EXHIBIT 152

Studying Data Repetition Through Scaling Laws and Mechanistic Interpretability

Danny Hernandez*

Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Nicholas Joseph,
Scott Johnston, Ben Mann, Neel Nanda, Chris Olah, Catherine Olsson,

Dario Amodei, Jared Kaplan, Sam McCandlish

Anthropic

Abstract

Recent large language models have been trained on vast datasets, but also often on repeated data, either intentionally for the purpose of upweighting higher quality data, or uninten tionally because data deduplication is not perfect and repeated data sneaks through at the sentence, paragraph, or document level. Some works have reported substantial negative effects of this repeated data. In this paper we attempt to study repeated data more systemat ically, and to understand its effects mechanistically. To do this, we train a family of models where most of the data is unique but a small fraction of it is repeated many times. We find a strong double descent phenomenon, in which a predictable range of repetition frequency leads to surprisingly severe degradation in performance. For instance, performance of a 800M parameter model can be degraded to that of a 2x smaller model (400M params) by repeating 0.1% of the data 100 times, despite the other 90% of the training tokens remain ing unique. We show that the region of damaged performance coincides with the advent of dramatically increased overfitting on the repeated fraction. Finally, we connect these observations to recent mechanistic interpretability work, by showing that data repetition disproportionately damages copying and internal structures associated with generalization, such as induction heads, providing a possible mechanism for the shift from generalization to memorization. Taken together, these results provide a hypothesis for why repeating a relatively small fraction of data in large language models could lead to disproportionately large harms to performance.

*Correspondence to: danny@anthropic.com
Author contributions are listed at the end of the paper

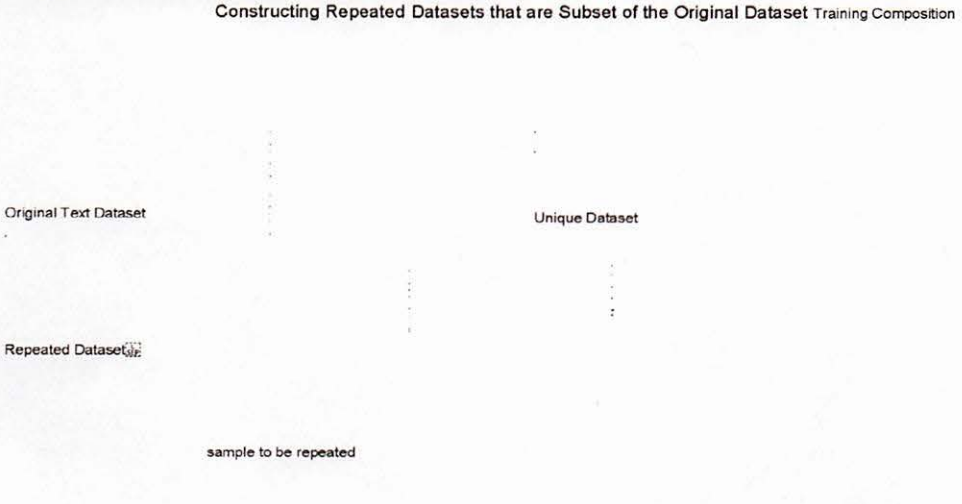




Figure 1 Experimental Setup. From a large original text dataset (left), we draw 90% of our desired training dataset in a non-repeated fashion, and 10% as repeats of a tiny portion of the original dataset (right). We hold constant that 10% of total training tokens will come from repeats, but we vary the repeated fraction in our runs. That is, the sample to be repeated might be very small, like 0.01% of the total training tokens repeated 1000x, or relatively large, like 1% of the total training tokens repeated 10x, and so on.

1 Introduction

Algorithmic innovation [Hernandez and Brown, 2020], compute [Amodei et al., 2018], and data are three of the major factors that drive the advance of AI. Large language models require massive quantities of data to be trained efficiently [Brown et al., 2020, Rae et al., 2021], and in fact recent work suggests that even more data than previously believed is required to train models optimally [Hoffmann et al., 2022]. Given this and the varied quality of different datasets, large models are often trained with the highest quality data delib erately repeated for multiple epochs. Also, large datasets often contain many copies of nearly identical or substantially overlapping documents, leading to "unintentional repeats". These can be reduced by dedupli cation but are difficult to completely eliminate. At the same time, recent results show that reduction of data repetition can improve language models substantially [Lee et al., 2021], and data repetition can also lead to memorization which may be associated with privacy issues [Carlini et al., 2020, Ziegler,] .

It therefore seems valuable to systematically study data repetition and its effects. Scaling laws have been found for model size, data quantity, and training time [Kaplan et al., 2020, Henighan et al., 2020] and such scaling laws can already predict the performance effects of repeating the entire dataset for multiple epochs. But frequent repeats of a small subset of the training data is a different situation, and may lead to effects more complicated than simple overfitting.

In this paper we study data repetition in language models through two lenses: the macroscopic lens of scaling laws, and the microscopic lens of mechanistic interpretability [Elhage et al., 2021, Catherine et al., 2022]. For the first lens, we train a series of language and code models on a dataset that consists of mostly unique data (for example 90%), plus a small amount of additional data repeated many times (Figure 1). We find a strong double-descent phenomenon [Advani and Saxe, 2017, Belkin et al., 2018, Nakkiran et al., 2019], such that there is a defined range of repetition frequency for which performance is harmed to a surprisingly large extent. The shape of the region is predictable and allows us to extrapolate to state-of-the-art language models and make predictions about what configurations of data repetition are likely to be most or least harmful.

For the second lens (mechanistic interpretability), we look inside the model and show that structures that have recently been associated with algorithmic generalization, such as induction heads [refs induction heads], are damaged by training on repeated data in a way that correlates with the double descent phenomenon. Behaviors associated with induction heads, such as copying sequences, are also shown to be strongly degraded.

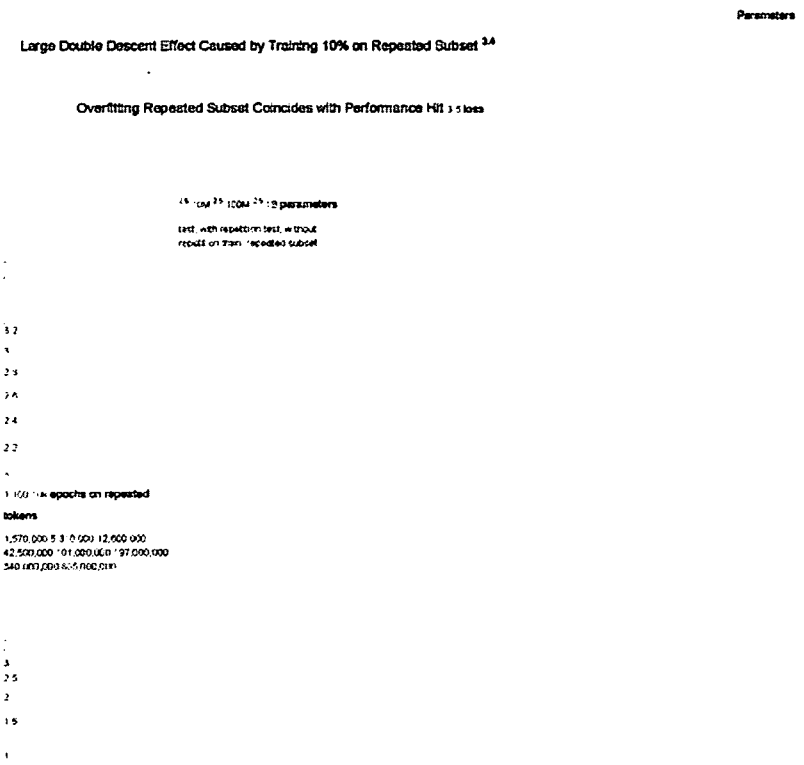


Figure 2 Models of different sizes show a degradation in performance at a specific range of repeats that depends on model size (left panel). At its peak the degradation sometimes reaches the equivalent of a 2x decrease in model size. The right panel shows that divergence (blue line) from a healthy, straight scaling law (red) lines up with when the models start to dramatically overfit the repeated subset (green curve). The blue line on the right corresponds to a vertical slice of models in the left diagram trained on the repeated subset for 120 epochs.

Together, the two lenses provide an integrated picture of how repeated data might be causing the network (or part of it) to shift from generalization to memorization, and mechanistically how this could be harming performance of the overall language model.

2 Summary of Results

To systematically study repeated data, we trained transformer language models on mostly unique data plus a small fraction of repeated data (Figure 1), varying the repeated dataset size, model size, and fraction of tokens trained on repeated data over 2-3 orders of magnitude. We examined the resulting models using both scaling laws and mechanistic interpretability tools. Our main findings were as follows:

- Repeated data induces a strong double-descent phenomenon [Advani and Saxe, 2017, Belkin et al., 2018, Nakkiran et al., 2019], in which data repeated a few times does not cause much damage to language model performance, data repeated very many times also does not cause much damage, but there is a peak in the middle where damage is surprisingly large. For instance, when we train an 800M parameter transformer [attention is all you need] with 10% of training tokens drawn from the repeated subset (yellow curve in Figure 2) we find the loss can be nearly as high as for the 340M parameter transformer (light green curve). We see a classic double descent learning curve in Figure 3. Figure 2 on the right shows that the peak performance hit coincides with where the train loss on the repeated data approaches zero, similar to previously observed double-descent phenomena [ref(s)]. This also provides a practical diagnostic for when repeated data is likely to be harming the model. Finally, extrapolating the region of peak degradation to larger model size predicts meaningful degradation of repeating data only 2 times for large (GPT-3 size) models.
- Repeated data can cause a divergence from power-law scaling. For the blue curve in Figure 2 right (122 repeated epochs), we see only a moderate impact to performance (line on log-log graph) until the model is scaled up to 100M parameters, after which we see a large divergence from power law scaling of cross entropy loss.
- Repeated data causes a disproportionately large performance hit to copying, a mechanism for in-context learning. We constructed a simple copying eval, the loss on the first paragraph of Harry Potter copied 12 times. We observe that using 1% repeated data at the worst number of repeated epochs caused a 49% reduction in effective model size on this task whereas it only caused a 2.5% reduction in effective model size on test loss.
- The disproportionate performance hit to copying coincides with a disproportionate degrada tion of induction heads. In line with [Catherine et al., 2022] we evaluated the models on repeated sequences of random tokens and observed the degree to which attention heads attend to earlier tokens that are preceded by a token that matches the present token.

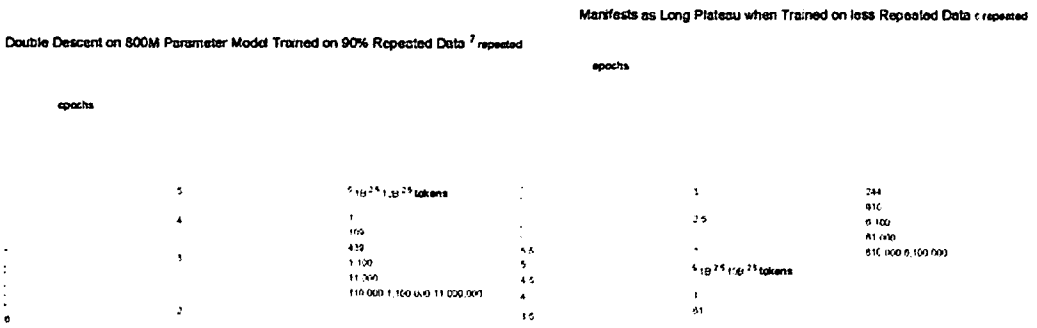


Figure 3 For the models on the left 10% of the tokens were drawn from the unique dataset and 90% of the tokens were drawn from repeated datasets of varying sizes. In this graph we see double descent curves. For several models test loss drops as normal during the beginning of training, but then starts to rise during the middle of training before dropping again. In the graph on the right the fraction of repeated data has been reduced to 50%, and we see that double descent bump has turned into a long plateaus for highly affected models.

- Repeated data causes a smaller but still disproportionate performance drop out of distribution, as measured by Python loss. Unlike our copying evals we mostly see the performance drop with higher levels of repetition, 10-90%.

- One and two-layer attention only models trained on repeated data are worse at copying and fuzzily copying proper names on inspection. When we zoom into smaller models we can see this degradation in a simple, understandable form of copying when looking at the per token losses in a paragraph of text.
- Training on repeated Python code creates a similar behavior. When training on Python we also observe a double descent phenomenon and a predictable poor performance region, though the shape of both curves are somewhat different.
- Pre-training on repeated data damages models. Pre-training with repeated data leads to worse performance than both training from scratch and fine-tuning from a control model. During fine tuning, the repeated data model forgets the repeated dataset, so we consider the model pre-trained with repeated data to be strictly worse than the model fine-tuned from the unique dataset.

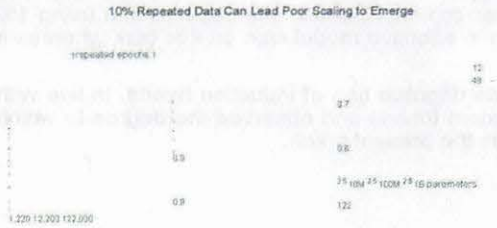
3 Results

Repeated data induces a strong double descent phenomena In Figure 3, we see midway during the training run the loss goes up and then down for most of the curves. As we lower the fraction of repeated data, it moves from a descent to a long plateau, but we consider it to be the same underlying phenomenon, we've simply weakened the effect. As in [Nakkiran et al., 2019] we see double descent effects caused by both increasing model size and epochs. As we increase the fraction of repeated data, the size of the performance drop goes up substantially, as seen in Figure 5

Repeated data can cause a divergence from power-law scaling. In Figure 4, we see that these regions can lead to sudden, serious degradation of scaling performance.

We also see in Figure 4, that models trained for 1220 repeated epochs show a dip in performance to about 0.55x model size equivalent, but then larger models recover to 0.83x since they are past the point of peak degradation.

A large region of reduced performance predicts a meaningful effect on the 2nd epoch of training for large models. In figure 4 above we consider the region of poor performance in which we observe at least 50% of the maximum effect (also see appendix A). We attempt to fit the boundaries of this region over a large sweep of model sizes, epochs on repeated tokens, and percent of tokens from repeated datasets. We observe a relatively large predictable region of reduced performance.



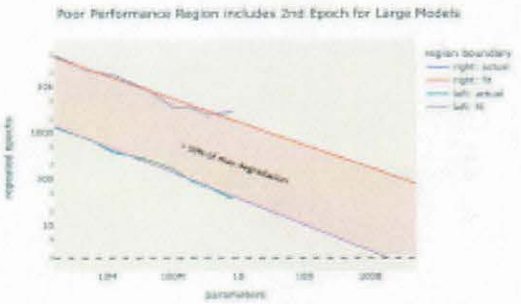


Figure 4 On the left we plot the same results as in Figure 2 re-parameterized in terms of the effective model size implied by the test loss. It's noteworthy that for the blue curve (122 repeated epochs), we see almost no performance deviation from a power law scaling law (line on log-log graph) until the model is scaled up to 100M parameters, after which we see a divergence. We see the same divergence around 400M parameters for 12,200 repeated epochs. The large predictable region on the right graph helps show why large models like GPT-3, x, and y need to be very careful about overfitting their high quality distributions like Wikipedia and books. Here the blue and green curves correspond to the right and left sides of the double descent curve where we observe 50% of the maximum effect. They are an aggregation of that curve for the scans where we trained on 3%, 10%, 20%, 50%, and 90% repeated data. The details of the fit are in Appendix B. A large number of runs needed to be aggregated to produce a clean fit.

$$E = k \cdot N^{a}$$

Where *E* corresponds to epochs of repetition and *N* corresponds to the parameters in the model. It's notable that the lines in figure 2b are relatively parallel. The fits for the above lines are given in the table below.

	k	*a*
right boundary	5.1e7	-.50
left boundary	4.2e6	-.56

In addition to being robust to task, model size, and architecture as shown in previous work [ ], it seems double descent is robust to being a sub-distribution and that it can have a large effect on overall performance even while being a modest fraction of training tokens.

Repeated data causes a disproportionately large performance hit to copying, a mechanism for in context learning. We constructed a simple eval in which copying is heavily emphasized eval, the loss on the first paragraph of Harry Potter copied 12 times.

The disproportionate performance hit to copying coincides with a disproportionate degradation of induction heads. In line with [Catherine et al., 2022] we evaluated the models on repeated sequences of random tokens, and observed the degree to which attention heads attend to earlier tokens that are preceded by a token that matches the present token.

We find it interesting that the sharp drop in prefix matching score for a 1.5M parameter model with 50% repetition corresponded to a complete breakdown of paragraph level copying.

This complete breakdown of paragraph level copying corresponds to a 1.5M parameter model having the effective overall performance of a 30,000 parameter model, while having an equivalent prefix matching score to a model with effectively 2,000 parameters.

Repeated data causes a smaller, disproportionate performance drop on our out-of-distribution eval (zero-shot python loss).

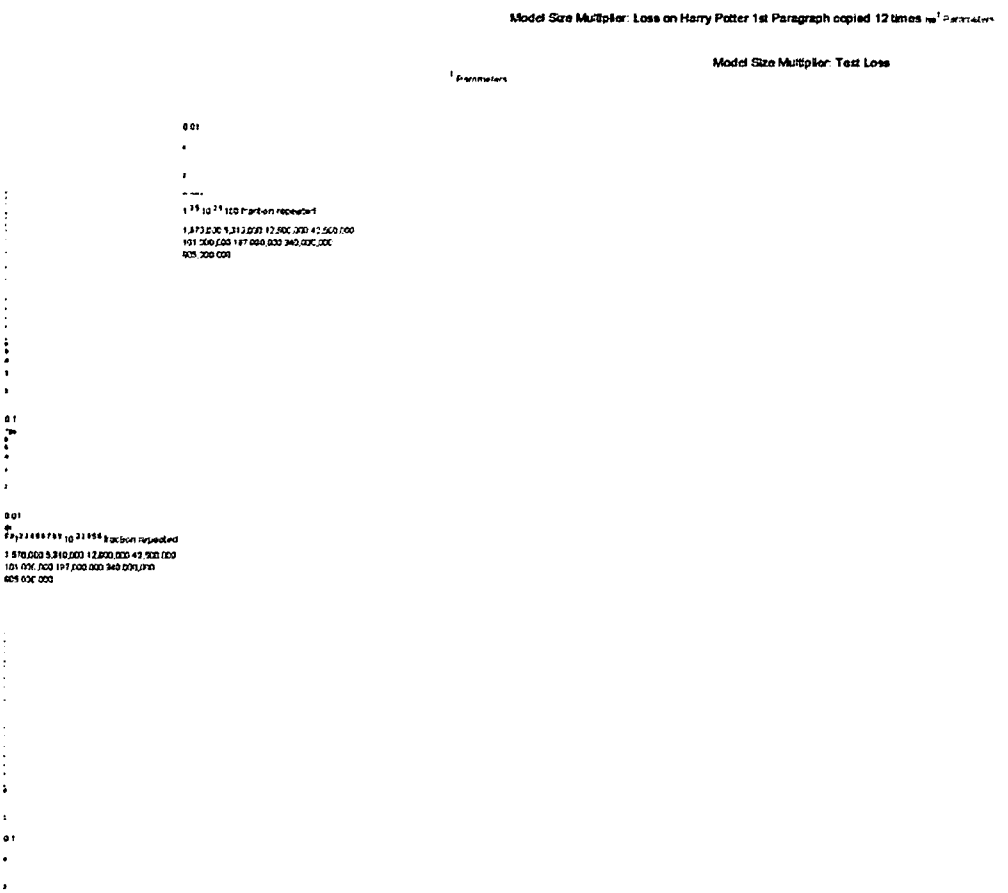


Figure 5 We observe that 1% repeated data at the worst number of repeated epochs caused a 49% reduction in effective model size on this task for a 340M parameter model, whereas it only caused a 2.5% reduction in effective model size on test loss. We see much larger effects on the copying eval than overall performance for repeat fractions between 1 and 20. The model size multiplier for copying is based on interpolation and the model size multiplier for test loss is based on a power law fit (see Appendix B for more details).

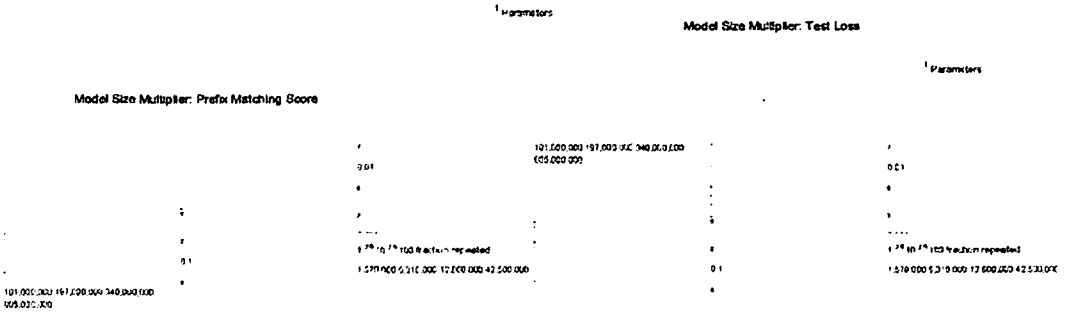


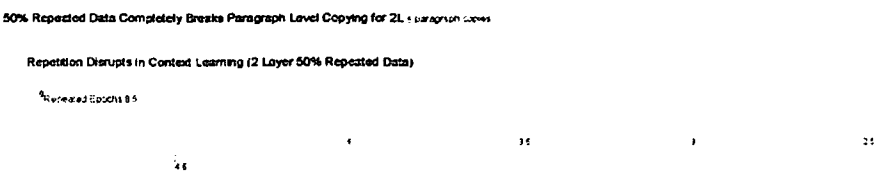
Figure 6 We observe that 1-3% repeated data causes something like a 2x model size multiplier are prefix matching score while causing less than a 1.1x model size reduction in effective model size on test loss. Again we see much larger effects on the copying eval than overall performance for repeat fractions between 1 and 20. The model size multiplier for prefix matching is based on a linear fit (see Appendix B for more details of fit). The test loss shown on the right is the same graph as in , but with differently scaled axes for ease of comparison.

Given that we overfit the model, we expected it to perform worse off distribution, which we do observe. We notice almost an opposite pattern to what we observed in the copying evals. We see most of the disproportion drop at 50% and 90% rather than 1-10%.

One and two-layer attention only models are worse at copying and fuzzily copying proper names on inspection. Additionally we trained one and two layer attention-only transformers, because we were inter ested in exploring the effects of repetition given the high degree of mechanistic interpretability small attention only models provide [Cite both Anthropic interpretability papers]. That is, we can reverse engineer a large portion of Transformers when they don't have MLPs with a circuits level understanding [reference circuits]. We also observe the double-descent phenomenon when using 1-layer attention-only architecture as shown in Appendix D.

For a 1-layer attention only models, when we zoom in and look at the tokens the repeated data models does worse on, we can see that these models are worse at a simple form of copying.

We can clearly see in Figure 9 that the repeated model is worse at names, for which copying previously seen tokens is extremely helpful. Here are two examples of the skip trigram pattern [interpretability paper] that illustrate why it's helpful for copying previously seen names.



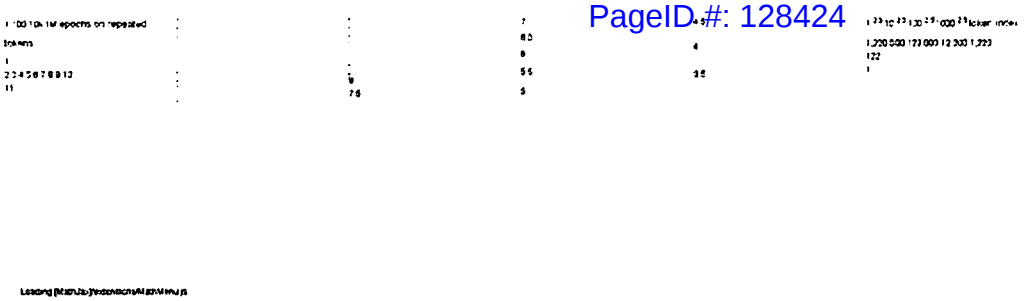


Figure 7 On the left we zoom in on the 2-layer models trained on 50% repeated data from Figure 5 and again evaluate loss on the first paragraph of Harry Potter repeated X times. In Appendix C, we explore shortening the length of the paragraph to verify the problem is with copying rather than long contexts. On the right we look at per token losses on the test set. We see dramatically reduced in-context learning. In-context learning appears to flatten out around token 1000 for this model

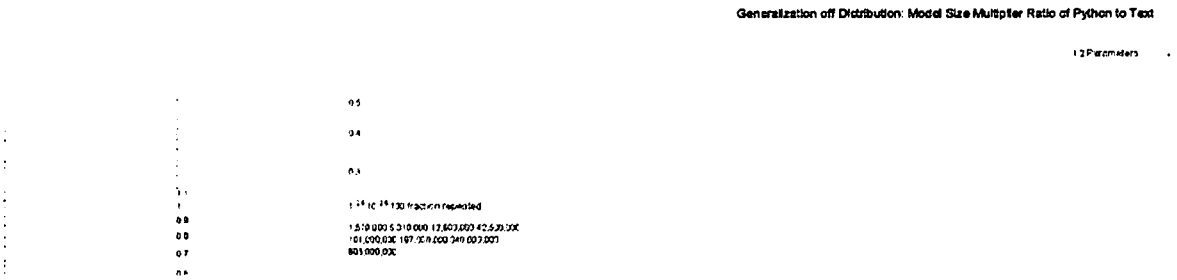


Figure 8 We observe that training on high levels of repeated data causes a disproportionate drop on out of distribution performance (python loss). For instance a 42M parameter model suffered trained on 50% and 90% repeated data suffered a 15% and 47% effective parameter drop on python loss (after taking into account the performance hit from python).

[*a*][*b*] . . . [*a*] => [*b*] [*P of*][*ter*] . . . [*P of*] => [*ter*]
[*a*][*b*] . . . [*a*] => [*b'*] [*P of*][*ter*] . . . [*P of*] => [*ters*]

If we plot the same visualization for a 2-layer attention-only model we can see a similar effect, where the control model is much better at fuzzy copying.

Visually it's less obvious that the 2-layer repeated model is worse at names. There are a few examples of it putting 1.1x less odds on the correct token. But on the other hand there are dramatic cases of the control model doing 500x times better (odds ratio on correct token) for fuzzy copying, like unDursleyish, which is exactly the kind of degradation we'd expect to see from disrupting induction heads.

We attempted to leverage logit attribution to see if the difference was primarily due to the induction head being less active or other heads interfering with it. We were unable to find clear evidence of either, but include our exploration of a 2 layer attention only model in Appendix X.

We observe a double descent phenomenon in models trained on python, but the python scans exhibit a somewhat different overall shape. If we use the same method to fit the poor performance region, we see

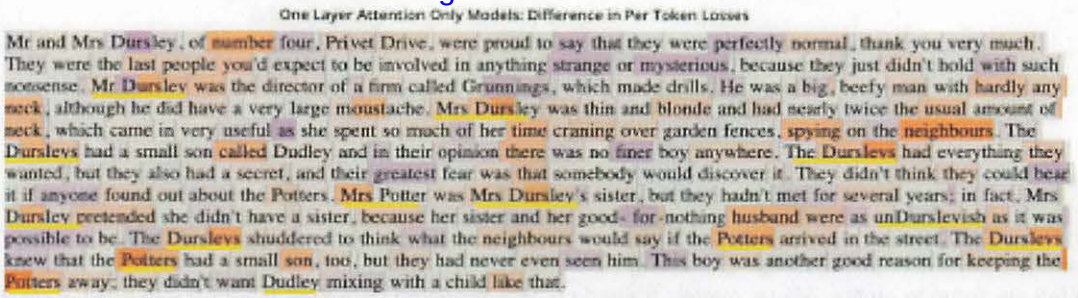


Figure 9 We visualized the difference in performance on the first paragraph of Harry Potter for control and 10% repeated data 1-layer attention-only model above. The color orange corresponds to the control model doing better, purple corresponds to the repeated data model doing better, and the intensity corresponds to the magnitude of the difference in per token losses. Proper names that appear earlier in the passage are great targets for copying. When we underlined all such names in yellow, it's clear that the control model is much better at copying. Often the difference is dramatic, we look at the last three appearances of Potters the control model puts a >97% chance on [ters] once it's seen [Pot], whereas the repeated data model puts <4% chance on that token.

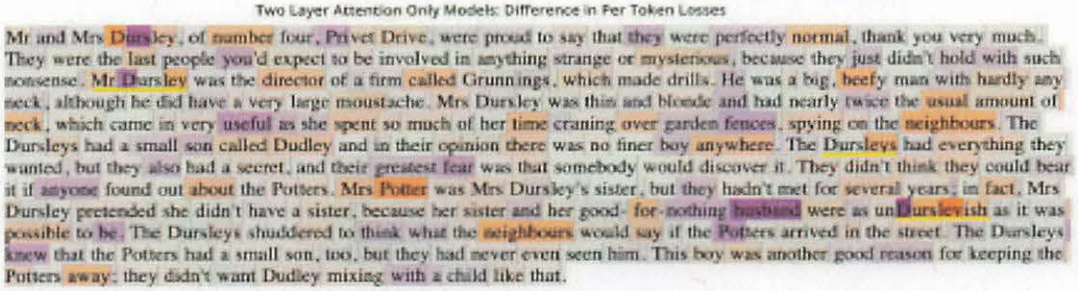


Figure 10 Similar to figure 7 but 2-layer attention-only models. While the repeated model sometimes does better on some repeated proper names where both models are quite uncertain, there are clear examples of the control being confident, correct, and much better. These examples are highlighted in yellow and discussed. On the token [ley] in the second appearance of [D][urs][ley] the control model puts a 92% chance on [ley] whereas the repeated data model puts a 10% chance. On the token [leys] in the second appearance of [D][urs][leys] the control model puts a 44% chance on [leys] whereas the repeated data model puts a 4.9% chance. On the [ley] in [un][D][urs][ley][ish] the control model puts a 68% chance on [ley] whereas the repeated data model puts a 0.4% chance.

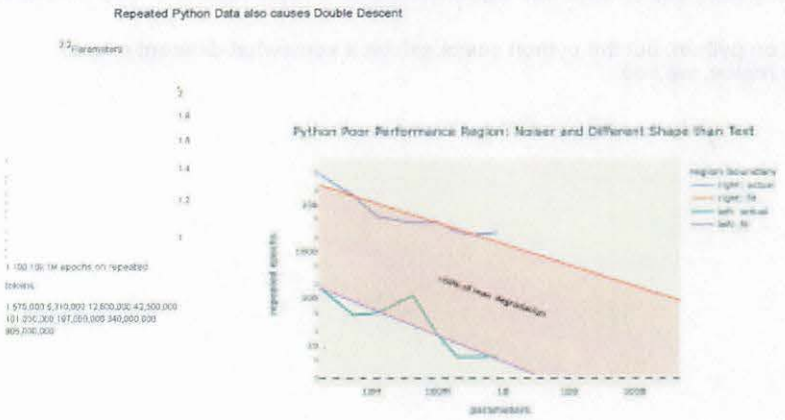


Figure 11 50% of the tokens in the models on the left were drawn from the python dataset described in methods. 50% of the tokens were drawn from repeated subsets of that distribution of varying sizes. We observe a double descent pattern with python data as we did in figure 1a. In figure 12b, we again see a large poor performance region as we did in Figure 4. Again the blue and green curves correspond to the right and left sides of the double descent curve where we observe 50% of the maximum effect.

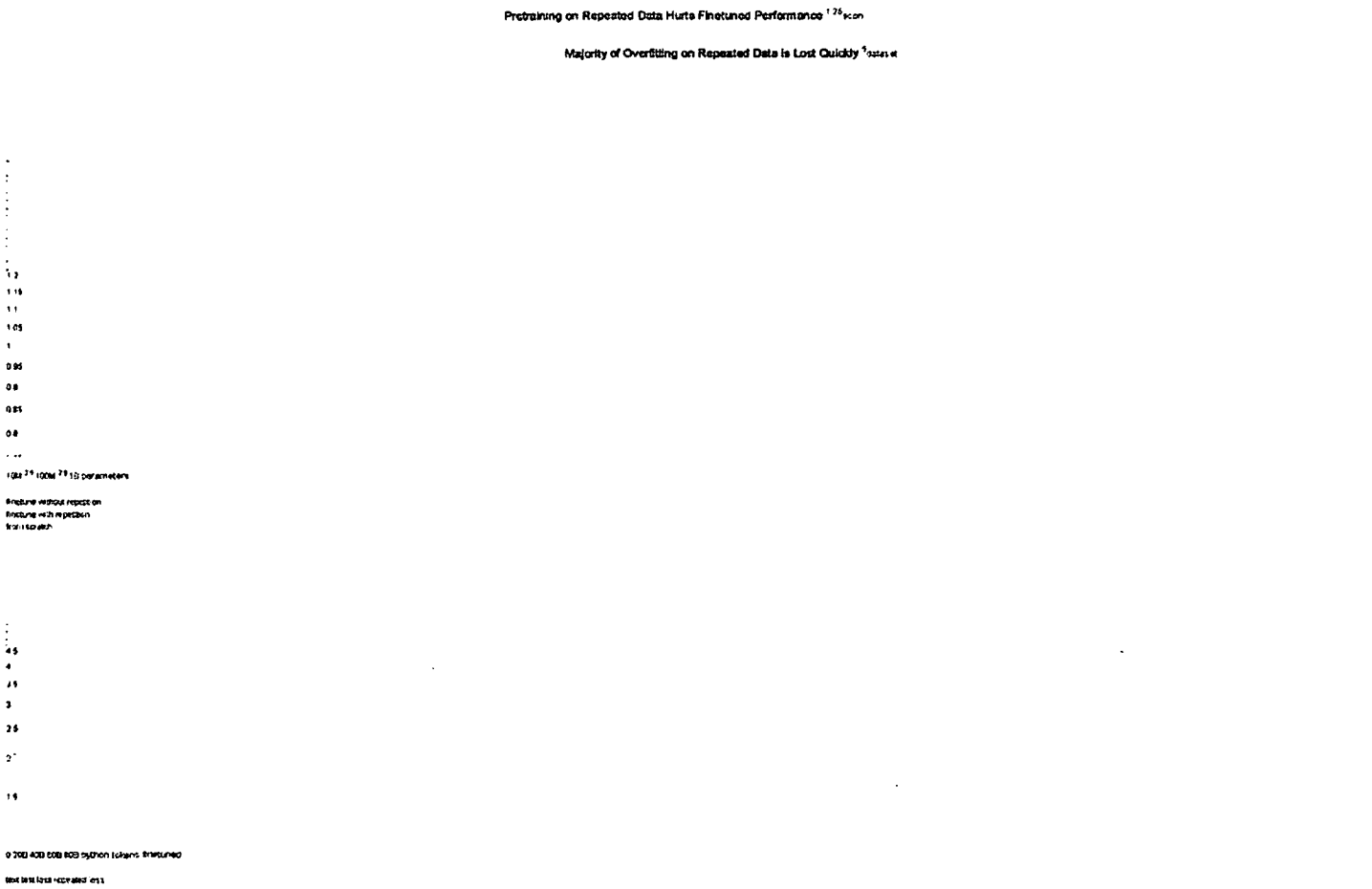


Figure 12 The red lines in both graphs correspond to models pre-trained on 10% original text dataset and 90% repeated data. For both of these runs, the repeated datasets were sized to maximize the performance hit. These runs are compared to two types of control runs, in green models trained on python from scratch and in blue models pre-trained on 100% large text datasets. All models were fine-tuned on the same python dataset. On the left we see a sizable performance hit from pre-training on 90% repeated data. In figure 5b we see that the repeated model forgets the repeated dataset with a modest amount of fine-tuning.

that a second epoch for today's large models (200B parameters) are robustly in the reduced performance region for python, but the fit is noisier than the fit for text and the two lines are no longer parallel.

The noise may partially be explained by the python fits being averaged over half as many settings for the fraction of tokens that are repeated data. It could also be that we need a higher resolution python scan to get a cleaner estimate for the poor performance region. Finally, the python data was trained on approximately 2 epochs as described in the methods section. Python also may have more unintentional repetition than text, from copying and pasting of example code and forking of codebases. Such repetition could change the shape of the region of poor performance. More analysis of the python experiments is shown in Appendix B.

Pre-training on repeated data hurts fine-tuned performance We find that the negative impact of repeated data persists after fine-tuning natural-language models on python.

It's noteworthy that the performance hit once fine-tuned is much smaller. An 800M model pre-trained on 90% repeated data from the double descent peak had its effective parameters reduced by 73x (shown in Appendix A). When we fine-tune from the repeated model we see a 1.6x reduction in effective parameters compared to training from scratch. This is still meaningful damage to the model, but it's recovered substantially. Since the repeated model forgets the repeated dataset after a modest amount of fine-tuning (Figure 12 , we consider the

fine-tuned model with repeated data pre-training to be dominated by the fine-tuned model from the unique dataset.

4 Methods

The decoder-only transformer models were trained on an 8192 token context with the same settings as de scribed in [alignment paper] for 100B tokens. Our language experiments utilized a 400B token dataset with 55% heavily filtered common crawl data (220B tokens), 32% internet books (128B tokens), and some smaller distributions including OpenWebText, Wikipedia, Stack Exchange most of which we sourced from The Pile [GBB+20], and leveraged the 50304 GPT-2 encoding [Radford et al., 2019][Hugging Face].

Code models were trained or fine-tuned on 45B tokens of python for 2.2 epochs. Fine-tuning experiments had the same hyperparameters as pre-training experiments, but with learning rates reduced by a factor of 2 and reduced warmups.

We varied model size, repeated dataset size, and the fraction of tokens trained on repeated data by 3, 2.5, and 2 orders of magnitude respectively.

5 Related Work

Scaling Laws to do adapt sclaing laws of transfer and societal impacts

Interpretability to do adapt sclaing laws of transfer and societal impacts

Rise of Large, Diverse Language Datasets
The engineering and science of large, diverse language datasets is relatively new. Pre-2017 many language models were trained on a single distribution of text, such as news articles (Jozefowicz et al., 2016), Wikipedia (Merity et al., 2016), or fiction books (Kiros et al., 2015)." GPT-2 leveraged webtext, outbound Reddit links with at least 3 upvotes in order to use human curation/filtration to ensure quality in addition to a broad distribution. GPT-2's capabilities are largely attributed to its scaled-up size and dataset (10x the parameters and 10x the data of GPT) [GPT-2 blog post]. The next generation of language models, GPT-3, Gopher, Chinchilla, etc, leveraged large, diverse datasets that consist of many sub-distributions. The Pile makes a high-quality distribution available widely [Pile].

Double Descent
The double descent phenomenon was first shown in generality by [Belkin et al. (2018)] where it was demon strated for decision trees, random features, and 2-layer neural networks. Similar behavior was previously observed in [Opper (1995; 2001), Advani and Saxe (2017), Spigler et al. (2018), and Geiger et al. (2019b), Nakkiran, et al].

Memorization
Repetition has mostly been studied in the context of . Prior work has shown the recovery of various forms of personal information like phone numbers [(Carlini et al., 2020; Ziegler, 2021),]. The amount of text a model can memorize has been given less attention [memorization], but it depends on entropy of the text the definition.

Data Quality Scaling Laws and Ablations
Constructing datasets for large language models [refs] includes a large number of decisions: choosing sam pling weights, quality filtering, de-duplication, de-duplication, fuzzy de-duplication, epochs per dataset, etc. There hasn't yet been a lot of public work that quantitatively shows the impact of such decisions, but the dataset ablations in Appendix A of the [Gopher] paper are notable. They clearly show the benefit of their dataset mixture, quality filter, exact de-duplication, and fuzzy de-duplication for 1.4B parameter models.

6 Discussion

6.1 Why does repeating a small fraction of data damage performance so much?

We showed that a dataset with only 10% repeated tokens can reduce model performance by an effective 2x in parameter count, much more than if that 10% of the data had simply never been trained on. The repeated data thus "sabotages" the model out of proportion to its share in the dataset. Why does this occur, and why only for a specific amount of repetition? One plausible hypothesis comes from looking at the model's "incentives" to

memorize vs generalize. To take some numbers that are roughly correct, a 800M parameter model typically has a loss of roughly 2.0 nats/token, a 400M parameter model has a loss of roughly 2.2 nats/token, and fully memorized data will have a loss of 0 nats/token. Now suppose a 800M model is trained on 90% unique data and 10% tokens consisting of repeated data. We can ask whether it is a "good tradeoff" for the model to memorize the repeated data (leading to 0 loss on 10% of the dataset), at the cost of degrading performance by the equivalent of a 2x multiple in model size (which raises loss on the other 90% from 2 to 2.2). Some simple arithmetic suggests that it is: $0.9 * 2.2 + 0.1 * 0 = 1.98 < 2.0$. Another way to say this is that zero loss is such a huge drop compared to the differences in entropy between model sizes that driving the loss to zero on even a tiny subset can incentivize enormous degradation in quality.

This however leaves open the question of when this tradeoff is necessary or possible – and here is where the double descent phenomenon comes in. If a lot of data is repeated only a few times (say 5% of the data repeated 2x) then the model may not have the capacity to memorize it, and also doesn't see it enough times during training to do so. If a tiny amount of data is repeated very many

times (say 0.01% of the data repeated 1000x), then the model will definitely memorize it, but because it is so small the model need not use much capacity to do so, so the degradation in quality will likely be small. There's a range in the middle where the data can be memorized *and* doing so consumes a large fraction of the model's capacity, and this is where the peak of degradation occurs.

6.2 Generalization, memorization, and induction heads

Our results show that overfitting on the repeated data results in worse test loss, and this co-occurs with a disproportionate degradation in the model's induction heads (prefix matching score) and its ability to copy text. Copying sequences can be seen as a form of generalization, as it requires algorithmic operations that are independent of the content of the data. [Elhage et al., 2021, Catherine et al., 2022] provided evidence for induction heads as the mechanism implementing copying and other pattern-matching. For the 2 Layer model shown in Figure 7 it seems as if the pressure to memorize the repeated dataset has led a skip tri-gram head to replace the induction head entirely. Thus our results tell a story where a type of generalization and its internal implementation are disrupted when the model memorizes repeated data – a vivid illustration of the memorization-generalization trade-off. Future work could take this even further, by measuring the number of parameters devoted to memorization and trying to observe them competing for space with induction heads. Finally, it's worth noting that the co-occurence of copying degradation and induction head degradation is itself some additional evidence for induction heads as the source of in-context learning; [Catherine et al., 2022] was not fully conclusive and our results further bolster the case.

6.3 Bridging mechanistic interpretability and scaling laws

The results connecting memorization to the degradation of mechanistic interpretability structures [Catherine et al., 2022] are an example of a "bridge" between microscopic phenomena inside the network and macroscopic trends in the loss. We view such connections as very fruitful tools for research, because they allow us to see the same thing through different lenses: the macroscopic behavior demonstrates the sig nificance of the microscopic mechanisms, and the microscopic mechanisms help explain how and why the macroscopic phenomena occur. Switching back and forth between the two allows for a deeper understanding of both, as well as more robust diagnostics if something goes wrong. We are aware of at least one other instance of such a bridge – the correspondence between the formation of induction heads and the boost in in-context learning near the beginning of training [Elhage et al., 2021, Catherine et al., 2022] – but such con nections remain rare so far, and we believe that finding more of them is a promising route to more deeply understanding neural nets.

6.4 Repeated data and fine-tuning

We hypothesized repetition might help explain why models trained from scratch sometimes outperformed models that were pre-trained and then fine-tuned [Hernandez et al., 2021]. For our purposes, we define ossi fication as any pre-training that leads a fine-tuned model to perform worse than a model trained from scratch (given a fixed compute and data budget). It required relatively extreme repetition in pre-training (90% train ing on repeated tokens at peak of double descent curve, 73x reduction in effective model size) to see a large ossification effect (1.6x reduction in effective model size) within our fine-tuning setup. We still think repe tition might explain a large fraction of ossification when we consider training on various types of repetition we didn't study here (sentence level, paragraph level, similar documents, distribution, etc). Overall, our

finding that repetition can induce ossification provides medium causal evidence to this hypothesis. We think ossification is an interesting phenomenon that merits further study.

6.5 Scaling Laws

6.6 Limitations

We attempt to discuss limitations throughout the text where appropriate, but for the reader's convenience, we enumerate them here. We attempt to list them in a loosely descending order of importance.

1. Our fits for region of poor performance were relatively noisy, and we only observed a clean trend by aggregating them. This is discussed in the Results section and further explored in Appendix A.
2. The data we repeated was a random subset of the original dataset, and is thus not directly applicable to the situation where higher quality data (such as Wikipedia) is intentionally repeated to improve quality. Nevertheless, it seems plausible that the results would carry over.
3. We measured loss, rather than downstream NLP evaluations. Overfitting doesn't always entail worse performance on downstream tasks [Instruct paper], so it's possible that the degradation we observe does not carry over to these tasks.
4. We didn't explore the effects of early stopping, dropout, weight decay, or other regularization.
5. We didn't investigate simpler systems than 1L attention-only models, which might contain more complete mechanistic insights.

6.7 Future Directions

Below are some promising future directions:

1. Varying the type of repetition. We could inject repeated sentences or paragraphs at the beginning or end of some fraction of contexts, or repeat chunks of documents in a different order. We could also explore cases where the repeated data has a different distribution than the unique data.
2. Further interpretability work. Are there neurons that tell the model what distribution it's in (unique or repeated)? Are there neurons through which we can observe and edit the repeated sequences?

3. Drill down on memorization and generalization. Could we measure the number of model parame ters taken up by memorization vs generalization, either behaviorally or by using mechanistic inter pretability to identify parameters that are storing memorized data? Can we measure how this varies across the double descent, and thus watch the competition between memorized data and induction heads for model capacity?

7 Conclusion

We've shown that small fractions of repeated data, if repeated at the right frequency, can cause surprisingly severe degradation to model performance. We show that this degradation scales predictably, occurs across datasets, and is associated with disproprotionate damage to internal mechanisms associated with generaliza tion, such as induction heads. In practical terms, these results provide a tool for predicting and diagnosing data-repetition-related problems in language models. In more conceptual terms, they are an example of a bridge between the macroscopic domain of scaling laws and the microscopic domain of mechanistic inter pretability, as well as a lens for gaining a more detailed understanding of how generalization and memoriza tion work. We believe these conceptual themes are promising ones, and hope to see more work that employs them.

Acknowledgments

We thank [tbd] for helpful feedback on the draft. We thank Daniela Amodei, Jamie Kerr, Jia Yuan Loke, Rebecca Raible, and Tim Telleen-Lawton for support with the project.

Author Contributions

References

[Advani and Saxe, 2017] Advani, M. S. and Saxe, A. M. (2017). High-dimensional dynamics of generaliza tion error in neural networks.

[Amodei et al., 2018] Amodei, D., Hernandez, D., Sastry, G., Clark, J., Brockman, G., and Sutskever, I. (2018). Ai and compute. *Heruntergeladen von https://blog. openai. com/aiand-compute*.

[Belkin et al., 2018] Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2018). Reconciling modern machine learning practice and the bias-variance trade-off.

[Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

[Carlini et al., 2020] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., Oprea, A., and Raffel, C. (2020). Extracting training data from large language models.

[Catherine et al., 2022] Catherine, O., Nelson, E., Neel, N., Nicholas, J., Nova, D., Tom, H., Ben, M., Amanda, A., Yuntao, B., Anna, C., Tom, C., Dawn, D., Deep, G., Zac, H.-D., Danny, H., Scott, J., Andy, J., Jackson, K., Liane, L., Kamal, N., Dario, A., Tom, B., Jack, C., Jared, K., Sam, M., and Chris, O. (2022). In-context learning and induction heads. *transfoerm-circuits.pub*.

[Elhage et al., 2021] Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread*. https://transformer-circuits.pub/2021/framework/index.html.

[Henighan et al., 2020] Henighan, T., Kaplan, J., Katz, M., Chen, M., Hesse, C., Jackson, J., Jun, H., Brown, T. B., Dhariwal, P., Gray, S., Hallacy, C., Mann, B., Radford, A., Ramesh, A., Ryder, N., Ziegler, D. M., Schulman, J., Amodei, D., and McCandlish, S. (2020). Scaling laws for autoregressive generative model ing.

[Hernandez and Brown, 2020] Hernandez, D. and Brown, T. B. (2020). Measuring the algorithmic efficiency of neural networks. *CoRR*, abs/2005.04305.

[Hernandez et al., 2021] Hernandez, D., Kaplan, J., Henighan, T., and McCandlish, S. (2021). Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*.

[Hoffmann et al., 2022] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. v. d., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. (2022). Training compute-optimal large language models.

[Kaplan et al., 2020] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models.

[Lee et al., 2021] Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. (2021). Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*.

[Nakkiran et al., 2019] Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. (2019). Deep double descent: Where bigger models and more data hurt.

[Radford et al., 2019] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

[Rae et al., 2021] Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., Rutherford, E., Hennigan, T., Menick, J., Cassirer, A., Powell, R., Driessche, G. v. d., Hendricks, L. A., Rauh, M., Huang, P.-S., Glaese, A., Welbl, J., Dathathri, S., Huang, S., Uesato, J., Mellor, J., Higgins, I., Creswell, A., McAleese, N., Wu, A., Elsen, E.,

Jayakumar, S., Buchatskaya, E., Budden, D., Sutherland, E., Simonyan, K., Paganini, M., Sifre, L., Martens, L., Li, X. L., Kuncoro, A., Nematzadeh, A., Gribovskaya, E., Donato, D., Lazaridou, A., Mensch, A., Lespiau, J.-B., Tsimpoukelli, M., Grigorev, N., Fritz, D., Sottiaux, T., Pajarskas, M., Pohlen, T., Gong, Z., Toyama, D., d'Autume, C. d. M., Li, Y., Terzi, T., Mikulik, V., Babuschkin, I., Clark, A., Casas, D. d. L., Guy, A., Jones, C., Bradbury, J., Johnson, M., Hechtman, B., Weidinger, L., Gabriel, I., Isaac, W., Lockhart, E., Osindero,

S., Rimell, L., Dyer, C., Vinyals, O., Ayoub, K., Stanway, J., Bennett, L., Hassabis, D., Kavukcuoglu, K., and Irving, G. (2021). Scaling language models: Methods, analysis, and amp; insights from training gopher.

[Ziegler,] Ziegler, A. Github copilot: Parrot or crow?