Reconciling Kaplan and Chinchilla Scaling Laws

Anonymous authors
Paper under double-blind review

Abstract

Kaplan et al. (2020) ('Kaplan') and Hoffmann et al. (2022) ('Chinchilla') studied the scaling behavior of transformers trained on next-token language prediction. These studies produced different estimates for how the number of parameters (N) and training tokens (D) should be set to achieve the lowest possible loss for a given compute budget (C). Kaplan: $N_{\text{optimal}} \propto C^{0.73}$, Chinchilla: $N_{\text{optimal}} \propto C^{0.50}$. This paper finds that much of this discrepancy can be attributed to Kaplan counting non-embedding rather than total parameters, combined with their analysis being performed at small scale. Simulating the Chinchilla study under these conditions produces biased scaling coefficients close to Kaplan's. Hence, this paper reaffirms Chinchilla's scaling coefficients, by explaining the primary cause of Kaplan's original overestimation. As a second contribution, the paper explains differences in the reported relationships between loss and compute. These findings lead us to recommend that future scaling studies use total parameters and compute.

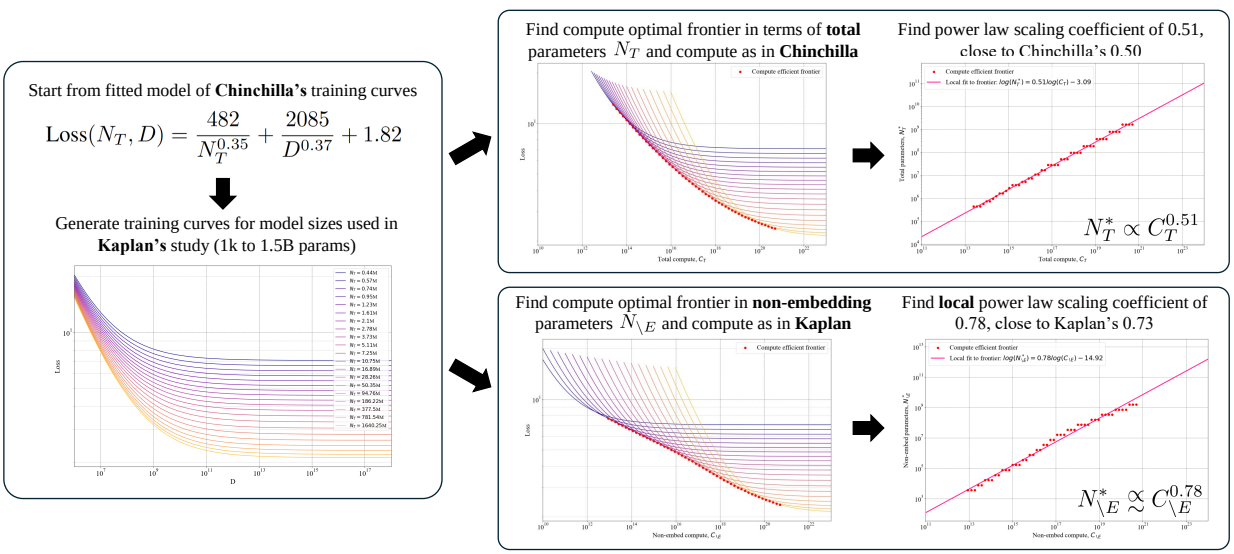

Figure 1: Overview of the approach used to reconcile the two studies.

1 Introduction

Kaplan et al. (2020) ('Kaplan') and Hoffmann et al. (2022) ('Chinchilla') provided two influential studies measuring the impact of scale in large language models (LLMs). Both informed large-scale efforts on how to trade off model parameters (N) and data size (D) for a given compute budget (C), but with conflicting advice. Kaplan's finding that $N_{\text{optimal}} \propto C^{0.73}, D_{\text{optimal}} \propto C^{0.27}$ led to the conclusion that *"big models may be more important than big data"*, and LLMs trained in the ensuing years committed more resources to model size and less to data size. The subsequent Chinchilla study found $N_{\text{optimal}} \propto C^{0.50}, D_{\text{optimal}} \propto C^{0.50}$,

leading to their main thesis *"for many current LLMs, smaller models should have been trained on more tokens to achieve the most performant model"*, sparking a trend towards LLMs of more modest model sizes being trained on more data.

What was the cause of the difference in these scaling coefficient estimates that led to vast amounts of compute (plus emissions and finances) being used inefficiently? There have been suggestions that differences could be explained by different optimization schemes (Hoffmann et al., 2022) or datasets (Bi et al., 2024). This note finds these suggestions incomplete, and offers a simple alternative explanation; much of the discrepancy can be attributed to Kaplan counting non-embedding rather than total parameters, combined with their analysis being performed at small scale.

We find that this methodological difference also plays a part in differences in the reported relationship between compute and loss. Kaplan: $L_{\text{optimal}} \propto C^{0.057}$, Chinchilla: $L_{\text{optimal}} - E \propto C^{0.178}$.

Concretely, this paper offers the following contributions.

- We develop an analytical approach comparing the scaling relationships reported in Kaplan and Chinchilla.
- This approach finds that Kaplan's reported relationship is locally consistent with Chinchilla's, if non-embedding parameters are used, and at smaller scale.
- Secondly, we study the reported relationships between compute and loss. Again non-embedding parameters and smaller scale models are the cause of Kaplan's biased estimate, combined with the absence of an offset term in their compute-loss form.
- We recommend that going forward the scaling community measures total parameters and compute, and uses an offset in the compute-loss relationship.

2 Set Up

Kaplan studied relationships in terms of non-embedding parameters ($N_{\setminus E}$) and non-embedding compute ($C_{\setminus E}$), excluding the linear layers embedding the vocabulary and position indices (N_E). By contrast, Chinchilla studied total parameters (N_T) and total compute (C_T). Define,

$$N_T = N_E + N_{\setminus E}, \tag{1}$$
$$N_E = (h + v)d, \tag{2}$$

where d is dimension of the transformer residual stream, v is vocab size, h is context length (only included when positional embeddings are learned). Using the common approximation for FLOPs $C = 6ND$, we define total and non-embedding compute,

$$C_T = 6N_T D = 6(N_E + N_{\setminus E})D, \tag{3}$$
$$C_{\setminus E} = 6N_{\setminus E} D. \tag{4}$$

The estimated scaling coefficients can be more precisely written (using * to signify 'optimal'),

$$\text{Kaplan:} N^*_{\setminus E} \propto C^{0.73}_{\setminus E}, \tag{5}$$
$$\text{Chinchilla:} N^*_T \propto C^{0.50}_T. \tag{6}$$

Our approach uses information and data from the Chinchilla and Kaplan studies to estimate the scaling laws that would emerge if the Chinchilla relationship had been expressed in terms of $N_{\setminus E}$ & $C_{\setminus E}$, and this had been done at the smaller model sizes used in Kaplan, as summarized in Figure 1.

We will see that for large N_T, N_E becomes a negligible portion of the model's parameters and compute cost. Hence in the large parameter regime the two coefficients directly conflict with each other. At smaller values

of N_T, N_E is *not* negligible (this is the regime considered in Kaplan's study – 768 to 1.5B parameters). We find that at the smaller end of this range, the relationship between $N^*_{\setminus E}$ & $C_{\setminus E}$ is not in fact a power law. However, fitting a "local" power law at this small scale, produces a coefficient that is close to Kaplan's, and hence roughly reconciles these two results.

Our approach in Section 3 is broken down as follows.

- **Step 1.** Fit a suitable function predicting $N_{\setminus E}$ from N_T.
- **Step 2.** Incorporate this function into a model predicting loss in terms of N_T & C_T.
- **Step 3.** Analytically derive the relationship between $N^*_{\setminus E}$ & $C_{\setminus E}$.
- **Step 4.** Simulate synthetic data from the Chinchilla loss model over the model sizes used Kaplan. Fit a local power law for $N^*_{\setminus E}$ in terms of $C_{\setminus E}$.

(Note that whilst this work focuses on the scaling coefficient for parameters, by subscribing to $C = 6ND$ the data coefficient is implied; $N \propto C^a \implies C/D \propto C^a \implies D \propto C^{1-a}$.)

Finally we empirically verify our analysis in Section 4, by training a set of language models at tiny scale and conducting scaling law analyses under various settings. We find that simply changing the basis N_T to $N_{\setminus E}$ produces coefficients inline with Chinchilla and Kaplan respectively, while multiple token budgets and decay schedules does not.

Section 5 presents a second, related contribution. We reconcile differences in the relationship between loss and compute proposed by the two studies. We leverage a similar analysis as above, but Step 3 & 4 are now performed with respect to the relationship between the optimal loss $L^*_{\setminus E}$, and compute $C_{\setminus E}$. So again, we begin from Chinchilla data, and correct for the exclusion of embedding parameters and compute, combined with the smaller model sizes used in Kaplan's study, and additionally a differing choice of fitting function. Through these corrections, we are able to roughly recover Kaplan's compute-loss coefficient, and hence reconcile the two studies.

3 Analysis: Compute-Parameter Scaling Coefficient

This Section presents our main analysis. We show that a *local* scaling coefficient of 0.74 to 0.78 (close to Kaplan's 0.73) can arise when computed in terms of non-embedding parameters in the small-parameter regime, whilst being consistent with Chinchilla's coefficient.

Step 1. *Fit a suitable function predicting $N_{\setminus E}$ from N_T.*

We require a suitable function relating non-embedding and total parameters. We propose to use the form

$$N_T = N_{\setminus E} + \omega N_{\setminus E}^{1/3}, \tag{7}$$

for some constant $\omega > 0$. Aside from having several nice properties (strictly increasing and $\lim_{N_T \to \infty} N_T = N_{\setminus E}$ [1]), it can be motivated from both the Kaplan and Chinchilla study.

Kaplan perspective. Consider Kaplan's method for parameter counting,

$$N_T = 12ld^2 + N_E, \tag{8}$$

where l is number of layers. Whilst Kaplan do not list their model configurations, they do study varying aspect ratio $A = d/l$ for a fixed size model. They find that models of a given size perform similarly over a range of aspect ratios, and range is not affected by model scale (their Figure 5). Hence, we could assume a

[1] **Proof.** $N_T = N_{\setminus E} + \omega N_{\setminus E}^{1/3} \implies N_T/N_{\setminus E} = 1 + \omega N_{\setminus E}^{-2/3}$. Examining the r.h.s., $\lim_{N_T \to \infty} 1 + \omega N_{\setminus E}^{-2/3} = 1$, hence we conclude on the l.h.s $\lim_{N_T \to \infty} N_{\setminus E}/N_T = 1$ or $N_{\setminus E} = N_T$.

sizing scheme with fixed aspect ratio ($A \approx 40$ appears sensible from their plots). Assuming this sizing allows us to state,

$$N_T = \frac{12}{A} d^3 + N_E. \tag{9}$$

Observing that $N_{\setminus E} = (12/A)d^3 \implies d = (N_{\setminus E}(A/12))^{1/3}$, and combining with $N_E = (v+h)d$,

$$N_T = N_{\setminus E} + (v+h)\left(\frac{A}{12}\right)^{1/3} N_{\setminus E}^{1/3}. \tag{10}$$

This is the same form as Eq. 7 with $\omega = (v+h)\left(\frac{A}{12}\right)^{1/3}$.

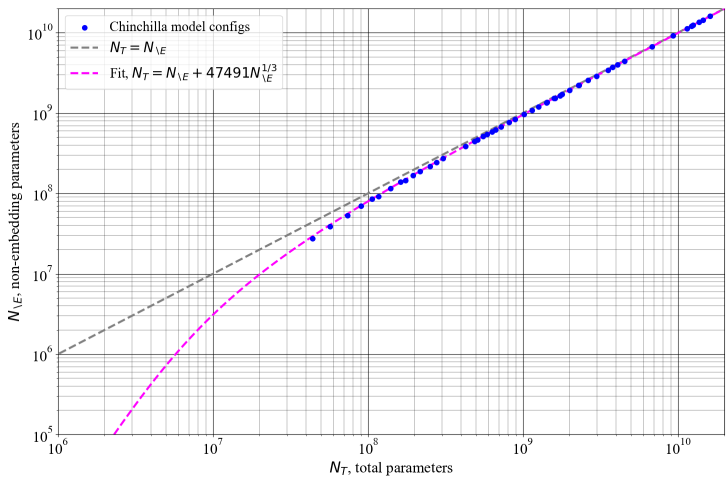

Figure 2: Total parameter count vs. non-embedding parameter count for the suite of models sizes used in the Chinchilla study, along with our fitted approximation. Note the curvature at model sizes below around 200M parameters.

Chinchilla perspective. We empirically fit a function $N_T = N_{\setminus E} + \omega N_{\setminus E}^{\delta}$ (note the learnable exponent) to the Chinchilla model configurations listed in Table A9 of Hoffmann et al. (2022) for a range of N_T (44M to 16B). We compute N_E from Eq. 2, using the reported vocab size of 32,000, but ignore the context length 2,048 since Chinchilla used non-learnable position embeddings (though their inclusion effects coefficients only slightly).

Figure 2 shows the configurations and relationship from a model fitted with numpy's `polyfit`, which produces coefficients, $\omega = 47491$ & $\delta = 0.34$. The exponent has come out close to 1/3, and an implied aspect ratio $A = 39.2$ (inferred from ω). Hence, this further supports the form in Eq. 7.

Step 2. *Incorporate this function into a model predicting loss in terms of N_T & C_T.*

Recall that whilst we are interested in the dependence of N_T^* on C_T, this arises only via their mutual relationship with loss

$$N_T^* = \underset{N_T \text{ s.t. } C_T = 6N_T D}{\operatorname{argmin}} L(N_T, C_T). \tag{11}$$

In order to analytically study their scaling relationship, we need an analytical from of loss. A functional form central to the Chinchilla study is

$$L(N_T, D) = \frac{N_c}{N_T^{\alpha}} + \frac{D_c}{D^{\beta}} + E, \tag{12}$$

$$L(N_T, C_T) = \frac{N_c}{N_T^{\alpha}} + \frac{D_c}{(C_T/6N_T)^{\beta}} + E, \tag{13}$$

where $N_c, D_c, \alpha, \beta > 0$ are constants, D is number of training tokens, E is the "unpredictable" loss of the sequence. By differentiating and setting to zero, then rearranging in terms of N_T we find

$$N_T^* = \left(\frac{\alpha}{\beta}\frac{N_c}{D_c}\right)^{\frac{1}{\alpha+\beta}} \left(\frac{C_T}{6}\right)^{\frac{\beta}{\alpha+\beta}} \quad \text{or simply} \quad N_T^* \propto C^{\frac{\beta}{\alpha+\beta}}. \tag{14}$$

We now modify Eq. 13 to be in terms of non-embedding parameters and compute. Note whilst N_T requires Eq. 7 from step 1, the second term avoids this as $D = C_T/6N_T = C_{\setminus E}/6N_{\setminus E}$.

$$L(N_{\setminus E}, C_{\setminus E}) = \frac{N_c}{(N_{\setminus E} + \omega N_{\setminus E}^{1/3})^\alpha} + \frac{D_c}{(C_{\setminus E}/6N_{\setminus E})^\beta} + E \tag{15}$$

Step 3. *Analytically derive the relationship between $N^*_{\setminus E}$ & $C_{\setminus E}$.*

To find the relationship between $N^*_{\setminus E}$ & $C_{\setminus E}$ we take the derivative of Eq. 15, set to zero and rearrange,

$$6N^*_{\setminus E}\left(N^*_{\setminus E} + \frac{\omega}{3}{N^*_{\setminus E}}^{1/3}\right)^{-\frac{1}{\beta}} \left(N^*_{\setminus E} + \omega {N^*_{\setminus E}}^{1/3}\right)^{\frac{1+\alpha}{\beta}} \left(\frac{\beta}{\alpha}\frac{D_c}{N_c}\right)^{\frac{1}{\beta}} = C_{\setminus E}. \tag{16}$$

This shows that in general the relationship between $N^*_{\setminus E}$ & $C_{\setminus E}$ is *not* a power law. However, we can consider a "local" power law approximation. That is, for some particular value of $N_{\setminus E}$, there is some constant g giving a first order approximation (denoted by $\propto\!\!\!\!\sim$) $N^*_{\setminus E} \propto\!\!\!\!\sim C^g_{\setminus E}$, where g is defined

$$\frac{1}{g} := \frac{d\log(C_{\setminus E})}{d\log(N^*_{\setminus E})} = 1 - \frac{1}{\beta}\frac{{N^*_{\setminus E}}^{2/3} + \frac{\omega}{9}}{{N^*_{\setminus E}}^{2/3} + \frac{\omega}{3}} + \frac{\alpha+1}{\beta}\frac{{N^*_{\setminus E}}^{2/3} + \frac{\omega}{3}}{{N^*_{\setminus E}}^{2/3} + \omega}. \tag{17}$$

Figure 3 plots Eq. 16 and 17, using coefficients for α, β, N_c, D_c from the Epoch AI specification (see later), and $\omega = 47491$. There are three phases.

- At small scale, $\lim_{N^*_{\setminus E}\to 0} \frac{1}{g} = \frac{\alpha/3+\beta}{\beta} \implies N^*_{\setminus E} \propto\!\!\!\!\sim C_{\setminus E}^{\frac{\beta}{\alpha/3+\beta}}$ [2].
- At large scale, $\lim_{N^*_{\setminus E}\to \infty} \frac{1}{g} = \frac{\alpha+\beta}{\beta} \implies N^*_{\setminus E} \propto\!\!\!\!\sim C_{\setminus E}^{\frac{\beta}{\alpha+\beta}}$ [3], as in the N_T case in Eq. 14.
- There is also a transition phase, where g briefly increases. This happens in between the two limits, when $N_E^{2/3}$ is of the same order as ω. Indeed at exactly the point $N_E^{2/3} = \omega$, we have $N_T = N_{\setminus E} + \omega N_{\setminus E}^{1/3} = N_T = 2N_{\setminus E}$, or a 50/50 split between embedding and non-embedding parameters. In Figure 3 we see this transition region occurs around this point; $N_{\setminus E} = \omega^{3/2} = 47491^{3/2} \approx 1 \times 10^7$.

Step 4. *Simulate synthetic data from the Chinchilla loss model over the model sizes used Kaplan. Fit a local power law for $N^*_{\setminus E}$ in terms of $C_{\setminus E}$.*

By reading g off Figure 3, we could estimate a local power law and hence scaling coefficient for a given value of $N^*_{\setminus E}$. However, it's not clear what $N^*_{\setminus E}$ point value is representative of the Kaplan study. We opt for a more faithful estimation procedure, generating synthetic training curves from Eq. 15 across the *range* of model sizes used in Kaplan, and fit coefficients using models falling on the compute efficient frontier. This will also verify our analytic expression for $N^*_{\setminus E}$ & $C_{\setminus E}$ in Eq. 16.

Figure 4 shows the synthetic training curves generated. We simulated 20 models with $N_{\setminus E}$ ranging from 790 parameters to 1.58B (Kaplan reports using model sizes *"ranging in size from 768 to 1.5 billion non-embedding parameters"*). For other constants in Eq. 15, there are two options. Whilst the original Chinchilla study

[2] **Proof.** As $N^*_{\setminus E} \to 0$, we can ignore $N^*_{\setminus E}$ terms and $1/g = 1 - (1/\beta)(3/9) + (\alpha+1)/3\beta = 1 + \alpha/3\beta$.

[3] **Proof.** As $N^*_{\setminus E} \to \infty$, we can ignore ω terms and $1/g = 1 - (1/\beta) + (\alpha+1)/\beta = 1 + \alpha/\beta$.

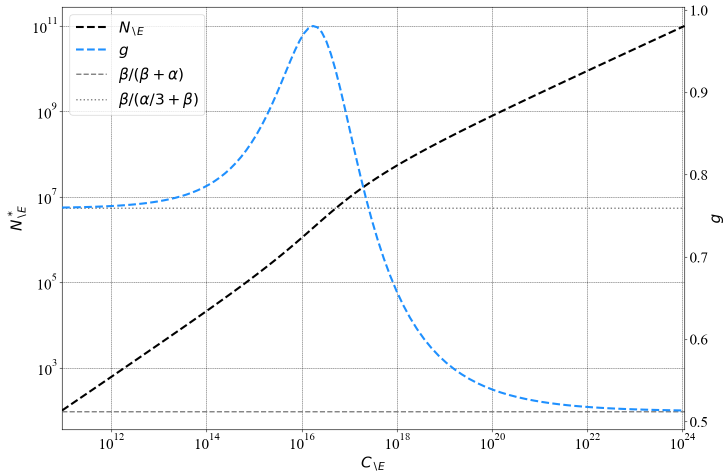

Figure 3: Visualization of Eq. 16 & 17, using the Epoch AI specification.

reported finding constants of $N_c = 406.4, Dc = 410.7, \alpha = 0.3392, \beta = 0.2849, E = 1.6934 \implies N_T^* \propto C_T^{0.46}$ ('Chinchilla specification'), Besiroglu et al. (2024) conducted a re-analysis, arguing that the following set of constants were more accurate $N_c = 482.01, Dc = 2085.43, \alpha = 0.3478, \beta = 0.3658, E = 1.8172 \implies N_T^* \propto C_T^{0.51}$ ('Epoch AI specification'). Our figures adopt the Epoch AI specification and $\omega = 47491$, though we report final results for the Chinchilla specification also.

Main result. Figure 5 shows the estimated scaling coefficient when fitting a power law to the compute optimal frontier (Chinchilla's Method 1) produced by these synthetic training curves. This marks our main result – beginning from a model taken from Chinchilla's study, and modifying two aspects to align with Kaplan's study ($N_T \to N_{\setminus E}$, small model sizes 0.79k – 1.58B parameters), we find *local* scaling coefficients,

$$\text{Epoch AI specification:} N^*_{\setminus E} \underset{\sim}{\propto} C_{\setminus E}^{0.78}, \tag{18}$$

$$\text{Chinchilla specification:} N^*_{\setminus E} \underset{\sim}{\propto} C_{\setminus E}^{0.74}, \tag{19}$$

which are close to the Kaplan coefficient of 0.73.

4 Experiments: Compute-Parameter Scaling Coefficient

We provide brief experiments verifying that our claims hold for models trained at small scale (millions of parameters).

Experiment 1. Firstly, we verify whether scaling coefficients come out close to Chinchilla's and Kaplan's when using N_T and $N_{\setminus E}$ respectively.

We trained five models of sizes, $N_T \in [0.8M, 1.6M, 2.1M, 3.3M, 4.6M]$ on the BookCorpus dataset. We used the GPT-2 tokenizer with vocab size of 50,257, and a context length of 16 (whilst much smaller than typical, our experiments suggest scaling coefficients are not affected by context length). Chinchilla's Method 1 was used to fit scaling coefficients, with the approximation $C = 6ND$.

Models were trained for updates $\in [4000, 4000, 4000, 8000, 8000]$, batchsize was 65,536 tokens per update, for total training tokens $D \in [262M, 262M, 262M, 524M, 524M]$. The best learning rate for each model size was chosen $\in [0.001, 0.005, 0.01, 0.05]$ and no annealing was applied.

Result 1. Table 1 shows that when coefficients are fitted to N_T, we find $N_T \propto C_T^{0.49}$ and for $N_{\setminus E}$, we find $N_{\setminus E} \propto C_{\setminus E}^{0.74}$. These match closely with the Chinchilla and Kaplan coefficients.

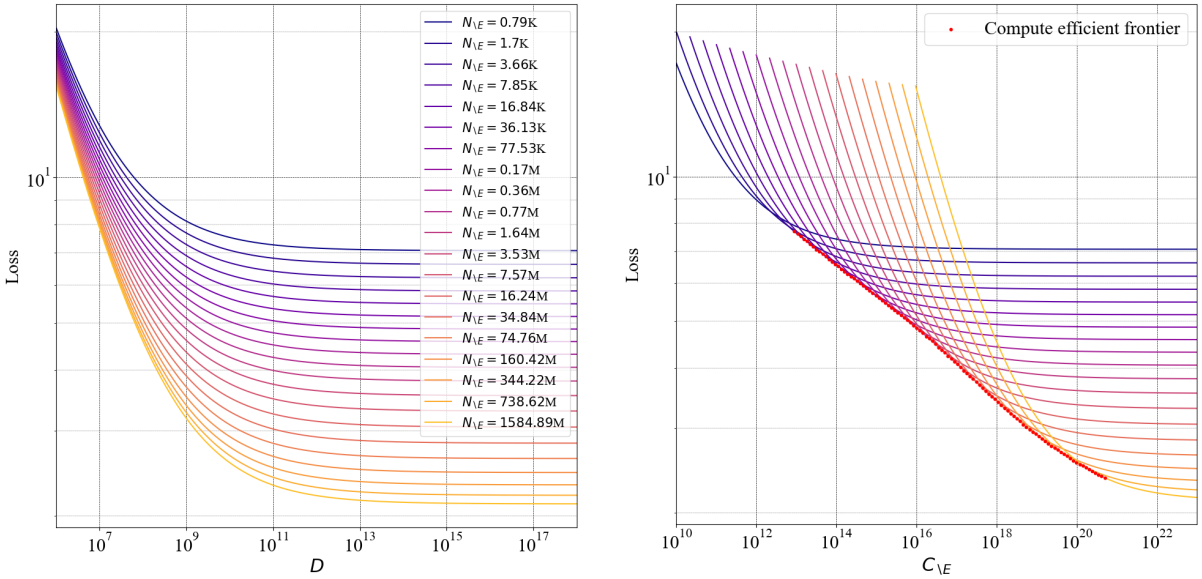

Figure 4: Synthetic training curves from Eq. 15 using the Epoch AI specification, across 20 logarithmically-spaced models matching Kaplan's size range. Left in terms of training tokens, right in terms of compute.

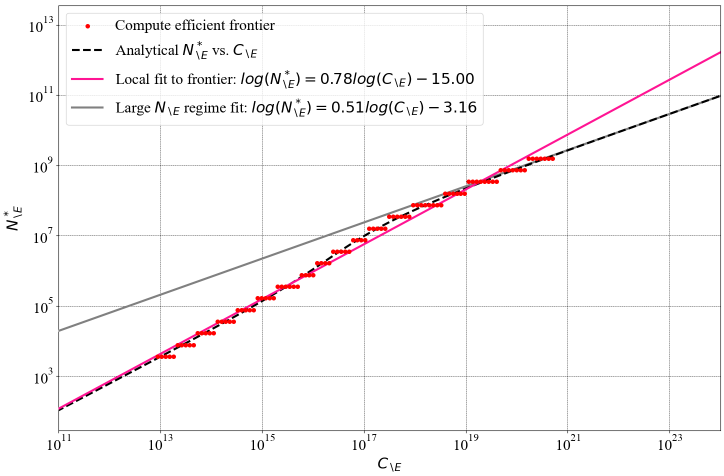

Figure 5: Using synthetic training curves generated in Figure 4, we empirically fit the frontier of compute efficient models using Chinchilla's Method 1. This gives our main result; synthetic training curves generated from Chinchilla's study, produce a local scaling coefficient $N^*_{\setminus E} \propto C^{0.78}_{\setminus E}$, close to Kaplan's coefficient of 0.73. The analytical function from Eq. 16 is also verified.

Experiment 2. We provide an ablation of optimization schemes demonstrating that using multiple training budgets per model affects coefficients only marginally (counter to Chinchilla's explanation).

- **Scheme 1.** A single learning rate of 0.001 is set for all models. A single model trained per size, and no annealing applied.
- **Scheme 2.** The best learning rate is chosen per model. A single model trained per size, and no annealing applied. (As in our N_T vs. $N_{\setminus E}$ comparison.)

- **Scheme 3.** The best learning rate is chosen per model. A single model trained per size, and cosine annealing applied at the update budget. (**Kaplan study used this.**)
- **Scheme 4.** The best learning rate is chosen per model. Six models trained per size at different budgets $\in [0.25D, 0.5D, 0.75D, 1.0D, 1.5D, 2.0D]$, and cosine annealing applied. (**Chinchilla study used this.**)

Result 2. Table 1 shows that optimization scheme has a smaller impact on scaling coefficients than switching from N_T to $N_{\setminus E}$. Using a single set of models with no annealing (scheme 2) produces the same coefficients as using the more computationally expensive scheme 4. Counter to Chinchilla's comment that moving from Kaplan's scheme 3 to scheme 4 would reduce the scaling coefficient, our experiment suggests the opposite is the case, increasing from 0.46 to 0.49. This might explain our slight overestimation of the scaling coefficients in Eq. 28 & 29.

Table 1: Comparison of different scaling coefficients from our experiments. Note that the change moving from N_T to $N_{\setminus E}$ has a much larger effect than moving between optimization schemes.

Experiment	a where $N_{\text{optimal}} \propto C^a$	b where $D_{\text{optimal}} \propto C^b$
Chinchilla, N_T	0.50	0.50
Kaplan, $N_{\setminus E}$	0.73	0.27
Ablating N_T vs $N_{\setminus E}$		
Ours, N_T & C_T	0.49	0.51
Ours, $N_{\setminus E}$ & $C_{\setminus E}$	0.74	0.26
Ablating optimization scheme		
Ours, N_T, scheme 1, single lrate, no anneal	0.58	0.42
Ours, N_T, scheme 2, best lrate, no anneal	0.49	0.51
Ours, N_T, scheme 3, best lrate, single-cosine anneal	0.46	0.54
Ours, N_T, scheme 4, best lrate, multi-cosine anneal	0.49	0.51

5 Analysis: Compute-Loss Scaling Coefficient

As well as using their analysis to calculate the compute-optimal parameter coefficient, Kaplan and Chinchilla also described the relationship between compute and loss, assuming parameters were scaled optimally. This optimal loss was again given in terms of non-embedding compute by Kaplan, and total compute by Chinchilla,

$$L^*_{\setminus E} = \min_{\text{s.t. } C_{\setminus E}=6N_{\setminus E}D} L(N_{\setminus E}, C_{\setminus E}), \tag{20}$$

$$L^*_T = \min_{\text{s.t. } C_T=6N_T D} L(N_T, C_T). \tag{21}$$

Concretely, the two studies reported the following forms and coefficients linking optimal loss and compute.

$$\text{Kaplan compute-loss form: } L^*_{\setminus E} = \left(\frac{C_{\setminus E}}{C_0}\right)^{-\gamma} \tag{22}$$

$$\text{Kaplan compute-loss fit: } L^*_{\setminus E} \propto C_{\setminus E}^{0.057} \tag{23}$$

$$\text{Chinchilla compute-loss form: } L^*_T = \left(\frac{C_T}{C_0}\right)^{-\gamma} + E \tag{24}$$

$$\text{Chinchilla compute-loss fit, Epoch AI spec: } L^*_T - E \propto C_T^{0.178} \tag{25}$$

$$\text{Chinchilla spec: } L^*_T - E \propto C_T^{0.155} \tag{26}$$

(See Section A.2 for Chinchilla's compute coefficient.) Similar to the compute-parameter scaling coefficient, on the surface Kaplan's coefficient of 0.057 appears quite far from Chinchilla's of 0.155–0.178. However,

we will again show that by beginning from the Chinchilla study, and adjusting for Kaplan's non-embedding compute, smaller scale, and additionally their compute-loss form, these two coefficients can be roughly reconciled.

Our analysis follows the four-step approach from Section 3. We can directly reuse Step 1 & 2, while Step 3 & 4 are now modified to study the relationship between optimal loss and compute, rather than optimal parameters and compute as previously.

Step 3. *Analytically derive the relationship between $L^*_{\setminus E}$ & $C_{\setminus E}$.*

We find that the relationship between $L^*_{\setminus E}$ & $C_{\setminus E}$ is not a power law (derived in Section A.1).

$$\frac{d\log(L^*_{\setminus E})}{d\log(C_{\setminus E})} = \frac{g}{L^*_{\setminus E}}\left[-\alpha N_c \frac{\left(N^*_{\setminus E} + \frac{1}{3}\omega\left(N^*_{\setminus E}\right)^{1/3}\right)}{\left(N^*_{\setminus E} + \omega\left(N^*_{\setminus E}\right)^{1/3}\right)^{\alpha+1}} + \beta D_c\left(\frac{C_{\setminus E}}{6N^*_{\setminus E}}\right)^{-\beta}\left(1-\frac{1}{g}\right)\right]. \tag{27}$$

However, again we can consider a local first-order approximation, $L^*_{\setminus E} \underset{\sim}{\propto} C^k_{\setminus E}$, where $k = \frac{d\log(L^*_{\setminus E})}{d\log(C_{\setminus E})}$. We visualize this in Figure 6.

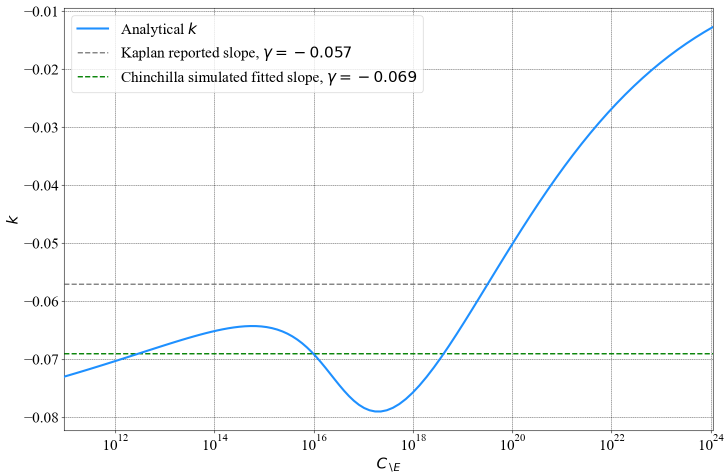

Figure 6: Visualization of Eq. 27, using the Epoch AI specification.

Step 4. *Simulate synthetic data from the Chinchilla loss model over the model sizes used Kaplan. Fit a local power law, for $L^*_{\setminus E}$ in terms of $C_{\setminus E}$, using Kaplan's compute-loss form.*

As in Section 3, we *could* use Eq. 27 to compute a point estimate for k in the relationship $L^*_{\setminus E} \underset{\sim}{\propto} C^k_{\setminus E}$. However, again we opt for the more faithful procedure of simulating data from the loss curves. We then fit Kaplan's compute-loss form $L^*_{\setminus E} = \left(\frac{C_{\setminus E}}{C_0}\right)^{-\gamma}$. For the two specifications, these give the following models,

$$\text{Epoch AI specification:} L^*_{\setminus E} \underset{\sim}{\propto} C^{-0.069}_{\setminus E}, \tag{28}$$

$$\text{Chinchilla specification:} L^*_{\setminus E} \underset{\sim}{\propto} C^{-0.066}_{\setminus E}, \tag{29}$$

which are roughly inline with Kaplan's reported coefficient of $L^*_{\setminus E} \underset{\sim}{\propto} C^{-0.057}_{\setminus E}$.

Figure 7 visualizes the fit of the Chinchilla compute-loss form vs. the Kaplan compute-loss form, when one counts total compute vs. non-embedding compute. We see that Kaplan's form provides a good fit of the data in the non-embedding compute plot, over the range of model sizes they considered. We speculate that this might be the motivation for Kaplan's selection of this simpler compute-loss form.

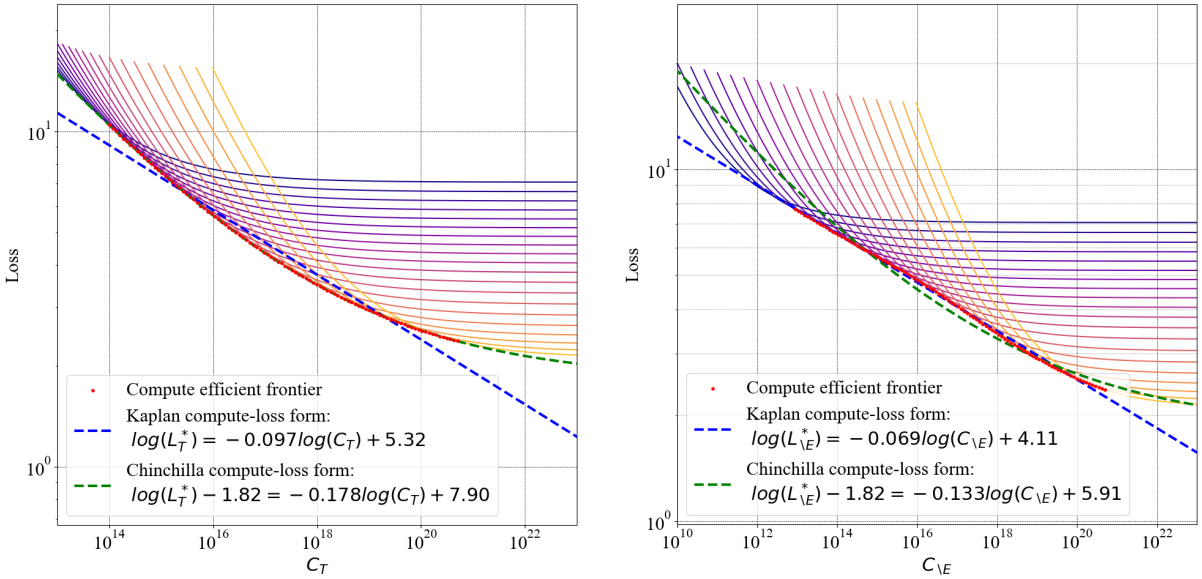

Figure 7: Synthetic data with the Epoch AI specification, as in Figure 4. Here we visualize the quality of the fits using both the Kaplan and Chinchilla compute-loss forms. Notice that Chinchilla's form provides a better fit under total compute, while Kaplan's form is a better match under non-embedding compute.

6 Related work

Following early works formalizing how language models improve with parameters, data, and training compute (Rosenfeld et al., 2019; Kaplan et al., 2020; Hoffmann et al., 2022), there has been investigation into whether these scaling laws arise in other domains (Henighan et al., 2020), and to explain their existence from a theoretical standpoint (Hutter, 2021; Maloney et al., 2022; Bahri et al., 2024).

Closer in spirit to our paper, are several concurrent works that have investigated various design decisions in scaling law analyses. Su et al. (2024) revisit the methodology used to find scaling coefficients. Hägele et al. (2024) found that multiple independent cosine schedules could be reproduced more efficiently through a constant learning rate with multiple short decays, or stochastic weight averaging. Our finding is subtly different; a simple fixed learning rate will recover very similar compute-parameter scaling coefficients as multiple cosine schedules. Bi et al. (2024) study the effect of various hyperparameters on scaling. They observe that different text datasets produce slightly different optimal coefficients, with 'cleaner' data leading to more parameter-hungry scaling behavior, which they speculate could partially explain the difference between Kaplan and Chinchilla coefficients.

Porian et al. (2024) provide a concurrent work with the same objective as our paper – explaining the differences between the Kaplan and Chinchilla coefficients. Through a set of large-scale experiments reproducing Kaplan's study, they determine that responsibility for the discrepancy can be attributed, in decreasing order of significance, to; 1) Kaplan counting non-embedding rather than total compute. 2) Kaplan using a fixed-length warmup period that was too long for smaller models, making them appear less efficient. 3) Kaplan not fully tuning optimization hyperparameters. We see these findings as complimentary to our own. We have been able to identify the primary 'first-order' reason using *only* information that was publicly available in the two papers, with a fully analytical approach. (Tiny-scale experiments were run post-hoc as verification.) This illustrates the promise of applying mathematical approaches to the empirical science of scaling.

7 Discussion

This note aimed to explain the difference between the Kaplan and Chinchilla scaling coefficients. We found two issues in Kaplan's study that combined to bias their estimated scaling coefficients; their choice to count only non-embedding parameters, and studying smaller sized model sizes. This means there is curvature in the true relationship between $N_{\setminus E}$ & N_T (Figure 5). At larger values of N_T, the embedding parameter counts become negligible, $N_T = N_{\setminus E}$, and differences would not arise. Alternatively, had Kaplan studied relationships directly in terms of N_T, this issue would also not arise, even at this smaller scale (confirmed by our Experiment 1 finding $N_T \propto C_T^{0.49}$ even for $N_T < 5M$). The form Kaplan used to predict loss from compute further contributed to differences in the reported compute-loss scaling coefficients.

Inconsistency across scaling studies. Existing literature on scaling is not consistent in its use of non-embedding vs. total compute. Some studies (Henighan et al., 2020; Gordon et al., 2021; Ghorbani et al., 2021; Fernandes et al., 2023; Hu et al., 2024; Bi et al., 2024; Su et al., 2024) follow Kaplan's approach, using non-embedding parameters or compute, while others (Clark et al., 2022; Que et al., 2024; Wang et al., 2023) adhere to the Chinchilla approach, using total parameters. Our work indicates that this choice can substantially alter scaling exponents, complicating cross-study comparisons. Similarly, the choice of compute-loss equation varies through the literature. Studies such as (Clark et al., 2022; Brown et al., 2020; Smith and Doe, 2024) opt for the Kaplan compute-loss form without offsets. In contrast, (Henighan et al., 2020; Gordon et al., 2021; Fernandes et al., 2023; Hu et al., 2024; Wang et al., 2023) employ the Chinchilla compute-loss form with non-zero offsets. Again, our work suggests that these methodological differences can lead to significant variations in scaling predictions and interpretations.

The lack of a standardized approach in scaling studies risks making comparisons misleading and insights less clear. We see our work as helping to understand certain decisions made in previous studies that should be standardized. Concretely, we advise future studies to report total, rather than non-embedding, parameters, and to include an offset in the compute-loss fitting models. We discuss motivation for these choices below. Furthermore, our initial evidence does not support using multiple cosine decays per model size – we find a single fixed learning rate per model size is sufficient.

Why should embedding parameters contribute to scaling behavior? Several works evidence that embedding parameters capture meaningful language properties. Word embeddings can be factorized into semantically interpretable factors (even the shallow Word2vec) (Mikolov et al., 2013a;b; Arora et al., 2018). LLMs learn linear embeddings of space and time across scales (Gurnee and Tegmark, 2024). Developing such meaningful embedding structures allows LLMs to perform high-level language operations, such as arithmetic (McLeish et al., 2024). Therefore, if one believes that the embedding layer does more than just 'translate' tokens to a vector of the correct dimension, we see no reason to exclude them in the parameter count.

Why should a non-zero offset be used in loss-compute predictions? The Chinchilla compute-loss form with a non-zero offset (Eq. 25), is a more appropriate form from the perspective of statistical learning. This approach accounts for the concept of irreducible risk (Hernandez et al., 2023), which posits a lower bound on achievable loss regardless of model or dataset size. This may arise from various factors: inherent biases or limitations in the learning algorithm, or noise in the original task. As a concrete example in language modeling, the best a model can do for the prediction of the first token in a sequence, is to estimate the marginal distribution of all tokens, which leads to a non-zero loss.

Limitations. We acknowledge several limitations of our analysis. We have aimed to capture the primary 'first order' reason for the difference between the Kaplan and Chinchilla scaling coefficients. But there are multiple other differences between the two studies that likely also affect scaling coefficients (Section 6); datasets (Kaplan used OpenWebText2, Chinchilla used MassiveText), transformer details (Kaplan used learnable position embeddings while Chinchilla's were fixed, also differing tokenizers, vocabularly sizes), optimization scheme (Kaplan used scheme 3, Chinchilla scheme 4, also differing warmup schedules), differences in computation counting (Kaplan used $C = 6ND$, Chinchilla's Method 1 & 2 used a full calculation). However, our work suggested these factors impact coefficients in a more minor way.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018. doi: 10.1162/tacl_a_00034. URL https://aclanthology.org/Q18-1034.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024.

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

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

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

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. In *International conference on machine learning*, pages 4057–4086. PMLR, 2022.

Patrick Fernandes, Behrooz Ghorbani, Xavier Garcia, Markus Freitag, and Orhan Firat. Scaling laws for multilingual neural machine translation. In *International Conference on Machine Learning*, pages 10053–10071. PMLR, 2023.

Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. Scaling laws for neural machine translation. *arXiv preprint arXiv:2109.07740*, 2021.

Mitchell A Gordon, Kevin Duh, and Jared Kaplan. Data and parameter scaling laws for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5915–5922, 2021.

Wes Gurnee and Max Tegmark. Language models represent space and time, 2024.

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. *arXiv preprint arXiv:2405.18392*, 2024.

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

Danny Hernandez, Jared Kaplan, Tom Henighan, Sam McCandlish, Tom B. Brown, Andrew Carr, Dieterich Lawson, Jacob Steinhardt, Chris Olah, and Dario Amodei. Scaling laws for neural language models. *Proceedings of the National Academy of Sciences*, 120(26), 2023. doi: 10.1073/pnas.2311878121. URL https://www.pnas.org/doi/10.1073/pnas.2311878121.

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.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

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

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

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

Sean McLeish, Arpit Bansal, Alex Stein, Neel Jain, John Kirchenbauer, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, Jonas Geiping, Avi Schwarzschild, and Tom Goldstein. Transformers can do arithmetic with the right embeddings, 2024.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013a. URL `https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf`.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013b. Association for Computational Linguistics. URL `https://aclanthology.org/N13-1090`.

Tomer Porian, Mitchell Wortsman, Jenia Jitsev, Ludwig Schmidt, and Yair Carmon. Resolving discrepancies in compute-optimal scaling of language models. *arXiv preprint arXiv:2406.19146*, 2024.

Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, et al. D-cpt law: Domain-specific continual pre-training scaling law for large language models. *arXiv preprint arXiv:2406.01375*, 2024.

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

J. Smith and A. Doe. The role of xyz in abc. *bioRxiv*, 2024. doi: 10.1101/2024.06.06.597716v1. URL `https://www.biorxiv.org/content/10.1101/2024.06.06.597716v1.full.pdf`.

Hui Su, Zhi Tian, Xiaoyu Shen, and Xunliang Cai. Unraveling the mystery of scaling laws: Part i. *arXiv preprint arXiv:2403.06563*, 2024.

Peihao Wang, Rameswar Panda, and Zhangyang Wang. Data efficient neural scaling law via model reusing. In *International Conference on Machine Learning*, pages 36193–36204. PMLR, 2023.

A Appendix

A.1 Derivation of compute-loss analytical form

This section derives k, defined as,

$$k = \frac{d\log(L^*_{\setminus E})}{d\log(C_{\setminus E})}. \tag{30}$$

Expanding with the chain rule we find,

$$k = \frac{d\log(L^*_{\setminus E})}{dL^*_{\setminus E}} \frac{dL^*_{\setminus E}}{dN^*_{\setminus E}} \frac{dN^*_{\setminus E}}{d\log(N^*_{\setminus E})} \frac{d\log(N^*_{\setminus E})}{d\log(C_{\setminus E})}, \tag{31}$$

$$= \frac{N^*_{\setminus E}}{L^*_{\setminus E}} \frac{dL^*_{\setminus E}}{dN^*_{\setminus E}} g, \tag{32}$$

where we previously derived $g = \frac{d\log(N^*_{\setminus E})}{d\log(C_{\setminus E})}$ in Eq. 17.

This leaves us with $\frac{dL^*_{\setminus E}}{dN^*_{\setminus E}}$ to find. First note that $L^*_{\setminus E}$ is given by Eq. 15 when the optimal model size is used, i.e., $N_{\setminus E} \leftarrow N^*_{\setminus E}$,

$$L^*_{\setminus E} = \frac{N_c}{(N^*_{\setminus E} + \omega {N^*_{\setminus E}}^{1/3})^\alpha} + \frac{D_c}{(C_{\setminus E}/6N^*_{\setminus E})^\beta} + E. \tag{33}$$

Before taking this derivative, we recall that $C_{\setminus E}$ is actually a function of $N^*_{\setminus E}$ (via Eq. 16). Hence, we tackle the derivative in two parts. We find the first term derivative is equal to,

$$\frac{d\frac{N_c}{(N^*_{\setminus E} + \omega {N^*_{\setminus E}}^{1/3})^\alpha}}{dN^*_{\setminus E}} = -\alpha N_c \frac{\left(1 + \frac{1}{3}\omega \left(N^*_{\setminus E}\right)^{-2/3}\right)}{\left(N^*_{\setminus E} + \omega \left(N^*_{\setminus E}\right)^{1/3}\right)^{\alpha+1}}. \tag{34}$$

The derivative of the second term, via the product rule, and spotting that $\frac{dC_{\setminus E}}{dN^*_{\setminus E}} = \frac{C_{\setminus E}}{gN^*_{\setminus E}}$, equals,

$$\frac{d\frac{D_c}{(C_{\setminus E}/6N^*_{\setminus E})^\beta}}{dN^*_{\setminus E}} = \beta D_c \frac{1}{N^*_{\setminus E}} \left(\frac{C_{\setminus E}}{6N^*_{\setminus E}}\right)^{-\beta} - \beta D_c \frac{1}{gN^*_{\setminus E}} \left(\frac{C_{\setminus E}}{6N^*_{\setminus E}}\right)^{-\beta} \tag{35}$$

$$= \beta D_c \frac{1}{N^*_{\setminus E}} \left(\frac{C_{\setminus E}}{6N^*_{\setminus E}}\right)^{-\beta} \left(1 - \frac{1}{g}\right). \tag{36}$$

Hence, combining these two terms we find,

$$\frac{dL^*_{\setminus E}}{dN^*_{\setminus E}} = -\alpha N_c \frac{\left(1 + \frac{1}{3}\omega \left(N^*_{\setminus E}\right)^{-2/3}\right)}{\left(N^*_{\setminus E} + \omega \left(N^*_{\setminus E}\right)^{1/3}\right)^{\alpha+1}} + \beta D_c \frac{1}{N^*_{\setminus E}} \left(\frac{C_{\setminus E}}{6N^*_{\setminus E}}\right)^{-\beta} \left(1 - \frac{1}{g}\right). \tag{37}$$

Combining this result into to Eq. 32 we get,

$$k = g \frac{N^*_{\setminus E}}{L^*_{\setminus E}} \frac{dL^*_{\setminus E}}{dN^*_{\setminus E}} \tag{38}$$

$$= \frac{g}{L^*_{\setminus E}} \left[-\alpha N_c \frac{\left(N^*_{\setminus E} + \frac{1}{3}\omega \left(N^*_{\setminus E}\right)^{1/3}\right)}{\left(N^*_{\setminus E} + \omega \left(N^*_{\setminus E}\right)^{1/3}\right)^{\alpha+1}} + \beta D_c \left(\frac{C_{\setminus E}}{6N^*_{\setminus E}}\right)^{-\beta} \left(1 - \frac{1}{g}\right) \right]. \tag{39}$$

A.2 Compute-loss coefficient derivation

We know from Eq. 14 $N_T^* \propto C^{\frac{\beta}{\alpha+\beta}}$, and similarly $D_T^* \propto C^{\frac{\alpha}{\alpha+\beta}}$. Substituting these into the loss form of Eq. 12, and for some new constants $\bar{N}_c, \bar{D}_c$ we find,

$$L_T^* = \frac{N_c}{{N_T^*}^{\alpha}} + \frac{D_c}{{D_T^*}^{\beta}} + E, \tag{40}$$

$$= \frac{\bar{N}_c}{C^{\frac{\alpha\beta}{\alpha+\beta}}} + \frac{\bar{D}_c}{C^{\frac{\alpha\beta}{\alpha+\beta}}} + E, \tag{41}$$

$$L_T^* - E \propto C^{\frac{-\alpha\beta}{(\alpha+\beta)}}. \tag{42}$$