# Datasets, Documents, and Repetitions: The Practicalities of Unequal Data Quality

**Alex Fang**[1,2] **Hadi Pouransari**[1] **Matt Jordan**[3]
**Alexander Toshev**[1] **Vaishaal Shankar**[4] **Ludwig Schmidt**[2] **Tom Gunter**[1]
[1]Apple [2]Stanford [3]work done while at UT Austin [4]work done while at Apple

## Abstract

Data filtering has become a powerful tool for improving model performance while reducing computational cost. However, as large language model compute budgets continue to grow, the limited data volume provided by heavily filtered and deduplicated datasets will become a practical constraint. In efforts to better understand how to proceed, we study model performance at various compute budgets and across multiple pre-training datasets created through data filtering and deduplication. We find that, given appropriate modifications to the training recipe, repeating existing aggressively filtered datasets for up to ten epochs can outperform training on the ten times larger superset for a single epoch across multiple compute budget orders of magnitude. While this finding relies on repeating the dataset for many epochs, we also investigate repeats within these datasets at the document level. We find that not all documents within a dataset are equal, and we can create better datasets relative to a token budget by explicitly manipulating the counts of individual documents. We conclude by arguing that even as large language models scale, data filtering remains an important direction of research.

## 1 Introduction

Scaling data and compute has been the winning recipe for producing state of the art machine learning models. As costs increase with diminishing returns, recent work often points to high-quality data as a key factor for high model performance [30, 6]. One method of obtaining high quality data is through data filtering. However, as frontier level large language models (LLMs) scale in compute through increases in both parameter count and token count, there are concerns that we will run out of training data. Data filtering reduces the dataset size, creating a tension between data quality and data quantity. It is currently unclear whether the performance gains from data filtering can be sustained at scale through repetitions as an attempt to compensate for the reduction in dataset size.

Recent work has studied data filtering as a means of efficiently training strong models with limited compute [8, 15]. However, it is unclear if the findings are practical for training larger models. Unfortunately, existing frontier level LLMs lack transparency when it comes to their data filtering pipelines [30, 6, 1, 16]. This gap in knowledge adds to concerns about the practicality of data filtering on a scale.

Understanding the interaction between data filtering and scaling would be made easier if we better understood what it means to be a better dataset. Currently, the quality of a dataset is determined by using standardized benchmarks to evaluate the performance of a model trained on it. However, there are other traits we desire in the data beyond the performance of the raw downstream model, such as repeatability and performance over replacement. Developing a better understanding of these traits would allow for a more principled construction of better datasets.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-0999

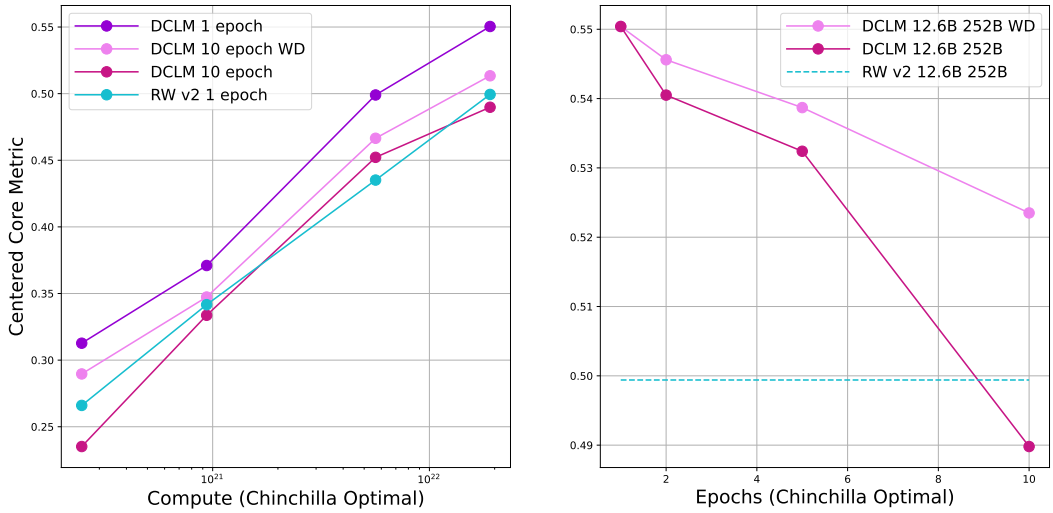



Figure 1: Repeating the aggressively filtered dataset DCLM-baseline for up to ten epochs consistently outperforms training on a single epoch of the ten times larger RefinedWeb supserset (cyan)—provided that we adapt the weight decay for the high-repetition runs. On the left, we show that this result appears consistent across compute budgets (model sizes of 1B, 3B, 7B, and 12.6B). On the right we show that, at the largest compute budget tested (12.6B parameters, 252B total tokens seen), adapting the weight decay as a function of repetition allows for a significantly better result versus training on the superset. Results are evaluated on the centered core metric from DCLM, which is a normalized average over 22 tasks. We also include a MMLU version of the right side in Appendix C.

In what follows, in Section 3 we study how datasets of different quality behave under repetition and explore how to improve multi-epoch performance. We compare DCLM-baseline filtering [15] with a superset comprising the strictly looser RefinedWeb filtering [21], as well as the commonly studied C4 dataset [23]. We find that while better datasets are not necessarily more repeatable, the interaction between dataset performance and repeatability depends on the compute allocation used to train the model. Furthermore, we study practical considerations for scaling up aggressively filtered datasets by repeating them for multiple epochs to account for the reduced number of tokens available. Datasets can be made more repeatable through regularizing factors, such as increasing weight decay and increasing tokens per parameter. We find that when training on the same number of total tokens, using ten epochs of the aggressively filtered DCLM-baseline can outperform a single epoch of its ten times larger superset—RefinedWeb.

Next, in Section 4 we investigate how to create better datasets by analyzing duplicate documents within the dataset. We find that web data filtering results in a bias towards documents that are more often partially duplicated, which was also reported in Penedo et al. [22]. Deduplicating the resulting dataset will therefore reduce the precision of the overall filtering step. We suggest simple methods that use a quality classifier to manipulate the individual counts of documents to improve dataset quality given a desired token budget, and show that this outperforms using only duplicate counts as in standard deduplication methods.

Lastly, we reflect on the role of data filtering for LLMs, and argue that it is a key ingredient for improving LLMs no matter the compute budget. We hope that this work not only clarifies some of the limitations of existing data filtering methods, but also demonstrates the value and practicality of data filtering, thus encouraging further research in this important direction.

## 2 Related work

**Data Filtering** We investigate whether data filtering is justifiable when it requires repeating to compensate for the decrease in tokens. Recent works such as DataComp-LM (DCLM) [15] and FineWeb-edu [22] have shown how data filtering can greatly improve performance. However, these datasets only retain a few trillion tokens, which may not be enough to train state-of-the-art models

https://doi.org/10.52202/085713-0999

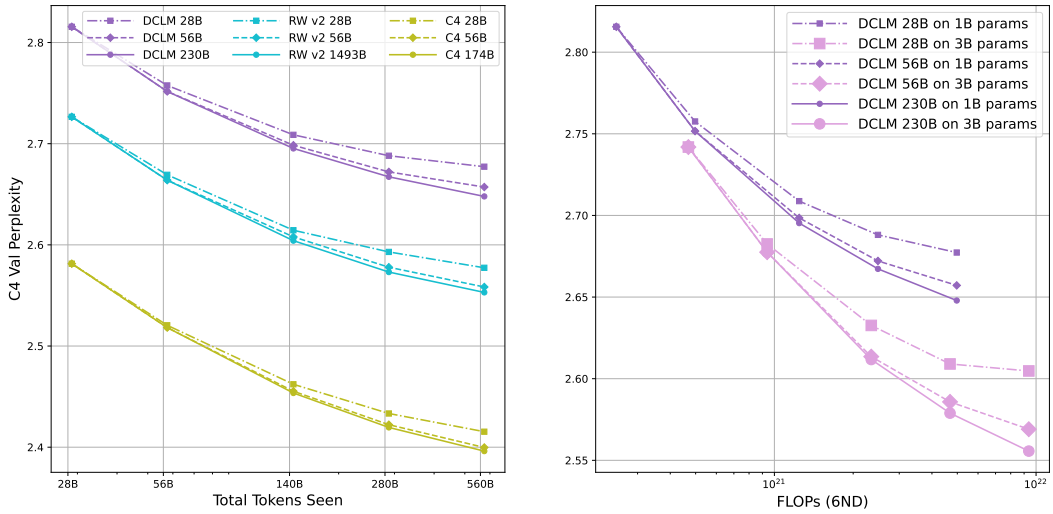



Figure 2: C4 perplexity as a function of training dataset and repetition count. Over-repeating data leads to diminishing returns irrespective of the training dataset chosen. For each dataset, we vary the number of unique tokens available, and then vary the training token budget. In the left plot we train 1B parameter models to show that heavy repetition results in a similar degradation in validation loss for all datasets tested. The right hand side shows that this effect holds at larger compute budgets for similarly over-trained models on the DCLM dataset.

unless trained for multiple epochs. Our study thus takes into account this scenario, where aggressively filtered datasets must be repeated, and compares them with larger but lower quality datasets.

**Repeatability** We build on the work in Muennighoff et al. [19], which studies training models that are data-constrained. They find that for a fixed compute budget, web-crawl data can be repeated four times before encountering significant diminishing returns compared to training on unique tokens. We will refer to this as repeatability, which is measured for a fixed training token budget by comparing the performance of repeating a subset for multiple epochs with that of a single epoch. Unlike this work, we examine the repeatability of crawl-data from two new angles: 1. whether dataset quality affects repetition resilience; 2. if regularization techniques can dampen the impact of over-repetition.

Goyal et al. [9] studies data filtering for CLIP models and argues that data filtering methods must take into account the computational budget. They find that training a ViT-B/32 for 640M samples on 12.8M LAION filtered samples is worse than 128M unfiltered samples, and that CLIP models trained on large amounts of compute should use less aggressive filtering. Although this differs from our findings, there are significant experimental differences due to modality and filtering techniques.

**Compute Allocation** Explicitly repeating tokens can lead to different optimal compute allocations. Hoffmann et al. [11] studies optimal allocation of parameters and tokens for a training computational budget. Their suggested ratio for parameters to tokens is called Chinchilla optimal, while training for more or less tokens can be considered overtraining or undertraining relative to that model size. Muennighoff et al. [19] find that for repeated data, overtraining is more efficient than staying at Chinchilla optimal.

**Deduplication** Prior work has shown that training on deduplicated data is desirable because it improves performance [13], reduces privacy risks [12], and reduces memorization [3]. However, more recent works like DCLM and FineWeb-edu use sharded deduplication for engineering and performance reasons. TxT360 [17] takes a slightly different approach, using global deduplication followed by upsampling by the natural distribution. We build on existing deduplication methods, and move toward upsampling based on quality metrics on a document level.


https://doi.org/10.52202/085713-0999

## 3 Repeating Filtered Datasets

In parallel to LLMs training on larger datasets, researchers are also focusing on filtering data to improve quality. Together, these factors accelerate concerns about running out of training data. In the earlier days of training LLMs, common practice was to train LLMs on a single epoch of data. This is in stark contrast to state-of-the-art vision models, where practitioners would repeat data for many epochs. Though single epoch training may have benefits such as preventing memorization [13, 3], in this work we study data repetition across datasets from a purely performance perspective. Muennighoff et al. [19] studied the repeatability of datasets in a data-constrained setting and found that datasets can be repeated for four epochs before noticing diminishing returns compared to training on fresh data. Building off this, we examine datasets of varying quality created through data filtering to understand whether these data filtering approaches affect repeatability. In this section we study datasets "as is" and treat them as a single unit rather than a collection of documents. We use original C4, DCLM's reproduction of RefinedWeb (we will refer to this as RefinedWeb and RW v2 interchangeably), and DCLM-baseline. See Appendix I to compare datasets "as is" with datasets that are deduplicated.

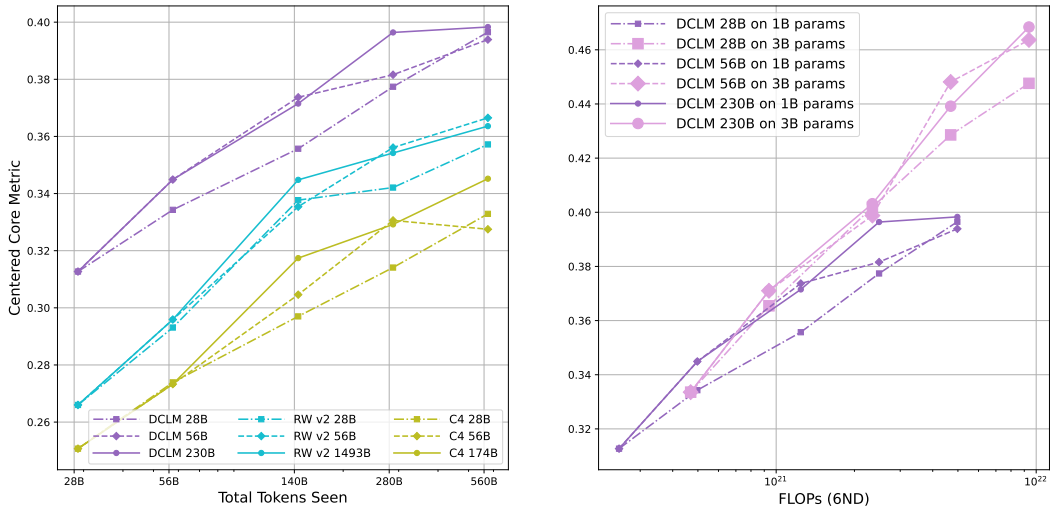



Figure 3: Downstream accuracy average of 22 tasks as different datasets are repeated and overtrained. The left side only contains 1B models across a variety of datasets and unique tokens, while the right side only trains on DCLM-baseline while varying in model size and unique tokens.

In Figure 2 we study a scenario similar to the findings in Muennighoff et al. [19], but instead start with Chinchilla optimal unique tokens for 1B parameter models. On the left side, we observe that the C4 validation perplexity eventually sees diminishing returns when training on C4 after repeating. In addition, we also train on higher quality datasets DCLM-baseline and RefinedWeb, and find that they have similar repeatability behavior when evaluating on C4 validation perplexity. In this setting we start with a Chinchilla optimal compute allocation and then repeat epochs by fixing the model size while increasing the token count, and compare this to datasets with more available tokens. On the right side we vary model size, but maintain the same dataset sizes that we repeat. This figure suggests that increasing model size while maintaining dataset size decreases repeatability, but there is not enough information to make a conclusion so we will explore this further later in the section. When we move to evaluating downstream accuracy across a 22 task subset from LLM Foundry (referred to as core centered metric in DCLM), we see in Figure 3 that quantifying repeatability is even less clear. It appears that when comparing using downstream accuracy, repeatability is more difficult to measure, and it is unclear if repeating meets the same diminishing returns seen when evaluating C4 perplexity. What is common between the two evaluation settings is that datasets that achieve better performance do not appear to be more repeatable than inferior datasets when examining diminishing returns.

Although better performing datasets are not more repeatable by preventing diminishing returns, in the overtraining setting (Figure 3) better datasets repeated multiple times clearly maintain better performance than an inferior dataset that is not repeated. However, it is important to note that

https://doi.org/10.52202/085713-0999

overtraining is practical only in certain settings with priorities such as minimizing the inference cost, and there is an assumption in Figure 3 that there are at least enough tokens to train Chinchilla optimal. This assumption does not always hold when LLMs are scaled up, especially when data filtering is involved.

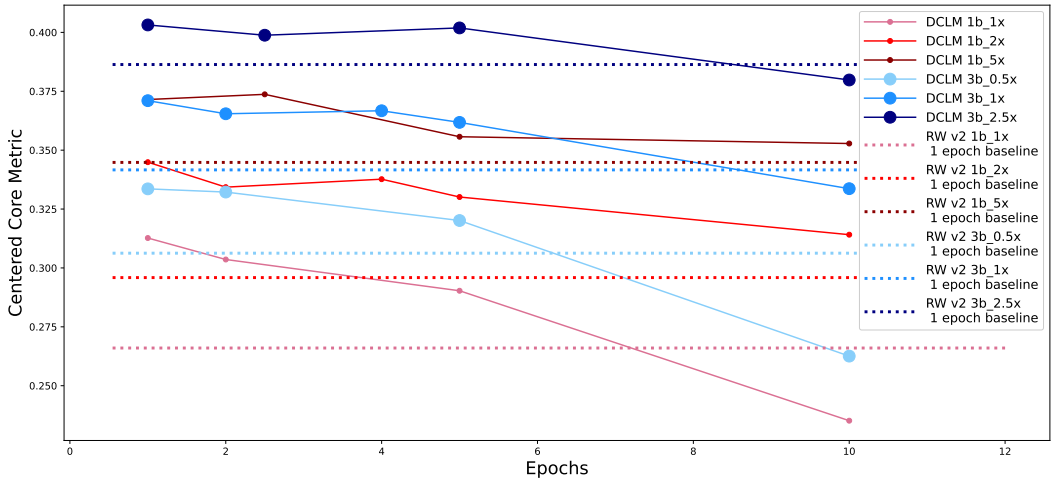



Figure 4: Downstream accuracy average of 22 tasks of DCLM-baseline dataset as we fix total tokens trained and vary epochs and unique tokens seen. Dotted horizontal line represents training one epoch on the RefinedWeb baseline, which we compare against because DCLM-baseline is additional filtering on top of RefinedWeb. Legend denotes model size and multiplier to scale tokens seen at Chinchilla optimal.

In contrast to repeating datasets by training for longer, we next examine fixing the token budget and repeating by decreasing the number of unique tokens. This scenario forces us to examine cases where there are fewer unique tokens available than Chinchilla optimal. In Figure 4 we compare for a fixed model size and total token budget how performance varies as you reduce the number of unique tokens available, causing an increase in the number of repeats for which the model sees the dataset. When there is no longer at least the Chinchilla optimal number of tokens available, repeating the DCLM-baseline dataset with too few unique tokens can lead to worse performance than training for a single epoch on its superset of RefinedWeb filtered data. On the other hand, compute allocations with high tokens per parameter seem to benefit more from higher quality datasets, since DCLM can be repeated more times to equalize performance with unrepeated RefinedWeb. Tokens per parameter is the main factor, as increasing model size while maintaining token count decreases DCLM repeatability to equalize with RefinedWeb, and increasing token count while maintaining tokens per parameter does not consistently increase DCLM repeatability to equalize performance with unrepeated RefinedWeb (Figure1, left). This would also explain why unrepeated RefinedWeb does worse than heavily repeated DCLM in Figure3. But there are increasing diminishing returns to performance with respect to compute when increasing tokens per parameter, so there still needs to be a balance when doing data filtering.

Table 1: Weight decay can reduce performance degradation from repeating data. All models are 12.6B parameters trained for 252B total tokens. We report results using the centered core metric. The default weight decay is 0.0316. Note that we confirm WD 1x is optimal for a single epoch of RefinedWeb, as well as for a single epoch of DCLM.

| Repeats | WD 1x | WD 2x | WD 3x |
|---|---|---|---|
| 10 epochs | 0.489 | 0.523 | 0.513 |
| 5 epochs | 0.532 | 0.539 | 0.528 |
| 2 epochs | 0.541 | 0.546 | 0.545 |
| 1 epochs | 0.550 | 0.547 | 0.546 |


https://doi.org/10.52202/085713-0999

We can mitigate performance degradation caused by repeating data with regularization in the form of increasing weight decay. In Table 1, we see that the optimal weight decay value increases as we increase the number of repeats. A reasonable schedule is to scale weight decay by roughly the square root of the number of repeats. With the optimal weight decay values, we see in Figure 1 that training for ten epochs of DCLM-baseline outperforms training on RefinedWeb filtered data for a single epoch. This is notable because DCLM-baseline takes the top 10% of documents of RefinedWeb as scored by a FastText classifier. However, it is important to note that this does not mean DCLM-baseline is a strictly better dataset than RefinedWeb filtering. When data is a much more constrained resource than compute, it is likely that diminishing returns from heavily repeating DCLM-baseline will perform worse than repeating RefinedWeb filtered data a couple of times.

## 4 Repeating Documents To Create Better Datasets

Table 2: 7B models trained on DCLM-baseline for 138B tokens (Chinchilla optimal) using different subsampling methods.

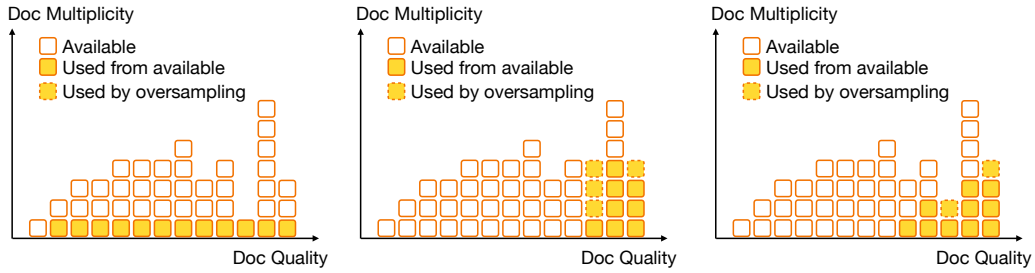
| Subsampling Method | CORE (22 task average) | MMLU |
|---|---|---|
| Duplicate-aware subsample | 34.1 | 25.1 |
| Global deduplication then subsample | 41.7 | 28.5 |
| Uniform subsample | 44.4 | 42.2 |
| Keep docs with duplicates $\geq 7$, then dedup | 45.8 | 40.0 |

We have established that, even as compute budgets increase, models can remain robust to dataset repetition given the right training recipe. We now investigate whether high quality data repetition may be best implemented at a document level, as opposed to increased epochs on the entire dataset. To better understand this question, we dive into DCLM-baseline, which contains many duplicates (fuzzy and exact) because it uses sharded deduplication due to engineering constraints. Penedo et al. [22] also uses sharded deduplication, observing that global deduplication does not always improve performance. This is in contrast to prior standard practices of deduplication, which recommended global deduplication and training for a single epoch on the resulting dataset.

The performance gains from data filtering clearly demonstrate that not all documents are of equal quality. We can also observe this by comparing subsampling strategies. The standard strategy is uniform subsampling, which results in a dataset that is diverse with very few duplicates, while more likely keeping documents that have more duplicates. A second strategy is to run minhash-based global fuzzy deduplication before subsampling down to the desired token count. Compared to uniform subsampling, incorporating global fuzzy deduplication treats documents with different duplicate counts equally, but both strategies should have few or no duplicates in the resulting subset if subsampling significantly. Lastly, we can mimic the duplication profile of the original dataset by using duplicate-aware subsampling. For each group of duplicates, this strategy keeps or removes the entire group with probability equivalent to the duplicate count normalized by the desired dataset reduction. This strategy results in a subset with a significant number of duplicates compared to the other subsampling strategies.

In Table 2 we observe that duplicate-aware subsampling performs the worst, suggesting that keeping a significant number of duplicates hurts performance. Uniform subsampling does significantly better than global deduplication before subsampling, which shows that documents in DCLM-baseline with high duplicate count are likely to be of higher quality. This is further supported by the fact that keeping unique copies for documents with greater than seven duplicates achieves competitive performance to uniform; however, this removes over 96% of the documents in DCLM-baseline.

Given a token budget and a metric correlated with document quality, we can take advantage of these findings by manipulating counts of individual documents through resampling. This is analogous to mixing but at document level instead of forming buckets. We use a function for assigning the counts of deduplicated documents within the dataset, adjust the range of the function by estimating how many more times you should repeat the best document in the dataset before including the worst, and adjust the domain of the function based on the number of documents desired. Figure 5 contains examples of such functions, which include greedily selecting the best documents with a constant function equal to a certain count (y=count), and a linear function which would increase the diversity




(a) Greedy 1 copy (b) Greedy 4 copies (c) Linear up to 4 copies

Figure 5: Examples of strategies for count manipulation. Note that Greedy 1 copy is similar to global deduplication, and can result in worse performance if document quality varies significantly. For example, if the dataset is split evenly between high quality documents and low quality documents, it may be better to repeat the high quality documents.

Listing 1: Pseudocode to sample documents using count manipulation with a simple greedy function.

```
def greedy_function_example(score):
    # given a score, return how many copies of the accopanying
    # document we want.
    # threshold is pre-calculated from all scores.
    # in this example, threshold keeps 1 copy of top
    # documents to reach desired output document count.
    if score >= threshold:
        return 1
    else:
        return 0

def sample_count_manipulation(documents, counts, scores, function):
    output = []
    for doc, count, score in zip(documents, counts, score):
        target_count = function(score)
        for _ in range(target_count):
            if np.random.rand() <= 1 / count:
                output.append(doc)
    return output
```

of documents when compared to the equivalent greedy strategy with the same maximum count value. Key to this approach is having a metric correlated with document quality. Here we use the worst ranking within the dataset between FastText score and pre-deduplication document count. In Table 3 we subsample from the 3.8T DCLM-baseline down to 138B tokens for training a 7B model, and our results show that doing basic document count manipulation can outperform the various baseline subsampling approaches. In this setting of extreme subsampling that reduces trillions of tokens to billions of tokens, the best strategy is greedily selecting high quality documents according to our metric without repeating.

Table 3: 7B models trained for 138B tokens on DCLM-baseline with different document count manipulation strategies. Figure 5 visualizes these strategies. Ensemble uses the worst rank between FastText score and duplication count.

| Subsampling Method | CORE (22 task average) | MMLU |
|---|---|---|
| DCLM-baseline uniform subsample | 44.4 | 42.2 |
| Greedy 1 copy (dup count) | 41.7 | 28.5 |
| Greedy 4 copies (ensemble) | 44.6 | 38.5 |
| Greedy 1 copy (ensemble) | 46.1 | 46.0 |
| Linear up to 4 copies (ensemble) | 46.0 | 44.2 |


https://doi.org/10.52202/085713-0999

Table 4: 7B models trained for 138B tokens by doing count manipulation on a duplicate-aware subsampled 280B tokens from DCLM-baseline. In this setting, the greedy yet diverse strategy sees each document approximately 3 times. Metric used is FastText score. Note that using duplication count as the metric would be ineffective here because there are not enough unique documents.

| Subsampling Method | CORE (22 task average) | MMLU |
|---|---|---|
| DCLM-baseline DA subsample | 34.1 | 25.1 |
| Greedy Diversity | 40.4 | 28.3 |
| Greedy 5 copies | 40.4 | 29.4 |
| Linear up to 5 copies | 41.2 | 30.4 |

The results in Table 3 are unusual in that we are taking an already highly filtered dataset and subsampling even further. Naturally, there are enough high quality tokens to train without repeats, hence the success of the greedy yet diverse strategy. However, this becomes impractical as we scale and there are no longer enough unique tokens to support this strategy. Therefore we now investigate the setting where the dataset we are trying to manipulate is closer in size to the desired token count. We duplicate-aware subsample DCLM-baseline down to 280B tokens and apply count manipulation to produce 138B tokens for training a 7B model. As we scale back up, this would allow us to use the strategy on the entire DCLM-baseline to create a dataset close to two trillion tokens. In Table 4, we see that the best strategy is applying a linear function to balance the tradeoff between exploring diverse documents and exploiting high quality documents. Contrasting the best strategy in the two settings suggests that using a good count manipulation strategy is more important in the setting where data is constrained.

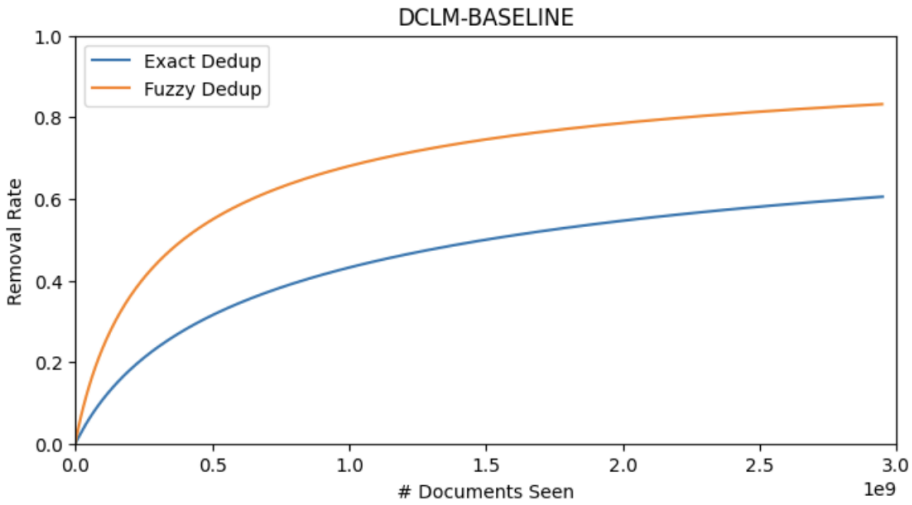



Figure 6: In DCLM-baseline, the proportion of duplicates inside the dataset increases as the number of documents increases. In Appendix E we see this trend also holds for RefinedWeb, but with lower removal rates.

Combining duplicate-aware subsampling with count manipulation is a powerful tool for studying datasets. Increasing dataset size typically refers to obtaining new tokens of the same quality. DCLM does this by increasing the number of Common Crawl WARC files at the same rate as increasing the training token budget. However, as seen in Figure 6, when the number of WARC files increases, so does the number of duplicates. We find that DCLM-baseline constructed from the DCLM 7B_2x pool contains 33% fuzzy duplicates, but DCLM-baseline constructed from 20 times more WARC files contains 83% fuzzy duplicates. By uniformly subsampling to run smaller scale experiments, this experiment design uses both new data and repetitions when scaling back up. Additionally, it is impractical to deduplicate unfiltered data due to engineering constraints caused by the amount of data, as well as design decisions like what type of deduplication to use. Duplicate-aware subsampling gives us the tools to design experiments where data scaling happens through providing new data of equal quality.

Count manipulation differs from filtering because filtering is restricted by the amount of documents available in the pool. Even with the perfect data quality metric, filtering alone may not create the best dataset because high quality documents do not necessarily appear many times in the dataset. Count manipulation allows for oversampling and is thus complementary to filtering, depending on traditional filtering techniques like deduplication and filtering based on a quality metric. Furthermore, while the experiments presented have built off of DCLM-baseline, it is important to remember that DCLM-baseline is an aggressively filtered dataset. Count manipulation should be just as if not more impactful on less filtered datasets because there is a wider range in quality of documents within the dataset, making it more likely that repetition of high quality documents improves performance more than seeing low quality documents for the first time.

Revisiting the ideas in Section 3, we can replace repeating entire datasets with count manipulation. Though it may appear that datasets as a whole can only repeat a set number of times before significant diminishing returns, when examining individual documents it is likely that repeating a high quality document many times is still better than seeing a low quality document for the first time. This tradeoff can be captured by the function applied to count manipulation.

## 5 How to Use Better Datasets

We would like to conclude by arguing that creating better datasets is still an important area of research. Our findings in Section 3 suggest that an aggressively filtered dataset like DCLM-baseline can be used as pre-training data at scale by repeating the dataset for multiple epochs with the right recipe adaptations, but this is just one of the many use cases for data filtering. Additionally, our findings in Section 4 can be used to improve existing datasets, and can be especially useful in situations where there is a constraint on the availability of unique data.

Better datasets are especially beneficial for training smaller scale LLMs. This can happen when practitioners want to reduce inference cost by training a smaller model or when there is a limit on the amount of compute available. In this setting, the amount of data available is no longer a key constraint, which places a greater importance on the tokens that are actually seen. Recent works such as DCLM [15] and FineWeb-edu [22] have shown how data filtering can greatly improve performance across models that only vary by the training data used.

High quality data is often used during continual pre-training. Works like DCLM and Llama 3 [6] use this technique to introduce higher quality data at the end of pre-training. This technique is separate and often complementary to fine-tuning techniques like instruction tuning. However, we find in Appendix D that for 7B models trained on 138B tokens, using DCLM-baseline for the last 10% or 20% of training performs the same as mixing that same proportion into RefinedWeb. Nonetheless, this technique can be used to introduce high quality data that was not ready at the beginning of the training run, and may help more for smaller amounts of data or different domains. Feng et al. [7] have shown that curriculum learning can be effective when targeting particular domains like math and code.

Lastly, while DCLM-baseline may not contain sufficient tokens for all use cases, it is important to remember that almost all pre-training datasets have a filtered component and we can change the filtering ratio based on needs. Throughout this work we have compared DCLM to RefinedWeb because DCLM is created through additional filtering on top of RefinedWeb filtering. As compute budgets for LLM pre-training increase, more emphasis will be placed on looser and more efficient data filtering that achieves high recall.

**Limitations** Data filtering techniques have their limitations as well. Our understanding of models is only as good as the evaluations we rely on. It is possible that data filtering reduces capabilities or removes knowledge that we desire in our models, but we can only hope to quantify this through evaluations that measure the effects that we care about. Additionally, we focus strictly on web-scraped datasets created for general language modeling. Dataset properties may differ when studying different domains such as code and math. Finally, our experiments are smaller scale than frontier level models, so our trends must continue to hold as we scale in order for these findings to apply to frontier level models.

 https://doi.org/10.52202/085713-0999

## Impact Statement

This paper studies data filtering and its role in improving Large Language Models. The societal consequences are similar to those typically discussed in Machine Learning and Large Language Models. One additional consequence we would like to highlight is the potential for bias amplification in dataset creation. Although we are not releasing additional datasets and models, repeating or resampling data can contribute toward certain biases, so researchers should take proper care to evaluate for such when developing and releasing datasets and models.

## References


[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

[3] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/forum?id=TatRHT_1cK`.

[4] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

[5] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[6] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL `https://doi.org/10.48550/arXiv.2407.21783`.

[7] Steven Feng, Shrimai Prabhumoye, Kezhi Kong, Dan Su, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Maximize your data's potential: Enhancing llm accuracy with two-phase pretraining. *arXiv preprint arXiv:2412.15285*, 2024.

[8] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Mitchell Wortsman Ryan Marten, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Mehdi Cherti Richard Vencu, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. `https://arxiv.org/abs/2304.14108`.

[9] Sachin Goyal, Pratyush Maini, Zachary C. Lipton, Aditi Raghunathan, and J. Zico Kolter. Scaling laws for data filtering– data curation cannot be compute agnostic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22702–22711, June 2024.

[10] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[11] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. https://arxiv.org/abs/2203.15556.

[12] Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR, 2022.

[13] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8424–8445. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.577. URL https://doi.org/10.18653/v1/2022.acl-long.577.

[14] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

[15] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*, 2024.

[16] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[17] Zhengzhong Liu, Bowen Tan, Hongyi Wang, Willie Neiswanger, Tianhua Tao, Haonan Li, Fajri Koto, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. Llm360 k2: Building a 65b 360-open-source large language model from scratch. *arXiv preprint arXiv:2501.07124*, 2025.

[18] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

[19] Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376, 2023.

[20] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

[21] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023.

[22] Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. *CoRR*, abs/2406.17557, 2024. doi: 10.48550/ARXIV.2406.17557. URL https://doi.org/10.48550/arXiv.2406.17557.

[23] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[24] P Rajpurkar. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[25] Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

[26] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI spring symposium series*, 2011.

 https://doi.org/10.52202/085713-0999

[27] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[28] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

[29] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

[30] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[31] Yury Tokpanov, Paolo Glorioso, Quentin Anthony, and Beren Millidge. Zyda-2: a 5 trillion token high-quality dataset. *arXiv preprint arXiv:2411.06068*, 2024.

[32] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[33] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

## A Training Setup

We use OpenLM (MIT license) for training. Standard model configurations and hyperparameters can be found in the DCLM code base. OpenLM is used for all 1B and 3B models, as well as 7B models in Section 4. These models are trained on Nvidia H100s.

We use AXLearn for 7B and 12B models in Section 3. These models are trained on TPUs.

The main datasets we trained on are DCLM (CC-by-4.0), RefinedWeb (v2, from DCLM) (CC-by-4.0), and C4 (CC-by-4.0 or odc-by depending on source).

## B Evaluation

We evaluate on the centered core metric from DCLM, which consists of the following 22 tasks: AGI Eval LSAT Analytical Reasoning [33], ARC Challenge [5], ARC Easy [5], BIG-bench cs algorithms [28], BIG-bench dyck languages [28], BIG bench language identification [28], BIG-bench operators [28], BIG-bench: wikidata [28], BIG-bench repeat copy logic [28], BoolQ [4], Commonsense QA [29], COPA [26], CoQA [25], HellaSwag (zero-shot and few-shot) [32], Jeopardy (MosaicML), LAMBADA [20], OpenBook QA [18], PIQA [2], SQuAD [24], Winograd Schema Challenge [14], Winogrande [27]

We also evaluate 7B and larger models on MMLU (few-shot) [10].

## C MMLU Repeatability

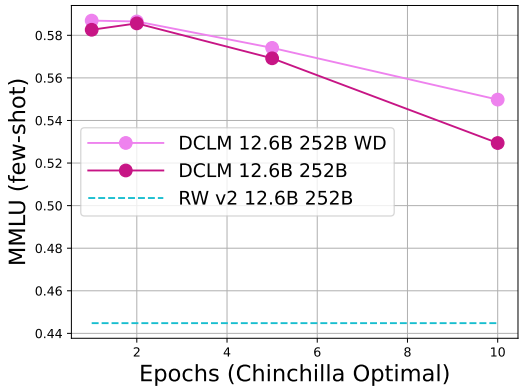



Figure 7: Right side of Figure 1 except measuring MMLU (few-shot) on the y-axis. We see that DCLM even with repeats significantly outperforms RefinedWeb on MMLU, even more so when comparing against centered core metric.

## D Continual Pre-training

We train 7B models for 138B tokens to investigate the effect of curriculum learning on general pre-training data of different quality.

| Dataset | Core | MMLU | Extended |
|---|---|---|---|
| Shuffle 90% RW and 10% DCLM | 41.3 | 25.7 | 22.4 |
| 90% RW followed by 10% DCLM | 41.0 | 26.2 | 21.9 |
| Shuffle 80% RW and 20% DCLM | 41.6 | 27.3 | 22.5 |
| 80% RW followed by 20% DCLM | 41.6 | 26.1 | 22.2 |


https://doi.org/10.52202/085713-0999

## E Deduplication Trends as Datasets Scale

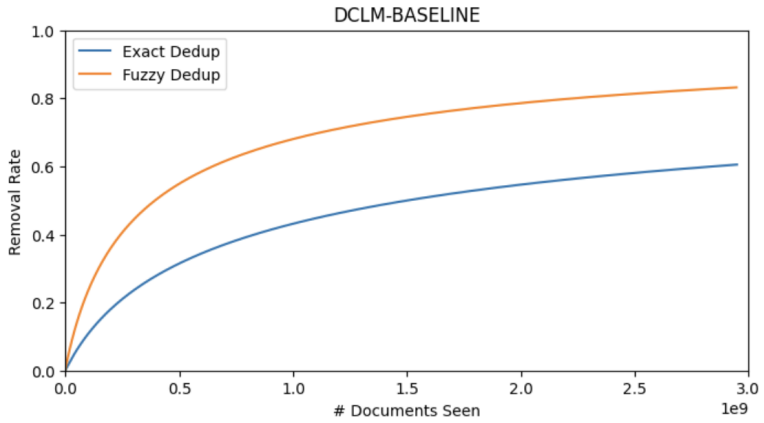



Figure 8: In DCLM-baseline, the proportion of duplicates inside the dataset increases as the number of documents increases. In Figure 9 we see this trend also holds for RefinedWeb, but with lower removal rates.

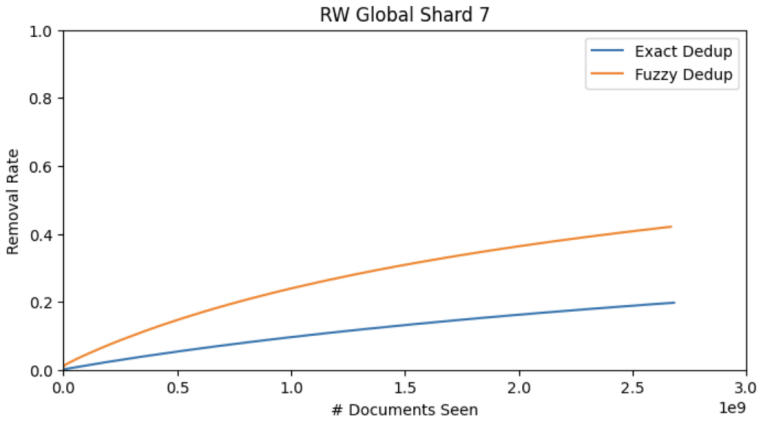



Figure 9: In RefinedWeb, the proportion of duplicates inside the dataset increases as the number of documents increases. Notice that these removal rates are lower than those of DCLM-baseline.

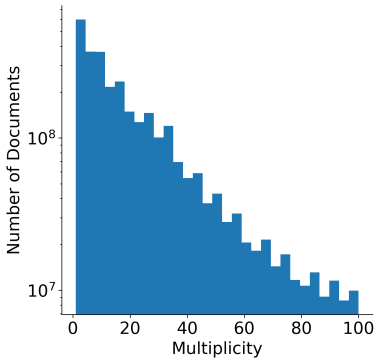
https://doi.org/10.52202/085713-0999

## F Fuzzy vs Exact Duplicates

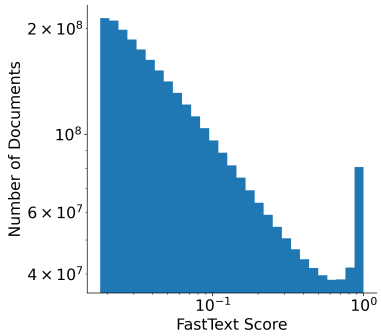
Some initial 7B parameter (Chinchilla optimal) experiments to explore the differences between fuzzy and exact duplicates. For the second pair of experiments, floor f refers to keeping documents that have at least f fuzzy copies, while ceil c refers to keeping no more than c copies. Additional work is required to better understand the differences between fuzzy and exact duplicates.

| Dataset | Core | MMLU | Extended |
|---|---|---|---|
| DCLM-baseline duplicate-aware subsample (fuzzy) | 34.1 | 25.1 | 17.5 |
| DCLM-baseline duplicate-aware subsample (exact) | 33.8 | 25.3 | 17.7 |
| DCLM-baseline floor 21 ceil 4 (fuzzy) | 42.7 | 37.4 | 25.6 |
| DCLM-baseline floor 21 ceil 1, 4 epochs (exact) | 43.3 | 32.4 | 24.0 |

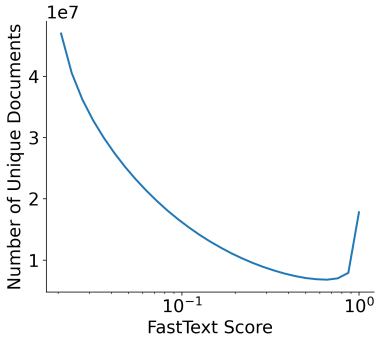
## G DCLM-baseline Count Manipulation Metric Statistics



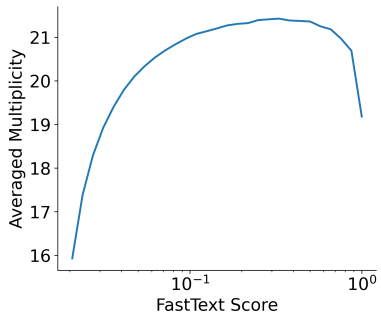

(a) DCLM-baseline distribution of fuzzy duplicates




(b) DCLM-baseline distribution of FastText scores




(c) DCLM-baseline unique documents by FastText score




(d) DCLM-baseline average fuzzy duplicate count by FastText score


https://doi.org/10.52202/085713-0999

## H Deduplication Methods Details and Comparisons

We use minhash-based fuzzy deduplication written in Rust. Our hyperparameters are 5-gram tokens and 14 bands of size 9.

Here we compare our fuzzy dedup results against the implementation from Tokpanov et al. [31] (Zyda-2 DCLM-dedup), as well as DCLM-baseline (no additional global dedup).

Table 5: 7B Chinchilla optimal deduplication comparison

| Dataset | Core | MMLU |
|---|---|---|
| DCLM Our Fuzzy Dedup | 41.7 | 28.5 |
| Zyda-2 DCLM-dedup | 43.6 | 29.4 |
| DCLM-baseline | 44.4 | 42.2 |

Table 6: 12B Chinchilla optimal deduplication comparison

| Dataset | Core | MMLU |
|---|---|---|
| Zyda-2 DCLM-dedup | 53.3 | 56.8 |
| DCLM-baseline | 55.0 | 58.3 |

## I Deduplication Interaction with Repeats

In Section 3, we study datasets as is, leaving the possibility of duplicates already existing in the datasets. Here we compare repeating a randomly sampled subset with repeating a global fuzzy deduplicated (Zyda-2 DCLM-dedup) subset. Note that randomly subsampling produces datasets where documents either have no or low repeats because the subset is much smaller than the pool that we subsample from, and the datasets studied in Section 3 already go through some sort of deduplication.

The results below suggest that low repeat counts favor DCLM-baseline, while high repeat counts favor the deduplicated version. At the compute regimes we study our findings should apply to both scenarios, but we hypothesize that deduplication or resampling becomes more important as we use higher repeat counts at larger scales.

| Epochs | DCLM Core | Dedup Core | DCLM MMLU | Dedup MMLU |
|---|---|---|---|---|
| 1 epoch | 55.0 | 53.3 | 58.3 | 56.8 |
| 2 epoch | 54.0 | 53.5 | 58.6 | 57.9 |
| 5 epoch | 53.2 | 51.9 | 56.9 | 55.5 |
| 10 epoch | 49.0 | 49.5 | 52.9 | 51.4 |

# J Additional Count Manipulation Details

## J.1 Count Manipulation Strategies

One way to determine the count function corresponding to each of the strategies in Figure 5 is as follows:

1. Greedy 1 copy: Find the threshold such that the desired number of documents after deduplication is above the threshold. Then sample all documents probabilistically by keeping each document with probability $\frac{1}{duplicate\_count}$ if the document's score is over the threshold.

2. Greedy 4 copies: Find the threshold such that the desired number of documents after deduplication is above the threshold. Then sample all documents probabilistically by looping 4 times per document over the threshold, with each trial attempting to keep that document with probability $\frac{1}{duplicate\_count}$.

3. Linear up to 4 copies: First, determine the maximum number of copies (in this case $max\_copies = 4$) you would like to keep for the best document. Next, determine the goal number of document ($goal\_docs$) you would like to keep. The number of unique documents for each copy count is $bucket\_size = \frac{goal\_docs}{\sum_{i=1}^{max\_copies} i}$. Using this, calculate the top $max\_copies$ thresholds using the deduplicated documents. Lastly, for each document (non-deduplicated), use the thresholds to identify which copy count bucket ($i$) it belongs to, and loop $i$ times, with each trial attempting to keep that document with probability $\frac{1}{duplicate\_count}$.

## J.2 Ensembling

The ensembling strategy in Table 3 is calculated by first finding each document's duplication count and DCLM FastText classifier score, then assigning each document its ranking (e.g. best score is 1, worst score is number of documents) for each metric, and finally taking the maximum of those values. In this new metric, lower is now better. It is important for each metric used in the ensemble to not only be correlated with data quality, but also be fine-grained. Otherwise, in Table 4 duplication count is no longer a useful metric because there are not enough unique documents.

## J.3 Metric Comprison Between FastText Score and Ensemble

Table 7: Ensemble results from Table 3 compared with using DCLM FastText score as metric.

| Subsampling Method | CORE (22 task average) | MMLU |
|---|---|---|
| DCLM-baseline uniform subsample | 44.4 | 42.2 |
| Greedy 1 copy (dup count) | 41.7 | 28.5 |
| Greedy 4 copies (ensemble) | 44.6 | 38.5 |
| Greedy 1 copy (ensemble) | 46.1 | 46.0 |
| Linear up to 4 copies (ensemble) | 46.0 | 44.2 |
| Greedy 4 copies (FastText) | 41.6 | 45.3 |
| Greedy 1 copy (FastText) | 45.5 | 45.4 |
| Linear up to 4 copies (FastText) | 42.1 | 46.7 |


https://doi.org/10.52202/085713-0999

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract includes our main findings, which are repeated and expanded upon in the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a limitations section at the end of our main paper, as well as discuss potential problems with data filtering.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not have theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experiment details (e.g. pseudocode, descriptions, hyperparameters) can be found throughout the paper and the appendix, and code needed to reproduce experiments are publicly available unless specifically omitted for now to maintain anonymity.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**


https://doi.org/10.52202/085713-0999

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The datasets we work with are available online, mainly through CommonCrawl's bucket and HuggingFace. One of the training frameworks we used is publicly available, while the other one will be available once anonymity is no longer necessary.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and test details are either specified in the paper, or are available through the publicly available training framework.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Experiments are large scale training runs, which means multiple runs to establish error bars is impractical.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We specify in the appendix the type of computer resources used for our training jobs; however, we do not give specific computational costs or total hours.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the code of ethics and we believe that our research follows the code.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include an impact section at the end of the main paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.


https://doi.org/10.52202/085713-0999

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release new data or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the datasets and other assets used, and mention their licenses in the appendix.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

 https://doi.org/10.52202/085713-0999

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We train LLMs and then evaluate on them, but do not use it as a tool in other parts of the research process.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.