# Scalable Data Ablation Approximations for Language Models through Modular Training and Merging

**Clara Na**[1,2] **Ian Magnusson**[1,3] **Ananya Harsh Jha**[1,3]
**Tom Sherborne**[4] **Emma Strubell**[1,2] **Jesse Dodge**[1] **Pradeep Dasigi**[1]
[1]Allen Institute for AI [2]Carnegie Mellon University [3]University of Washington
[4]Cohere
csna@cs.cmu.edu

## Abstract

Training data compositions for Large Language Models (LLMs) can significantly affect their downstream performance. However, a thorough data ablation study exploring large sets of candidate data mixtures is typically prohibitively expensive since the full effect is seen only after training the models; this can lead practitioners to settle for sub-optimal data mixtures. We propose an efficient method for approximating data ablations which trains individual models on subsets of a training corpus and reuses them across evaluations of combinations of subsets. In continued pre-training experiments, we find that, given an arbitrary evaluation set, the perplexity score of a single model trained on a candidate set of data is strongly correlated with perplexity scores of parameter averages of models trained on distinct partitions of that data. From this finding, we posit that researchers and practitioners can conduct inexpensive simulations of data ablations by maintaining a pool of models that were each trained on partitions of a large training corpus, and assessing candidate data mixtures by evaluating parameter averages of combinations of these models. This approach allows for substantial improvements in amortized training efficiency – scaling only linearly with respect to new data – by enabling reuse of previous training computation, opening new avenues for improving model performance through rigorous, incremental data assessment and mixing.

## 1 Introduction

As Large Language Models (LLMs) and their training corpora have grown in scale, it is increasingly costly not only to *train* an LLM given a fixed recipe, but also to *develop* recipes for training new language models with improved capabilities. Design decisions can span many factors such as model architecture, optimization techniques, and others; one critical aspect of LLM development is *training data composition*.

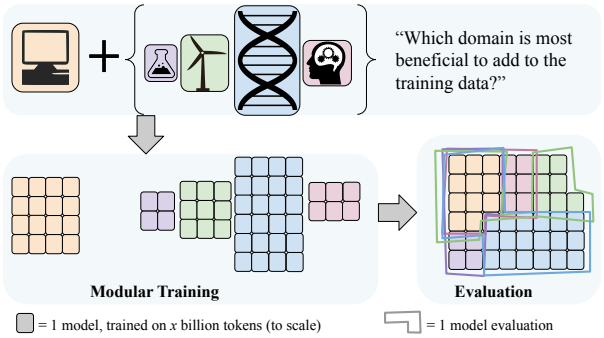



Figure 1: Given a corpus containing multiple subsets of data, a traditional approach to studying the effects of training on different data mixtures calls for training models on each candidate data mixture. Our strategy *reuses* training on data shared across candidate mixtures, by 1) conducting **modular training** of many models on equally sized "base unit" data partitions and 2) performing **evaluation** on *parameter averages* of model combinations. We find that this yields useful *proxy* metrics for predicting perplexity scores on arbitrary evaluation domains, such that we can simulate comprehensive data ablation studies at a fraction of the cost.

It is especially important to find a suitable training data recipe as decisions made in the pre-training stage can propagate downstream to domain- and task-specific settings (Yadlowsky et al., 2023). Meanwhile, it is common for organizations that develop models to release reports that describe only the optimal configurations that were eventually found (if that) and the capabilities of the final models, with little to no mention of the development process and the suboptimal configurations that were also assessed.

This leaves open questions about the impacts of different pre-training data compositions on language model performance, which is increasingly expensive to explore rigorously as models and corpora grow larger. Though previous studies have empirically tested the effects of different pre-training data compositions (Rae et al., 2022; Longpre et al., 2023; Muennighoff et al., 2023) or presented strategies for data selection for training data mixtures



*Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21125–21141
November 12-16, 2024 

(Xie et al., 2023; Xia et al., 2024), scaling empirical experiments beyond a handful of candidate mixtures tends to be prohibitively expensive. Moreover, there has been limited exploration into *why* certain domains and data mixtures are more suited for certain evaluation domains than others.

We propose an efficient method for approximating language model perplexity performance for a large collection of pre-training data compositions, using only a fraction of the computational cost of training on the full set of data compositions considered. Specifically, we find that a model's perplexity performance on a set of held-out or arbitrary out-of-domain data tends to be strongly correlated to the perplexity score of a *parameter average* (Izmailov et al., 2018; Wortsman et al., 2022) of individual models trained on distinct partitions of the data.

Since in our approach assessing the utility of new data requires training only on the new data, the amount of training needed to assess data mixtures grows only linearly as a function of new data to be assessed, compared to polynomial or combinatorial growth in the naive approach; see §2.1.1.

We run extensive experiments on pre-training language models, primarily in a continued pre-training setting, and evaluating both in- and out-of-domain performance of mixtures of text from different domains. Throughout our experiments (§5), we vary the compositions of our data mixtures with respect to total size, component data and model size, and component diversity, in order to explore a set of research questions:

1. Can we predict the perplexity evaluation performance of a model trained on a mixture of data using the performance of models trained on component data *partitions* (§5.1)?

2. Can we simulate data mixing experiments with unevenly sized partitions (§5.2)?

3. Can we simulate data mixing experiments combining high-level sources (§5.3)?

4. How do our methods apply to larger models (§5.4)?

Some of our main findings include:

1. On arbitrary evaluation domains of interest (which may be out of distribution), a *parameter average* of individual models, trained in parallel on subsets of a candidate data mixture, can predict perplexity evaluation scores of a model trained on the full data mixture.

2. If candidate mixtures are comprised of domains that are uneven in size, divide the training corpus they belong to into evenly sized fundamental units to be used as component subsets. To evaluate the candidate data mixtures, consider the performance of *weighted* parameter averages of the component units' models for each mixture.

3. Expect more reliability at the "optimal" end of the performance distribution. It is often easier to find the most favorable data mixtures by our suggested proxy metrics than the least favorable data mixtures.

Additionally, we release training code, datasets, and models[1]. Although the current study is necessarily an exploratory investigation limited to 130 million and 1.1 billion parameter models in continued pre-training (Gururangan et al., 2020; Gupta et al., 2023) settings with carefully defined domains, we hope that our results will prompt further study across additional settings and scales of data and model size. Ultimately, our goal is to enable researchers and practitioners to simulate more principled and thorough data ablation studies as they develop new LLMs and curate new training data corpora for them.

## 2 Modular training for data ablations

We characterize our proposed approach and its asymptotic complexity vs the traditional method for conducting data ablations.

The asymptotic efficiency advantage of our method is comparable to that found comparing a naive recursive implementation of an algorithm with combinatorial complexity, to one that uses memoization. Our method analogously benefits from an effective decomposition of a large training corpus into subsets whose corresponding trained models are reused repeatedly across many candidate data mixtures ("overlapping subproblems"). Our method also similarly requires additional cache storage for component models, relatively inexpensive compared to additional compute.

### 2.1 Formalism

We consider a set of $n$ partitions $\{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_n\}$ comprising a corpus $\mathcal{C}$, where we are interested in understanding the effect of training a model on different combinations of partitions. We define a *data*

[1] https://github.com/clarana/ez-data-ablations

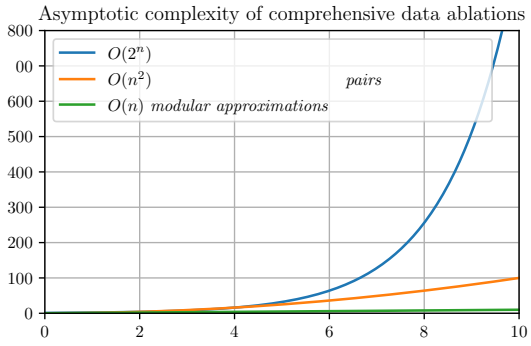


Figure 2: Our method (green) has only linear runtime complexity with respect to the number of unique partitions in a corpus. This allows us to simulate *comprehensive* data ablations at extremely low cost compared to naive training on all possible partition combinations.

*mixture*, $\vec{d}$, as a vector of labels indicating presence or absence for each partition $i$ in the mixture, $\vec{d}_i = \mathbb{1}_{\vec{d}}(P_i)$. We define a *data ablation study* by 1) a *set* of data mixtures $\{\vec{d}\} \subseteq \{0,1\}^n$ that we are interested in evaluating and 2) a set of evaluation domains.

Let $k = ||\vec{d}||_1$ denote the number of partitions included in a data mixture, where $k \leq n$. Note that for a fixed $k$, $|\{\vec{d} \in \{0,1\}^n : \sum_{i=1}^n \vec{d}_i = k\}|$ is equal to $\binom{n}{k}$. The total number of possible $\vec{d}$ for a given $\mathcal{C}$ is the exponential $|\{0,1\}^n| = 2^n$.

#### 2.1.1 Complexity analysis

Using the naive approach, the training cost of evaluating a set of training data mixtures $\{\vec{d} \in \{0,1\}^n : \sum_{i=1}^n \vec{d}_i = k\}$ in full for a fixed $k$ scales with polynomial complexity, $O(n^k)$; the cost of evaluating *all* $\vec{d} \in \{0,1\}^n$ scales exponentially, $O(2^n)$.

In comparison, our approximation method's cost is $O(n)$ and calls for the caching and reuse of models trained on partitions common across across candidate mixtures $\vec{d}$; thus, training is required only once for each partition. Additional evaluations of data mixtures only require additional training for any previously *unseen* partitions (and caching of the resulting new models).[2]

Though in practical settings we may not necessarily want to compare *all* possible data mixtures $\vec{d}$ (and indeed, we do not explicitly verify whether it is possible to simulate an ablation study where $\exists\, \vec{d}_a, \vec{d}_b \in \{\vec{d}\}$ such that $||\vec{d}_a||_1 \neq ||\vec{d}_b||_1$), the efficiency gains from our approximation method are substantial even when we limit ourselves to studying $\{\vec{d} \in \{0,1\}^n : \sum_{i=1}^n \vec{d}_i = k\}$ for some fixed $k$ (in Fig. 2, we have $k = 2 \rightarrow O(n^2)$).

[2]Reaping the full computational efficiency benefits of our method requires *caching* of all trained individual models; space complexity is $O(n)$ in our method, whereas the naive method is $O(1)$ and does not require caching trained models.

### 2.2 Method setting and description

In our work, we explore corpora $\mathcal{C}$ that can be divided into partitions $\mathcal{P}$ using metadata, for example along high-level source domains (e.g. academic documents vs source code), topic (e.g. biology vs sociology), or temporality (e.g. news articles by year). In conducting a training data ablation study $\{\vec{d}\} \subseteq \{0,1\}^n$, one might wish to evaluate data mixtures' alignments with a handful of specific domains (e.g. certain domains in Paloma; Magnusson et al. (2023)), investigate the differential effect of training on particular types of text (e.g. the newest batch of data from a social media website, or scientific documents vs patents), or identify particularly influential or unhelpful subsets of training data for an evaluation domain of interest.

In practice, partitions of interest may be uneven in size. We find that training and parameter averaging more models on smaller component partitions is favorable to training and parameter averaging fewer models on larger partitions within a larger, unevenly distributed data mixture, which we show in §5.2 and §5.3. Formally, our recommended strategy is as follows:

1. Given a corpus $\mathcal{C}$, identify data partitions of interest $\mathcal{P}_1, \mathcal{P}_2, .., \mathcal{P}_n \in \mathcal{C}$, based on ablation studies of interest. These partitions can be of any size. However, assume that within a single ablation study $\{\vec{d}\} \subseteq \{0,1\}^n$, we have some fixed $k = ||\vec{d}_a||_1 = ||\vec{d}_b||_1 \forall\, \vec{d}_a, \vec{d}_b \in \{\vec{d}\}$.

2. Further *split* (or recombine) partitions $\mathcal{P}_i$ into *equally sized* "base unit" partitions $\mathcal{P}'_1, \mathcal{P}'_2, ..., \mathcal{P}'_m \in \mathcal{C}$. *Train* a copy of a language model on each of the $m$ partitions $\mathcal{P}'_j$, on an equal number of tokens $t$.

3. For each ablation study $\{\vec{d}\} \subseteq \{0,1\}^n$, evaluate data mixtures of $k$ partitions on arbitrary evaluation domains of interest using *parameter averages* of the base component models from Step 2. We posit that researchers and practitioners can use the resulting proxy perplexity scores to identify data mixtures $\vec{d}' \in \{0,1\}^n$ that are better or worse fit to these evaluation domains of interest.

## 3 Data

The training datasets for our main experiments are derived from the Semantic Scholar Open Research

Corpus (S2ORC) (Lo et al., 2020) and M2D2 Wikipedia (Reid et al., 2022). Additionally, for a given model size, we begin by pre-training a model on a general "seed" corpus of Gutenberg books and English Wikipedia. All models of the same size used in the current study are initialized from a copy of this seed model before being trained on a combination S2ORC and M2D2 Wikipedia text. We evaluate our models using held-out sets from the continued pre-training data, as well as a subset of the validation sets from Paloma (Magnusson et al., 2023), an evaluation dataset for assessing language model fit on various domain-specific text. Table 1 describes characteristics and sizes of these datasets.

| Purpose | Data source | # Part. | Tokens |
|---|---|---|---|
| Pretraining (PT) | Wiki+Gutenberg | — | 10.6B |
| Continued PT, Eval | S2ORC | 128 | 43.9B |
| Continued PT, Eval | M2D2 Wiki | 11 | 2.9B |
| Evaluation only | Paloma[3] | 9 | 27M |

Table 1: Statistics of datasets. Token counts use the GPT-NeoX-20B tokenizer[4] (Black et al., 2022), and include training and held-out sets when applicable.

As ours is an exploratory study, we experiment primarily at 1) smaller scales and 2) with data mixtures of carefully constructed domains *within* datasets of continual pre-training scale.

Data is partitioned topically and temporally in the case of S2ORC, and topically in M2D2 Wikipedia[5]. For M2D2 Wikipedia, we use the L1 (top-level) domains as originally defined by Reid et al. (2022) with a median of 282 million tokens each. As S2ORC is not pre-partitioned, we construct 22 partitions $\mathcal{P}$ for S2ORC based on metadata for field of study (FoS), and 128 "base unit" partitions $\mathcal{P}'$ of roughly equal size, around 287 million tokens each, based on FoS and year.

Some documents are tagged with multiple fields of study (see Table 5 in Appendix A.1) in which case we use the first FoS listed. We treat the 1% of documents without an explicit tagging as their own FoS, indicated by "NA". Documents' publication years range from 1970 to 2022. All partitions are associated with one or more publication years and a single field of study (with "NA" treated as its own FoS), with two exceptions: Geography was merged into Environmental Science, and Law was merged into Political Science, due to their high co-occurrence and relatively small standalone size.

Table 6 in Appendix A.1 shows partition counts for each field of study: fields of study with larger partition counts are larger than fields of study with smaller partition counts. Humanities fields are overall less heavily represented in the corpus; many have only a single partition (containing all documents from 1970 through 2022), while some of the largest partitions contain over 1 billion tokens from papers published in a single year within a field of study. In practice, in §5, we *upsample* smaller partitions when training such that data mixtures from multiple partitions within S2ORC include roughly equal representation from each partition.

All seed and continued pre-training data was *decontaminated* against Paloma evaluation data, and all continued pre-training data was decontaminated against its respective evaluation splits. Following Soldaini et al. (2024)'s standard for decontamination on Paloma (Magnusson et al., 2023), a Bloom filter was used to exclude all documents containing least one "contaminated" paragraph (at least 13 tokens long and found in the evaluation set). In general, we use S2ORC and M2D2 Wikipedia as examples of datasets at continued pre-training scale that can be partitioned along natural inherent groupings in their distributions, but we encourage future work exploring different data.

## 4 Methods

For our base experiments, we train small (130m parameter) decoder-only models with a PaLM-like architecture, on independent partitions of the Semantic Scholar Open Research Corpus (S2ORC) (Lo et al., 2020) and M2D2 Wikipedia (Reid et al., 2022), defined as discussed in §3. In a subset of our experiments, we involve larger (1.1b parameter) models. Models share an initial optimization trajectory of 42 billion tokens of "seed" pre-training on a general corpus of 10 billion tokens consisting of Gutenberg books and English Wikipedia. Each model is then trained on a specific partition within S2ORC or M2D2 Wikipedia for some multiple of 1 billion tokens, depending on the experimental setup. We evaluate models using per-token perplexity scores on domain-specific validation sets

[3]Paloma (Magnusson et al., 2023) contains 16 top-level sources and 546 domains, with 123.68 million tokens total; we use a diverse subset of the validation sets, listed in Table 3.

[4]Following Groeneveld et al. (2024), with the addition of special tokens.

[5]Reid et al. (2022) also introduce M2D2 S2ORC, a corpus of academic documents partitioned hierarchically into high- and low-level topical domains. We instead use the larger S2ORC corpus (Lo et al., 2020) in its decontaminated, Dolma-like (Soldaini et al., 2024) format, which retains document boundaries not easily recoverable from the M2D2 format

as well as out-of-domain sets from Paloma (Magnusson et al., 2023); all perplexity scores reported in this study are on held-out sets. We use the GPT-NeoX-20B tokenizer (Black et al., 2022) as it is used by Groeneveld et al. (2024) with the addition of 3 special tokens for masking personally identifiable information[6].

See Appendix A.3 for more detailed reporting on training settings, hyperparameters, and hardware.

## 5 Experiments

In our experiments, we have:

$$\mathcal{C} = \bigcup_{f=1}^{22} \mathcal{P}_{S2ORC_f} \cup \bigcup_{g=1}^{11} \mathcal{P}_{Wiki_g} = \bigcup_{i=1}^{128} \mathcal{P}'_{S2ORC_i} \cup \bigcup_{g=1}^{11} \mathcal{P}'_{Wiki_g}$$

Partitions $\mathcal{P}' \in \mathcal{C}$ are disjoint, but we have no evidence to suggest that this is a strict requirement.[7] Total number of partitions is $n = 22 + 11 = 33$ in most of our experiments, except in §5.1 where we sample partitions directly from $\{\mathcal{P}' \in \mathcal{C}\}$ and so $n = 128 + 11 = 139$. The set of evaluation domains in each experiment includes 1) in-domain datasets from each data mixture $\vec{d}$'s respective combined held-out dataset and 2) nine domains from Paloma as OOD or domain-shifted data.

### 5.0.1 Figure information

In all experiments, for each evaluation dataset, we plot the perplexity evaluation performance of the model trained sequentially[8] on the entire data mixture (SEQ) to various *proxy* perplexity evaluations: each point lies along the same $y = a$ as at least one other point associated with the same data mixture (and a different proxy signal). We compare SEQ model scores to the average of the perplexity scores of the individual (IND) models trained on the components of the data mixture, as well as to the perplexity scores of (MERGED) parameter averages of base component models. These IND models' average scores are written as "mean IND" in the figures. "IND ID" scores are similar but differ in that they include only in-domain held-out evaluations in the average).

[6] gpt-neox-olmo-dolma-v1_5

[7] In fact, our $\mathcal{P}' \in \mathcal{C}$ almost certainly have *distributional* overlap, though very significant overlap may reduce our method's effectiveness.

[8] We describe training as "sequential" to distinguish from the modular training on individual partitions of data that can be performed in parallel, *not* to refer to any particular curricular ordering of the training data.

In §5.2, we discuss effective vs ineffective partitioning and weighting for data mixtures with larger or uneven domains. In particular, we compare parameter averages of $k = ||\vec{d}||_1 = 2$ unevenly trained component models with "macro"- and "micro"-(parameter) averages of many more, evenly trained "base unit" models trained on individual base components $\mathcal{P}'$. In a macro-MERGED model, the $k$ high-level component models are themselves parameter averages of unequal numbers of equally trained base unit models; a micro-MERGED model is the direct uniform parameter average of the equally trained base unit models. We find that macro-MERGED model performance tends to be the most reliable proxy metric in settings where the partitions are sampled from the same distribution of sizes, while micro-MERGED model performance is more reliable in settings with an expected skew.

Overall, we find that SEQ scores on arbitrary evaluation domains (which may be OOD or domain-shifted with respect to the training data) are strongly correlated with the perplexity scores of appropriately composed MERGED models, though there are settings where mean scores of IND models are also strongly correlated.

### 5.1 Base partition studies

We begin with a controlled continued pre-training setting, in which which we select training data mixtures comprised of *pairs* ($k = 2$) or *triplets* ($k = 3$) of equally sized data partitions $\mathcal{P}'$ (not $\mathcal{P}$ as we do in other experiments) from the same high-level data source domain. We replicate the $k = 2$ experiments in both S2ORC ($|\{\vec{d}'\}| = 50$) and M2D2 Wikipedia ($|\{\vec{d}'\}| = 20$), where data mixtures $\vec{d}'$ are sampled uniformly and without replacement from the 128 topical-temporal partitions in S2ORC we have defined in §3 (and none from Wikipedia; $\{\vec{d}' \subseteq \{0,1\} : ||\vec{d}'_{S2ORC}||_1 = 2, ||\vec{d}'_{Wiki}||_1 = 0\}$), and from the 11 L1 topical partitions as originally defined by Reid et al. (2022) ($\{\vec{d}' \subseteq \{0,1\} : ||\vec{d}'_{S2ORC}||_1 = 0, ||\vec{d}'_{Wiki}||_1 = 2\}$). For the $k = 3$ experiment, we sample 50 mixtures from $\{\vec{d}' \subseteq \{0,1\} : ||\vec{d}'_{S2ORC}||_1 = 3, ||\vec{d}'_{Wiki}||_1 = 0\}$.

Starting with the same seed model initialization in all cases, we continue pre-training on these mixtures for 1 billion tokens for each partition.

Figure 3 displays the strong relationships we find between SEQ models' performance and our proxy metrics of choice for all three of these settings, and Table 2 reports the correlation scores. The data

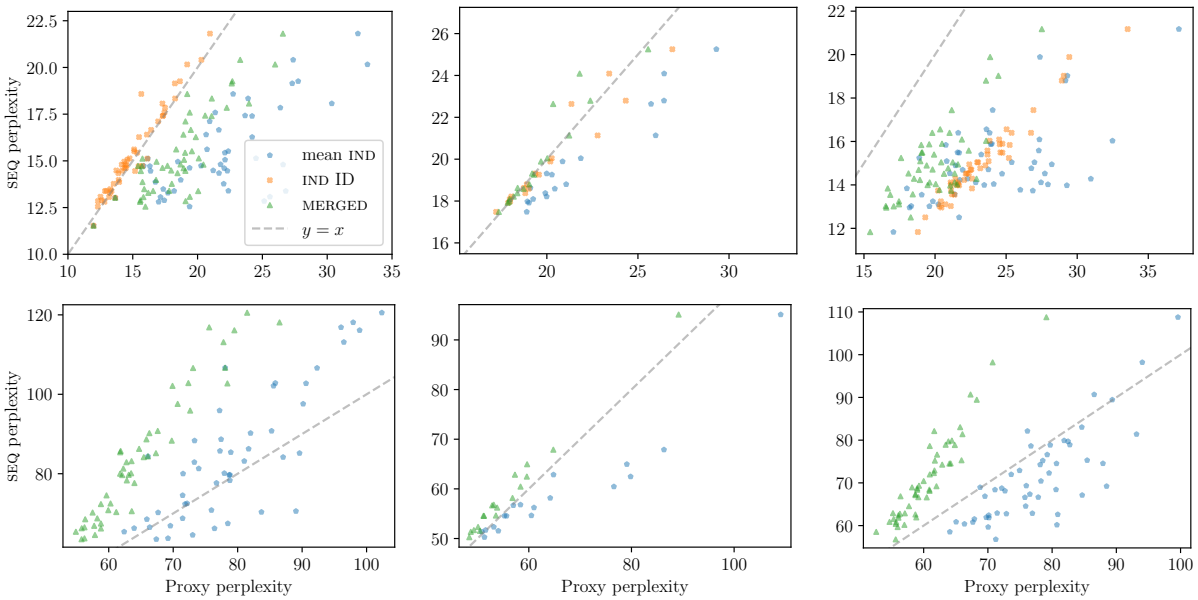



Figure 3: SEQ(uentially trained), MERGED, and IND(ividual) models trained on random pairs of S2ORC base partitions (**left**), pairs of M2D2 Wikipedia partitions (**middle**) and triples of S2ORC base partitions (**right**), evaluated on respective held-out sets of the same (**top**) and on nine subsets of Paloma (**bottom**). Each point lies along the same $y-$value for SEQ score as the other proxy evaluation score(s) for the same data mixture. For held-out sets of the training data, the SEQ performance on the combined dataset correlates most strongly with the average of the component models' respective in-domain evaluations ("IND ID"), compared to MERGED model performance or mean IND scores. However, on OOD Paloma data, the Pearson's correlation is highest between SEQ and MERGED models. See Table 2 for correlation values and §5.0.1 for definitions of terms.

mixtures vary in component domain similarity – for example, in the S2ORC settings, some pairs consist of older and newer partitions from the same FoS, while other pairs differ in both recency and topic. Although greater intra-mixture similarity is related to lower perplexity scores on the held-out sets of training data mixtures, the strong correlation between SEQ and proxy model perplexities is seen in both low and high perplexity scores.

| $\mathcal{P}'$ Expt | Eval | IND ID | mean IND | MERGED |
|---|---|---|---|---|
| S2ORC | ID | **0.968** | 0.711 | 0.846 |
| $k = 2$ | Paloma | - | 0.819 | **0.961** |
| Wiki | ID | 0.962 | **0.966** | 0.948 |
| $k = 2$ | Paloma | - | 0.942 | **0.993** |
| S2ORC | ID | **0.977** | 0.584 | 0.77 |
| $k = 3$ | Paloma | - | 0.799 | **0.951** |

Table 2: Pearson's correlation scores between SEQ model perplexity scores and proxy perplexity scores for experiments in §5.1.

In the following sections, we present studies exploring larger data mixture sizes (§5.2), lower intra-mix similarity (§5.3), and larger model sizes (§5.4)

## 5.2 Varying data mixture partition sizes

We analyze a setting with larger, unevenly sized $\mathcal{P}$ with $k = 2$. We verify that models trained

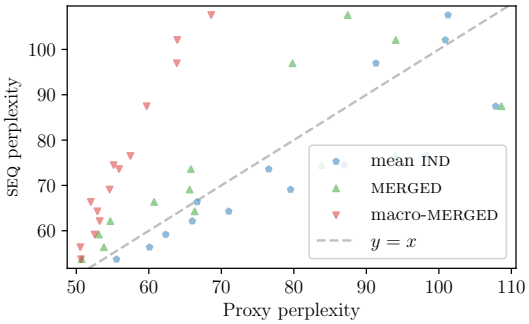



Figure 4: SEQ vs. proxy macro-averaged Paloma perplexity scores of models trained on differently sized partitions $\mathcal{P}$. We see that the macro-MERGED scores are the most reliable metric (Pearson's correlation 0.984).

for 1 billion tokens each on base unit partitions $\mathcal{P}' \in \mathcal{P}$ exhibit evaluation scores in their parameter averaged forms predictive of SEQ scores $\forall\ \mathcal{P} \in \mathcal{C}$ (see Appendix A.4). We then move to sampling from the $n = 33$ higher-level partitions $\mathcal{P}$, starting first with pairs ($k = 2$) of fields of study from S2ORC: we sample 14 data mixtures from $\{\vec{d} \subseteq \{0, 1\} : ||\vec{d}_{S2ORC}||_1 = 2, ||\vec{d}_{Wiki}||_1 = 0\}$.

Different from before, we compare *multiple* types of parameter averaged models as candidate proxies for the SEQ model in each data mixture.

| | Average IND scores | | MERGED scores | | |
|---|---|---|---|---|---|
| | MERGED | SEQ | SEQ | macro- | micro- |
| $\mathcal{P}_1$ | 0.844 | 0.826 | 0.763 | 0.929 | 0.937 |
| $\mathcal{P}_2$ | 0.909 | 0.930 | 0.914 | 0.959 | 0.946 |
| $\mathcal{P}_1 + \mathcal{P}_2$ | 0.856 | 0.844 | 0.755 | 0.944 | 0.938 |
| M2D2 S2 | 0.869 | 0.908 | 0.608 | 0.894 | **0.918** |
| M2D2 Wiki | 0.905 | 0.886 | 0.822 | **0.966** | 0.858 |
| Wiki-103 | 0.927 | 0.885 | 0.783 | **0.983** | 0.867 |
| PTB | 0.880 | 0.835 | 0.694 | **0.929** | 0.773 |
| 4chan | 0.874 | 0.883 | 0.866 | **0.905** | 0.785 |
| c4-en | 0.905 | 0.863 | 0.798 | **0.985** | 0.849 |
| mc4-en | 0.844 | 0.836 | 0.770 | **0.969** | 0.829 |
| RedPajama | 0.790 | 0.902 | 0.838 | **0.930** | 0.852 |
| Manosphere | 0.917 | 0.919 | 0.895 | **0.976** | 0.867 |
| Avg (macro) | 0.910 | 0.882 | 0.790 | **0.984** | 0.848 |

Table 3: Pearson correlation scores for our experiment mixing unevenly sized partitions, described in §5.2. We report correlation scores between SEQ models' perplexity evaluation scores vs candidate proxy evaluation scores, on various held-out and OOD sets. The first three rows are evaluations on held-out sets from respective data mixtures $\vec{d}$ and have IND ID scores > 0.98. The two labels in the first row distinguish between averages of "IND" perplexity evaluations (which may themselves be of MERGED parameter averages or individual SEQ models), and evaluations of "MERGED" parameter averages of models (whose components may be SEQ models, MERGED models, or base component models).

In one variation (MERGED), we evaluate the parameter average of the two models that were each trained for potentially *unequal* optimization trajectory lengths on *unequal* amounts of data. Each component model in this case was trained on data for an entire FoS for a *multiple* of 1 billion tokens, where the multiplier was $|\mathcal{P}|$, the number of base units $\mathcal{P}'$ comprising the field of study partition $\mathcal{P}$.

We describe the other two types of parameter averages as *macro*-MERGED and *micro*-MERGED models, where the fundamental component models were each trained for only 1 billion tokens on a single base unit temporal partition $\mathcal{P}'$ belonging to the field of study $\mathcal{P}$. Each macro-MERGED model in this experiment is created by 1) forming the uniform parameter average using base models trained on each $\mathcal{P}' \in \mathcal{P}$ from $\vec{d}$, and then 2) forming the uniform parameter average of the $k = 2$ resulting parameter averaged models. Each micro-MERGED model is a uniform parameter average of all individual base models trained on each $\mathcal{P}'$ in both $\mathcal{P}$ in the data mixture. Each SEQ model in this study was trained for $(|\mathcal{P}_1| + |\mathcal{P}_2|) * 1$ billion tokens.

Table 3 shows Pearson correlation scores between SEQ and proxy metrics for the experiment depicted in Figure 4. Macro-MERGED model evaluations are the most useful proxy for SEQ perplexity evaluation scores in this setting, where, notably, though the data mixtures can contain partitions of uneven size, the *expected values* are equal because they are sampled from the same distribution.

### 5.3 Mixed sources

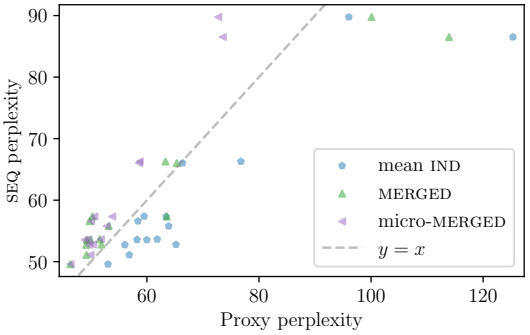



Figure 5: SEQ vs. proxy macro-averaged Paloma perplexity scores of models trained on data mixtures containing data from multiple sources. As in §5.2, it is beneficial to merge models trained on similar amounts of data, but here, *micro*-MERGED models are more useful – 0.989 corr. with SEQ, vs. 0.963 (MERGE of the SEQ models for each $\mathcal{P}$), 0.916 (mean IND scores), 0.807 (macro-MERGED), or 0.759 (mean MERGED scores).

We present a setting where data mixtures have multiple high-level source domains in their partitions, such that the documents in a data mixture are not necessarily similar in format. We again have $k = 2$ and $n = 33$, and we sample 15 data mixtures from $\{\vec{d} \subseteq \{0, 1\} : ||\vec{d}_{S2ORC}||_1 = 1, ||\vec{d}_{Wiki}||_1 = 1\}$. We assign training durations to our component and SEQ models according to the rules of §5.2, but as $\mathcal{P}_2$ is always a Wikipedia L1 domain, we have $|\mathcal{P}_2| = 1 \; \forall \; \vec{d}$ and therefore *more uneven* data mixtures. In Figure 5, we show that when when data mixtures combine these high-level domains, the micro-MERGED models for each data mixture are highly correlated with SEQ models in their perplexity scores on arbitrary evaluation domains.

#### 5.3.1 Fixing one data source

In a variation of the previous experiment, we fix the specific L1 domain (we choose $\mathcal{P}'_{Tech}$: "Technology and applied sciences" $\subset \mathcal{P}_{Wiki}$) across all experiments: we use all 22 data mixtures from $\{\vec{d} \subseteq \{0, 1\} : ||\vec{d}_{S2ORC}||_1 = 1, \mathcal{P}'_{Tech} \in \vec{d}\}$.

We argue that our approximation method can be applied in a practical setting where, for example, we wish to find auxiliary data that is beneficial to some existing training data, with respect to some evaluation domain(s); training on a performant data mixture selected by proxy metric yields a performant SEQ model. Results are discussed with Figure 8 in Appendix A.5.

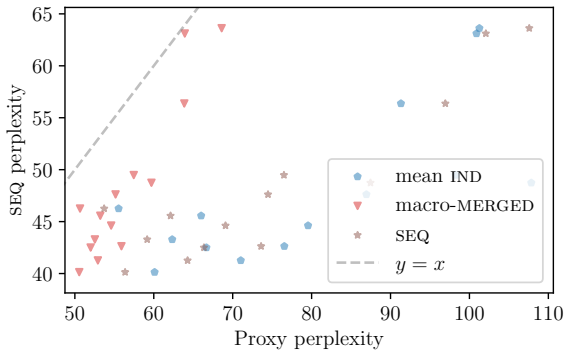



Figure 6: Macro-averaged Paloma perplexity scores of 1.1b SEQ models vs. proxy scores of 130m models trained on differently sized partitions $\mathcal{P}$. We again see that the macro-MERGED scores are the most reliable metric, with a Pearson's correlation of 0.926. Notably, macro-MERGED scores of the 130m models are on average more predictive of the 1.1b models' SEQ scores than the 130m models' own SEQ scores.

| | Avg IND scores | | MERGED scores | | | IND |
|---|---|---|---|---|---|---|
| | MERG. | SEQ | SEQ | macro | micro | SEQ |
| $\mathcal{P}_1$ | 0.785 | 0.751 | 0.695 | 0.898 | 0.872 | 0.983 |
| $\mathcal{P}_2$ | 0.896 | 0.915 | 0.882 | 0.952 | 0.908 | 0.990 |
| $\mathcal{P}_1 + \mathcal{P}_2$ | 0.802 | 0.785 | 0.701 | 0.911 | 0.891 | 0.986 |
| M2D2 S2 | 0.720 | 0.829 | 0.467 | 0.716 | 0.740 | 0.909 |
| M2D2 Wiki | 0.864 | 0.855 | 0.810 | 0.945 | 0.861 | 0.972 |
| Wiki-103 | 0.881 | 0.853 | 0.755 | 0.954 | 0.851 | 0.977 |
| PTB | 0.709 | 0.636 | 0.475 | 0.780 | 0.576 | 0.791 |
| 4chan | 0.778 | 0.613 | 0.631 | 0.914 | 0.760 | 0.849 |
| c4-en | 0.800 | 0.720 | 0.647 | 0.935 | 0.789 | 0.930 |
| mc4-en | 0.660 | 0.487 | 0.429 | 0.793 | 0.650 | 0.739 |
| RedPajama | 0.787 | 0.808 | 0.738 | 0.938 | 0.862 | 0.926 |
| Manosphere | 0.854 | 0.832 | 0.816 | 0.948 | 0.833 | 0.967 |
| Avg | 0.810 | 0.729 | 0.635 | 0.926 | 0.772 | 0.907 |

Table 4: For the experiment described in §5.4, Pearson correlation scores between 1.1b SEQ models' perplexity evaluation scores vs candidate proxy evaluation scores of 130m models, on various held-out and OOD sets. Here, the SEQ scores of 130m SEQ models are included for comparison.

### 5.4 Applicability to larger model scales

We also experiment with larger (1.1 billion parameter) models and present evidence that our approximation methods are useful beyond the smaller model scale we primarily experiment with, in both the sense that 1) proxy evaluations of 1.1b parameter models can be used to select performant data mixtures, and that 2) proxy evaluations of 130m parameter models can be used to select performant data mixtures for 1.1b parameter models. The latter finding has significance as an *orthogonally* beneficial feature of our proposed approximation method,[9] along with the asymptotic complexity benefit. We highlight the latter case by adapting the previously studied experimental setup with $k = 2$ larger, unevenly sized $\mathcal{P}$ from S2ORC (depicted in Figure 4 and Table 3 from §5.2) to compare the same proxy model metrics with 1.1b SEQ model scores on the same candidate data mixtures. The results of this experiment are shown in Figure 6 and Table 4.

In Appendix A.6, we discuss additional experiments with 1.1b models, where we study all $\{\vec{d} \subseteq \{0,1\} : ||\vec{d}_{S2ORC}||_1 = 2, ||\vec{d}_{Wiki}||_1 = 0, |\mathcal{P}| = 1 \; \forall \; \mathcal{P} \in \vec{d}\}$.

### 5.5 Efficiency analysis

In general, our runtime complexity analysis from §2.1.1 translates reliably into empirical efficiency gains. We provide concrete examples of computational cost savings from our experiments. Over the 14 candidate mixtures sampled for the experiment in §5.2, we formed the macro-MERGED models using 104 total "base unit" 130m models that were each trained for 1 billion tokens in around 5 hours on a single A100 GPU, amounting to 520 total GPU hours. The SEQ models for the candidate data mixtures required 163 billion tokens of training in total. Then, a lower bound estimate for the number of GPU hours used to train the corresponding 14 data mixtures is $163 * 5 = 815$ – in reality, the sum was higher due to communication overhead from training SEQ models on multiple GPUs.

Notably, by our proposed paradigm, we can evaluate the remaining 157 pairs of fields of study with *no additional training*. We can evaluate all 231 possible FoS-pair data mixtures with only 120 additional GPU hours (to include the 3 fields of study that were not represented in the 14 candidate mixtures we sampled), which brings the total GPU hours needed to only 640. In contrast, naively training SEQ models for all possible candidate mixtures would have cost us 2688 GPU hours total.

That being said, *illustrating* that there exists a strong correlation trend is computationally expensive; our bottleneck is training the very SEQ models that our method allows us to avoid, or at the very least reduce. Thus, our empirical experiments involve only a fraction of the total search space for their respective settings.

Our method's efficiency advantage is greater and more straightforward to calculate for the experiments in §5.1, since we sample partitions directly

[9] In fact, Ye et al. (2024) shows that smaller (SEQ) models can predict larger (SEQ) model performance on a data mixture. While we focus on efficiency gains from reusing training performed on decomposed subparts of an arbitrary data mixture, we show that these efficiency gains are *compatible*.

from "base components" $\mathcal{P}' \in \mathcal{C}$. For the S2ORC-only $k = 2$ and $k = 3$ experiments, we can use the same 128 base component models from §5.2 (no additional training cost). Naively training models on all possible combinations of data would result in $\binom{128}{2} * 2$ =16,256 billion tokens of training over 8128 candidate mixtures, for 81,280 GPU hours for $k = 2$. For $k = 3$, we would have $\binom{12,8}{3} * 3$ =1,024,128 billion tokens over 341,376 candidate mixtures, for 5,120,640 GPU hours total.

## 6 Related Work

**Pretraining mixtures** Existing literature includes many open corpora that are presented as viable training data mixtures (or components of them) for pre-training LLMs (Dodge et al., 2021; Gao et al., 2020; Computer, 2023; Penedo et al., 2023; Soldaini et al., 2024). The individual components of these mixtures and their relative sizes are typically chosen based on some intrinsic measure of data quality as it is often prohibitively expensive to perform thorough data ablations to create mixtures optimizing for downstream performance.

**Efficient data selection** Given that data ablations on large language models is expensive, one class of approaches relies on approximating them on smaller models. Relevant work studies scaling laws for model parameters vs training tokens (Hoffmann et al., 2022; Biderman et al., 2023), empirical effects of including or excluding different sources of data (Longpre et al., 2023), and the effects of training over multiple epochs vs new training tokens (Muennighoff et al., 2023). Previous work has also explored improving domain-specific fit via continued pre-training (Gururangan et al., 2020), predicting domain fit using lexical features (Reid et al., 2022), or improving general test-time adaptation via dynamic data selection, either by distributionally robust optimization with a small proxy model (Oren et al., 2019; Xie et al., 2023) or online using a multi-armed bandit approach (Albalak et al., 2023). Additional previous works aim to adapt to known downstream tasks via data selection, including at the individual example level (Wang et al., 2020) or even by explicitly fine-tuning models on many tasks (Aghajanyan et al., 2021). Notable recent and concurrent work proposes scalable influence functions, traditionally used at only very small scales, as a method for selecting better training data (Choe et al., 2024; Yu et al., 2024). Another concurrent preliminary work (Thrush et al., 2024) proposes a training-free method of selecting data to improve performance on downstream tasks. See Albalak et al. (2024) for a survey on data selection methods. We note that our strategies may be compatible with many of the existing methods for dataset selection, potentially leading to cumulative efficiency improvements – in particular, we show that our proxy metrics are compatible with Ye et al. (2024)'s strategy of predicting larger models' performance on a data mixture using a smaller, proxy model's performance.

**Model averaging** Merging models via weight-space averaging is more commonly done in the fine-tuning stages of language model training (Izmailov et al., 2018; Wortsman et al., 2022), typically for improving robustness and mitigating cross-task interference. When used for pretraining language models (Li et al., 2022; Chronopoulou et al., 2023), the goal is to efficiently adapt to new domains at inference time. To the best of our knowledge, ours is the only work that studies model merging for efficient data ablations.

**Linear mode connectivity** Relevant to our work, the mode connectivity perspective argues models share behavior when linearly connected on the loss surface (Frankle et al., 2020; Juneja et al., 2023; Neyshabur et al., 2020). Notably, Garipov et al. (2018) previously demonstrated that model ensembling is most effective when combining models on the equivariant loss curve on the loss surface. This applies to our work as all our models share a seed training phase, and we can subsequently presume each expert is close on the loss surface.

## 7 Conclusion

We lay groundwork for a promising new strategy whereby one can efficiently simulate principled, fine-grained ablations of training datasets and data mixtures. We are hopeful that researchers and practitioners can follow our recommendations to find beneficial data mixtures for their own models and data. Moreover, we encourage others to experiment in additional settings and report observations specific to their own assumptions and settings, particularly in relation to larger scales of models and data and limitations of the current method.

## Limitations

Although our proposed strategies may yield immediate practical benefits certain settings, there are several limitations in the current study that call for further investigation:

**The role of a shared optimization trajectory.** In our experiments, our models share a substantial amount of shared initial "seed" pre-training (42 billion tokens), and comparatively little continued pre-training ($\leq$ 27 billion tokens). Although there is precedent for multi-stage pre-training of LLMs, where greater care is taken in later stages to curate appropriate training datasets (Li et al., 2022; Gururangan et al., 2023)[10], there are several unanswered questions about the necessity and role of the initial optimization shared across our models – e.g. the *minimum* amount of shared initial "seed" pre-training necessary, or the *maximum* amount of continued pre-training allowed, for models trained in parallel on partitions of a data mixture to remain "mergeable" and predictive of a model trained sequentially on the entire data mixture.

**Varying data mixture proportions.** From the present study's results, one might reasonably hypothesize that a data mixture that is predicted to be particularly performant on an evaluation set(s) of interest should be upsampled if used as part of a larger training corpus. However, we leave to future work the explicit investigation of proxy metrics for predicting perplexity performance on different proportions of the same dataset components.

**Varying total amount of training.** In our settings, we assume similar amounts of data (or logical partitions) are being mixed in each candidate configuration. Future work may investigate the viability of our method for predicting performance in lifelong learning settings, modeling the effect of adding training data to a mixture progressively.

**Scaling up beyond 1.1b parameters.** The overall scale of our models and data is smaller than what is commonly used in practice today. Although we have promising evidence that models at the 1.1b parameter scale do not behave drastically differently from our 130m parameter models with respect to our reported trends (and that the smaller models' MERGED performance has a useful correlation with the 1.1b models' SEQ performance), we acknowledge the possibility that the patterns may be less reliable in models at 7b or larger scales, or that it may be necessary to adopt certain strategies to handle additional logistical challenges of working with larger models and corpora (e.g. larger model storage size on disk, and more copies stored for larger corpora). We are hopeful that follow-up work can allow the community to develop a stronger sense of these limitations and necessary adaptations.

**Downstream task evaluations.** In our study, we evaluate our models with per-token perplexity score on a diverse set of evaluation sets. The small scale of most of our models are such that useful behavior on standard downstream tasks typically requires few-shot learning or fine-tuning. Although perplexity scores from the same model can be useful for comparison and can be an informative signal about model *fit* to various textual domains, we recognize that perplexity alone does not necessarily capture the aspects of language model *behavior* that may be of interest, and perplexity scores by themselves are not necessarily interpretable in comparisons between domains, especially as perplexity can be highly impacted by formatting and structure of text.

## Ethical Considerations

While our work is aimed at reducing the computational requirements of experimentation with pretraining data mixtures, our experimentation itself used significant amounts of compute. We estimate that, given access to similar hardware, replicating our experiments on our small models alone would require at minimum 2,000 GPU hours (partially due to the necessity of training SEQ models for all experiments). In general, our approach should enable more frequent experimentation and testing of data mixtures due to lowered computational costs of conducting them, which may ideally enable a more nuanced understanding of the role of different types of data in different data mixtures, including with respect to fit to various evaluation domains. However, we recognized that, by Jevons Paradox, the efficiency gains introduced by our methods could lead to increased overall resource consumption rather than a reduction.

[10] Additionally, the the OLMo 1.7 blog post describes using a 2 stage curriculum that our method may be considered an approximation of.

## Acknowledgments

The authors are grateful to Luca Soldaini and Kyle Lo for assistance with procuring training data, members of the AllenNLP team for input during early stages of the project, AI2 IT for critical technical support, and the anonymous reviewers and SAC for their time and helpful feedback. We would like to thank Joel Mire, Akhila Yerukola, Cathy Jiao, and Jared Fernandez for extremely helpful feedback. This work was supported in part by the National Science Foundation Graduate Research Fellowship Program under grant DGE2140739.

## References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*. https://arxiv.org/abs/2402.16827.

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. 2023. Efficient online data mixing for language model pre-training. In *R0-FoMo:Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *Preprint*, arXiv:2304.01373.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, Jeff Schneider, Eduard Hovy, Roger Grosse, and Eric Xing. 2024. What is your data worth to gpt? llm-scale data valuation with influence functions. *Preprint*, arXiv:2405.13954.

Alexandra Chronopoulou, Matthew Peters, Alexander Fraser, and Jesse Dodge. 2023. AdapterSoup: Weight averaging to improve generalization of pre-trained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2054–2063, Dubrovnik, Croatia. Association for Computational Linguistics.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. 2020. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3259–3269. PMLR.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *ArXiv*, abs/2101.00027.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. 2018. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Olmo: Accelerating the science of language models. *Preprint*, arXiv:2402.00838.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. 2023. Continual pre-training of large language models: How to re-warm your model? In *Workshop on Efficient Systems for Foundation Models @ ICML2023*.

Suchin Gururangan, Margaret Li, Mike Lewis, Weijia Shi, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2023. Scaling expert language models with unsupervised domain discovery. *ArXiv*, abs/2303.14177.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Pavel Izmailov, Dmitrii Podoprikhin, T. Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence*.

Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. 2023. Linear connectivity reveals generalization strategies. In *The Eleventh International Conference on Learning Representations*.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2022. Branch-train-merge: Embarrassingly parallel training of expert language models. *ArXiv*, abs/2208.03306.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Michael Kinney, and Daniel S. Weld. 2020. S2orc: The semantic scholar open research corpus. In *Annual Meeting of the Association for Computational Linguistics*.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2023. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. *Preprint*, arXiv:2305.13169.

Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, Dirk Groeneveld, Iz Beltagy, Hannaneh Hajishirzi, Noah A. Smith, Kyle Richardson, and Jesse Dodge. 2023. Paloma: A benchmark for evaluating language model fit. *Preprint*, arXiv:2312.10523.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. 2020. What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems*, volume 33, pages 512–523. Curran Associates, Inc.

Yonatan Oren, Shiori Sagawa, Tatsunori B. Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4227–4237, Hong Kong, China. Association for Computational Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. In *Advances in Neural Information Processing Systems*, volume 36, pages 79155–79172. Curran Associates, Inc.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2022. Scaling language models: Methods, analysis & insights from training gopher. *Preprint*, arXiv:2112.11446.

Machel Reid, Victor Zhong, Suchin Gururangan, and Luke Zettlemoyer. 2022. M2D2: A massively multi-domain language modeling dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 964–975, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*.

Tristan Thrush, Christopher Potts, and Tatsunori Hashimoto. 2024. Improving pretraining data using perplexity correlations. *Preprint*, arXiv:2409.05816.

Xinyi Wang, Hieu Pham, Paul Michel, Antonios Anastasopoulos, Graham Neubig, and Jaime Carbonell. 2020. Optimizing data usage via differentiable rewards.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2024. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations*.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. *ArXiv*, abs/2305.10429.

Steve Yadlowsky, Lyric Doshi, and Nilesh Tripuraneni. 2023. Pretraining data mixtures enable narrow model selection capabilities in transformer models. *Preprint*, arXiv:2311.00871.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Yunhua Zhou, Jun Zhan, and Xipeng Qiu. 2024. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *Preprint*, arXiv:2403.16952.

Zichun Yu, Spandan Das, and Chenyan Xiong. 2024. Mates: Model-aware data selection for efficient pretraining with data influence models. *Preprint*, arXiv:2406.06046.

# A Appendix

## A.1 S2ORC dataset details

Tables 5 and 6 contain additional information about the S2ORC text dataset.

| # FoS | Document Count | % |
|---|---|---|
| 0 | 210 | 1.0% |
| 1 | 15844 | 76.8% |
| 2 | 4318 | 20.9% |
| 3 | 259 | 1.3% |
| 4 | 7 | $< 0.1\%$ |
| 5 | 1 | $< 0.1\%$ |
| *All (sample)* | 20,639 | 100% |

Table 5: Distribution of number of tagged fields of study in S2ORC documents, the 20k documents in our validation set. The original S2ORC corpus contains 81.1M documents in total. We use only the first field of study for our partitioning purposes.

## A.2 Licenses

S2ORC and M2D2 have CC BY-NC licenses. Out of the other Paloma subsets we used, most are licensed under AI2 ImpACT License - Low Risk Artifacts, excepting Wikitext-103 (CC BY-SA) and RedPajama. Our use of the datasets is for research purposes and aligns with their intended uses.

## A.3 Training details

For all models, we preserve optimizer states between seed and continued pre-training, use an Adam optimizer with 0.1 weight decay and (0.9, 0.95) betas, and train on batch sizes of 1024 comprised of packed sequences of 2048 tokens, or 2M tokens per batch.

In seed pre-training, we use a cosine learning rate with warmup up to a maximum of $6e-4$ followed by annealing to $6e-5$ by the end of seed training. However, we find it beneficial to jump back up to the maximum LR when starting from a seed model initialization and switching to optimization on the domain-specific partitions.

All experiments use at most a single node at a time. We used between 1 GPU (modular training) and 8 GPUs (longest SEQ training) per model training job. The GPU hardware available to us was a mixture of A6000s, L40s, and 80GB A100s.

### A.3.1 A note on model architecture

We describe our model as "PaLM-like" to distinguish between sequential and parallel use of attention and MLP blocks: Llama models have sequential blocks, while PaLM and our models have parallel blocks. We do not believe that the use of sequential vs parallel MLP and attention blocks would fundamentally affect our results or imply any loss of generality. On the other hand, one benefit to using a parallel architecture was that we observed superior throughputs on our setup early on, and this allowed us to run more data mixture experiments with our small models.

## A.4 Modeling entire fields of study

We select partitions $\mathcal{P}$, each of which contains between 1 and 20 "base unit" partitions $\mathcal{P}'$. Note that here, there is no overlap between different candidate data mixtures. Instead, we present this as a transition between the earlier settings mixing evenly sized $\mathcal{P}'$ and the later settings mixing unevenly sized $\mathcal{P}$. We focus on 1) providing further evidence of the ID vs OOD behavior seen in the previous experiments, and 2) providing a basis for the "macro"-MERGED model evaluation strategy we see in experiments testing uneven data mixtures.

In Figure 7 we see that SEQ scores are indeed strongly correlated with the performance of the aggregated MERGED model for each OOD field of study.

## A.5 Fixing one partition

This experiment is a practical extension and variation of the experiment in §5.3. Figure 8 contains the results of this study. We again see that micro-MERGED model scores are the most useful proxy score overall. For 3 out of 9 Paloma splits, selecting the "best" mixture by proxy metric gives us the "best" model as measured by evaluating full SEQ model for the data mixture on the same evaluation set, and in another 3, the "best" mixture appears in the top 3 mixtures chosen by the proxy ranking (out of 22 candidates). Selecting the top candidate by proxy gives us a median "true" rank of 2, with a maximum "true" rank of 5, across the 9 Paloma splits we evaluate on.

## A.6 Larger models

Figure 9 plots SEQ vs. proxy metrics in 1.1b parameter models. We study all $\{\vec{d} \subseteq \{0,1\} : ||\vec{d}_{S2ORC}||_1 = 2, ||\vec{d}_{Wiki}||_1 = 0, |\mathcal{P}| = 1 \; \forall \; \mathcal{P} \in \vec{d}\}$.

Though the correlation values themselves are weaker, this is also true in the replication of these experiments with smaller models as both SEQ and proxy models, seen in Appendix A.6.2. Moreover, in all settings and evaluation domains, the data

| Field of Study | # Partitions | Field of Study | # Partitions |
|---|---|---|---|
| Agricultural and Food Sciences | 5 | History | 1 |
| Art | 1 | Linguistics | 1 |
| Biology | 18 | Materials Science | 5 |
| Business | 4 | Mathematics | 12 |
| Chemistry | 4 | Medicine | 14 |
| Computer Science | 11 | NA | 2 |
| Economics | 4 | Philosophy | 1 |
| Education | 3 | Physics | 20 |
| Engineering | 5 | Politics | 2 |
| Environmental Science | 7 | Psychology | 6 |
| Geology | 1 | Sociology | 1 |
| **Total** | **128** | | |

Table 6: We partition S2ORC topically and temporally. There are 128 partitions total, with a median token count of 287 million tokens. Partition sizes vary no more than about one order of magnitude in token count.

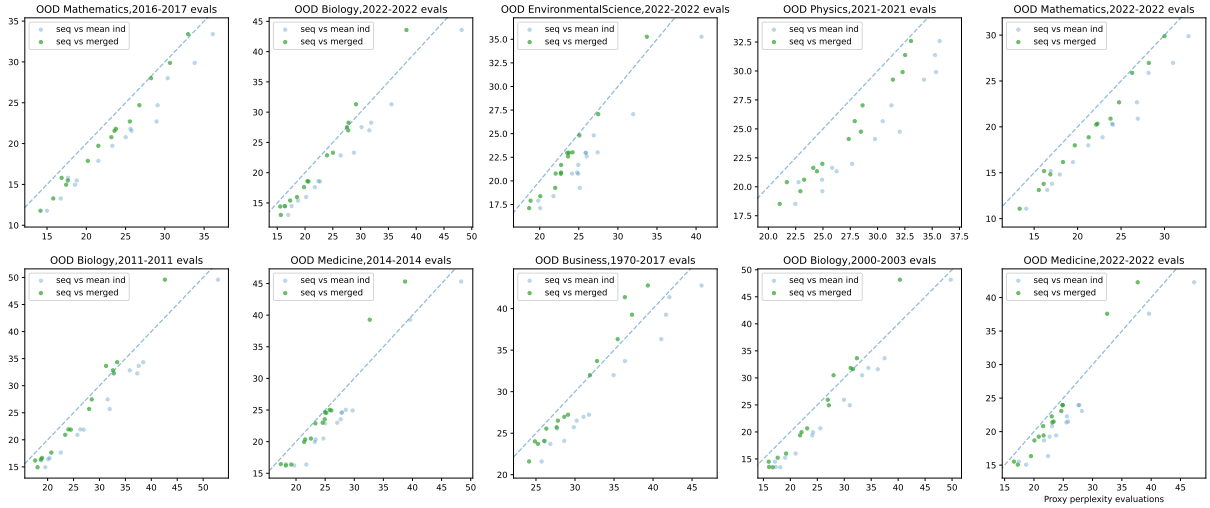



Figure 7: Here, we consider entire S2ORC fields of study as their own data mixtures. We display ten randomly selected S2ORC domains' evaluation scores (each scatterplot corresponds to one evaluation domain). SEQ models are trained on the entire field of study, and MERGED models are the parameter average of multiple models, as few as 2 or as many as 20. IND scores are the macro-average of the individual models' scores on the evaluation domain. Each point in a figure corresponds to the MERGED or IND score plotted against the SEQ score for a single field of study. As before, the y-axis shows SEQ perplexity evaluation performance, while the x-axis shows proxy perplexity scores.

mixture with the lowest proxy perplexity score also leads to one of the lowest perplexity scores in SEQ models.

### A.6.1 Predicting larger model performance

Here, we use these same candidate mixtures to present another version of the experiment shown in §5.4, where we explore the possibility of predicting SEQ model performance of larger models using smaller, proxy models. Figure 11 depicts the results of this experiment.

### A.6.2 Replication in 130m parameter models

We replicate the experiments from §A.6 and §A.6.1 using solely 130m parameter models. We observe that this set of data mixtures yields noisy correlations compared to the other settings we explore at this smaller scale. We conjecture that this may be in part due to the types of fields of study that are small enough to have only one base component partition: Art, Geology, History, Linguistics, Philosophy, and Sociology. These are mostly humanities fields with high overlap and similarity as measured by pairwise cosine distances between mean SentenceBERT (Reimers and Gurevych, 2019) embeddings of held-out examples from each of these partitions. Specifically, we observe separately that relaxing the definition of a data mixture $\vec{d}$ to allow for multiple copies of the same partition in a single mixture can cause the associated mixture to deviate significantly from the other points in the study – accordingly, it could be the case that partitions that are too similar to each other (such that, for example,

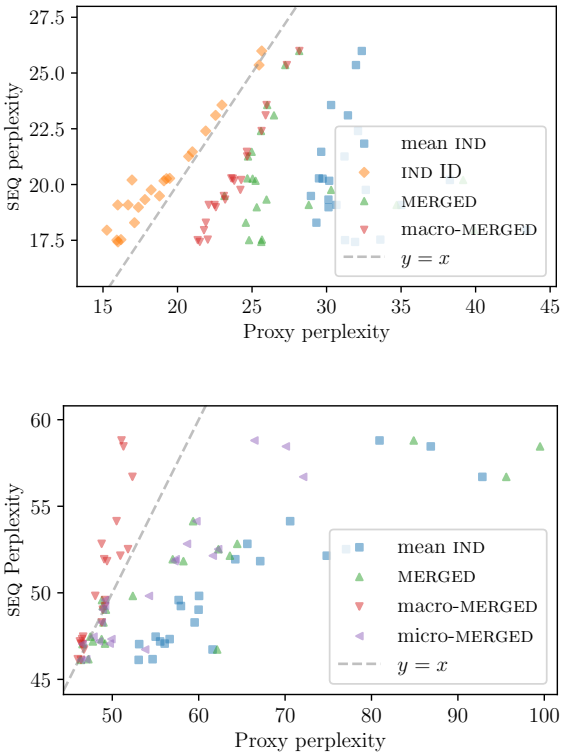
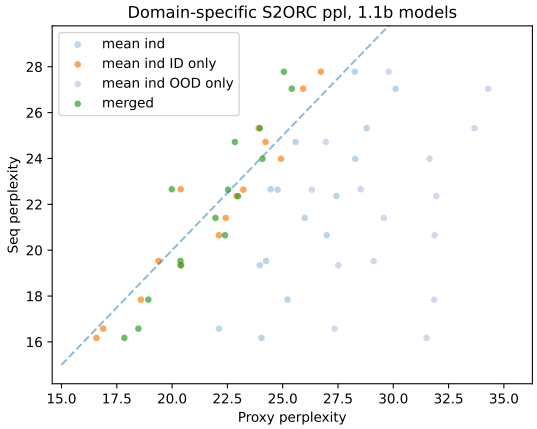


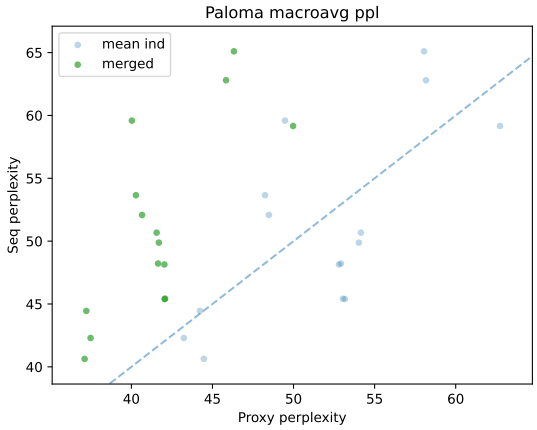

Figure 8: SEQ vs. proxy macro-averaged Paloma perplexity scores of models trained on data mixtures where all contain the same L1 Wikipedia domain. It is again beneficial to use *micro*-MERGED models: 0.935 corr. with SEQ, vs. 0.889 (MERGE of the SEQ models for each $\mathcal{P}$), 0.920 (mean IND scores), 0.882 (macro-MERGED), or 0.794 (mean MERGED scores)

.

a human might have trouble distinguishing between them as separate domains) may behave differently in a data mixture compared to other, less similar partitions.

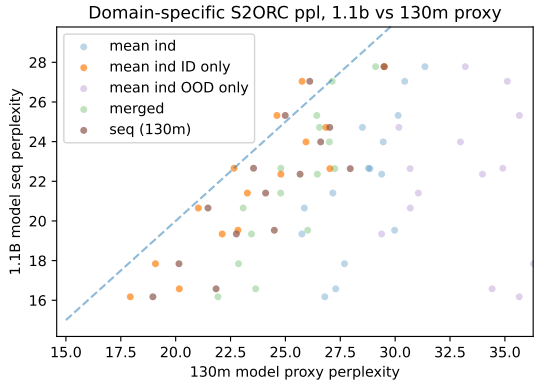



(a) Here, the means of the in-domain scores of IND models is not as clearly better than the MERGED models for predicting SEQ performance on data mixtures, but the correlation is still strong.

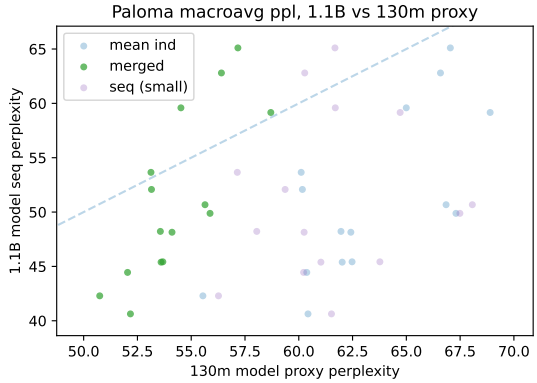



(b) On OOD Paloma evals the correlation is even weaker, but we still see that we can pick the "best" data mixtures by picking the "best" model mixtures.

Figure 9: At larger 1.1b model scale, the correlations we see are weaker but still usable in practice if one only wishes to select the most performant data mixtures. Some of the increased variance seems to come from the sample itself (just the smallest fields of study $\mathcal{P}$ from S2ORC). We see a similarly increased amount of variance in the 130m models when replicating on the same data mixtures (see Figure A.6.2).

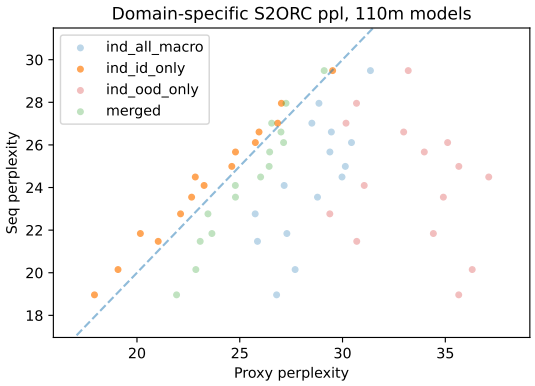



(a) These are the weakest correlation so far for predicting in-domain SEQ performance, but still a recognizable trend. The weaker correlation is unsurprising given that we are comparing models different in scale by a factor of almost 10.




(b) Interestingly, scores of the smaller MERGED models are actually much *more strongly correlated* with SEQ model scores than scores of smaller SEQ models!

Figure 10: Concurrent work by Ye et al. (2024) shows that smaller (SEQ) models can predict larger (SEQ) model performance on a data mixture. While we focus on asymptotic efficiency gains from reusing training performed on decomposed subparts of an arbitrary data mixture, we show that these efficiency gains are *compatible*.



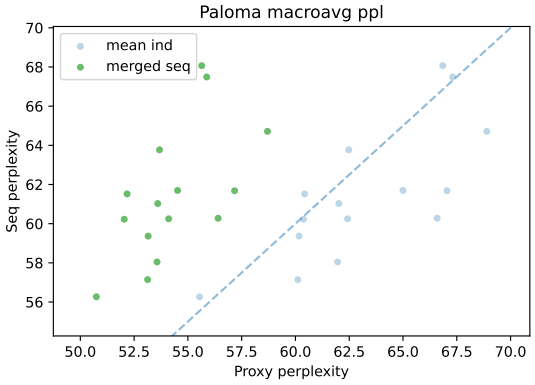




Figure 11: We see that the weaker correlations seen in Figures 9 and 11 seem unlikely to be solely due to the larger model scale, as the correlation is weaker also when 130m models are used as both SEQ and proxy models.