^{\alpha}\alpha A}\lambda\mu^{\alpha}\rho^{\alpha+1} = 0$$

Retrieving the definitions for $\lambda$ and $\mu$ allows us to get get the following expression to simplify the last term

$$2^{-\alpha}\mu^{\alpha}\lambda = A + B\kappa = A\frac{\alpha}{\beta}(1+\frac{\beta}{\alpha})$$

which eventually brings us to

$$\left(\frac{Q_1}{Q_0}\right)^{\frac{1}{\beta}} + 3\left(\frac{\beta}{\alpha}+1\right)\rho - 3\left(\frac{\beta}{\alpha}+1\right)\rho^{\alpha+1} = 0$$

or

$$\left(\frac{Q_1}{Q_0}\right)^{\frac{1}{\beta}} + 3\left(\frac{\beta}{\alpha}+1\right)\left(\rho - \rho^{\alpha+1}\right) = 0$$

We approximate $\rho^{1+\alpha}$ around $\rho = 1$ with Taylor expansion $\rho^{1+\alpha} \approx 1+(1+\alpha)(\rho-1) = \rho+\alpha\rho-\alpha$ which brings

$$\left(\frac{Q_1}{Q_0}\right)^{\frac{1}{\beta}} + 3\left(\beta+\alpha\right)\left(-\rho+1\right) \approx 0$$

or the final form

$$\rho \approx 1 + \frac{1}{3\left(\beta+\alpha\right)}\left(\frac{Q_1}{Q_0}\right)^{\frac{1}{\beta}} \tag{7}$$

resulting in

$$M \approx M(Q_1)\left(1 + \frac{1}{3\left(\beta+\alpha\right)}\left(\frac{Q_1}{Q_0}\right)^{\frac{1}{\beta}}\right) \tag{8}$$

The solution to the one-stage problem shows that the optimal model trained on synthetic data is undertrained compared to the TCO regime, which is drastically different from the total compute-optimal regime discussed in Section C. This is a consequence of considering $n = 1$ and taking the compute spent on inference with the target model out of the scope.

# C Training-inference optimality

The problem considered in this section was first formulated by [25] and provided here for the sake of completeness. Our notation here is consistent with the rest of the paper: $M$ is the number of model parameters, $D$ is the number of data tokens processed during the model training, and $N$ is the number of data tokens generated by the model over the inference phase of its lifetime. The generated tokens are served directly to the consumer, no data quality improvement is being applied to the generated data, and no further training is assumed for the data generated.

The amount of training compute spent over the lifetime of a model can be approximated as

$$6MD + 2MN$$

The zero-stage problem of training a model of a given target quality $Q$ in a way that would minimize the amount of computing over the lifetime can be written in the form

$$\begin{array}{ll} \min_{M,D} & 6MD + 2MN \\ \text{s.t.} & \frac{A}{M^{\alpha}} + \frac{B}{D^{\beta}} = \frac{1}{Q} \end{array}$$

The gradient of the objective function can be written as

$$\nabla C(M,N) = \begin{pmatrix} 6D + 2N \\ 6M \end{pmatrix}$$

While the gradient of the constraint is

$$\nabla L(M,N) = \begin{pmatrix} -\alpha A M^{-\alpha-1} \\ -\beta B D^{-\beta-1} \end{pmatrix}$$

Leveraging the first-order Karush-Kunh-Tucker necessary conditions of local optimality, we search for the values of the primal variables $M$ and $N$ and the dual variable $\lambda$ such that the following system is satisfied

$$\begin{cases} \nabla C(M,N) - \lambda \nabla L(M,N) = 0 \\ \frac{A}{M^\alpha} + \frac{B}{D^\beta} = \frac{1}{Q} \end{cases}$$

The first equation gets expanded as $6D + 2N = -\lambda\alpha A M^{-\alpha-1}$ and $6M = -\lambda\beta B D^{-\beta-1}$. The dual variable can be eliminated, yielding

$$(6D + 2N)M^{\alpha+1} = \frac{\alpha A}{\beta B} 6M D^{\beta+1}$$

or equivalently

$$\frac{A}{M^\alpha} = \frac{3\beta B D^{-\beta} + N\beta B D^{-\beta-1}}{3\alpha}.$$

Substituting this expression into the constraint gives the following equation on $D$

$$\frac{1}{Q} - \left(\frac{\beta B}{\alpha} + B\right) D^{-\beta} - \frac{N\beta B}{3\alpha} D^{-\beta-1} = 0 \tag{9}$$

As noticed by [25], this problem cannot be solved analytically, but finding the numerical solution for specific values of the parameters is possible and we analyze the numerical solution in Section C.2. Although the exact analytical solution is not possible, we can provide an approximate analytical solution.

### C.1 Approximate analytical solution

Let us introduce the function $D(Q)$ that takes the value of the amount of data that is necessary to train a model of quality $Q$ in the TCO regime. The explicit form of this function is

$$D(Q) = D(T(Q)) = (\lambda Q)^{\frac{1}{\beta}} \delta$$

Let us introduce $Q_N$ which is the quality of a model that is trained on $N$ natural tokens in the TCO regime. In other words, we define $Q_N = (N\delta)^\beta / \lambda$ to satisfy

$$N = D(Q_N) = (\lambda Q_N)^{\frac{1}{\beta}} \delta$$

Let us look for the solution of (9) in the form

$$D = \rho D(Q)$$

where $\rho$ is an arbitrary function of $Q$ and $Q_N$, which does not restrict the space of possible solution. We can rewrite the equation (9) as

$$D^{\beta+1} - Q\left(\frac{\beta B}{\alpha} + B\right) D - Q\frac{N\beta B}{3\alpha} = 0$$

Substituting the newly introduced quantities into it yields

$$\left((\lambda Q)^{\frac{1}{\beta}} \delta\rho\right)^{\beta+1} - Q\left(\frac{\beta B}{\alpha} + B\right)(\lambda Q)^{\frac{1}{\beta}} \delta\rho - Q\frac{\beta B}{3\alpha}(\lambda Q_N)^{\frac{1}{\beta}} \delta = 0$$

or

$$\delta\lambda^{\frac{1}{\beta}}Q^{\frac{1}{\beta}+1}\lambda\delta^{\beta}\rho^{\beta+1} - \delta\lambda^{\frac{1}{\beta}}Q^{\frac{1}{\beta}+1}\left(\frac{\beta B}{\alpha}+B\right)\rho - \delta\lambda^{\frac{1}{\beta}}Q^{\frac{1}{\beta}+1}\frac{\beta B}{3\alpha}Q_N^{\frac{1}{\beta}}Q^{-\frac{1}{\beta}} = 0$$

Dividing by $\delta\lambda^{\frac{1}{\beta}}Q^{\frac{1}{\beta}+1}$ results in a simplified expression

$$\lambda\delta^{\beta}\rho^{\beta+1} - \left(\frac{\beta B}{\alpha}+B\right)\rho - \frac{\beta B}{3\alpha}\left(\frac{Q_N}{Q}\right)^{\frac{1}{\beta}} = 0$$

We can show that $\lambda\delta^{\beta}/B = 1+\beta/\alpha$ by expanding

$$\lambda\delta^{\beta} = \kappa^{-\frac{\beta}{\alpha+\beta}}\left(A\kappa^{-\frac{\alpha}{\alpha+\beta}} + B\kappa^{\frac{\beta}{\alpha+\beta}}\right) = B + \frac{A}{\kappa} = B(1+\frac{A}{B}\frac{\beta B}{\alpha A})$$

Which further simplifies the equation for $\rho$ into

$$(\alpha+\beta)(\rho^{1+\beta}-\rho) - \frac{\beta}{3}\left(\frac{Q_N}{Q}\right)^{\frac{1}{\beta}} = 0$$

The first-order Taylor approximation $\rho^{1+\beta} \approx \rho + \beta\rho - \beta$ for $\rho^{1+\beta}$ around $\rho = 1$ allows to linearize the equation into

$$\rho \approx 1 + \frac{1}{3(\alpha+\beta)}\left(\frac{Q_N}{Q}\right)^{\frac{1}{\beta}} \tag{10}$$

which is remarkably similar to (7). It yields the approximate solution to (9)

$$D \approx D(Q)\left(1 + \frac{1}{3(\alpha+\beta)}\left(\frac{Q_N}{Q}\right)^{\frac{1}{\beta}}\right) \tag{11}$$

and allows to analytically estimate the optimal value of the training-inference optimal model size to be

$$M = M(Q)\left(1+\frac{\alpha}{\beta}(1-\rho^{-\beta})\right)^{-\frac{1}{\alpha}}$$

This can be shown by rewriting the model quality constraint in the form

$$M = A^{\frac{1}{\alpha}}\left(Q^{-1} - B(\rho D(Q))^{-\beta}\right)^{-\frac{1}{\alpha}}$$

and substituting the expression for $D(Q)$

$$M = A^{\frac{1}{\alpha}}\left(Q^{-1} - B\left(\lambda Q\right)^{-1}(\rho\delta)^{-\beta}\right)^{-\frac{1}{\alpha}}$$

Taking $\lambda Q$ outside of the parentheses and multiplying by 1 results in

$$M = (\lambda Q)^{\frac{1}{\alpha}}A^{\frac{1}{\alpha}}\left(\lambda - B(\rho\delta)^{-\beta}\right)^{-\frac{1}{\alpha}}2\mu^{-1}\mu/2$$

Note that the TCO model size that corresponds to the model quality $Q$ can be written as

$$M(Q) = (\lambda Q)^{\frac{1}{\alpha}}\mu/2$$

Which implies that

$$M = M(Q)A^{\frac{1}{\alpha}} \left(2^{-\alpha}\mu^{\alpha}\lambda - 2^{-\alpha}\mu^{\alpha}\delta^{-\beta}B\rho^{-\beta}\right)^{-\frac{1}{\alpha}}$$

Taking a closer look at the terms in the parentheses yields

$$(\mu/2)^{\alpha}\lambda = (6^{-\frac{\beta}{\alpha+\beta}}\kappa^{\frac{1}{\alpha+\beta}})^{\alpha}6^{\frac{\alpha\beta}{\alpha+\beta}}\left[A\kappa^{\frac{-\alpha}{\alpha+\beta}} + B\kappa^{\frac{\beta}{\alpha+\beta}}\right] = A(1+\frac{\alpha}{\beta})$$

and

$$(\mu/2)^{\alpha}\delta^{-\beta} = (6^{-\frac{\beta}{\alpha+\beta}}\kappa^{\frac{1}{\alpha+\beta}})^{\alpha}(6^{-\frac{\alpha}{\alpha+\beta}}\kappa^{\frac{-1}{\alpha+\beta}})^{-\beta} = \kappa = \frac{\alpha A}{\beta B}$$

which eventually brings us to the expression

$$M = M(Q)\left(1 + \frac{\alpha}{\beta}(1-\rho^{-\beta})\right)^{-\frac{1}{\alpha}}$$

Linearly approximating $1-\rho^{-\beta}$ around $\rho = 1$ yields

$$M \approx M(Q)\left(1+\alpha(\rho-1)\right)^{-\frac{1}{\alpha}}$$

Using the linear approximation of $(1+\alpha(\rho-1))^{-\frac{1}{\alpha}}$ around $\rho = 1$ which is equal to $2-\rho$ and substituting the expression for $\rho$ we get

$$M \approx M(Q)\left(1 - \frac{1}{3(\alpha+\beta)}\left(\frac{Q_N}{Q}\right)^{\frac{1}{\beta}}\right) \tag{12}$$

As expected, the training-inference optimal model size for this scenario is smaller than the training compute optimal model size.

### C.2 Application to GPT-3.5

The total demand from human users over the life span of a model is difficult to predict in advance of training. However, it is possible to conduct the posterior analysis. In this subsection, we propose a rough estimate of the real-world compound demand for a model of a specific quality and use the estimate to evaluate the quality of the approximate analytic solution proposed earlier.

It has been speculated [13] that the GPT-3.5 base model has consumed approximately $T_{\text{GPT-3.5}} = 2.6 \times 10^{24}$ FLOP during training and contained approximately $M_{\text{GPT-3.5}} = 1.75 \times 10^{1}1$ parameters. This allows us to estimate the training dataset size to be around $D_{\text{GPT-3.5}} = 1.5 \times 10^{1}3$ tokens. The quality value of such a model would be around $Q_{\text{GPT-3.5}} = 11.378$.

Since GPT-3.5 remained the free model accessible through the widely popular ChatGPT application up until the introduction of GPT-4o [24], which we consider as the deprecation point for GPT-3.5. Thus, the life span of GPT-3.5 can be estimated to be around 530 days.

In early February 2024, Open AI representatives reported [2] satisfying the daily demand of approximately $10^{11}$ generated tokens. Given that the number of ChatGPT users with paid subscriptions that had access to the only alternative model GPT-4 was around 0.14%, we assume that all of the demand was coming for the GPT-3.5. The API price difference between GPT-3.5 and GPT-4 also likely made GPT-3.5 a favorable choice for the generation of large token quantities. Making a rough assumption that the demand for the GPT-3.5 generation remained constant over its lifespan, we arrive at the estimated figure for the total generation demanded from GPT-3.5 of $N_{\text{GPT-3.5}} \approx 5 \times 10^{13}$ tokens.

After substituting $Q_{\text{GPT-3.5}}$ and $N_{\text{GPT-3.5}}$ into (9) and solving it exactly we get the training-inference optimal model size $M_{\text{tvi}} \approx 1.6 \times 10^{11}$ parameters and dataset size $D_{\text{tvi}} \approx 1.86 \times 10^{13}$ tokens, which are fairly close to the values that could have been used in practice. Using the approximation

(11) by substituting $Q_{N_{\text{GPT-3.5}}} = (N_{\text{GPT-3.5}}\delta)^\beta/\lambda \approx 25.4$ and $Q_{\text{GPT-3.5}}$ into it, we get the estimation $\hat{D}_{\text{tvi}} \approx 2.88 \times 10^{13}$ which is approximately 50% larger than the result of the exact solution $D_{\text{tvi}}$. While not being exact for direct usage, the analytical solution allows us to build the intuition about the training regime appropriate in the multi-stage synthetic generation scenario, further discussed in Section D.

# D Omitting the TCO assumption

In Section B we quantified the optimal model size in the one-stage synthetic data generation setup, which neglects the fact that the target model will be used to generate new synthetic data. The resulting model size was larger than the corresponding TCO model size. In Section C we quantified the optimal model size in the simple training-inference setup, which neglects the fact that the data generated during the inference phase will be used in training of the future model. The resulting model size was smaller than the corresponding TCO model size. Hence, if we take into account both that the target model will be used to generate new synthetic data and that the data generated will be used in training the future model, we expect to end up with the optimal training regime being close to the TCO regime. In this Section, we describe our experiment to check that. We formulate the problem of training a chain of synthesizer models omitting the TCO assumption, solve it numerically, and compare the optimal objective value to the one obtained for the solution of the 2.

To formulate the multistage synthesizer training problem deprived of any prior assumption on the training regime, we define the parameters and the variables of the optimization problem.

Parameters: $n, \alpha, \beta, A, B, Q_0, Q_n, D_0$

Variables: $Q_i$ for $i = 1 \dots n-1$(quality of model $i$);
$M_i$ for $i = 0 \dots n$ (size of model $i$);
$D_i$ for $i = 1 \dots n$ (training data amount for model $i$);
$N_i$ for $i = 0 \dots n-1$ (generated data amount by model $i$);
$\gamma_i$ for $i = 0 \dots n-1$ (training-inference trade-off factor for model $i$).

The objective for the optimization problem is the total amount of compute spent through the stages of training and synthesis:

$$2\sum_{i=0}^{n-1} \gamma_i M_i N_i + 6\sum_{i=1}^{n} M_i D_i$$

The constraint defining model quality can be written through the scaling law expression

$$\frac{A}{M_i^\alpha} + \frac{B}{D_i^\beta} = \frac{1}{Q_i}$$

The data quality condition that appears as $Q(T_i) = Q(f(\gamma_{i-1})T_{i-1})$ in (2) can be written as a constraint in terms of model quality $Q$ and training-inference trade-off factor $\gamma$

$$Q_i = f(\gamma_{i-1})^{\frac{\alpha\beta}{\alpha+\beta}} Q_{i-1}$$

The condition on the training data amount takes the form

$$D_i = D_0 + N_{i-1}$$

which is the last component of the problem formulation. To simplify the problem we make substitutions of the equality conditions into the objective function. The model quality constraint:

$$Q_i = \frac{M_i^\alpha D_i^\beta}{AD_i^\beta + BM_i^\alpha}$$

can be substituted into the data quality constraint

$$\frac{M_i^\alpha D_i^\beta}{AD_i^\beta + BM_i^\alpha} = f(\gamma_{i-1})^{\frac{\alpha\beta}{\alpha+\beta}} \frac{M_{i-1}^\alpha D_{i-1}^\beta}{AD_{i-1}^\beta + BM_{i-1}^\alpha}$$

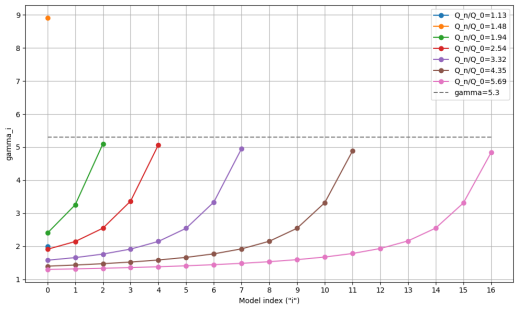
or equivalently

$$\left(\frac{M_i}{M_{i-1}}\right)^{\alpha}\left(\frac{D_i}{D_{i-1}}\right)^{\beta}\frac{AD_{i-1}^{\beta}+BM_{i-1}^{\alpha}}{AD_i^{\beta}+BM_i^{\alpha}}=f(\gamma_{i-1})^{\frac{\alpha\beta}{\alpha+\beta}}$$

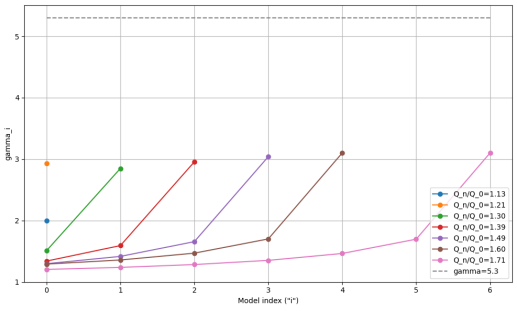
which allows to express the training-inference trade-off factor

$$\left(\frac{M_{i+1}}{M_i}\right)^{\frac{\alpha+\beta}{\beta}}\left(\frac{D_{i+1}}{D_i}\right)^{\frac{\alpha+\beta}{\alpha}}\left(\frac{AD_i^{\beta}+BM_i^{\alpha}}{AD_{i+1}^{\beta}+BM_{i+1}^{\alpha}}\right)^{\frac{\alpha+\beta}{\alpha\beta}}=f(\gamma_i)$$

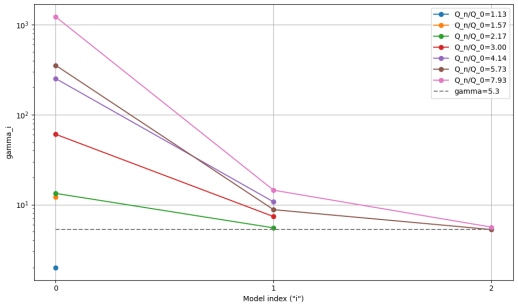
We attempted to numerically solve this optimization problem for specific values of the parameters and compare the obtained solution of (2) in terms of the optimal objective value. While we were not able to recover the TCO regime as the optimal regime of training, the solution of (2) always resulted in a better or equal value of the objective, which we attribute to flatness and non-convexity in the optimization landscape of the full multistage problem, since the optimization algorithm successfully converges to a near-stationary point. Thus, we conclude that the assumption of TCO training regime does not lead to a dramatic increase in the compute costs, which justifies the use of this assumption.



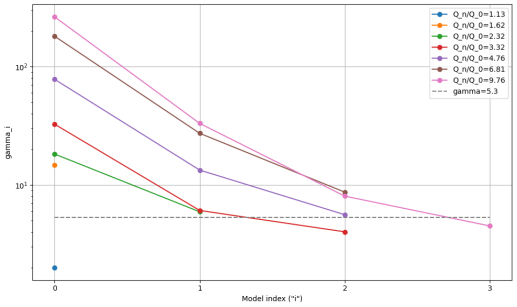

(a) $f(\gamma) = 0.3\gamma + 0.7$



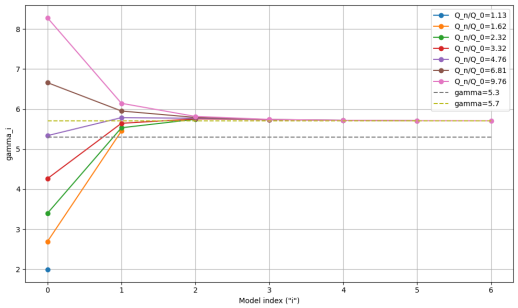

(b) $f(\gamma) = e^{\gamma - 1}$



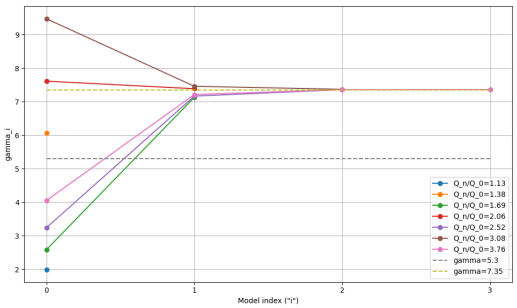

(c) $f(\gamma) = 1.5\gamma - 0.5$



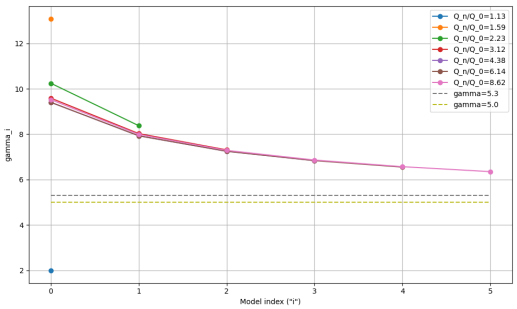

(d) $f(\gamma) = 1 + \log \gamma$




(e) $f(\gamma) = \gamma$




(f) $f(\gamma) = \gamma^{\frac{3}{2}}$




(g) $f(\gamma) = \gamma^{\frac{1}{10}}$

Figure 4: Similarly to Figure 2a, we plot the values $\gamma_i(n^\star(\frac{Q_n}{Q_0}))$ for $i$ from 0 to $n^\star(\frac{Q_n}{Q_0}) - 1$ that correspond to the optimal solution of (3) for different values of the relative quality of the base and the target models $Q_n/Q_0$ and different forms of inference-training trade-off captured by the function $f(\gamma)$.

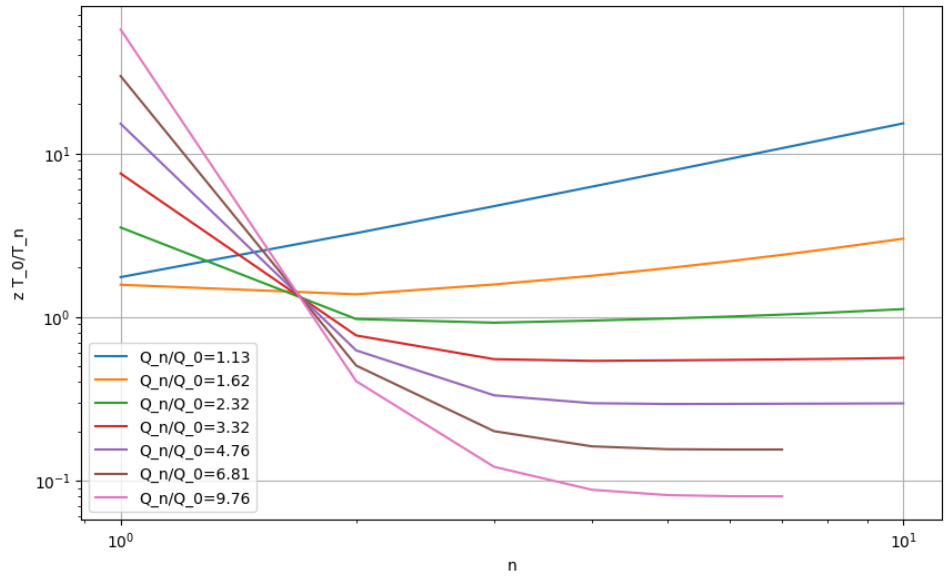
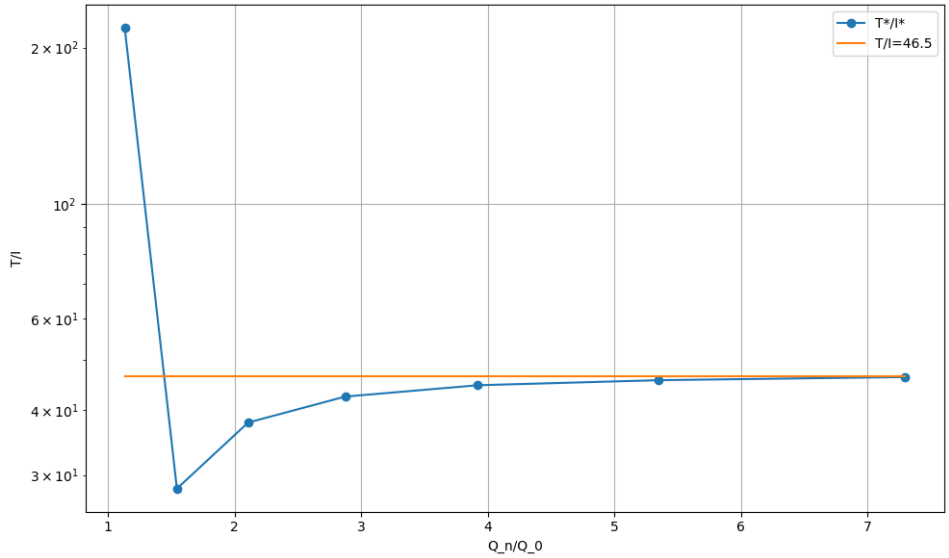
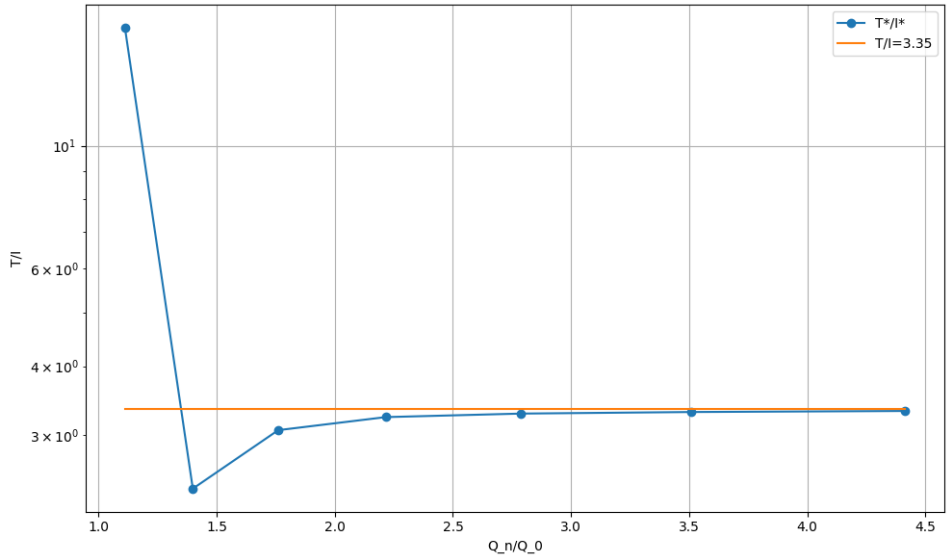



Figure 5: The behavior of $z(n)\left(\frac{Q_0}{Q_n}\right)^{\frac{\alpha+\beta}{\alpha\beta}}$ for different values of $\frac{Q_n}{Q_0}$. The trends and minima positions of $Z(n) = (\lambda Q_0)^{\frac{\alpha+\beta}{\alpha\beta}}\left[\frac{z(n)}{3} + \left(\frac{Q_n}{Q_0}\right)^{\frac{\alpha+\beta}{\alpha\beta}}\right]$ are the same.




(a) Based on the coefficients from Table 1 obtained by [4]




(b) Based on the coefficients from Table 1 obtained by [19]

Figure 6: The ratio of the total amount of training compute to the total amount of inference compute for the synthesizing chain of $n^\star(Q_n/Q_0)$ models as a function of $Q_n/Q_0$, as obtained from the solution of (2) with $f(\gamma) = \gamma^{\frac{\alpha+\beta}{\alpha+2\beta}}$