IMPROVING PRETRAINING DATA USING PERPLEXITY CORRELATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Quality pretraining data is often seen as the key to high-performance language models. However, progress in understanding pretraining data has been slow due to the costly pretraining runs required for data selection experiments. We present a framework that avoids these costs and selects high-quality pretraining data without *any* LLM training of our own. Our work is based on a simple observation: LLM losses on many pretraining texts are correlated with downstream benchmark performance, and selecting high-correlation documents is an effective pretraining data selection method. We build a new statistical framework for data selection centered around estimates of perplexity-benchmark correlations and perform data selection using a sample of 90 LLMs taken from the Open LLM Leaderboard on texts from tens of thousands of web domains. In controlled pretraining experiments at the 160M parameter scale on 8 benchmarks, our approach outperforms DSIR on every benchmark, while matching the best data selector found in DataComp-LM, a hand-engineered bigram classifier.

1 INTRODUCTION

Dataset curation is increasingly crucial for training high-quality large language models (LLMs). As pretraining datasets have grown, from under 200B tokens in 2020 (Raffel et al., 2020; Gao et al., 2020) to 240T tokens today (Li et al., 2024), it has become critical to identify subsets of the available data that will lead to the best LLMs, and a wide range of methods have arisen to meet these needs (Ilyas et al., 2022; Xie et al., 2023a;b; Engstrom et al., 2024; Everaert & Potts, 2024; Liu et al., 2024; Llama Team, 2024). However, data-driven approaches to data selection typically involve expensive model retraining steps that limit their effectiveness, and no algorithm has been reported to consistently beat or match hand-crafted classifiers for data selection (Li et al., 2024).

Is training new LLMs necessary for data selection? Instead of training our own models, can we use the growing collection of publicly available, high-performance LLMs (Wolf et al., 2019; Beeching et al., 2023) to perform data valuation and selection? This would have significant benefits: we could leverage the millions of dollars collectively spent on building these LLMs, and we would have coverage over a large, heterogeneous collection of high-performance models varying in size, architectures, and pretraining data distribution.

Despite these advantages, using existing models for pretraining data selection is challenging, as the training data for these models are often unknown and heterogeneous. Our key observation is that data selection can be done using two observable features of *all* public models today: 1) all open-weight models produce a causal language modeling loss for a given text, and 2) all of them can be evaluated on benchmarks. Prior work has found systematic relationships between web corpus loss and benchmark performance (Wei et al., 2022; Huang et al., 2024), which suggests the possibility of using correlations between perplexity and benchmark scores as the basis for a data selection policy.

In the present paper, we pursue this possibility and find a radically simple approach that is also effective: we select data via *perplexity correlations* (Figure 1), where we select data domains (e.g. wikipedia.org, stackoverflow.com, etc.) for which LLM log-probabilities are highly correlated with downstream benchmark performance. To enable our approach, we complement our algorithm with a statistical framework for correlation-based data selection and derive correlation estimators that perform well over our heterogeneous collection of LLMs.

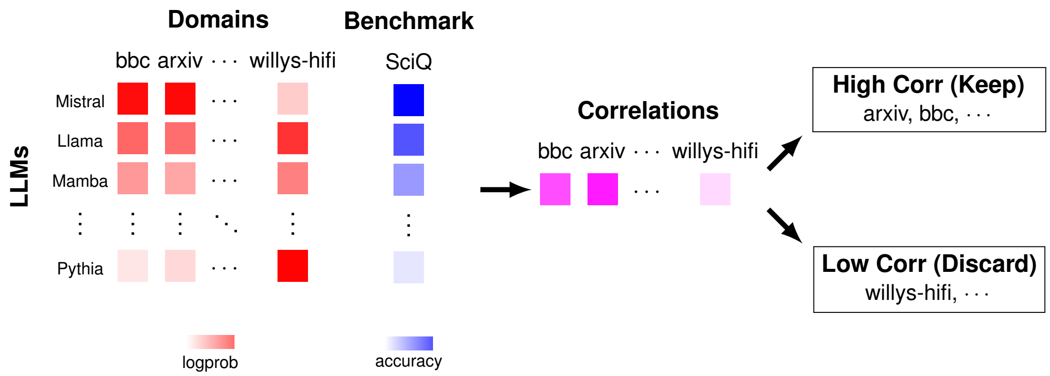


Figure 1: We pretrain on domains where lower loss is generally correlated with higher downstream performance. Our approach does this by taking public, pretrained LLMs and measuring correlations across their log-likelihoods (left, red matrix) and performance on a target benchmark (center, blue vector). We then perform data selection by training a fastText classifier that distinguishes high correlation domains from others. This approach is on par with the best-known data selection methods in our experiments, despite requiring no human selection of high-quality domains.

We validate our approach using a collection of pretrained causal LLMs on the Hugging Face Open LLM Leaderboard (Beeching et al., 2023) and find that perplexity correlations are predictive of an LLM's benchmark performance. Importantly, we find that these relationships are robust enough to enable reliable data selection that targets downstream benchmarks. In controlled pretraining experiments at the 160M parameter scale on eight benchmarks, our approach strongly outperforms DSIR (Xie et al., 2023b) (a popular training-free data selection approach based on n-gram statistics) while generally matching the performance of the best method validated at scale by Li et al. (the OH-2.5 +ELI5 fastText classifier (Joulin et al., 2016)) without any parameter tuning or human curation.

2 Related Work

To go beyond the status quo of deduplication, perplexity filtering, and hand-curation (Laurençon et al., 2022; BigScience, 2023; Abbas et al., 2023; Groeneveld et al., 2024; Soldaini et al., 2024; Penedo et al., 2024; Llama Team, 2024), targeted methods have been proposed to filter pretraining data so that the resulting LLM will achieve higher scores on given benchmarks. There are lightweight approaches that use n-gram overlap (Xie et al., 2023b) or embedding similarity (Everaert & Potts, 2024) to select training data that is similar to data from a given benchmark. There are also less-scalable methods that require training proxy LLMs on different data mixtures (Ilyas et al., 2022; Xie et al., 2023a; Engstrom et al., 2024; Liu et al., 2024; Llama Team, 2024).

Given the high costs of proxy-based data selection methods, they have primarily been used to select among human-curated pretraining data mixtures (Llama Team, 2024; Li et al., 2024) rather than a high dimensional space of mixtures. Our work takes an orthogonal approach and builds upon recent observational studies that have found scaling relationships that hold across collections of uncontrolled and diverse LLMs (Owen, 2024; Ruan et al., 2024). While these studies do not examine loss-to-performance relationships or derive useful data selection methods from them, we know that losses and performance are generally highly correlated. Validation losses on samples of text corpora are commonly used as a proxy for downstream performance when comparing LLMs pretrained on the same data distribution (Kaplan et al., 2020; Hoffmann et al., 2022; Wei et al., 2022), even if they have different architectures (Poli et al., 2023; Peng et al., 2023; Gu & Dao, 2024).

According to a recent survey of data selection approaches by Li et al. (2024), the heavier-weight pretraining data selection methods have not shown large gains, and the current state-of-the-art across many tasks is primitive: a fixed fastText classifier (Joulin et al., 2016) combined with an English filter as a final layer after extensive deduplication and filtering. Are we missing important information that we can efficiently extract from a diverse collection of already trained models, larger and more diverse than any single organization is likely to produce? We show evidence supporting this hypothesis – simple loss-performance correlation coefficients are effective when used for data selection.

3 PROBLEM SETTING

Our goal is to build predictive models of how pretraining data distributions affect downstream benchmark performance and use them to build better language models. Unfortunately, this task is challenging and computationally expensive. A standard approach adopted in paradigms such as datamodeling (Ilyas et al., 2022) is to obtain N different pretraining distributions $\{\mathbf{p}_i : i \in [N], \mathbf{p}_i \in \mathbb{R}_0^{+D}\}$ over $D \gg N$ domains (e.g. arxiv.org, stackoverflow.com, etc.), pretrain and measure model errors on a target benchmark $y_i \in [0, 1]$, and fit a model $\mathbf{p} \to y$. This approach requires N LLM training runs, performed at a scale sufficient to obtain non-random performance on y. This can cost tens to hundreds of millions of dollars for hard benchmarks such as MMLU, where even the performance of 1B parameter LLMs often do not exceed random chance (Beeching et al., 2023).

Instead, our work considers the following *observational* setting that requires no training. We obtain N pretrained, high-performance LLMs that vary in pretraining data, tokenizer, architecture, and scale (e.g. models on Huggingface's OpenLLM leaderboard). Now, if we could train a predictor $\mathbf{p} \to y$ on these N models, we could avoid large scale model training. Unfortunately, this is impossible as the training data for these models is often proprietary, and so we have no knowledge of $\mathbf{p}$.

The key observation of our work is that we can replace $p_{i,j}$ (the unobserved sampling probability of model i's data selection policy on document j) with an observable surrogate $x_{i,j}$, which is the negative log-likelihood of document j under model i.[1] We can then build a regression model that relates negative log-likelihood $\mathbf{x}_i$ and benchmark error y_i. Using this model, we can select pretraining data from domains j for which decreasing the loss $x_{i,j}$ is predicted to rapidly decrease error y_i.

The perplexity-performance hypothesis. We formulate the task of predicting errors y_i from negative log-probabilities $\mathbf{x}_i$ as a single-index model (SIM),

$$y_i = f(\langle \boldsymbol{\theta}^*, \mathbf{x}_i \rangle + \epsilon_i) \tag{1}$$

where $f : \mathbb{R} \mapsto \mathbb{R}$ is some unknown monotonically increasing univariate function, ϵ_i is zero-mean noise which is independent of $\mathbf{x}$, and $\boldsymbol{\theta}^* \in \mathbb{R}^D$ are unknown weights over D domains.

A single index model is highly flexible (due to the arbitrary, monotone f) and has the advantage that we do not need to estimate the nonlinear function f if our goal is to optimize model performance. We can see this directly from the monotonicity of f as

$$\langle \boldsymbol{\theta}^*, \mathbf{x}_i \rangle + \epsilon_i < \langle \boldsymbol{\theta}^*, \mathbf{x}_j \rangle + \epsilon_j \iff f(\langle \boldsymbol{\theta}^*, \mathbf{x}_i \rangle + \epsilon_i) < f(\langle \boldsymbol{\theta}^*, \mathbf{x}_j \rangle + \epsilon_j). \tag{2}$$

Data selection from perplexity correlations. The weights $\boldsymbol{\theta}^*$ tell us which domain perplexities are correlated with downstream performance. However, this isn't sufficient for data selection. Even if we know how model likelihoods relate to model performance, we do not know how data selection affects likelihoods. Even worse, this data mixture to likelihood relationship *cannot* be learned observationally, as we do not know the data mixture of any of our models.

Despite this, we show that there is a clean approach for optimizing the data mixture. Our core observation is the following: *if we find a nonnegative $\boldsymbol{\theta}^*$, sampling proportional to $\boldsymbol{\theta}^*$ is always a good choice.* More formally, we see that this sampling distribution defines the pretraining loss such that optimizing the training loss directly optimizes the downstream task via the single index model.

Proposition 1 *Suppose that $\boldsymbol{\theta}^*$ weights are non-negative. Then, for models with associated likelihoods $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^D$, the minimizer of the pretraining loss over the $\boldsymbol{\theta}^*$ sampling distribution $\mathbb{E}_{j\sim\boldsymbol{\theta}^*}[x_j]$ also has the lowest expected downstream error according to the single index model:*

$$\arg\min_{\mathbf{x}\in\mathcal{X}} \mathbb{E}_{j\sim\boldsymbol{\theta}^*}[x_j] = \arg\min_{\mathbf{x}\in\mathcal{X}} \mathbb{E}[f(\langle \boldsymbol{\theta}^*, \mathbf{x} \rangle + \epsilon)].$$

This observation follows directly from the fact that we can normalize any non-negative $\boldsymbol{\theta}^*$ into a distribution (and shift the normalization constant into f) which allows us to write the inner product in the single-index model as a monotone function of the expected pretraining loss:

$$y = f(\langle \boldsymbol{\theta}^*, \mathbf{x} \rangle + \epsilon) = f(\mathbb{E}_{j\sim\boldsymbol{\theta}^*}[x_j] + \epsilon). \tag{3}$$

[1] To be precise, we use bits-per-byte, which normalizes the sequence negative log-likelihood with the number of UTF-8 bytes. This is defined in terms of the length of the string in tokens L_T, the length of the string in UTF-8 bytes L_B, and the cross entropy loss ℓ as BPB $= \frac{L_T \ell}{L_B \ln(2)}$

Proposition 1 allows us to entirely avoid the task of finding the optimal data mixture for a target likelihood. Instead, we pick sampling distributions that make the pretraining loss a monotone function of the predicted downstream error. Afterward, we can rely on our ability to optimize the loss to optimize downstream performance.

This view gives us a straightforward roadmap for data selection in the remainder of the paper: estimate a set of domains where loss and downstream benchmark performance is highly correlated, and then constrain our $\boldsymbol{\theta}^*$ estimates to be a pretraining data sampling distribution.

4 Methods

We now describe the details of our approach, starting by presenting the algorithm itself and the intuitions behind it, followed by a more precise and mathematical justification for the various steps.

4.1 Algorithm

Estimating $\boldsymbol{\theta}^*$. The parameter θ_j^* measures the relationship between log-likelihoods in domain j and downstream performance. Because of this, we might naturally expect θ_j^* to be related to nonlinear correlation coefficients between x and y. Our work uses a simple correlation measure,

$$\gamma_j = \sum_{\substack{1\leq k,l\leq n \\ k\neq l}} \text{sign}(y_k - y_l)(\text{rank}_j(x_{k,j}) - \text{rank}_j(x_{l,j}))$$

where $\text{rank}_j(x)$ is the rank of x among $\{x_{1,j} \dots x_{N,j}\}$. This formula is intuitive: when model k does better than model l, what percentile is model k's log-likelihood compared to model l's? While this is not the only correlation coefficient that performs well (see Appendix G), this functional form has the additional benefit of being a principled estimate of $\boldsymbol{\theta}^*$. In particular, we show in sections below that in expectation, the ranking of domains in $\boldsymbol{\gamma}$ exactly matches those of $\boldsymbol{\theta}^*$ (under standard high-dimensional regression assumptions; see Section 4.2 for a complete discussion).

Selecting pretraining data. Suppose that we have an accurate estimate γ_j which is nonnegative. In this case, we could use γ_j directly as a data selection procedure and Proposition 1 would ensure that minimizing the population pretraining loss minimizes downstream errors. Unfortunately, γ_j can be negative and the finite number of tokens per domain can make it difficult to minimize the population pretraining loss. Thus, we must project γ_j onto the set of reasonable pretraining data distributions that are nonnegative and account for the per-domain token counts.

What is a good way to project a set of domain rankings estimated via $\boldsymbol{\gamma}$ into a pretraining sampling distribution? Intuitively, if wikipedia.org has a $\gamma_j = 0.5$ and arxiv.org is $\gamma_k = 0.9$, it would be natural to select tokens in order of $\boldsymbol{\gamma}$, preferring tokens from arxiv.org over tokens from wikipedia.org.

Having established the ordering of domains, the remaining question is how many tokens we take for each domain. We follow recent observations that repeating data degrades performance (Abbas et al., 2023) to arrive at a simple selection algorithm: select domains in greatest to least $\boldsymbol{\gamma}$, taking all the tokens in each domain once, until we exhaust our total pretraining token budget.

Full algorithm. Together, these steps result in a simple, parameter-free algorithm that calculates our rank correlation coefficient, and selects domains in order from largest to smallest coefficient. We show this process explicitly with pseudocode in Algorithm 1 (see Appendix A), and additionally show an extra step where we train a fastText (Joulin et al., 2016) classifier (using standard settings and bigram features from Li et al. (2024)) which distinguishes our selected documents and domains from the rest of the pool. The fastText classifier allows us to perform data selection at a single-page level, and scale the selection process to larger datasets. We also found the classifier to slightly improve downstream performance over directly selecting the documents. More information on the specifics of the data selection approaches that we tested is given in Appendix F.

4.2 Theory

We now study the approach closely and show that our choices for the correlation coefficient and projection step are extensions of the classic, high-dimensional single index model estimator of Plan et al. (2016). We describe the basic single-index model estimators first, describe our extensions, and then conclude with a discussion on how our estimator and results deviate from the theory. A discussion of other potential estimation paradigms is provided in Appendix D.

4.2.1 HIGH-DIMENSIONAL ESTIMATION OF SINGLE INDEX MODELS

For our theory, we consider the standard high-dimensional regression setting of Plan et al. (2016) and Chen & Banerjee (2017). Here, our goal is to estimate the unknown weights $\boldsymbol{\theta}^*$ in a single-index model $y_i = f(\langle \boldsymbol{\theta}^*, \mathbf{x}_i \rangle + \epsilon_i)$, with $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $\|\boldsymbol{\theta}^*\|_2 = 1$ (assumed without loss of generality, as $\|\boldsymbol{\theta}^*\|_2$ can be absorbed by f).

Our starting point is the classic result of Plan et al. (2016), who showed

$$\mathbb{E}\left[y_k \mathbf{x}_k\right] = c\boldsymbol{\theta}^*, \tag{4}$$

for some positive constant c and $1 \leq k \leq N$. Closely related is the result of Chen & Banerjee (2017) who showed a robust estimator quite similar to ours,

$$\mathbb{E}\left[\text{sign}(y_k - y_l)(\mathbf{x}_k - \mathbf{x}_l)\right] = \beta\boldsymbol{\theta}^* \tag{5}$$

for any $1 \leq k, l \leq N$ (where $k \neq l$) and some positive constant β. Both of these results clearly identify that for the high-dimensional single-index model in the Gaussian setting, generalized correlation coefficients provide consistent estimates of the true regression coefficient $\boldsymbol{\theta}^*$.

4.2.2 DERIVING OUR ESTIMATOR

Both Plan et al. and Chen & Banerjee provide moment-matching style estimators that consistently recover $\boldsymbol{\theta}^*$ in high-dimensional, sparse settings. However, we found that both estimators directly use the values of x, and this resulted in brittle estimates due to outliers in language model log-likelihoods. While outlier removal is one possibility, we found that a simpler approach was to robustify the estimator of Chen & Banerjee (2017) to outliers in x.

Recall that our estimate $\boldsymbol{\gamma}$ is a U-statistic, defined as pairwise sums of

$$\text{sign}(y_i - y_j)(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)), \tag{6}$$

for any $1 \leq i, j \leq N$ (where $i \neq j$), where Φ is the empirical CDF of the $\mathbf{x}$ values. This estimate is significantly less sensitive to outliers than that of Chen & Banerjee (2017), as the empirical CDF is bounded between zero and one, and no single model can make the estimator degenerate.

We study this estimate theoretically in the Gaussian setting, where we consider the asymptotically equivalent estimator with Φ as the CDF of the standard Gaussian. In this case, we can show that this modified estimator is also consistent in recovering $\boldsymbol{\theta}^*$.

Theorem 1 *When $\epsilon \sim \mathcal{N}(0, \sigma^2)$, we have:*

$$\mathbb{E}[\text{sign}(y_i - y_j)(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))] = \frac{2}{\pi}\sin^{-1}\left(\frac{\boldsymbol{\theta}^*}{2\sqrt{1+\sigma^2}}\right). \tag{7}$$

We provide the proof in Appendix B. Because we assume $||\boldsymbol{\theta}^*||_2 = 1$ and the expected value in Equation 7 must be between -1 and 1, we are always within the domain of $\sin^{-1}$ and able to invert it. After inverting, we get:

$$\hat{\boldsymbol{\theta}} \propto \sin\left(\frac{\pi}{2}\mathbb{E}\left[\text{sign}(y_i - y_j)(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))\right]\right) \tag{8}$$

as an estimate for $\boldsymbol{\theta}^*$, where the constant $2\sqrt{1+\sigma^2}$ term due to noise has been dropped.

Beyond the fact that our estimator is consistent, we can show an even tighter connection to the Chen & Banerjee estimator: our estimates agree when running the original estimator on rank-transformed data. More specifically, for two models $\mathbf{x}_i$ and $\mathbf{x}_j$ with the estimated model rankings $\langle \hat{\boldsymbol{\theta}}, \mathbf{x}_i \rangle > \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_j \rangle$, the expected ranking under rank-transformation (i.e. $\Phi(\mathbf{x})$) match this ranking.

Corollary 1 *Suppose that $\hat{\boldsymbol{\theta}}$ is any vector of fixed weights and $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then, conditioning on the event $\langle \hat{\boldsymbol{\theta}}, \mathbf{x}_i \rangle < \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_j \rangle$, we have with probability 1 that:*

$$\langle \hat{\boldsymbol{\theta}}, \mathbb{E}[\Phi(\mathbf{x}_i) \mid \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_i \rangle < \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_j \rangle] \rangle < \langle \hat{\boldsymbol{\theta}}, \mathbb{E}[\Phi(\mathbf{x}_j) \mid \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_i \rangle < \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_j \rangle] \rangle. \tag{9}$$

This proof follows from the same calculations as Theorem 1 and is given in Appendix B.

4.2.3 SELECTING DATA FOR PRETRAINING

Recall that our algorithm for data selection is to constrain $\boldsymbol{\gamma}$ to be a valid sampling distribution (nonnegative, at the very least) and then sample directly from this estimate. For now, we focus on constraining $\hat{\boldsymbol{\theta}}$, and we will see at the end of this section that we can apply the same constraint to $\boldsymbol{\gamma}$ directly to get the same result. The theory of constrained estimation for $\hat{\boldsymbol{\theta}}$ is simple and well-understood, with both Plan et al. (2016) and Chen & Banerjee (2017) extensively studying the problem of estimating $\hat{\boldsymbol{\theta}}$ under a known convex constraint set C. In particular, Plan et al. (2016) show that performing a L_2 projection via $\hat{\boldsymbol{\theta}}^{\text{proj}} = \arg\min_{\boldsymbol{\theta}\in C} \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_2$ provides improved convergence rates that depend on the Gaussian mean width of C rather than the ambient dimension, and Chen & Banerjee (2017) show similar results when maximizing the linear correlation $\hat{\boldsymbol{\theta}}^{\text{proj}} = \arg\min_{\boldsymbol{\theta}\in C\subseteq B_D} -\langle\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}\rangle$.

We take a similar approach here. We define a convex constraint set C that forces $\hat{\boldsymbol{\theta}}$ to be a reasonable sampling distribution and find the best sampling distribution via the linear correlation approach.

We define C as the combination of two sets of constraints. First, we must have a valid sampling distribution, so we constrain $\hat{\boldsymbol{\theta}}$ to lie in the simplex. As we noted above, it is well-known that duplicating data harms performance (Abbas et al., 2023), and so we constrain $\hat{\boldsymbol{\theta}}$ to avoid data duplication by limiting the maximum weight on domains. Concretely, if want to pretrain on m tokens overall, we enforce $\theta_i^* \leq \tau_i, \forall i \in [1, D]$, where τ_i is set so $\tau_i m$ is the number of tokens from the i-th domain that we can access for training.

The resulting linear program has a simple solution and takes the form of initializing $\hat{\boldsymbol{\theta}}^{\text{proj}}$ to $\mathbf{0}$ and then iterating through the values in $\hat{\boldsymbol{\theta}}$ from largest to smallest, setting the value at the corresponding index of $\hat{\boldsymbol{\theta}}^{\text{proj}}$ to the maximum allowable value, until $\hat{\boldsymbol{\theta}}^{\text{proj}}$ sums to 1 (see Appendix C for a proof).

Theorem 2 *Suppose we want to solve:*

$$\hat{\boldsymbol{\theta}}^{\text{proj}} = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^D} -\langle\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}\rangle,$$

subject to:

$$\sum_{i=1}^{D} \theta_i = 1$$
$$0 \leq \theta_i \leq \tau_i, \forall i \in [1, D],$$

where $\tau_i > 0$ are fixed values. Then, the solution is:

$$\hat{\theta}_k^{\text{proj}} = \begin{cases} \tau_k & \text{if } \sum_{j:\, r_j(\hat{\theta}_j) \geq r_k(\hat{\theta}_k)} \tau_j \leq 1 \\ 1 - \sum_{j:\, r_j(\hat{\theta}_j) > r_k(\hat{\theta}_k)} \tau_j & \text{if } \sum_{j:\, r_j(\hat{\theta}_j) \geq r_k(\hat{\theta}_k)} \tau_j \geq 1 \wedge \sum_{j:\, r_j(\hat{\theta}_j) > r_k(\hat{\theta}_k)} \tau_j \leq 1 \\ 0 & \text{otherwise} \end{cases}, \tag{10}$$

where r is some function that breaks all ties between $\hat{\theta}_j$ and $\hat{\theta}_k$ for $k \neq j$, and otherwise leaves the ordinal relationships the same.

We note that while the use of this linear program is in line with the constrained estimators proposed in Chen & Banerjee (2017), the L_2 projection is arguably more natural, and does not require assuming that $\|\hat{\boldsymbol{\theta}}\|_2 = 1$ for asymptotic recovery conditions. We derive similar closed-form expressions for this quadratic case in Appendix C, but do not use this approach for two separate reasons.

First, the L_2 projection depends on the L_2 norm of $\hat{\boldsymbol{\theta}}$, unlike the linear program which only depends on the ranks of the values in $\hat{\boldsymbol{\theta}}$. The challenge with determining the norm is that the exact recovery result in Equation (7) requires knowledge of the noise level, and the trigonometric functions rely strongly on the Gaussian structure of x. Because of this, we are unlikely to be able to estimate the norm of $\hat{\boldsymbol{\theta}}$ with any accuracy, and the only way to avoid this would be to treat the norm as a hyperparameter, which adds unnecessary complexity. The second reason is empirical (although possibly a consequence of the first) – we found that the linear projection performed better across a wide range of benchmarks and conditions (see Appendix G).

We conclude by relating our theory to the full algorithm in Section 4.1. The estimation step for γ is the finite sample, U-estimate of the expectation in Equation (8), dropping the nonlinear transform $\sin$ and $\pi/2$ as these two terms do not change the rankings of the domains. The data selection step directly applies our projection in Equation (10), and we make use of the fact that this projection only relies on rankings among the domains to use γ rather than an exact estimate for θ^*.

5 Results

We empirically validate our approach to predicting downstream performance and data selection. Our validation consists of three sets of experiments: we first pretrain 160M-parameter LLMs from scratch to study our primary goal of selecting pretraining data to improve downstream performance, followed by analyzing the ability of losses to predict downstream performance. Throughout our experiments, we use the same single-index model that we train using Algorithm 1. As shown in the algorithm, we train the fastText classifier on selected vs unselected domains and use the classifier to filter the pretraining data at the page-level.

Input data matrix X. To build the input data matrix, $\mathbf{X}$, we collected byte normalized loss values from a sample of 90 Open LLM Leaderboard (Beeching et al., 2023) LLMs that we could run without errors. Concretely, these values are defined as bits-per-byte $\frac{L_T \ell}{L_B \ln(2)}$ where L_T is the token count, L_B is the number of UTF-8 bytes, and ℓ is the per-token cross-entropy (Gao et al., 2020). We collected these values on "sample" subset[2] of the RedPajama V2 (RPJv2) dataset (Together Computer, 2023) for all domains with ≥ 25 pages in the sample. There are 9,841 domains/features. Specifics are in Appendix E. A detailed principal components analysis of $\mathbf{X}$, which reveals a variety of salient embedded information in the losses, is in Appendix J.

Target benchmark performance y. We constructed a target vector, $\mathbf{y}$, for LAMBADA (Paperno et al., 2016), ARC Easy (Clark et al., 2018), PIQA (Bisk et al., 2020), and SciQ (Welbl et al., 2017). These are all of the tasks reported in the Pythia scaling experiments for which a model in the 160M parameter range could meaningfully perform above chance. We also constructed target vectors for $\text{LAMBADA}_{\text{IT}}$, $\text{LAMBADA}_{\text{FR}}$, $\text{LAMBADA}_{\text{DE}}$, and $\text{LAMBADA}_{\text{ES}}$, which are subsets of LAMBADA translated into Italian, French, German, and Spanish by Black (2023). These languages match those in RPJv2 where each page is conveniently tagged as one of five languages: English, Spanish, French, German, and Italian. The correspondence between our target benchmark languages and the RPJv2 metadata is convenient, as it allows us to easily include language filtering baselines.

5.1 Pretraining

We begin by validating our algorithm in the end-to-end task of pretraining data selection with controlled experiments at the 160M parameter, 3.2B token scale. The low compute requirements of this setting allow us to more extensively study replicates and ablations in Appendix G within the timeframe of a few days. While 160M models are small, this is far from an easy setting for our data selection algorithm. Most of the Open LLM Leaderboard models are 10 to 100$\times$ larger than the 160M scale, and our single index model must extrapolate substantially from $\approx$7B scale models to our small-scale validation setting (see Appendix I for a histogram of model sizes).

Pretraining data and setting. For pretraining, we used the "sample-100B" subset of RPJv2. This is larger than the sample that we used to compute our estimate. We filtered this data so it contains only the domains used for our estimate, and then tokenized the data with the Pythia tokenizer. The vast majority of the domains from our BPB matrix were present in this larger sample of text. However, 42 (out of 9,841) were not, and so we removed them from our estimate. For every data selection method that we tested, the task was to further select 3.2B tokens for pretraining, which is Chinchilla-optimal (Hoffmann et al., 2022) for the 160M-parameter LLM used in our tests.

Baselines. We compare against several baseline data-selection methods. First, we present the results of uniformly sampling from the available pretraining data. Then we use the language tags present in RPJv2 to filter only for the language matching the target task. In addition to these commonsense baselines, we also run DSIR (Xie et al., 2023b): a lightweight training data selection technique based on n-gram overlaps that Li et al. (2024) found to be competitive with proxy LLM-based techniques

[2] https://huggingface.co/datasets/togethercomputer/RedPajama-Data-V2

Table 1: Average rankings of each data selection method (lower is better) across 8 benchmarks shows that correlation-based filtering beats baselines by a wide margin, and matches the current best open data filter from Li et al. (2024). Our approach significantly beats the default filter in Li et al. (2024) with the EN filter and loses slightly after additional manual language filtering that depends on the target task (+ manual Lang Filter).

Method	None	Lang Filt	DSIR (Xie et al., 2023b)	Handcrafted fastText + EN Lang Filter (Li et al., 2024)	Handcrafted fastText w/o Lang Filter	Handcrafted fastText + manual Lang Filter	Perplexity Correlations
Avg. Rank	3.750	4.000	4.500	3.750	3.250	1.375	1.750

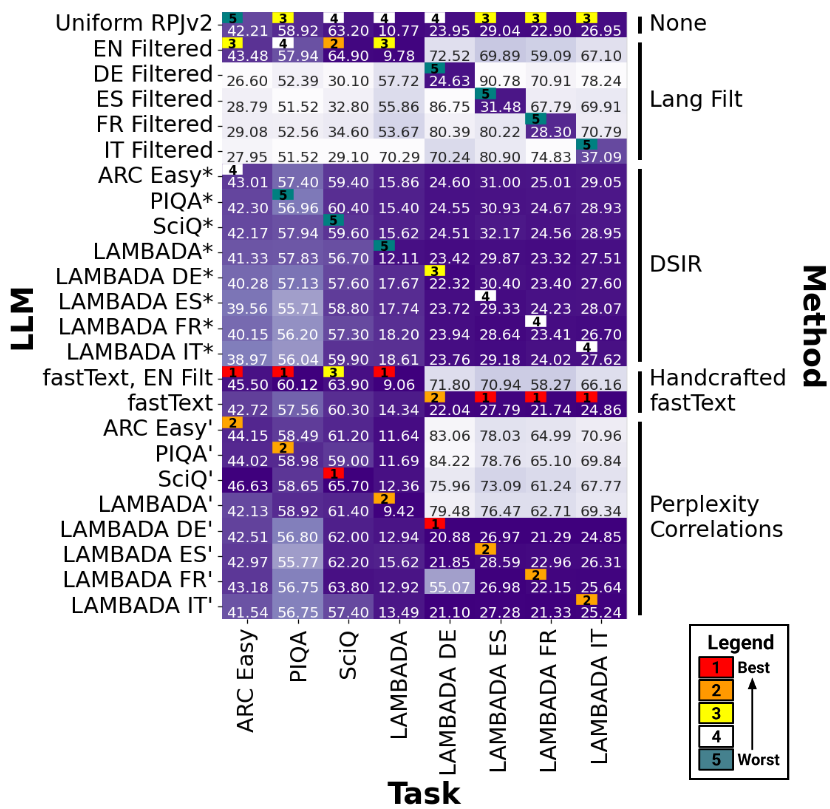


Figure 2: Pretraining results with different data selection methods. Each row is an LLM, and each column is a task. The number in the upper left indicates the ranking of the method when targeting that benchmark compared to other methods (lower is better). Numbers within the heatmap denote accuracy for all benchmarks except the LAMBADA tasks for which the values are log perplexities (where lower scores are better). We find that our approach appropriately optimizes data mixes for the target language and benchmark, and matches the fastText baseline across most benchmarks.

and was also validated at scale (Parmar et al., 2024). Finally, we run the state-of-the-art method for pretraining data quality filtering found by Li et al., which is a fastText classifier that beats all of the heavier-weight proxy-LLM methods tested. The classifier was trained on a benchmark-agnostic and handcrafted objective, which is to classify data as Common Crawl[3] (low quality) or OH2.5 (Teknium, 2023) and Reddit ELI5 (Fan et al., 2019) (high quality). It is combined with an English filter in Li et al.; we present results for this fastText filter with and without the English filter.

Model and hyperparameters. We use the Pythia 160M LLM configuration from Biderman et al. (2023) and optimize the hyperparameters including learning rate, weight decay, and warmup to minimize loss on the uniform sampling (no selection algorithm) baseline. Training hyperparameters were fixed across all methods. We provide additional training and evaluation details in Appendix F.

[3] https://commoncrawl.org

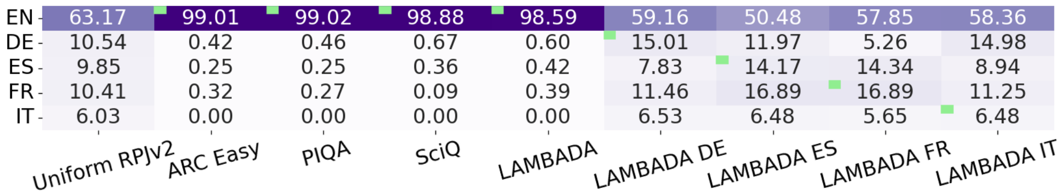


Figure 3: Language distributions of pretraining data selected by perplexity correlations. The default RPJv2 distribution is given in the left column for reference. The English benchmark targets often exclusively select English but the reverse is not the case. In every case, our approach selects more data than the default from the benchmark-matched language (shown as a green box in each column).

Results. We report average rankings over all benchmarks in Table 1, and we find that our approach significantly outperforms the basic baselines of random sampling, language filtering, and DSIR. Compared to the existing state of the art from Li et al. (2024), our approach beats the performance of the default, English-filtered fastText classifier, but loses slightly once we add in a manual language filtering step to enable better performance on the multilingual LAMBADA datasets. For the maintext comparisons, we use the optional fastText classifier from our algorithm to select pretraining data at the page levels, but we show ablations without the classifier in Appendix G.

Figure 2 shows how each data selection method affects benchmark performance in more detail. Each block of rows represents a data selection method, while an individual row represents an LLM within a method that targets a particular benchmark or set of benchmarks. Columns represent benchmarks. We see that language filtering and perplexity correlations both clearly optimize for the target benchmark: within each block, the benchmark column matching each row typically performs best. The pattern is much less obvious for DSIR – the heatmap looks more uniform across LLMs with different task targets. We also see that while language filtering has significant impacts on model performance, our performance significantly exceeds the impact of language filtering across all tested benchmarks.

Figure 3 shows the distribution of languages in pretraining data selected by our method, targeting each benchmark. Our algorithm provides significant enrichment of the corresponding languages for the multilingual benchmarks (LAMBADA_*), but we also find that it does not *exclusively* select domains in one language. In contrast, for English benchmarks our approach selects nearly exclusively English data, likely due to the large quantity of high-quality English data in our pretraining data pool. There are significantly fewer tokens in non-English languages in the pretraining data pool and our τ constraint to prevent duplication has a large impact on the weights when the benchmarks are non-English. We provide the same figure when the τ values are made $5\times$ as large in Appendix H.

Finally, we note that our results are somewhat insensitive to the specifics of the perplexity-correlation procedure we present in Algorithm 1. We show in Appendix G that varying the projection method (linear, L_2) and even using Spearman rank correlations (Spearman, 1904) often work better than the baselines. This suggests that the performance of our approach is not dependent on the precise form of the estimator that is coupled to our theory results, but holds broadly across perplexity-correlation relationships. Additionally, our approach performs better with the optional fastText classifier that our algorithm trains, possibly because it operates at the page-level instead of the domain-level

5.2 Performance Rank Predictions

We have shown that our approach succeeds at selecting useful pretraining data, but how good are the single index model's predictions? A good map of loss to benchmarks would be helpful in selecting among candidate pretraining data mixtures generally, even without using our specific algorithm.

Comparing model performance rankings predicted by our regression to the ground truth, we find generally accurate predictions. Figure 4 shows 5-fold leave-out plots for PIQA, and $\text{LAMBADA}_{\text{FR}}$ with the rank predictions given by $\langle \hat{\boldsymbol{\theta}}^{\text{proj}}, \Phi(\mathbf{x}) \rangle$. Every point in the plot is a held-out point: we estimated $\boldsymbol{\theta}^*$ five times, holding out a different 20% of the data each time, and plotted the prediction for every point when it was held out.

We find that our estimator achieves high ordinal prediction performance across all target tasks. We include 5-fold leave-out R^2 scores for all tasks in Figure 5. However, we complement these strong

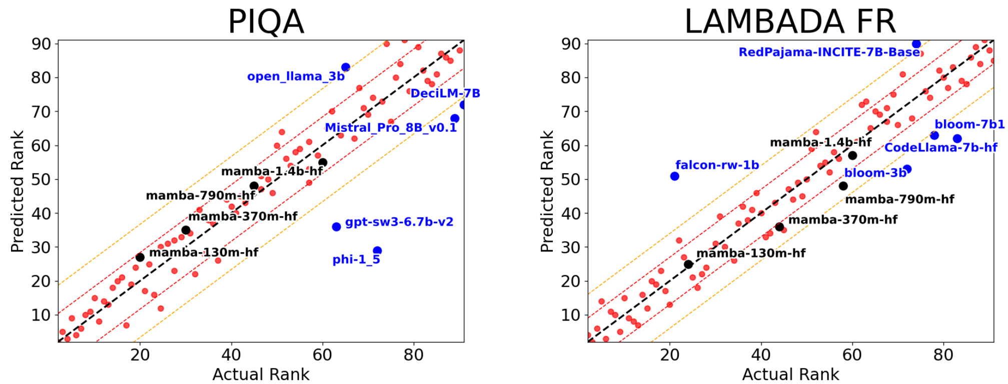


Figure 4: Rank predictions given by $\langle \hat{\boldsymbol{\theta}}^{\text{proj}}, \Phi(\mathbf{x}) \rangle$ for PIQA and LAMBADA FR. A standard deviation (σ) from the ideal fit is shown in red. 2σ is shown in orange. Many models outside 2σ (shown in blue) are trained on atypical data such as multilingual data, code, or GPT-4 (Brown et al., 2020) outputs. Models with atypical architectures (i.e. Mamba (Gu & Dao, 2024)) are shown in black. Generally, our estimate tightly predicts ordinal benchmark performance from web corpus losses.

	ARC Easy	PIQA	SciQ	LAMBADA	LAMBADA DE	LAMBADA ES	LAMBADA FR	LAMBADA IT
Proj. Estimate	89.1±3.6	89.7±3.0	78.1±5.0	92.6±1.7	89.9±2.4	84.3±4.1	92.8±1.8	91.8±2.0
Estimate	85.7±4.6	86.3±3.8	75.2±6.4	91.1±2.0	89.0±2.3	80.6±4.7	91.2±2.1	92.0±1.8
Mean Loss	86.1±4.5	86.3±3.9	75.0±6.3	91.3±2.0	88.8±2.4	80.7±4.8	91.6±2.0	92.1±1.7

Figure 5: Held-out R^2 score of our raw correlation estimate $\hat{\boldsymbol{\theta}}$, our projected estimate $\hat{\boldsymbol{\theta}}^{\text{proj}}$, and the average loss baseline. The 95% bootstrapped confidence intervals are wide enough that no individual comparison is significant. Across benchmarks, $\hat{\boldsymbol{\theta}}^{\text{proj}}$ has statistically significant gains over the baseline (p=0.035) as it is unlikely that $\hat{\boldsymbol{\theta}}^{\text{proj}}$ beats mean loss 7 times out of 8 by chance.

results with the additional observation that simply taking the *mean* loss across all domains is a strong predictor of model performance (bottom row). The surprising effectiveness of average loss over uniformly sampled documents has been discussed extensively (Owen, 2024; Wei et al., 2022; Kaplan et al., 2020) and our results further suggest that regressions with correlations only slightly above the mean loss baseline still can result in effective data selection methods.

Finally, we discuss outliers in our prediction of model performance. Our predictions are accurate for LLMs with usual architectures (e.g. Mamba (Gu & Dao, 2024)), the smallest/largest vocabulary sizes, context sizes, and parameter sizes. However, we also see that LLMs that were trained on unusual data are not as well predicted by our approach (e.g. Phi (Gunasekar et al., 2023)). We may simply require a bigger or more diverse pretraining data pool and set of models to find estimates that work well for models that expect different styles of text.

6 Conclusion

Does high-performance data selection require careful hand-crafted heuristics or prohibitively expensive model training runs? Our work demonstrates an alternative, viable approach – leveraging existing, public models as a source of information for data selection. Pretraining experiments suggest that a simple, correlation-based approach to selecting data can be effective, but more broadly, we show how to 1) use single-index models as a surrogate for downstream performance and 2) build models that relate *losses* to downstream performance and use these surrogates effectively in data selection.

REFERENCES

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv*, 2023.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. *ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv*, 2023.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf, 2023. URL https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard. Open LLM Leaderboard.

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. Pythia: A suite for analyzing large language models across training and scaling. *arXiv*, 2023.

BigScience. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv*, 2023.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. *AAAI*, 2020.

Sid Black, 2023. URL https://huggingface.co/datasets/EleutherAI/lambada_openai. Multilingual LAMBADA.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv*, 2020.

Sheng Chen and Arindam Banerjee. Robust structured estimation with single-index models. *ICML*, 2017.

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*, 2018.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. *ICML*, 2008.

Logan Engstrom, Axel Feldmann, and Aleksander Madry. Dsdm: Model-aware dataset selection with datamodels. *arXiv*, 2024.

Dante Everaert and Christopher Potts. Gio: Gradient information optimization for training dataset selection. *ICLR*, 2024.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: long form question answering. *arXiv*, 2019.

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. The pile: An 800gb dataset of diverse text for language modeling. *arXiv*, 2020.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation. *Zenodo*, 2023.

Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, 2023. URL https://github.com/openlm-research/open_llama.

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. Olmo: Accelerating the science of language models. *arXiv*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv*, 2024.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need. *arXiv*, 2023.

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. Training compute-optimal large language models. *arXiv*, 2022.

Yuzhen Huang, Jinghan Zhang, Zifei Shan, and Junxian He. Compression represents intelligence linearly. *COLM*, 2024.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. *ICML*, 2022.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv*, 2016.

Adam Tauman Kalai and Ravi Sastry. The isotron algorithm: High-dimensional isotonic regression. *COLT*, 2009.

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

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gerard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, Giada Pistilli, Olivier Nguyen, Somaieh Nikpoor, Maraim Masoud, Pierre Colombo, Javier de la Rosa, Paulo Villegas, Tristan Thrush, Shayne Longpre, Sebastian Nagel, Leon Weber, Manuel Muñoz, Jian Zhu, Daniel Van Strien, Zaid Alyafeai, Khalid Almubarak, Minh Chien Vu, Itziar Gonzalez-Dios, Aitor Soroa, Kyle Lo, Manan Dey, Pedro Ortiz Suarez, Aaron Gokaslan, Shamik Bose, David Adelani, Long Phan, Hieu Tran, Ian Yu, Suhas Pai, Jenny Chim, Violette Lepercq, Suzana Ilic, Margaret Mitchell, Sasha Alexandra Luccioni, and Yacine Jernite. The bigscience roots corpus: A 1.6tb composite multilingual dataset. *NeurIPS Datasets and Benchmarks*, 2022.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv*, 2024.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. *arXiv*, 2024.

Llama Team. The llama 3 herd of models. *arXiv*, 2024.

Edward W. Ng and Murray Geller. A table of integrals of the error functions. *Journal of Research of the Natianal Bureau of Standards, Section B: Mathematical Sciences*, 1968.

David Owen. How predictable is language model benchmark performance? *arXiv*, 2024.

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. *ACL*, 2016.

Jupinder Parmar, Shrimai Prabhumoye, Joseph Jennings, Bo Liu, Aastha Jhunjhunwala, Zhilin Wang, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Data, data everywhere: A guide for pretraining dataset construction. *arXiv*, 2024.

Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 1901.

Guilherme Penedo, Hynek Kydlíček, 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. *arXiv*, 2024.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Jiaju Lin, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanislaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. Rwkv: Reinventing rnns for the transformer era. *arXiv*, 2023.

Yaniv Plan, Roman Vershynin, and Elena Yudovina. High-dimensional estimation with geometric constraints. *Information and Inference: A Journal of the IMA*, 2016.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. *arXiv*, 2023.

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*, 1–67, 2020.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. Observational scaling laws and the predictability of language model performance. *arXiv*, 2024.

Shai Shalev-Shwartz and Yoram Singer. Efficient learning of label ranking by soft projections onto polyhedra. *JMLR*, 2006.

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. Dolma: an open corpus of three trillion tokens for language model pretraining research. *arXiv*, 2024.

Charles Spearman. *The Proof and Measurement of Association between Two Things*. The American Journal of Psychology, 1904.

Teknium. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants, 2023. URL https://huggingface.co/datasets/teknium/OpenHermes-2.5.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Together Computer, 2023. URL https://github.com/togethercomputer/RedPajama-Data. RedPajama: an Open Dataset for Training Large Language Models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv*, 2023.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *JMLR*, 2008.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *TMLR*, 2022.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. *W-NUT*, 2017.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *arXiv*, 2019.

Jeffrey M. Wooldridge. *Econometric Analysis of Cross Section and Panel Data*. MIT Press, 2010.

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

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. *NeurIPS*, 2023b.

A MAIN ALGORITHM

Algorithm 1 Perplexity Correlation Based Data Selection

Input: Benchmark error vector $\mathbf{y} \in [0,1]^N$, log-loss matrix normalized as bits-per-byte $\mathbf{X} \in \mathbb{R}_0^{+N \times D}$, available tokens per domain $\mathbf{a} \in \mathbb{N}^D$, and pretraining token target $b \in \mathbb{N}$.
Output: Target token counts per domain $\mathbf{t} \in \mathbb{N}_0^D$, a fastText classifier to filter pretraining data.
Initialize: $\boldsymbol{\gamma} \leftarrow \mathbf{0} \in \mathbb{R}^D, \mathbf{t} \leftarrow [0 \ldots] \in \mathbb{N}_0^D$, counter $\leftarrow 0$.
$\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_N \leftarrow \text{rank}(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N)$ ▷ 1. Compute the $\boldsymbol{\gamma}$ correlation coefficient
for $i, j \in 0$ **to** N **do**
  $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} + \text{sign}(y_i - y_j) \cdot (\mathbf{r}_i - \mathbf{r}_j)$
for $i \in$**ArgSort**($\boldsymbol{\gamma}$, descending=True) **do** ▷ 2. Select most to least correlated domains
  $t_i \leftarrow \min(a_i, b -$ counter$)$
  counter $\leftarrow$ counter $+ a_i$
  if counter $\geq b$ **then**
    Break
classifier = trainFastText(positive $= 1_{t>0}$, negative $= 1_{t=0}$)
Return $\mathbf{t}$, classifier

B ESTIMATOR SOLUTION

B.1 LEMMA 1

Statement of Lemma 1 *Define the PDF of* HalfNormal *as* $f(x;\sigma) = \frac{\sqrt{2}}{\sigma\sqrt{\pi}} e^{-\frac{x^2}{2\sigma^2}}$ *for* $x > 0$ *and 0 otherwise. Now, suppose:*

- $\boldsymbol{\beta}$ *is a vector with* $||\boldsymbol{\beta}||_2 = 1$
- $\mathbf{Z}_1, \mathbf{Z}_2$ *are vectors* $\sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- $\epsilon \sim \mathcal{N}(0, \sigma^2)$
- $Z' \sim \mathcal{N}(0, 1)$
- $Z_+ \sim$ HalfNormal(1).

Then we have:

$$Z_{1j} | \langle \mathbf{Z}_1 - \mathbf{Z}_2, \boldsymbol{\beta} \rangle + \epsilon > 0 \stackrel{d}{=} Z' \sqrt{1 - \frac{\beta_j^2}{2 + \sigma^2}} + \frac{\beta_j}{\sqrt{2+\sigma^2}} Z_+,$$

where Z_{1j} *is the* j*-th entry of* $\mathbf{Z}_1$.

Proof: First, note:

$$Z_{1j} | \langle \mathbf{Z}_1 - \mathbf{Z}_2, \boldsymbol{\beta} \rangle + \epsilon > 0 \stackrel{d}{=} Z_{1j} | \left\langle \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \epsilon/\sigma \end{bmatrix}, \begin{bmatrix} \boldsymbol{\beta} \\ -\boldsymbol{\beta} \\ \sigma \end{bmatrix} \right\rangle > 0 \stackrel{d}{=} Z_{1j} | \left\langle \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \epsilon/\sigma \end{bmatrix}, \begin{bmatrix} \boldsymbol{\beta} \\ -\boldsymbol{\beta} \\ \sigma \end{bmatrix} / \sqrt{2+\sigma^2} \right\rangle > 0,$$

where $\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix}$ denotes the vector-valued result of concatenating vectors and scalars. For readability, we set $\mathbf{Z}_c = \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \epsilon/\sigma \end{bmatrix}$ and $\boldsymbol{\beta}_c = \begin{bmatrix} \boldsymbol{\beta} \\ -\boldsymbol{\beta} \\ \sigma \end{bmatrix} / \sqrt{2+\sigma^2}$.

Given that $\boldsymbol{\beta}_c$ is unit-norm (by supposition, $\boldsymbol{\beta}$ is unit-norm), and every element of $\mathbf{Z}_c$ is $\sim \mathcal{N}(0,1)$ (even ϵ/σ), we can easily split a conditional random vector containing Z_{1j} into a conditionally dependent component and independent component:

$$\mathbf{Z}_c | \langle \mathbf{Z}_c, \boldsymbol{\beta}_c \rangle > 0 \stackrel{d}{=} (\mathbf{I} - \boldsymbol{\beta}_c \boldsymbol{\beta}_c^\top)\mathbf{Z}'' + \boldsymbol{\beta}_c \mathbf{Z}_+.$$

The first term is orthogonal to $\boldsymbol{\beta}_c$ and so it is the part of $\mathbf{Z}_c$ that is not subject to the condition. In the unconditional case, $\mathbf{Z}_c \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and so $\mathbf{Z}'' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The second term is the part of $\mathbf{Z}_c$ that is in the direction of $\boldsymbol{\beta}_c$. $\mathbf{Z}_+ \sim$ HalfNormal$(\mathbf{I})$ because our dot product condition is satisfied for half of the possible non-orthogonal $\mathbf{Z}_c$ values. Now, we focus on finding $\mathbf{Z}_c | \langle \mathbf{Z}_c, \boldsymbol{\beta}_c \rangle > 0$ for a single index j. We have (for C defined to be the dimensionality of $\boldsymbol{\beta}_c$):

$$\begin{aligned}((\mathbf{I} - \boldsymbol{\beta}_c \boldsymbol{\beta}_c^\top)\mathbf{Z}'')_j + (\boldsymbol{\beta}_c \mathbf{Z}_+)_j &= Z''_j(1 - {\beta_c}_j^2) - \sum_{\substack{1 \le i \le C \\ i \ne j}} Z''_i \beta_{cj} \beta_{ci} + \beta_j Z_{+j} \\ &= Z''_j - \sum_{i=1}^{C} Z''_i \beta_{cj} \beta_{ci} + \beta_j Z_{+j}.\end{aligned}$$

Now, note that $Z''_j - \sum_{i=1}^{C} Z''_i \beta_{cj} \beta_{ci}$ is the sum of independent zero-mean Gaussians with variances given by 1 and ${\beta_c}_j^2 {\beta_c}_i^2$, so it itself is a zero-mean Gaussian $Y \sim \mathcal{N}(0, 1 - \sum_{i=1}^{C} {\beta_c}_j^2 {\beta_c}_i^2)$. We can also use the fact that $\sum_{i=1}^{C} {\beta_c}_i^2 = 1$ (recall that $\boldsymbol{\beta_c}$ is unit norm) to get: $Y \sim \mathcal{N}(0, 1 - {\beta_c}_j^2)$. So we have that the conditional Z_{1j} is given by:

$$Z' \sqrt{1 - {\beta_c}_j^2} + \beta_{cj} Z_+ = Z' \sqrt{1 - \frac{\beta_j^2}{2 + \sigma^2}} + \frac{\beta_j}{\sqrt{2 + \sigma^2}} Z_+,$$

for $Z' \sim \mathcal{N}(0,1)$. As a corollary, we can see that Z_{2j} under the same condition is given by:

$$Z' \sqrt{1 - \frac{\beta_j^2}{2 + \sigma^2}} + \frac{-\beta_j}{\sqrt{2 + \sigma^2}} Z_+.$$

B.2 LEMMA 2

Statement of Lemma 2 *Suppose that Φ is the CDF of a standard Gaussian, a and c are constants, and $Z \sim \mathcal{N}(0,1)$. Then we have:*

$$\mathbb{E}[\Phi(aZ + c)] = \Phi\left(\frac{c}{\sqrt{1 + a^2}}\right).$$

Proof: By the definition of the CDF of a standard Gaussian, we have:

$$\mathbb{E}[\Phi(aZ + c)] = \mathbb{E}[P(X \le aZ + c)],$$

where $X \sim \mathcal{N}(0,1)$. Continuing, we have:

$$= \mathbb{E}[P(X - aZ - c \le 0)].$$

Now, note that $X - aZ - c$ is the sum of independent Gaussian random variables with given mean and variance; it itself is a Gaussian random variable $\sim \mathcal{N}(-c, a^2 + 1)$. To find $P(X - aZ - c \le 0)$, we can evaluate its CDF at 0:

$$= \mathbb{E}\left[\Phi\left(\frac{c}{\sqrt{a^2 + 1}}\right)\right] = \Phi\left(\frac{c}{\sqrt{a^2 + 1}}\right).$$

B.3 LEMMA 3

Statement of Lemma 3 *Suppose Φ is the standard Gaussian CDF, $Z_+ \sim$ HalfNormal(1), and b and a are constants. Then we have:*

$$\mathbb{E}\left[\Phi\left(\frac{Z_+ b}{\sqrt{a^2 + 1}}\right)\right] = \frac{1}{2} + \frac{1}{\pi} \tan^{-1}\left(\frac{b}{\sqrt{a^2 + 1}}\right).$$

Proof: By the definition of expected value, we can take the following integral where f_{Z_+} is the PDF of Z_+. We integrate from 0 instead of $-\infty$ because the PDF of the Standard Half Normal is 0 in the domain below 0:

$$\begin{aligned}
\mathbb{E}\left[\Phi\left(\frac{Z_+ b}{\sqrt{a^2+1}}\right)\right] &= \int_0^\infty \Phi\left(\frac{zb}{\sqrt{a^2+1}}\right) f_{Z_+}(z)dz \\
&= \int_0^\infty \Phi\left(\frac{zb}{\sqrt{a^2+1}}\right) \frac{\sqrt{2}}{\sqrt{\pi}} e^{\frac{-z^2}{2}} dz \\
&= \frac{1}{\sqrt{2\pi}} \left(\int_0^\infty e^{\frac{-z^2}{2}} dz + \int_0^\infty \operatorname{erf}\left(\frac{zb}{\sqrt{2}\sqrt{a^2+1}}\right) e^{\frac{-z^2}{2}} dz\right) \; \textbf{(*)}.
\end{aligned}$$

The second integral is generally non-trivial to solve, but luckily we can solve it by using Equation 2 in Section 4.3 of the integral table from Ng & Geller (1968), which states:

$$\int_0^\infty \operatorname{erf}(cx) e^{-d^2x^2} dx = \frac{\sqrt{\pi}}{2d} - \frac{1}{d\sqrt{\pi}} \tan^{-1}\left(\frac{d}{c}\right)$$

Where c and d are real and positive. We split the solution by cases: $b > 0$, $b = 0$, and $b < 0$. We find that in every case, we can manipulate our integral so that the solution is trivial or the constant inside the erf$(\cdot)$ is positive (and so we can use the integral table). In every case, we find that the solution is $\frac{1}{2} + \frac{1}{\pi}\tan^{-1}\left(\frac{b}{\sqrt{a^2+1}}\right)$.

Case 1: $b > 0$. We can use the integral table directly:

$$\begin{aligned}
\textbf{(*)} &= \frac{1}{\sqrt{2\pi}} \left(\frac{\sqrt{\pi}}{\sqrt{2}} + \frac{\sqrt{\pi}}{\sqrt{2}} - \frac{\sqrt{2}}{\sqrt{\pi}} \tan^{-1}\left(\frac{\sqrt{a^2+1}}{b}\right)\right) \\
&= \frac{1}{2} + \frac{1}{2} - \frac{1}{\pi} \tan^{-1}\left(\frac{\sqrt{a^2+1}}{b}\right).
\end{aligned}$$

Then, using the identity:

$$\tan^{-1} x + \tan^{-1} \frac{1}{x} = \frac{\pi}{2} \text{ if } x > 0,$$

we find the following:

$$= \frac{1}{2} + \frac{1}{\pi} \tan^{-1}\left(\frac{b}{\sqrt{a^2+1}}\right).$$

Case 2: $b = 0$. Note that $\operatorname{erf}(0) = 0$; we do not have to use the integral table:

$$\begin{aligned}
\textbf{(*)} &= \frac{1}{\sqrt{2\pi}} \left(\frac{\sqrt{\pi}}{\sqrt{2}} + 0\right) \\
&= \frac{1}{2}.
\end{aligned}$$

Because $\tan^{-1}(0) = 0$, we have:

$$= \frac{1}{2} + \frac{1}{\pi} \tan^{-1}\left(\frac{b}{\sqrt{a^2+1}}\right).$$

Case 3: $b < 0$. Because erf$(\cdot)$ is an odd function, we can pull the negative out:

$$\textbf{(*)} = \frac{1}{\sqrt{2\pi}} \left(\int_0^\infty e^{\frac{-z^2}{2}} dz - \int_0^\infty \operatorname{erf}\left(\frac{z|b|}{\sqrt{2}\sqrt{a^2+1}}\right) e^{\frac{-z^2}{2}} dz\right).$$

Now we can use the integral table as in the $b > 0$ case:

$$\begin{aligned}
&= \frac{1}{\sqrt{2\pi}} \left(\frac{\sqrt{\pi}}{\sqrt{2}} - \frac{\sqrt{\pi}}{\sqrt{2}} + \frac{\sqrt{2}}{\sqrt{\pi}} \tan^{-1}\left(\frac{\sqrt{a^2+1}}{|b|}\right)\right) \\
&= \frac{1}{2} + \frac{1}{2} - \frac{1}{\pi} \tan^{-1}\left(\frac{\sqrt{a^2+1}}{|b|}\right).
\end{aligned}$$

We can then use the same identity again:

$$\tan^{-1} x + \tan^{-1} \frac{1}{x} = \frac{\pi}{2} \text{ if } x > 0$$

to get:

$$= \frac{1}{2} - \frac{1}{\pi} \tan^{-1} \left(\frac{|b|}{\sqrt{a^2+1}} \right).$$

Because $\tan^{-1}$ is an odd function, we can put the negative inside of it:

$$= \frac{1}{2} + \frac{1}{\pi} \tan^{-1} \left(\frac{b}{\sqrt{a^2+1}} \right).$$

B.4 Full Proof

Here, we prove:

$$\mathbb{E}[\text{sign}(y_1 - y_2)(\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2))] = \frac{2}{\pi} \sin^{-1} \left(\frac{\boldsymbol{\theta}^*}{\sqrt{4 + 2\sigma_1^2 + 2\sigma_2^2}} \right)$$

with $y_1, y_2, \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2)$, and $\boldsymbol{\theta}^*$ defined in the main text, for the case where ϵ_1 and ϵ_2 are zero-mean Gaussian noise $\sim \mathcal{N}(0, \sigma_1^2)$ and $\sim \mathcal{N}(0, \sigma_2^2)$, respectively.

It is easy to see that this is a more general version of the following theorem.

Theorem 1 *When $\epsilon \sim \mathcal{N}(0, \sigma^2)$, we have:*

$$\mathbb{E}[\text{sign}(y_i - y_j)(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))] = \frac{2}{\pi} \sin^{-1} \left(\frac{\boldsymbol{\theta}^*}{2\sqrt{1+\sigma^2}} \right). \tag{7}$$

Proof: By symmetry, we have:

$$\begin{aligned}
&\mathbb{E}[\text{sign}(y_1 - y_2)(\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2))] \\
&= \frac{1}{2}\mathbb{E}[\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2) | \text{sign}(y_1 - y_2) > 0] + \frac{1}{2}\mathbb{E}[-(\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2)) | \text{sign}(y_1 - y_2) < 0].
\end{aligned}$$

By increasing monotonicity of f, we have $\text{sign}(y_1 - y_2) > 0 \iff \langle \mathbf{x}_1 - \mathbf{x}_2, \boldsymbol{\theta}^* \rangle + \epsilon_\Delta > 0$, for $\epsilon_\Delta = \epsilon_1 - \epsilon_2 \sim \mathcal{N}(0, \sigma_1^2 + \sigma_2^2)$. So:

$$\begin{aligned}
=&\frac{1}{2}\mathbb{E}[\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2) | \langle \mathbf{x}_1 - \mathbf{x}_2, \boldsymbol{\theta}^* \rangle + \epsilon_\Delta > 0] \\
&+ \frac{1}{2}\mathbb{E}[-(\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2)) | \langle \mathbf{x}_1 - \mathbf{x}_2, \boldsymbol{\theta}^* \rangle + \epsilon_\Delta < 0].
\end{aligned}$$

Because $\mathbf{x}_1 \stackrel{d}{=} \mathbf{x}_2$ and $\epsilon_\Delta \stackrel{d}{=} -\epsilon_\Delta$, the two expected values above are the same:

$$= \mathbb{E}[\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2) | \langle \mathbf{x}_1 - \mathbf{x}_2, \boldsymbol{\theta}^* \rangle + \epsilon_\Delta > 0].$$

By linearity of expectation:

$$= \mathbb{E}[\Phi(\mathbf{x}_1) | \langle \mathbf{x}_1 - \mathbf{x}_2, \boldsymbol{\theta}^* \rangle + \epsilon_\Delta > 0] - \mathbb{E}[\Phi(\mathbf{x}_2) | \langle \mathbf{x}_1 - \mathbf{x}_2, \boldsymbol{\theta}^* \rangle + \epsilon_\Delta > 0].$$

Now, we focus on finding the overall estimate for a single index j. By Lemma 1, we have, for $Z \sim \mathcal{N}(0, 1)$ and $Z_+ \sim \text{HalfNormal}(1)$:

$$\Phi(x_{1j}) | \langle \mathbf{x}_1 - \mathbf{x}_2, \boldsymbol{\theta}^* \rangle + \epsilon_\Delta > 0 \stackrel{d}{=} \Phi(Za + Z_+ b_1).$$

Here, $a = \sqrt{1 - \frac{(\theta_j^*)^2}{2+\sigma_1^2+\sigma_2^2}}$ and $b_1 = \frac{\theta_j^*}{\sqrt{2+\sigma_1^2+\sigma_2^2}}$. As a corollary of Lemma 1, we can see:

$$\Phi(x_{2j}) | \langle \mathbf{x}_1 - \mathbf{x}_2, \boldsymbol{\theta}^* \rangle + \epsilon_\Delta > 0 \stackrel{d}{=} \Phi(Za + Z_+ b_2).$$

Where $b_2 = -\frac{\theta_j^*}{\sqrt{2+\sigma_1^2+\sigma_2^2}}$. So for the index j, our estimate is:

$$\begin{aligned}&\mathbb{E}[\Phi(Za + Z_+b_1)] - \mathbb{E}[\Phi(Za + Z_+b_2)]\\&= \mathbb{E}[\mathbb{E}[\Phi(Za + c)|c = Z_+b_1]] - \mathbb{E}[\mathbb{E}[\Phi(Za + c)|c = Z_+b_2]].\end{aligned}$$

Using Lemma 2, we have:

$$= \mathbb{E}\left[\Phi\left(\frac{Z_+b_1}{\sqrt{a^2+1}}\right)\right] - \mathbb{E}\left[\Phi\left(\frac{Z_+b_2}{\sqrt{a^2+1}}\right)\right].$$

Then, using Lemma 3, we have:

$$\begin{aligned}&= \frac{1}{2} + \frac{1}{\pi}\tan^{-1}\left(\frac{b_1}{\sqrt{a^2+1}}\right) - \frac{1}{2} - \frac{1}{\pi}\tan^{-1}\left(\frac{b_2}{\sqrt{a^2+1}}\right)\\&= \frac{1}{\pi}\tan^{-1}\left(\frac{b_1}{\sqrt{a^2+1}}\right) - \frac{1}{\pi}\tan^{-1}\left(\frac{b_2}{\sqrt{a^2+1}}\right).\end{aligned}$$

Using the fact that $\tan^{-1}$ is an odd function and $b_2 = -b_1$, we get:

$$= \frac{2}{\pi}\tan^{-1}\left(\frac{b_1}{\sqrt{a^2+1}}\right).$$

Now, we write a and b_1 in terms of θ_j^*:

$$\begin{aligned}&= \frac{2}{\pi}\tan^{-1}\left(\frac{\frac{\theta_j^*}{\sqrt{2+\sigma_1^2+\sigma_2^2}}}{\sqrt{2 - \frac{(\theta_j^*)^2}{2+\sigma_1^2+\sigma_2^2}}}\right)\\&= \frac{2}{\pi}\tan^{-1}\left(\frac{\frac{\theta_j^*}{\sqrt{4+2\sigma_1^2+2\sigma_2^2}}}{\sqrt{1 - \left(\frac{\theta_j^*}{\sqrt{4+2\sigma_1^2+2\sigma_2^2}}\right)^2}}\right).\end{aligned}$$

Using the identity $\sin^{-1} x = \tan^{-1}\left(\frac{x}{\sqrt{1-x^2}}\right)$, we have:

$$= \frac{2}{\pi}\sin^{-1}\left(\frac{\theta_j^*}{\sqrt{4+2\sigma_1^2+2\sigma_2^2}}\right).$$

B.5 Corollary 1

Corollary 1 *Suppose that $\hat{\boldsymbol{\theta}}$ is any vector of fixed weights and $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then, conditioning on the event $\langle \hat{\boldsymbol{\theta}}, \mathbf{x}_i \rangle < \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_j \rangle$, we have with probability 1 that:*

$$\langle \hat{\boldsymbol{\theta}}, \mathbb{E}[\Phi(\mathbf{x}_i) \mid \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_i \rangle < \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_j \rangle] \rangle < \langle \hat{\boldsymbol{\theta}}, \mathbb{E}[\Phi(\mathbf{x}_j) \mid \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_i \rangle < \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_j \rangle] \rangle. \tag{9}$$

To see this, we can find:

$$\mathbb{E}[\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2) | \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_1 \rangle + \epsilon_1 > \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_2 \rangle + \epsilon_2] = \mathbb{E}[\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2) | \langle \hat{\boldsymbol{\theta}}, \mathbf{x}_1 - \mathbf{x}_2 \rangle + \epsilon_\Delta > 0]$$

Note that we have already computed this expected value in the proof above; for an index j, it is:

$$\frac{2}{\pi}\sin^{-1}\left(\frac{\hat{\theta}_j}{\sqrt{4+2\sigma_1^2+2\sigma_2^2}}\right).$$

Because $\sin^{-1}$ is an odd function, the above expression has the same sign as $\hat{\theta}_j$. Because the values at every index of $\mathbb{E}[\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2)]$ under our condition and $\hat{\boldsymbol{\theta}}$ are the same sign, we have $\langle \mathbb{E}[\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2)], \hat{\boldsymbol{\theta}} \rangle > 0$, so $\langle \hat{\boldsymbol{\theta}}, \mathbb{E}[\Phi(\mathbf{x}_1)] \rangle > \langle \hat{\boldsymbol{\theta}}, \mathbb{E}[\Phi(\mathbf{x}_2)] \rangle$.

C Optimal projected weights solutions

C.1 Linear Projection

Theorem 2 *Suppose we want to solve:*

$$\hat{\boldsymbol{\theta}}^{\text{proj}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^D}{\arg\min} -\langle \boldsymbol{\theta}, \hat{\boldsymbol{\theta}} \rangle,$$

subject to:

$$\sum_{i=1}^{D} \theta_i = 1$$
$$0 \leq \theta_i \leq \tau_i, \forall i \in [1, D],$$

where $\tau_i > 0$ are fixed values. Then, the solution is:

$$\hat{\theta}_k^{\text{proj}} = \begin{cases} \tau_k & \textit{if } \sum_{j:\, r_j(\hat{\theta}_j) \geq r_k(\hat{\theta}_k)} \tau_j \leq 1 \\ 1 - \sum_{j:\, r_j(\hat{\theta}_j) > r_k(\hat{\theta}_k)} \tau_j & \textit{if } \sum_{j:\, r_j(\hat{\theta}_j) \geq r_k(\hat{\theta}_k)} \tau_j \geq 1 \wedge \sum_{j:\, r_j(\hat{\theta}_j) > r_k(\hat{\theta}_k)} \tau_j \leq 1 \\ 0 & \textit{otherwise} \end{cases}, \quad (10)$$

where r is some function that breaks all ties between $\hat{\theta}_j$ and $\hat{\theta}_k$ for $k \neq j$, and otherwise leaves the ordinal relationships the same.

Proof: We proceed by considering each of the three cases from Equation 10.

Case 1. Suppose for the sake of contradiction that the optimal solution is $\hat{\boldsymbol{\theta}}^{\text{proj}}$ and yet $\hat{\theta}_k^{\text{proj}} < \tau_k$ for some $\hat{\theta}_k^{\text{proj}}$ falling under the first case of Equation 10. Now suppose that we construct a $\boldsymbol{\theta}'$ also satisfying the projection constraints that is the same as $\hat{\boldsymbol{\theta}}^{\text{proj}}$ except in these places:

$$\theta'_k = \hat{\theta}_k^{\text{proj}} + \Delta = \tau_k$$
$$\theta'_p = \hat{\theta}_p^{\text{proj}} - \delta_1 \geq 0$$
$$\vdots$$
$$\theta'_q = \hat{\theta}_q^{\text{proj}} - \delta_n \geq 0$$

for some $\Delta = \sum_{i=1}^{n} \delta_i > 0$ where $\hat{\theta}_p \geq \cdots \geq \hat{\theta}_q$ are all of the $\hat{\theta}$ values which do not fall under the first condition and where the corresponding $\hat{\theta}^{\text{proj}}$ values are nonzero. We know that there must be some $\hat{\theta}_p^{\text{proj}}, \cdots, \hat{\theta}_q^{\text{proj}}$ from which we can subtract $\delta_1, \cdots, \delta_n$ (and so from which we can take the Δ) because $\sum_{j:\, r_j(\hat{\theta}_j) \geq r_k(\hat{\theta}_k)} \tau_j \leq 1$. Now, we have:

$$\begin{aligned}
&\langle \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}^{\text{proj}} \rangle - \langle \hat{\boldsymbol{\theta}}, \boldsymbol{\theta}' \rangle \\
&= \hat{\theta}_k \hat{\theta}_k^{\text{proj}} + \hat{\theta}_p \hat{\theta}_p^{\text{proj}} + \cdots + \hat{\theta}_q \hat{\theta}_q^{\text{proj}} - \hat{\theta}_k \hat{\theta}_k^{\text{proj}} - \hat{\theta}_k \Delta - \hat{\theta}_p \hat{\theta}_p^{\text{proj}} - \cdots - \hat{\theta}_q \hat{\theta}_q^{\text{proj}} + \hat{\theta}_p \delta_1 + \cdots + \hat{\theta}_q \delta_n \\
&= -\hat{\theta}_k \Delta + \hat{\theta}_p \delta_1 + \cdots + \hat{\theta}_q \delta_n \\
&\leq \hat{\theta}_p (\delta_1 + \cdots + \delta_n) - \hat{\theta}_k \Delta \\
&= \hat{\theta}_p \Delta - \hat{\theta}_k \Delta \\
&\leq 0.
\end{aligned}$$

At this point, the only way to avoid the contradiction result would be if $\hat{\theta}_k = \hat{\theta}_p = \cdots = \hat{\theta}_q$. Otherwise, the above non-strict inequality would be a strict inequality. If $\hat{\theta}_k = \hat{\theta}_p = \cdots = \hat{\theta}_q$, then we know that $\hat{\theta}_k$ is the smallest $\hat{\theta}$ value satisfying condition 1 and all of the other greater $\hat{\theta}$ values satisfying condition 1 must be projected to their τ threshold value (otherwise we would get the contradiction result). In this edge case can see above that rearranging the remaining weight among equal $\hat{\theta}$ values does not change the dot product, so all of the solutions that we can get without the contradiction result are equivalently optimal (including the solution from Equation 10).

Case 3. This is analogous to case 1. Suppose for the sake of contradiction that the optimal solution is $\hat{\boldsymbol{\theta}}^{\text{proj}}$ and yet $\hat{\theta}_k^{\text{proj}} > 0$ for some $\hat{\theta}_k^{\text{proj}}$ falling under the third case of Equation 10. Now suppose that we construct a $\boldsymbol{\theta}'$ also satisfying the projection constraints that is the same as $\hat{\boldsymbol{\theta}}^{\text{proj}}$ except in these places:

$$\begin{aligned}
\theta'_k &= \hat{\theta}_k^{\text{proj}} - \Delta = 0 \\
\theta'_p &= \hat{\theta}_p^{\text{proj}} + \delta_1 \leq \tau_p \\
&\vdots \\
\theta'_q &= \hat{\theta}_q^{\text{proj}} + \delta_n \leq \tau_q
\end{aligned}$$

for some $\Delta = \sum_{i=1}^{n} \delta_i > 0$ where $\hat{\theta}_p \geq \cdots \geq \hat{\theta}_q$ are all of the $\hat{\theta}$ values which do not fall under the third condition and where the corresponding $\hat{\theta}^{\text{proj}}$ values are not at their thresholds. By construction we know that there must be some $\hat{\theta}_p^{\text{proj}}, \cdots, \hat{\theta}_q^{\text{proj}}$ to which we can add $\delta_1, \cdots, \delta_n$. Now, we have:

$$\begin{aligned}
&\langle \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}^{\text{proj}} \rangle - \langle \hat{\boldsymbol{\theta}}, \boldsymbol{\theta}' \rangle \\
&= \hat{\theta}_k \hat{\theta}_k^{\text{proj}} + \hat{\theta}_p \hat{\theta}_p^{\text{proj}} + \cdots + \hat{\theta}_q \hat{\theta}_q^{\text{proj}} - \hat{\theta}_k \hat{\theta}_k^{\text{proj}} + \hat{\theta}_k \Delta - \hat{\theta}_p \hat{\theta}_p^{\text{proj}} - \cdots - \hat{\theta}_q \hat{\theta}_q^{\text{proj}} - \hat{\theta}_p \delta_1 - \cdots - \hat{\theta}_q \delta_n \\
&= \hat{\theta}_k \Delta - \hat{\theta}_p \delta_1 - \cdots - \hat{\theta}_q \delta_n \\
&\leq -\hat{\theta}_q (\delta_1 + \cdots + \delta_n) + \hat{\theta}_k \Delta \\
&= -\hat{\theta}_q \Delta + \hat{\theta}_k \Delta \\
&\leq 0.
\end{aligned}$$

At this point, the only way to avoid the contradiction result would be if $\hat{\theta}_k = \hat{\theta}_p = \cdots = \hat{\theta}_q$. Otherwise, the above non-strict inequality would be a strict inequality. If $\hat{\theta}_k = \hat{\theta}_p = \cdots = \hat{\theta}_q$, then we know that $\hat{\theta}_k$ is the largest $\hat{\theta}$ value satisfying condition 3 and all of the other smaller $\hat{\theta}$ values satisfying condition 3 must be projected to 0 (otherwise we would get the contradiction result). In this edge case, we can see above that rearranging the remaining weight among equal $\hat{\theta}$ values does not change the dot product, so all of the solutions that we can get without the contradiction result are equivalently optimal (including the solution from Equation 10).

Case 2. Above, we show that both Case 1 and Case 3 are true. So, the remaining weight must be given to the single value of $\hat{\boldsymbol{\theta}}^{\text{proj}}$ not covered by either case.

C.2 QUADRATIC PROJECTION

C.2.1 LEMMA 4

Statement of Lemma 4 *Suppose that $\hat{\boldsymbol{\theta}}^{\text{proj}}$ is the optimal solution to:*

$$\hat{\boldsymbol{\theta}}^{\text{proj}} = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^D} ||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}||_2^2,$$

subject to:

$$\begin{gathered}
\sum_{i=1}^{D} \theta_i = 1 \\
0 \leq \theta_i \leq \tau_i, \forall i \in [1, D],
\end{gathered}$$

where $\tau_i > 0$ are fixed values. Then, $\hat{\theta}_s^{\text{proj}} = 0$ implies that any j with $\hat{\theta}_s > \hat{\theta}_j$ must have $\hat{\theta}_j^{\text{proj}} = 0$.

Proof: This is similar to Lemma 2 from Shalev-Shwartz & Singer (2006). Assume for the sake of contradiction $\hat{\theta}_s^{\text{proj}} = 0$ and $\hat{\theta}_s > \hat{\theta}_j$, yet we have $\hat{\theta}_j^{\text{proj}} > 0$.

Now we can construct another vector $\boldsymbol{\theta}'$ that is the same as $\hat{\boldsymbol{\theta}}^{\text{proj}}$, except in two places:

$$\theta'_s = \hat{\theta}^{\text{proj}}_s + \Delta$$
$$\theta'_j = \hat{\theta}^{\text{proj}}_j - \Delta,$$

for some Δ satisfying $0 < \Delta < \min(\hat{\theta}^{\text{proj}}_j, \tau_s - \hat{\theta}^{\text{proj}}_s)$. This bound on Δ ensures that θ' is still within the thresholds. We know that Δ can exist because $\min(\hat{\theta}^{\text{proj}}_j, \tau_s - \hat{\theta}^{\text{proj}}_s) > 0$ (by supposition, $\tau_s - \hat{\theta}^{\text{proj}}_s = \tau_s - 0 > 0$ and $\hat{\theta}^{\text{proj}}_j > 0$).

Now we can compute:

$$\begin{aligned}
||\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}^{\text{proj}}||_2^2 - ||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}'||_2^2 &= (\hat{\theta}_s - \hat{\theta}^{\text{proj}}_s)^2 + (\hat{\theta}_j - \hat{\theta}^{\text{proj}}_j)^2 - (\hat{\theta}_s - (\hat{\theta}^{\text{proj}}_s + \Delta))^2 - (\hat{\theta}_j - (\hat{\theta}^{\text{proj}}_j - \Delta))^2 \\
&= 2\Delta((\hat{\theta}_s - \hat{\theta}^{\text{proj}}_s) - (\hat{\theta}_j - \hat{\theta}^{\text{proj}}_j) - \Delta) \\
&> 2\Delta((\hat{\theta}_s - \hat{\theta}^{\text{proj}}_s) - (\hat{\theta}_j - \hat{\theta}^{\text{proj}}_j) - \min(\hat{\theta}^{\text{proj}}_j, \tau_s - \hat{\theta}^{\text{proj}}_s)) \\
&\geq 2\Delta((\hat{\theta}_s - \hat{\theta}^{\text{proj}}_s) - (\hat{\theta}_j - \hat{\theta}^{\text{proj}}_j) - \hat{\theta}^{\text{proj}}_j) \\
&= 2\Delta(\hat{\theta}_s - \hat{\theta}_j) \\
&> 0.
\end{aligned}$$

So $\hat{\boldsymbol{\theta}}^{\text{proj}}$ cannot be the optimal solution.

C.2.2 Lemma 5

Statement of Lemma 5 *Suppose that $\hat{\boldsymbol{\theta}}^{\text{proj}}$ is the optimal solution to:*

$$\hat{\boldsymbol{\theta}}^{\text{proj}} = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^D} ||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}||_2^2,$$

subject to:

$$\sum_{i=1}^{D} \theta_i = 1$$
$$0 \leq \theta_i \leq \tau_i, \forall i \in [1, D],$$

where $\tau_i > 0$ are fixed values. Then, $\hat{\theta}^{\text{proj}}_s = \tau_s$ implies $\hat{\theta}^{\text{proj}}_j = \tau_j$ for any $\hat{\theta}_j - \tau_j > \hat{\theta}_s - \tau_s$.

Proof: Again, this is similar to Lemma 2 from Shalev-Shwartz & Singer (2006). Assume for the sake of contradiction $\hat{\theta}^{\text{proj}}_s = \tau_s$ and $\hat{\theta}_j - \tau_j > \hat{\theta}_s - \tau_s$, yet we have $\hat{\theta}^{\text{proj}}_j < \tau_j$.

Now we can construct another vector $\boldsymbol{\theta}'$ that is the same as $\hat{\boldsymbol{\theta}}^{\text{proj}}$, except in two places:

$$\theta'_s = \hat{\theta}^{\text{proj}}_s - \Delta$$
$$\theta'_j = \hat{\theta}^{\text{proj}}_j + \Delta,$$

for some Δ satisfying $0 < \Delta < \min(\hat{\theta}^{\text{proj}}_s, \tau_j - \hat{\theta}^{\text{proj}}_j)$. This bound on Δ ensures that θ' is still within the thresholds. We know that Δ can exist because $\min(\hat{\theta}^{\text{proj}}_s, \tau_j - \hat{\theta}^{\text{proj}}_j) > 0$ (by supposition, $\tau_j - \hat{\theta}^{\text{proj}}_j > 0$ and $\hat{\theta}^{\text{proj}}_s = \tau_s > 0$).

Now we can compute:

$$\begin{aligned}
||\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}^{\text{proj}}||_2^2 - ||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}'||_2^2 &= (\hat{\theta}_s - \hat{\theta}^{\text{proj}}_s)^2 + (\hat{\theta}_j - \hat{\theta}^{\text{proj}}_j)^2 - (\hat{\theta}_s - (\hat{\theta}^{\text{proj}}_s - \Delta))^2 - (\hat{\theta}_j - (\hat{\theta}^{\text{proj}}_j + \Delta))^2 \\
&= 2\Delta((\hat{\theta}_j - \hat{\theta}^{\text{proj}}_j) - (\hat{\theta}_s - \hat{\theta}^{\text{proj}}_s) - \Delta) \\
&> 2\Delta((\hat{\theta}_j - \hat{\theta}^{\text{proj}}_j) - (\hat{\theta}_s - \hat{\theta}^{\text{proj}}_s) - \min(\hat{\theta}^{\text{proj}}_s, \tau_j - \hat{\theta}^{\text{proj}}_j)) \\
&\geq 2\Delta((\hat{\theta}_j - \hat{\theta}^{\text{proj}}_j) - (\hat{\theta}_s - \hat{\theta}^{\text{proj}}_s) - (\tau_j - \hat{\theta}^{\text{proj}}_j)) \\
&= 2\Delta((\hat{\theta}_j - \tau_j) - (\hat{\theta}_s - \hat{\theta}^{\text{proj}}_s)) \\
&= 2\Delta((\hat{\theta}_j - \tau_j) - (\hat{\theta}_s - \tau_s)) \\
&> 0.
\end{aligned}$$

So $\hat{\boldsymbol{\theta}}^{\text{proj}}$ cannot be the optimal solution.

C.2.3 Full Proof

Theorem 3 *Suppose we want to solve:*

$$\hat{\boldsymbol{\theta}}^{\text{proj}} = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^D} ||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}||_2^2,$$

subject to:

$$\sum_{i=1}^{D} \theta_i = 1$$
$$0 \le \theta_i \le \tau_i, \forall i \in [1, D],$$

where $\tau_i > 0$ are fixed values. Then the solution is:

$$\hat{\theta}_k^{\text{proj}} = \min(\max(\hat{\theta}_k - \lambda, 0), \tau_k),$$

where λ is found (through e.g. bisection search) to satisfy:

$$\sum_{i=1}^{D} \min(\max(\hat{\theta}_i - \lambda, 0), \tau_i) = 1.$$

Proof: Note that this problem is the same as the simplex projection problem from Shalev-Shwartz & Singer (2006) and Duchi et al. (2008), except here we have additional $\theta_i \le \tau_i$ constraints. The Lagrangian for this problem is[4]:

$$\mathcal{L}(\boldsymbol{\theta}, \mu, \zeta, \lambda) = \frac{1}{2}||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}||_2^2 + \lambda\left(-1 + \sum_{i=1}^{N}\theta_i\right) - \langle \mu, \boldsymbol{\theta}\rangle + \langle \zeta, \boldsymbol{\theta} - \tau\rangle.$$

To find the optimality condition with respect to a single index of $\boldsymbol{\theta}$, we set the derivative to zero:

$$\frac{d\mathcal{L}}{d\theta_i} = \theta_i - \hat{\theta}_i + \lambda - \mu_i + \zeta_i = 0.$$

The complimentary slackness KKT condition gives us that $\zeta_i = \mu_i = 0$ when $0 < \theta_i < \tau_i$, so for θ_i not at the boundary of our constraints, we get:

$$\theta_i = \hat{\theta}_i - \lambda.$$

So, we have that for all $\theta_i \in (0, \tau_i)$, there is a shared value λ which we subtract from $\hat{\theta}_i$ to get the value of θ_i. How do we know which θ_i are 0 and which θ_i are τ_i, though?

Assume that we know λ. By Lemma 4, we can characterize the optimal solution as:

$$\hat{\theta}_k^{\text{proj}} = \max(\hat{\theta}_k - \lambda, 0),$$

for $\hat{\theta}_k^{\text{proj}} \ne \tau_k$. By Lemma 5, we can characterize the optimal solution as:

$$\hat{\theta}_k^{\text{proj}} = \min(\hat{\theta}_k - \lambda, \tau_k),$$

for $\hat{\theta}_k^{\text{proj}} \ne 0$. So, we can combine these two forms to get:

$$\hat{\theta}_k^{\text{proj}} = \min(\max(\hat{\theta}_k - \lambda, 0), \tau_k).$$

Now recall that we have the following constraint:

$$\sum_{i=1}^{D} \min(\max(\hat{\theta}_i - \lambda, 0), \tau_i) = 1.$$

Given this constraint, we can find λ through search (moving the value up or down). We can see this by noticing that $\sum_{i=1}^{D} \min(\max(\hat{\theta}_i - \lambda, 0), \tau_i)$ is a strictly decreasing function of λ between the setting of λ that makes $\hat{\theta}_i - \lambda > 0$ for at least one i, and the setting of λ that makes $\hat{\theta}_i - \lambda < \tau_i$ for at least one i. So in this range, there is only one setting of λ that satisfies this equation. We can only choose a λ outside of this range when $\sum_{i=1}^{D} \tau_i = 1$, and in this case the solution is trivial: $\hat{\theta}_i^{\text{proj}} = \tau_i$ for all i.

[4] Note that multiplying $||\hat{\boldsymbol{\theta}}^{\text{proj}} - \boldsymbol{\theta}||_2^2$ by $\frac{1}{2}$ does not change the minimization problem and enables us to get rid of a factor of 2 after taking the derivative of the Lagrangian.

D Alternative Methods

Our estimator is far from the only reasonable high-dimensional, single-index model estimator. We briefly discuss some alternatives and the tradeoffs involved before moving to experimental results.

We could use classic low-dimensional methods regularized for the high-dimensional setting. This includes ordinal regression (Wooldridge, 2010) and the isotron algorithm (Kalai & Sastry, 2009). We found these methods to underperform correlation-based estimators, and tuning hyperparameters added additional complexity that was not needed in the correlation-based approaches.

Another class of methods involve scaling laws (Kaplan et al., 2020; Llama Team, 2024; Ruan et al., 2024). We could transform the y values via an inverse sigmoid or power law, and fit high-dimensional linear regression methods (e.g. ridge, partial least squares, or Lasso). We initially found this approach promising, but the inverse transforms were unstable, and the combination of fitting the nonlinear transform and regularization required significant amounts of tuning.

Rank-correlation methods, including our robustified version of the estimator from Chen & Banerjee (2017), and even the standard Spearman correlation (Spearman, 1904) (see Appendix G) performed well. We believe that in general, robust per-feature correlations are likely to perform well as $D \gg N$, and extreme levels of regularization are needed to obtain reasonable models. Sparse methods such as the Lasso (Tibshirani, 1996) are one classic answer, but we cannot necessarily assume that the underlying correlations $\boldsymbol{\theta}^*$ are sparse, and we did not find these techniques to perform well.

E Loss Matrix Computation Specifics

For all of our experiments, we computed the loss matrix as follows. For efficiency purposes, we sampled only 25 pages for a domain's bits-per-byte (BPB) computation even if a domain had more than 25 pages. To get an LLM's BPB on a page, we split the page into chunks of text that were 512 tokens according to a reference tokenizer (we used the Llama 2 7B tokenizer; Touvron et al. 2023). These text chunks turned out to be small enough to fit in the context of every LLM we tested. We then averaged BPB across chunks for each page and then across pages for each domain.

F Additional Details for Pretraining Experiments

In this section, we specify hyperparameters and methods used for LLM pretraining and evaluation for our LLM pretraining experiments. We also specify settings used for the data-selection methods.

F.1 LLM Pretraining

We trained each LLM on 4 NVIDIA A100 GPUs. At 3.2B tokens, each training run took under 3 hours with the Hugging Face Trainer (Wolf et al., 2019) and appropriate PyTorch (Ansel et al., 2024) compile flags. We provide pretraining hyperparameters in Table 2. Given our per-device batch size, we found the learning rate by increasing it by a factor of 2 until we saw instability and then using the highest learning rate where no instability was observed. Refer to the Pythia paper (Biderman et al., 2023) for more information; we initialized the model from scratch using their 160M model configuration at `https://huggingface.co/EleutherAI/pythia-160m`. Other hyperparameters can be assumed to be Hugging Face Trainer defaults at the time of this writing.

F.2 LLM Evaluation

At the end of the pretraining script, we used the Eleuther AI Eval Harness (Gao et al., 2023). For efficiency, we set the sample limit to 5000 examples per benchmark. Elsewhere, we used the default settings. On 4 NVIDIA A100s, it took only a few minutes per LLM to compute evaluation results for SciQ, ARC Easy, PIQA, LAMBADA, and all of the translations of LAMBADA.

F.3 DSIR

DSIR (Xie et al., 2023b), despite its simplicity, requires some tuning. A decision must be made about how to format the bemchmark data into a single piece of text per example so that it can be compared with potential pretraining data in terms of n-gram overlap. The LAMBADA tasks only

Table 2: LLM Pretraining Hyperparameters

Parameter	**Value**
Per-device Batch Size	128
Learning Rate	5×10^{-3}
Warmup Ratio	0.1
Adam β_1	0.9
Adam β_2	0.95
Adam ϵ	1×10^{-8}
Weight Decay	0.1
LR Scheduler	cosine
Max Grad Norm	1.0
BF 16	True
Distributed Backend	nccl
Gradient Accumulation Steps	1

Table 3: Unique pretraining tokens selected per benchmark, from DSIR.

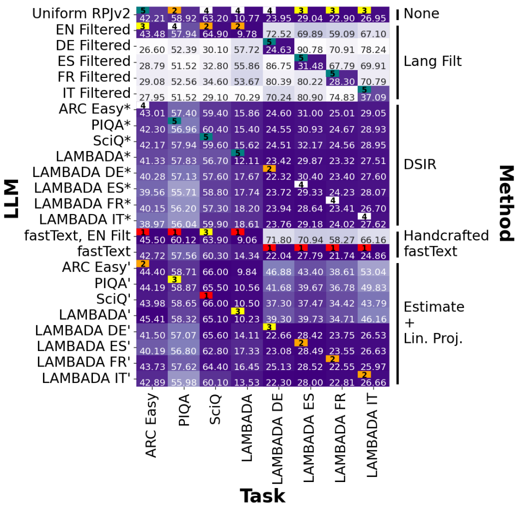
Benchmark	**Tokens**
ARC Easy	2,905,206,499
PIQA	2,910,486,295
SCIQ	2,920,734,042
LAMBADA	3,022,219,424
$\text{LAMBADA}_{\text{DE}}$	3,210,986,137
$\text{LAMBADA}_{\text{ES}}$	3,396,528,704
$\text{LAMBADA}_{\text{FR}}$	3,413,930,081
$\text{LAMBADA}_{\text{IT}}$	3,384,854,845

have one text column per example, so the decision here is trivial. Examples from the other tasks each have a question, possibly a context, and a set of multiple choice answers to choose from. We chose to concatenate all of these columns together with spaces to form one piece of text per example, duplicating the same question as a prefix for each different answer.

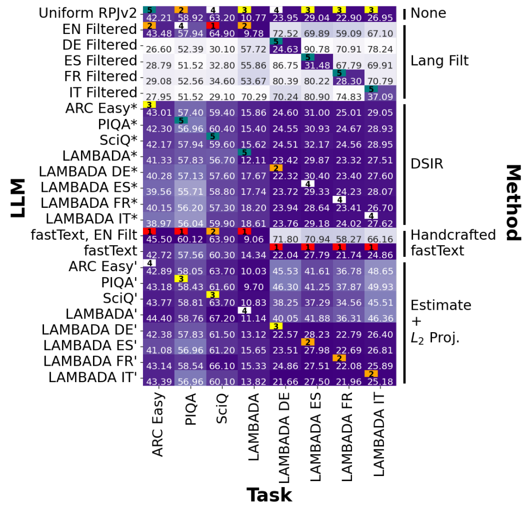
DSIR does not allow the user to specify the exact number of unique tokens desired for pretraining. It only allows the specification of the number of unique pages, which can have wildly varying token counts. For every DSIR job, we set the desired number of pages to 3325589, which we found through binary search to produce slightly more than 3.2B unique tokens for $\text{LAMBADA}_{\text{FR}}$. It was expensive to find this number for even one bechmark, because for each iteration of the binary search, we had to run DSIR and then the Pythia tokenizer to know how many tokens resulted from the input page number parameter. We provide the number of unique tokens from DSIR for each task in Table 3. We pretrained on 3.2B tokens for every LLM regardless of whether all of them were unique.

F.4 FASTTEXT

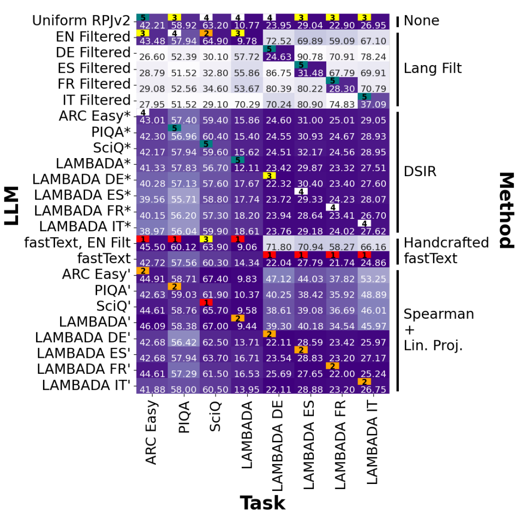
The “SOTA” fastText model from Li et al. (2024) is available here: https://huggingface.co/mlfoundations/fasttext-oh-eli5. We used this model to filter data by sorting pages by the model’s “high quality” score, including the top pages in order until we had either reached or gone



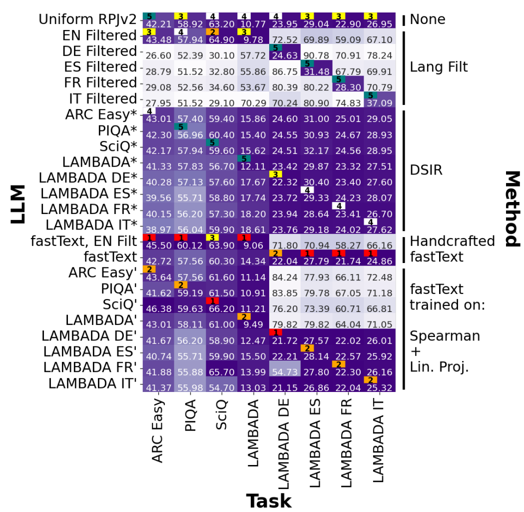
(a) Estimate with linear projection. This is our algorithm from the main text without training the additional fastText filter.



(b) Estimate with quadratic projection. Same as (a) except the linear projection is replaced with the quadratic projection.



(c) Spearman rank correlation with linear projection. Same as (a) except we replaced our estimator with the Spearman rank correlation.



(d) fastText filter trained on data selected in (c). This is the same as our algorithm in the main text, replacing our estimator with the Spearman rank correlation.

Figure 6: Pretraining results for different methods within our paradigm. Overall, we see that many rank-correlation pretraining data selection approaches perform well.

slightly over 3.2B unique tokens. This aligns with the data-selection procedure in the original paper, and is also essentially the same as running the linear projection (Equation 10) at the page-level. We also applied this method when selecting data using our own fastText filter trained by our algorithm.

G Additional Pretraining Results

In Figure 6, we present additional pretraining results for methods in our loss-performance correlation data selection paradigm. We find that using Spearman rank correlation (Spearman, 1904) in place of our estimator achieves comparable performance. On some tests, it performs even better than our estimator. We also find that using the quadratic projection, while perhaps more intuitive, leads to worse performance than the linear projection.

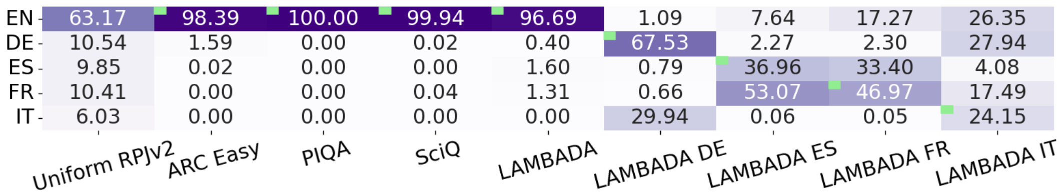


Figure 7: This figure is analogous to Figure 3, except the τ thresholds have been multiplied by 5. We see that our approach selects even more relevant data when the selection pool is larger.

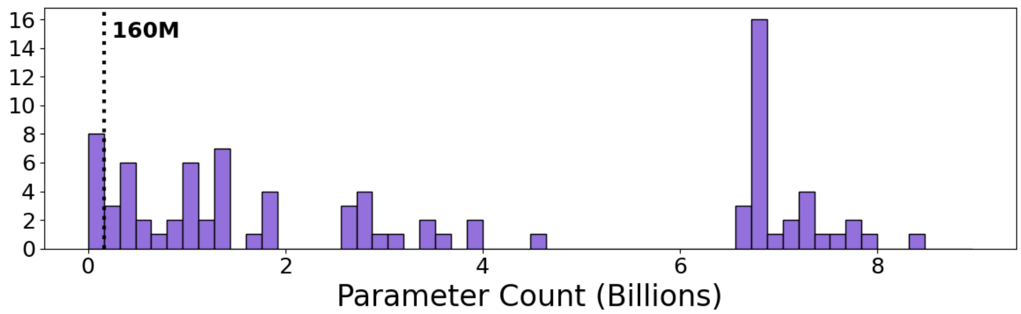


Figure 8: The parameter-count histogram of the 90 models from the Open LLM Leaderboard (Beeching et al., 2023) that we used to compute our estimate for pretraining data selection. Bar widths are 160M. The smallest model in the sample has ≈33M parameters and the largest has ≈9B. The spike around 6.7B parameters is due to a large number of partially trained Pythia (Biderman et al., 2023) checkpoints from the same training run at that scale. Our algorithm has the hard task of selecting pretraining data for 160M parameter models, which is abnormally small in the set of models used to compute the estimate.

H Pretraining Token Distribution with $5 \times \tau$

Figure 7 shows what the projected estimate in our pretraining experiments would be if we had a pretraining data pool $5\times$ as large. We see here that the estimate does an even better job at selecting pretraining data with the language that matches the target task.

I Parameter Count Distribution for Estimator LLMs

In Figure 8, we present the parameter-count histogram of the 90 models from the Open LLM Leaderboard (Beeching et al., 2023) that we used to compute our estimate for pretraining data selection. Only 8 models here are less than 160M parameters. Despite this, our estimate can be used to effectively pretrain 160M parameter LLMs.

J Analysis of the model-loss matrix $\mathbf{X}$

What information is contained in the matrix of model losses $\mathbf{X}$? Clearly, it must contain semantically meaningful information about the data, such as the language that a piece of text is in. We performed PCA (Pearson, 1901) and t-SNE (van der Maaten & Hinton, 2008) on $\mathbf{X}$ and plotted the first two components for each of our 9,841 domains. As shown in the first row of Figure 9, we found two components with relatively high singular values. The first component clearly corresponds with the language of a domain. The second component corresponds with the average bits-per-byte or entropy of a domain. The t-SNE components show the same general pattern as well as showing that the language clusters are very well separated. As shown in our plots, there are several salient clusters within the language clusters. Within the English cluster, we found a subcluster for luxury goods, another for legal services and information, another for academic research, and even a cluster for funeral homes.

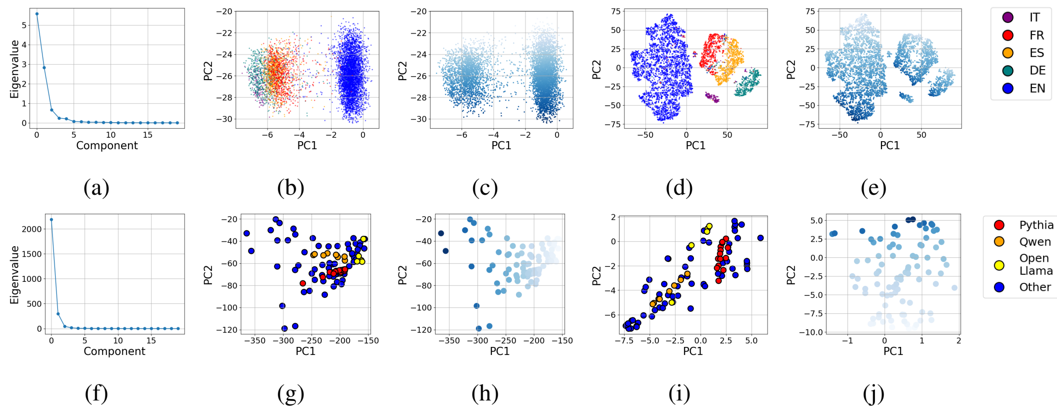


Figure 9: Analysis of the loss matrix. The first row treats domains as examples to be projected via PCA, while the second row treats models as examples. Panels (a): eigenvalue decay for the eigendecomposition of the $D \times D$ covariance matrix resulting from the loss matrix; a few dominant PCs are seen. (b) and (c): domains plotted by the first two PCA components showing separation of language in b and entropy in c. (d,e) show analogous plots in t-SNE with a clearer separation of language. (f): eigenvalue decay analogous to (a). (g,h): models plotted by the first two PCA components showing clustering by model family (clusters show Pythia (Biderman et al., 2023), Qwen (Bai et al., 2023), and OpenLlama (Geng & Liu, 2023) derivatives – the three largest clusters in our data), and average model loss. (i,j) show analogous results under t-SNE where (i) is normalized to remove per-model entropy differences.

The second row of Figure 9 shows plots for the loss matrix when we take the principal components of the other dimension, where points correspond to the 90 LLMs. For PCA, PC1 corresponds to entropy. For both cases, it is less clear what the other PCs are, but when we color the three largest families of models in our data (Pythia (Biderman et al., 2023), Qwen (Bai et al., 2023), and OpenLlama (Geng & Liu, 2023)), we see that model families are clustered together in the PC graphs.