Harmonizing Scaling Laws: Bridging the Gap Between Kaplan and Chinchilla

Abstract

Studies by Kaplan et al. (2020) and Hoffmann et al. (2022) examined the scaling characteristics of transformers in next-token language prediction, yielding different recommendations for configuring the number of parameters (N) and training tokens (D) to minimize loss within a set compute budget (C). Kaplan suggested an optimal parameter count scaling with Noptimal $\propto C^{0.73}$, whereas Chinchilla proposed Noptimal $\propto C^{0.50}$. This paper demonstrates that a significant portion of this difference can be traced back to Kaplan's focus on non-embedding parameters, rather than the total parameter count, along with their study's concentration on a smaller scale. When the Chinchilla study is simulated under similar circumstances, biased scaling coefficients similar to those of Kaplan are produced. As a result, this work confirms Chinchilla's scaling coefficients by clarifying the primary reason for Kaplan's initial overestimation. Additionally, this research clarifies variations in the stated correlations between computational loss and budget. As a result of these findings, we advocate for upcoming scaling investigations to utilize total parameter counts and overall computational resources.

1 Introduction

Two important studies by Kaplan et al. (2020) and Hoffmann et al. (2022) examined how scale affects large language models (LLMs). Both studies provided advice on how to balance model parameters (N) and training tokens (D) for a fixed computing budget (C), but their suggestions conflicted. The conclusion drawn from Kaplan's discovery that Noptimal $\propto C^{0.73}$ and Doptimal $\propto C^{0.27}$ was that "large models might be more crucial than extensive data." Subsequently, LLMs trained in the following years allocated more resources to model size and less to data size. The Chinchilla research that came after that discovered that Noptimal $\propto C^{0.50}$ and Doptimal $\propto C^{0.50}$, which resulted in their main argument that "for many current LLMs, smaller models should have been trained on more tokens to achieve the most performant model." This sparked a trend in which LLMs with smaller model sizes were trained using more data.

What caused the discrepancy in these scaling coefficient estimates, which resulted in a significant waste of computer resources, emissions, and money? There have been theories suggesting that variations in optimization techniques or datasets might account for the differences. This paper argues that these explanations are insufficient and proposes a straightforward substitute: the majority of the discrepancy is caused by Kaplan's decision to count non-embedding parameters instead of total parameters, together with the limited scale of their investigation.

Additionally, it is discovered that this methodological discrepancy contributes to variations in the stated correlation between loss and compute.

Specifically, this research provides the following:

- An analytical method is created to assess the scaling relationships described in the studies (Section 3). If non-embedding parameters are utilized, and at a smaller scale, this method demonstrates that Kaplan's documented relationship is locally compatible with Chinchilla's.

.

- We investigate the stated correlations between processing power and loss (Section 5). Once more, the cause of Kaplan's skewed estimate is the use of non-embedding parameters and smaller scale models, together with the lack of an offset term in their compute-loss equation.
- It is suggested that the scaling community use total parameters, total compute, and an offset in the compute-loss equation going forward.

2 Preliminaries

This section provides some foundational information and definitions (Section 2.1), summarizes the analytical method used for our primary finding (Section 2.2), and documents our assumptions (Section 2.3).

2.1 Set Up

Kaplan et al. (2020) and Hoffmann et al. (2022) conducted empirical studies to model the relationships between the number of parameters (N), training tokens (D), training compute (C), and loss (L) in transformers used for language modeling. The primary functional relationship explored was a power law, y = axb, which is frequently employed in various scientific fields to illustrate the connection between two quantities (x and y) that span multiple orders of magnitude.

The two studies differed in their definitions of N and C. Kaplan investigated relationships regarding non-embedding parameters (N
E) and non-embedding compute (C
E), excluding contributions from embedding layers for vocabulary and position indices (NE). In contrast, Chinchilla studied total parameters (NT) and total compute (CT). We define,

$$NT = NE + NE, \tag{1}$$
$$NE = (h + v)d, \tag{2}$$

where d represents the transformer residual stream's dimension, v denotes the vocabulary size, and h stands for the context length (included only when positional embeddings are learned). Utilizing the typical approximation for training compute FLOPs C = 6ND (where a factor of 6 accounts for a forward and backward pass), we establish total and non-embedding compute as:

$$CT = 6NTD = 6(NE + NE)D, \tag{3}$$
$$CE = 6NED. \tag{4}$$

The definition of compute, C = 6ND, indicates a direct trade-off between the number of parameters and training tokens for a specified compute budget. The focus of the two research studies is on "compute optimal" configurations, which are the parameter and token combinations that result in the lowest loss for a given compute budget. This is expressed as follows for total parameters (using $\star$ to denote "optimal"):

$$NT = argminL(NT, CT). \tag{5}$$

Subject to:

$$CT = 6NTD \tag{6}$$

With this notation, the estimated scaling coefficients can be written more precisely as:

$$Kaplan : NEC0.73E, Chinchilla : NTC0.50T. \tag{7}$$

(It should be noted that although this study concentrates on the scaling coefficient for parameters, the data coefficient is inferred by subscribing to C = 6ND; N $\propto$ C^a $\rightarrow$ C/D $\propto$ C^a $\rightarrow$ D $\propto$ C^{1-a}.)

An important functional form relating NT, D, and L, as used in the Chinchilla study, is:

$$L(NT, D) = NcNT + DcD\theta 3b2 + E, \tag{8}$$

where Nc, Dc, α, $\beta > 0$ are empirically determined constants, and E represents the irreducible loss inherent in language. This equation conveniently generates power-law relationships: N T $\propto$ C^a T with a = β / (α + β), D T $\propto$ C^b T with b = α / (α + β), and L T - E $\propto$ C$^{-\gamma}$ T with γ = $\alpha\beta$ / (α + β).

There are two possible specifications based on the constants in Equation 8: those originally reported in the Chinchilla study and those from a re-analysis by Besiroglu et al. (2024), which claims to correct minor errors in the fitting procedure. Our work presents results using both specifications.

$$Chinchilla : Nc = 406.4, Dc = 410.7, = 0.3392, \Phi 3b2 = 0.2849, E = 1.693 = \Phi 21d2NTC0.46T, \tag{9}$$

$$EpochAI : Nc = 482.0, Dc = 2085.43, = 0.3478, \Phi 3b2 = 0.3658, E = 1.817 = \Phi 21d2NTC0.51T. \tag{10}$$

2.2 Analysis Overview

In our analysis, we use data and insights from the Chinchilla and Kaplan studies to predict the scaling laws that would result if the Chinchilla relationship were stated in terms of N
E and C
E, and this was done using the smaller model sizes used in Kaplan's study.

It will be demonstrated that when NT is large, N
E becomes an insignificant component of the model's parameters and computing cost. As a result, the two coefficients are in direct opposition to one another in the large parameter regime. The embedding parameters are not insignificant when NT is smaller (this is the regime examined in Kaplan's study, which used parameters ranging from 768 to 1.5B). We discover that the relationship between N
E and C
E is not, in fact, a power law at the lower end of this range. However, fitting a "local" power law at this modest scale yields a coefficient that is comparable to Kaplan's, roughly reconciling these two findings.

Our approach in Section 3 is broken down as follows:

- Step 1. Fit a suitable function predicting N
  E from NT.
- Step 2. Incorporate this function into a model predicting loss in terms of NT and CT.
- Step 3. Analytically derive the relationship between N
  E and C
  E.
- Step 4. Simulate synthetic data from the Chinchilla loss model over the model sizes used in the Kaplan study. Fit a local power law for N
  E in terms of C
  E.

Section 4 provides experimental validation of our analysis by training a set of language models at a very small scale and examining scaling laws under different settings. Simply changing the basis from NT to N
E yields coefficients consistent with Chinchilla and Kaplan, respectively, while varying token budgets and decay schedules does not.

A second, connected contribution is made in Section 5. The two studies' suggested relationships between loss and computation are reconciled by us. In order to examine the relationship between the ideal loss L
E and compute C
E, Steps 3 and 4 are carried out once more using a similar analysis as before. To do this, we start with Chinchilla data and adjust for the smaller model sizes utilized in Kaplan's investigation, the exclusion of embedding parameters and compute, and a different fitting function option. We are able to roughly recover Kaplan's compute-loss coefficient and reconcile the two studies by making these adjustments.

2.3 Assumptions

For transparency, we list the assumptions and approximations made in our analysis.

- We assume C
  E = 6N
  ED and CT = 6NT D.
- We assume a fixed functional form between total and non-embedding parameters in Equation 11, and fit ω empirically using Chinchilla model configurations.
- We assume a fixed functional form between loss, total parameters, and training data given by Equation 8. We report results using both the Chinchilla (Equation 9) and Epoch AI (Equation 10) fitted constants.
- We approximate Kaplan's models with 20 logarithmically spaced model sizes from 0.79k to 1.58B non-embedding parameters.

3 Analysis: Compute-Parameter Scaling Coefficient

This section presents our core analysis. We demonstrate that a local scaling coefficient ranging from 0.74 to 0.78 (close to Kaplan's 0.73) can emerge when calculated in terms of non-embedding parameters within the small-parameter regime, while remaining consistent with Chinchilla's coefficient.

Step 1. Fit a suitable function predicting N
E from NT.

We need a suitable function connecting non-embedding and total parameters. We propose to use the form:

$$NT = NE + \Phi 3c9N^{1/3}E \tag{11}$$

for some constant $\omega > 0$. Apart from having several desirable properties (strictly increasing and lim NT $\to \infty$ NT = N
E^2), it can be supported by findings from both the Kaplan and Chinchilla studies.

Kaplan perspective. Consider Kaplan's approach to parameter counting:

$$NT = 12ld^2 + NE, \tag{12}$$

where l represents the number of layers. While Kaplan does not explicitly list their model configurations, they do explore varying the aspect ratio A = d/l for a fixed model size. They determine that models of a given size exhibit similar performance across a range of aspect ratios, and this is not influenced by model scale (their Figure 5). Consequently, we could propose a sizing scheme with a fixed aspect ratio (A $\approx$ 40 appears reasonable from their plots). Assuming this sizing allows us to state (with l = d/A in Equation 12):

$$NT = \frac{12}{A}d^3 + NE. \tag{13}$$

Observing that N
E = $\frac{12}{A}$ $d^3 \to$ d = (N
E $\frac{A}{12}$)$1/3$, and combining with NE = (v + h)d,

$$NT \approx NE + (v+h)(\frac{A}{12})1/3N1/3E. \tag{14}$$

This takes the same form as Equation 11 with ω = (v + h)($\frac{A}{12}$)$1/3$.

Chinchilla perspective. We empirically fit a function NT = N
E + ωN^{δ}
E (note the learnable exponent) to the Chinchilla model configurations listed in Table A9 of Hoffmann

et al. (2022) for a range of NT (44M to 16B). We calculate NE from Equation 2, using the reported vocabulary size of 32,000, but disregard the context length of 2,048 since Chinchilla used non-learnable position embeddings (though their inclusion only slightly affects the coefficients).

Fitting a model with numpy's polyfit yields coefficients ω = 47491 and δ = 0.34. The exponent is close to 1/3, with an implied aspect ratio A = 39.2 (inferred from ω). This further supports the form in Equation 11.

Step 2. Incorporate this function into a model predicting loss in terms of NT and CT.

It should be remembered that although we are interested in how N T depends on CT, this only occurs because of how they both relate to loss.

$$NT = argminL(NT, CT). \tag{15}$$

Subject to:

$$CT = 6NTD \tag{16}$$

To analytically examine their scaling relationship, we need a mathematical expression for loss, for which we utilize the functional form from the Chinchilla study. Substituting CT = 6NT D into Equation 8 yields:

$$L(NT, CT) = NcNT + Dc(CT/6NT)\Phi 3b2 + E. \tag{17}$$

By differentiating Equation 16 with respect to NT, setting the result to zero, and rearranging in terms of NT, we obtain:

$$NT = CT^{\frac{\beta}{\alpha+\beta}}(\frac{\beta Dc}{\alpha 6^{\beta} Nc})^{\frac{1}{\alpha+\beta}}, orsimplyNT \propto C^{\frac{\beta}{\alpha+\beta}} \tag{18}$$

We now modify Equation 16 to be in terms of non-embedding parameters and compute. While NT requires Equation 11 from step 1, the second term avoids this because D = CT / 6NT = C
E / 6N
E.

$$L(NE, CE) = Nc(NE + \Phi 3c9N^{1/3}E)^{\alpha} + Dc(CE/6NE)^{\beta} + E \tag{19}$$

Step 3. Analytically derive the relationship between N
E and C
E.

To determine the relationship between N
E and C
E, we take the derivative of Equation 18 with respect to N
E, set it to zero, and rearrange:

$$CE = 6NE(NE + \omega(NE)1/3)^{\alpha}(\frac{\beta Dc}{\alpha Nc})(\frac{1}{1 + \frac{\omega}{3}(NE)^{-2/3}}$$

$+ \alpha)^{-1}$ (20)

This indicates that, generally, the relationship between N
E and C
E is not a power law. Nevertheless, we can think about a "local" power law approximation. That is, for a specific value of N
E, there exists a constant g that provides a first-order approximation (denoted by $\propto$) N
E, where g is defined as:

$$g := \frac{dlog(CE)}{dlog(NE)} = \frac{1}{1 - \frac{1}{\beta}\frac{\omega}{3}(NE)^{-2/3}}$$

$\frac{}{1+\frac{\omega}{3}(NE)^{-2/3}} + \alpha + 1 \frac{}{\beta\frac{\omega}{3}(NE)^{-2/3}} \frac{}{1+\frac{\omega}{3}(NE)^{-2/3}}$. (21)

The derivation is detailed in Appendix A.1. There are three phases.

- At a small scale, lim N
  E $\to$ 0 g = $\frac{\alpha/3+\beta}{\beta} \to$ N
  E $\propto$ C$^{\frac{\beta}{\alpha/3+\beta}}$
  E.
- At a large scale, lim N
  E $\to \infty$ g = $\frac{\alpha+\beta}{\beta} \to$ N
  E $\propto$ C$^{\frac{\beta}{\alpha+\beta}}$
  E, consistent with the NT case in Equation 17.
- A transition phase exists where g briefly increases. This occurs between the two limits when N$^{2/3}$
  E is of the same order as ω. Indeed, at exactly the point N$^{2/3}$
  E = ω, we have NT = N
  E + ωN$^{1/3}$
  E = NT = 2N
  E, indicating a 50/50 split between embedding and non-embedding parameters.

Step 4. Simulate synthetic data from the Chinchilla loss model over the model sizes used in the Kaplan study. Fit a local power law for N
E in terms of C
E.

By reading g, we could estimate a local power law and thus a scaling coefficient for a specific value of N
E. However, it is unclear which N
E value is representative of the Kaplan study. We choose a more accurate estimation approach, creating synthetic training curves from Equation 18 over the range of model sizes employed in the Kaplan study, and fitting coefficients using models that lie on the compute-efficient frontier. This will also validate our analytical expression for N
E and C
E in Equation 19.

We simulated 20 models with N
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 Equation 18, we adopt the Epoch AI specification (Equation 10) and ω = 47491, though we also report results for the Chinchilla specification (Equation 9).

Main result. The estimated scaling coefficient is shown when a power law is fitted to the compute optimal frontier (Chinchilla's Method 1) generated by these synthetic training curves. This represents our primary finding - by starting with a model from the Chinchilla study and modifying two aspects to match Kaplan's study (NT $\to$ N
E, small model sizes 0.79k - 1.58B parameters), we obtain local scaling coefficients:

$$EpochAI : NEC0.78E, \tag{22}$$

$$Chinchilla : NEC0.74E, \tag{23}$$

which are close to the Kaplan coefficient of 0.73. Therefore, this demonstrates that the Chinchilla coefficient is largely consistent with Kaplan's coefficient, given these two adjustments. This constitutes the paper's main result, reconciling these two apparently conflicting results.

4 Experiments: Compute-Parameter Scaling Coefficient

We offer concise experiments to confirm that our assertions are valid for models trained on a limited scale (millions of parameters).

Experiment 1. First, we confirm if scaling coefficients approximate those of Chinchilla and Kaplan when employing NT and N
E, respectively.

Five models with sizes NT $\in$ [0.8M, 1.6M, 2.1M, 3.3M, 4.6M] were trained using the BookCorpus dataset. The GPT-2 tokenizer was used, with a vocabulary size of 50,257 and a context length of 16 (although this is much less than normal, our tests indicate that context length has no impact on scaling coefficients). To estimate scaling coefficients, Chinchilla's Method 1 was applied, using the approximation C = 6ND.

Models were trained for updates $\in$ [4000, 4000, 4000, 8000, 8000], with a batch size of 65,536 tokens per update, for a total of training tokens D $\in$ [262M, 262M, 262M, 524M, 524M]. For each model size, the optimal learning rate was selected from $\in$ [0.001, 0.005, 0.01, 0.05], and no annealing was implemented.

Result 1. When coefficients are fitted to NT, we obtain NT $\propto$ $C^{0.49}$T, and for N
E, we obtain N
E $\propto$ $C^{0.74}$
E. These closely match the Chinchilla and Kaplan coefficients, respectively.

Experiment 2. We present an ablation of optimization schemes, demonstrating that using multiple training budgets per model has a negligible impact on coefficients (contrary to Chinchilla's explanation).

- Scheme 1. A single learning rate of 0.001 is set for all models. A single model is trained per size, and no annealing is applied.
- Scheme 2. The best learning rate is chosen per model. A single model is trained per size, and no annealing is applied. (As in our NT vs. N
  E comparison.)
- Scheme 3. The best learning rate is chosen per model. A single model is trained per size, and cosine annealing is applied at the update budget. (Kaplan study used this.)
- Scheme 4. The best learning rate is chosen per model. Six models are trained per size at different budgets $\in$ [0.25D, 0.5D, 0.75D, 1.0D, 1.5D, 2.0D], and cosine annealing is applied. (Chinchilla study used this.)

Result 2. The optimization technique has less of an influence on scaling coefficients than switching from NT to N
E. Using a single set of models without annealing (scheme 2) yields coefficients that are identical to those of the more computationally demanding scheme 4. In contrast to Chinchilla's assertion that switching from Kaplan's scheme 3 to scheme 4 would lower the scaling coefficient, our research indicates the opposite, with an increase from 0.46 to 0.49. This might account for our minor overestimation of the scaling coefficients in Equations 21 and 22.

Table 1: Comparison of different scaling coefficients from our experiments. Note that the change moving from NT to N
E has a much larger effect than moving between optimization schemes.

5 Analysis: Compute-Loss Scaling Coefficient

In addition to examining the compute-optimal parameter scaling, Kaplan and Chinchilla also characterized the scaling relationship between compute and loss, assuming optimal parameter scaling. Kaplan expressed this optimal loss in terms of non-embedding compute, while Chinchilla used total compute.

$$LE = minL(NE, CE), s.t. CE = 6NED, \tag{24}$$

$$LT = minL(NT, CT), s.t. CT = 6NTD. \tag{25}$$

Specifically, the two studies reported the following forms and coefficients linking optimal loss and compute:

$$Kaplan : LE = (\frac{CE}{C_o})^{\gamma} \tag{26}$$

$$Kaplan : LEC0.057E \tag{27}$$

$$Chinchilla : LTE = (\frac{C_T}{C_o})^{-\gamma} \tag{28}$$

$$Chinchilla : LTEC0.155T \tag{29}$$

$$EpochAI : LTEC0.178T \tag{30}$$

(Refer to Section A.3 for Chinchilla's compute coefficient.) Similar to the compute-parameter scaling coefficient, Kaplan's coefficient of 0.057 initially appears significantly different from Chinchilla's range of 0.155 to 0.178. However, we will again demonstrate that by starting with the Chinchilla study and adjusting for Kaplan's non-embedding compute, smaller scale, and their compute-loss form, these two coefficients can be largely reconciled.

Our analysis follows the same four-step approach as in Section 3. We can directly reuse Steps 1 and 2, while Steps 3 and 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
E and C
E.

We determine that the relationship between L
E and C
E is not a power law (derived in Section A.2).

$$\frac{dlog(LE)}{dlog(CE)} = NE(NE + \omega(NE)^{1/3}$$
$$)^{\alpha} \ (1\frac{}{1+\frac{\omega}{3}(NE)^{-2/3}})\frac{}{LE(6NE)^{\beta}} \tag{31}$$

Nevertheless, we can once more take into account a local first-order approximation, L
E $\propto$ C^k
E, where k = $\frac{dlog(LE)}{dlog(CE)}$.

Step 4. Simulate synthetic data from the Chinchilla loss model over the model sizes used in the Kaplan study. Fit a local power law for L
E in terms of C
E, using Kaplan's compute-loss form.

As in Section 3, we could use Equation 30 to calculate a point estimate for k in the relationship L
E $\propto$ C^k
E, and then fit. However, we again opt for the more faithful procedure of simulating data from the loss curves.

Using Kaplan's compute-loss form L
E = $(\frac{CE}{C_o})^{\gamma}$, we obtain the following models for the two specifications:

$$EpochAI : LECE, \tag{32}$$

$$Chinchilla : LECE, \tag{33}$$

which are roughly in line with Kaplan's reported coefficient of L
E $\propto$ C$^{-0.057}$
E.

We observe that Kaplan's form provides a good fit of the data in the non-embedding compute plot at a small scale, 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.

6 Related work

After early research that established how language models get better with parameters, data, and training computation, there has been research into the theoretical underpinnings of these scaling laws and whether they apply to other domains.

Several concurrent studies that have looked at how different design decisions affect scaling law analyses are more closely related to the spirit of our work. The methodology for determining scaling coefficients is revisited by Su et al. (2024). Hagele et al. (2024) discovered that multiple short decays with a constant learning rate or stochastic weight averaging may be used to recreate numerous independent cosine schedules more effectively. Our discovery is subtly different; a straightforward fixed learning rate will recover extremely comparable compute-parameter scaling coefficients as many cosine schedules. The impact of different hyperparameters on scaling laws is examined by Bi et al. (2024). They point out that different text datasets yield somewhat different optimal coefficients, with "cleaner" data exhibiting more parameter-hungry scaling behavior, which they believe may partially account for the discrepancy between the Kaplan and Chinchilla coefficients.

The goal of Porian et al. (2024)'s concurrent work is to clarify the discrepancies between the Kaplan and Chinchilla coefficients, which is the same goal as that of our paper. They conduct a number of large-scale experiments that replicate Kaplan's study, and they come to the conclusion that the discrepancy is caused by, in decreasing order of importance: 1) Kaplan's use of non-embedding compute rather than total compute; 2) Kaplan's use of an excessively long fixed-length warmup period for smaller models, which made them appear less efficient; and 3) Kaplan's failure to fully optimize hyperparameters. We believe that these findings complement our own. We have used an entirely analytical method to identify the main "first order" cause using just the data that was made publicly available in the two papers. (As a form of verification, tiny-scale experiments were conducted post-hoc.) This shows how mathematical techniques can be used in scaling's empirical science.

7 Discussion

This study sought to account for the disparity between the scaling coefficients of Kaplan and Chinchilla. We discovered two problems with Kaplan's study that, when taken together, biased their estimated scaling coefficients: they focused on smaller model sizes and only counted coefficients: they focused on smaller model sizes and only counted non-embedding parameters. This implies a curvature in the actual relationship between Nand NT (Figure 5). At greater values of NT, the embedding parameter counts become negligible, NT = N, and differences would not arise. Alternatively, had Kaplan investigated relationships directly in terms of NT, this issue would also not occur, even at this smaller scale (confirmed by our Experiment 1 finding NT $(\propto)C(^{0.49})Tevenfor NT < 5M).Theform Kaplanusedtopredictlossfromcomputefurthercontributedtodifferencesinthereportedcompute-lossscalingcoefficients.$

Inconsistency across scaling studies. Existing literature on scaling is not consistent in its use of non-embedding vs. total compute. Some studies follow Kaplan's approach, using non-embedding parameters or compute, while others 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 opt for the Kaplan compute-loss form without offsets. In contrast, 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 for measuring compute-optimal parameter coefficients.

Why should embedding parameters contribute to scaling behavior? Several works provide evidence that embedding parameters capture meaningful language properties. Word embeddings can be factorized into semantically interpretable factors (even the shallow Word2vec). LLMs learn linear embeddings of space and time across scales. Developing such meaningful embedding structures allows LLMs to perform high-level language operations, such as arithmetic. 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 (Equation 27) is a more appropriate form from the perspective of statistical learning. This approach accounts for the concept of irreducible risk, 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 and 2 used a full calculation). However, our work suggested these factors impact coefficients in a more minor way.

8 Appendix

8.1 Derivation of Equation 20

This section derives Equation 20:

$$g := \frac{dlog(C)}{dlog(N)} = \frac{1}{1 - \frac{1}{\beta}\frac{\frac{\omega}{3}(N)(^{-2/3})}{1+\frac{\omega}{3}(N)(^{-2/3})} + \frac{\alpha+1}{\beta}\frac{\frac{\omega}{3}(N)(^{-2/3})}{1+\frac{\omega}{3}(N)(^{-2/3})}}. \tag{34}$$

First note that

$$\frac{dlog(C)}{dlog(N)} = \frac{dlog(C)}{dN}\frac{dN}{dlog(N)} = \frac{dlog(C)}{dN}N \tag{35}$$

Recall the definition of Cfrom Equation 19:

$$C = 6N(N + \omega(N)(1/3))(^{\alpha})((\frac{\beta Dc}{\alpha Nc}))((\frac{1}{1 + \frac{\omega}{3}(N)(^{-2/3}) + \alpha}))(^{-1}) \tag{36}$$

$$log(C) = log(N) - \frac{1}{\beta}log(1 + \frac{\omega}{3}(N)(^{-2/3})) + \frac{\alpha+1}{\beta}log(N + \omega(N)(^{1/3})) + const \tag{37}$$

where const. does not depend on N . We now can take the derivative of each term. Derivative of term 1:

$$\frac{dlog(N)}{dN} = \frac{1}{N} \tag{38}$$

Derivative of term 2:

$$\frac{d}{dN}(-\frac{1}{\beta}log(1+\frac{\omega}{3}(N)(^{-2/3}))) = -\frac{1}{\beta}\frac{1}{1+\frac{\omega}{3}(N)(^{-2/3})}\frac{\omega}{3}(-\frac{2}{3})(N)(^{-5/3}) \tag{39}$$

Derivative of term 3:

$$\frac{d}{dN}(\frac{\alpha+1}{\beta}log(N+\omega(N)(^{1/3}))) = \frac{\alpha+1}{\beta}\frac{1}{N+\omega(N)(^{1/3})}(1+\frac{\omega}{3}(N)(^{-2/3})) \tag{40}$$

Then assemble all terms and multiply by N as per Equation 35.

8.2 Derivation of compute-loss analytical form in Equation 30

This section derives k, defined as:

$$k = \frac{dlog(L)}{dlog(C)}. \tag{41}$$

Expanding with the chain rule we find:

$$k = \frac{dlog(L)}{dL}\frac{dL}{dN}\frac{dN}{dlog(N)}\frac{dlog(N)}{dlog(C)} = \frac{N}{L}\frac{dL}{dN}g, \tag{42}$$

where we previously derived g = (d log(C)$\overline{dlog(N))inEquation20.}$

This leaves us with (dL $\overline{dN)tofind.FirstnotethatLisgivenbyEquation18whentheoptimalmodelsizeisused,i.e.,N(\rightarrow)N:}$

$$L = Nc(N+\omega(N)(^{1/3}))(^{\alpha}) + Dc(C/6N)(^{\beta}) + E. \tag{43}$$

Before taking this derivative, we recall that Cis actually a function of N (via Equation 19). Hence, we tackle the derivative in two parts. We find the first term derivative is equal to:

$$\frac{d}{dN}(Nc(N+\omega(N)(^{1/3}))(^{\alpha})) = \alpha Nc(1+\frac{\omega}{3}(N)(^{-2/3}))(N+\omega(N)(^{1/3}))(^{\alpha-1}) \tag{44}$$

The derivative of the second term, via the product rule, and spotting that (dC$\overline{dN)=(\frac{Cg}{N}),equals:}$

$$\frac{d}{dN}(Dc(C/6N)(^{\beta})) = \beta Dc(\frac{C}{6N})(^{\beta-1})((\frac{1}{6N})(\frac{Cg}{N}) - (\frac{C}{6(N)(^2)})) = \beta Dc(\frac{C}{6N})(^{\beta})(\frac{g-1}{N}) \tag{45}$$

Hence, combining these two terms we find:

$$\frac{dL}{dN} = \alpha Nc(1+\frac{\omega}{3}(N)(^{-2/3}))(N+\omega(N)(^{1/3}))(^{\alpha-1}) + \beta Dc(\frac{C}{6N})(^{\beta})(\frac{g-1}{N}) \tag{46}$$

Combining this result into to Equation 43 we get:

$$k = \frac{N}{L}\frac{dL}{dN}g = \frac{g}{L}(\alpha Nc(1+\frac{\omega}{3}(N)(^{-2/3}))(N+\omega(N)(^{1/3}))(^{\alpha}) + \beta Dc(g-1)(\frac{C}{6N})(^{\beta})) \tag{47}$$

8.3 Compute-loss coefficient derivation

We know from Equation 17 N T $(\propto)C(\frac{\beta}{\alpha+\beta}), andsimilarlyDT(\propto$ $)C(\frac{\alpha}{\alpha+\beta}).Substitutingtheseintothelossformof Equation8, andforsomenewconstants(\bar{N}c), (\bar{D}c)wefind, LT = Nc(NT)(^{\alpha}) + Dc(DT)(^{\beta}) + E$(48)

$$LT = \bar{N}cC(\tfrac{\alpha\beta}{\alpha+\beta}) + (\bar{D}c)C(\tfrac{\alpha\beta}{\alpha+\beta}) + E \tag{49}$$

$$LT - E(\propto)C(\tfrac{-\alpha\beta}{\alpha+\beta}) \tag{50}$$