Selection of LLM Fine-Tuning Data Based on Orthogonal Rules

Xiaomin Li[1*], **Mingye Gao**[2], **Zhiwei Zhang**[3], **Chang Yue**[4], **Hong Hu**[5]
[1]Harvard University
[2]Massachusetts Institute of Technology
[3]Pennsylvania State University
[4]Princeton University
[5]Washington University in St. Louis

Abstract

High-quality training data is critical to the performance of large language models (LLMs). Recent work has explored using LLMs to rate and select data based on a small set of human-designed criteria (rules), but these approaches often rely heavily on heuristics, lack principled metrics for rule evaluation, and generalize poorly to new tasks. We propose a novel rule-based data selection framework that introduces a metric based on the orthogonality of rule score vectors to evaluate and select complementary rules. Our automated pipeline first uses LLMs to generate diverse rules covering multiple aspects of data quality, then rates samples according to these rules and applies the determinantal point process (DPP) to select the most independent rules. These rules are then used to score the full dataset, and high-scoring samples are selected for downstream tasks such as LLM fine-tuning. We evaluate our framework in two experiment setups: (1) alignment with ground-truth ratings and (2) performance of LLMs fine-tuned on the selected data. Experiments across IMDB, Medical, Math, and Code domains demonstrate that our DPP-based rule selection consistently improves both rating accuracy and downstream model performance over strong baselines.

Code & Data — https://github.com/XiaominLi1998/Submission-OrthoRules

Extended version — https://arxiv.org/abs/2410.04715

1 Introduction

Large language models (LLMs) have been widely adopted across a diverse range of applications. Training these models—both during pre-training and fine-tuning—typically requires large and varied datasets. Prior work has shown that data quality plays a critical role in the effectiveness of LLM training (Brown 2020; Chowdhery et al. 2023; Du et al. 2022; Dubey et al. 2024; Wenzek et al. 2019). For example, Meta's LIMA paper (Zhou et al. 2024) demonstrated that just 1K carefully curated samples can outperform a much larger set of 50K original samples. Similar findings have emerged from other studies, where selecting high-quality data subsets improves both training efficiency and model performance (Cao, Kang, and Sun 2023; Hsieh et al. 2023; Xie et al. 2024; Sachdeva et al. 2024; Zhang et al. 2023).

*Correspondence to: Xiaomin Li (xiaominli@g.harvard.edu).

A recent trend involves using LLM-as-a-judge to rate data quality based on a set of human-designed metrics, referred to here as "rules" (Yuan et al. 2024; Wettig et al. 2024; Bai et al. 2022; Mu et al. 2024). For instance, Wettig et al. (2024) used LLMs to score pre-training data according to four predefined rules. RedPajama (Together AI 2023) built a rule set with over 40 criteria for evaluating LLM training data. In specialized domains like safety, Constitutional AI (Bai et al. 2022) defined a set of "constitutions" for generating safe synthetic data; Huang et al. (2024) later expanded this to 133 rules. OpenAI's Rule-based Rewarding (Mu et al. 2024) similarly introduced 21 general safety rules as part of the reinforcement learning from human feedback (RLHF) pipeline. These rule-based approaches provide better interpretability by assigning granular, rule-specific scores rather than a single opaque quality label, and studies have shown this fine-grained strategy leads to more accurate assessments (Yuan et al. 2024; Wettig et al. 2024; Bai et al. 2022; Mu et al. 2024).

Despite this progress, several challenges remain. First, designing an effective rule set is difficult, as acknowledged in the prior work mentioned above. Most current rules rely heavily on human heuristics and are often too broad to yield useful signal. Second, there is a lack of principled metrics for evaluating rules, and little understanding of how rule quality and quantity affect downstream outcomes. Prior work (Bai et al. 2022; Wettig et al. 2024; Together AI 2023) typically randomly selects a subset of rules for scoring, a decision that can substantially impact the resulting data quality. Additionally, many rules are correlated, introducing redundancy and potential bias in the ratings (Wettig et al. 2024). This raises an important question: with a "constitution" (a pool of rules) in hand, exactly which "laws" (a subset of task-related rules) should be applied to a specific task? Random selection as in Bai et al. (2022) may not be the optimal strategy. A third major drawback is the limited flexibility of these rules; they are often designed for specific settings, such as safety tasks, and lack general applicability across diverse settings.

In our work, we aim to address these challenges. We first leverage GPT-4 (Achiam et al. 2023) to automatically generate candidate rules, prompting it with descriptions of both the target task and the source dataset. The generated rules are then manually reviewed to ensure clarity and validity. At this stage, some generated rules are found to be repetitive or redundant. Our strategy is to first generate a rule set that can

comprehensively cover various data quality aspects, which may include many correlated and redundant rules. The second step is to filter out repetitive ones, and select a subset of rules that are relatively uncorrelated/independent. This is achieved by first using the rules to rate a batch of data, creating one score vector for each rule, and then assessing the independence of rule subsets through the overall orthogonality of their corresponding score vectors. We propose the formula in Section 3 to measure the orthogonality and use determinant point process (DPP) sampling (Macchi 1975; Borodin and Olshanski 2000) to identify a subset of independent rules. Once the rules are determined, the third step is to apply them to rate the full dataset and select the high-quality subset. Combining rule generation, rule-based rating, rule selection by DPP, and data selection, our method establishes a fully automated framework for rule-based data selection (illustrated in Figure 1). To our knowledge, we are the first to introduce a mathematical evaluation metric for rules based on score vector orthogonality. Moreover, this pipeline can be applied to new tasks with minimal manual effort, addressing the shortcomings of existing methods.

We validate our approach across four domains: IMDB, Medical, Math, and Code, by fine-tuning LLMs on data selected by our framework. We show that our rule-based method consistently improves data rating accuracy and leads to better model performance. Below is a list of main contributions of our work:

1. **Rule-free vs. Rule-based Rating.** We provide the first systematic comparison showing that fine-grained, rule-based evaluation outperforms rule-free methods in data quality assessment.
2. **Rule Evaluation Metric:** We introduce a novel rule evaluation metric designed to promote low correlation and high diversity among rules, the first rule evaluation metric for rule-based LLM-as-a-judge to our knowledge. We also propose the method of using DPP on task-aware score vectors to select a subset of independent rules.
3. **Automated Rule-based Selection Pipeline.** We confirm that LLMs are effective rule generators, eliminating the need for manual rule crafting. Our automated pipeline generates the rules, selects the rules, and then uses them to identify high-quality data.
4. **Cross-Domain and Cross-Model.** We validate our method both by comparing against ground-truth ratings and by benchmarking LLMs trained on selected data across multiple domains (IMDB, Medical, Math, Code) and model families (Pythia-1B and Llama3-8B with LoRA), confirming the generality of our approach.

2 Related Work

Rule-based rating. Some studies adopt a more fine-grained approach to data quality by distilling it into a finite set of metrics, which we refer to as "rules". For instance, RedPajama (Together AI 2023) provides over 40 quality rules as basic filtering criteria. More relevant to our research, several studies apply this rule-based idea to rate LLM data. Yuan et al. (2024) rates each sample from 1 to 5 based on how many of five predefined criteria it satisfies. Wettig et al. (2024) uses four general rules to rate and select pre-training data, while Sun et al. (2024) proposes 16 handcrafted rules to assess response quality. Rule-based evaluation is also used in safety applications. Constitutional AI (Bai et al. 2022) applies a random subset of 16 safety critique rules (called "constitutions") to iteratively revise synthetic data. In Mu et al. (2024), rule-based scores from 21 safety rules are directly integrated into the RLHF reward, while Wang et al. (2024) trains a composite reward model using rule-based ratings. As noted earlier, existing rule designs often rely heavily on human heuristics, lack principled evaluation metrics and exploration of rule sizes, and are not easily adaptable. Our framework addresses these limitations.

LLM data selection. There are different genres of data selection approaches for LLMs. Basic filtering, such as setting thresholds on word length, is used in many studies to eliminate low-quality data (Soldaini et al. 2024; Wenzek et al. 2019; Raffel et al. 2020; Penedo et al. 2023). Fuzzy deduplication removes repetitive or similar samples (Allamanis 2019; Lee et al. 2021; Gao et al. 2020; Jiang et al. 2022). Another method is *heuristic classification*, selecting data based on similarity to high-quality sources such as Wikipedia (Brown 2020; Touvron et al. 2023; Chowdhery et al. 2023; Du et al. 2022; Gao et al. 2020; Wenzek et al. 2019). More recently, querying LLMs to rate data directly has become a standard practice (Li et al. 2023; Chen et al. 2023; Bai et al. 2022; Wettig et al. 2024; Yuan et al. 2024; Dubois et al. 2024). Other methods include coreset and optimization-based subset selection (Xia et al. 2022; Borsos, Mutny, and Krause 2020).

3 Methodology

3.1 Definitions and Notations

We introduce the definitions of the primary objects considered in our method:

- R: the total number of available rules.
- r: the number of selected rules, using a specified rule selection method.
- $\mathcal{D}$: the set of all data samples, with its size denoted by $N \stackrel{\text{def}}{=} |\mathcal{D}|$.
- $\mathcal{B} \subseteq \mathcal{D}$: a batch of data samples, randomly selected for evaluating the correlation of rules during the rule selection step, with its size denoted by $n \stackrel{\text{def}}{=} |\mathcal{B}|$ (we use $n = 10^4$ in experiments in Section 5).
- $\boldsymbol{S} \in \mathbb{R}^{n \times R}$: the rating matrix $\boldsymbol{S}$ where each entry $S_{i,j}$ represents the score of the i-th data sample according to the j-th rule and is constrained to the interval $[0, 1]$.
- $\bar{\boldsymbol{S}} \in \mathbb{R}^{n \times r}$: a submatrix of $\boldsymbol{S}$ consisting of the r selected columns from $\boldsymbol{S}$, corresponding to the r selected rules.

Measure orthogonality: To guide rule selection, we propose a metric based on the orthogonality of score vectors. To achieve this, we introduce a mathematical framework to quantify the degree of orthogonality or correlation among a given set of score vectors. Given $\bar{\boldsymbol{S}} \in \mathbb{R}^{n \times r}$, define its sample mean as $\mu_i \stackrel{\text{def}}{=} \frac{1}{n} \sum_{k=1}^{n} S_{k,i}$, and sample covariance

matrix $\widehat{\boldsymbol{\Sigma}}(\bar{\boldsymbol{S}}) \in \mathbb{R}^{r\times r}$ by

$$\widehat{\boldsymbol{\Sigma}}_{i,j}(\bar{\boldsymbol{S}}) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{k=1}^{n}\left(S_{k,i}-\mu_i\right)\left(S_{k,j}-\mu_j\right).$$

Then define the *sample correlation matrix* $\widehat{\boldsymbol{C}}(\bar{\boldsymbol{S}}) \in \mathbb{R}^{r\times r}$ as

$$\widehat{\boldsymbol{C}}_{i,j}(\bar{\boldsymbol{S}}) \stackrel{\text{def}}{=} \frac{\widehat{\boldsymbol{\Sigma}}_{i,j}(\bar{\boldsymbol{S}})}{\sqrt{\widehat{\boldsymbol{\Sigma}}_{i,i}(\bar{\boldsymbol{S}})\,\widehat{\boldsymbol{\Sigma}}_{j,j}(\bar{\boldsymbol{S}})}}, \quad 1\le i,j\le r.$$

To quantify the degree of correlation/dependence for a given rating submatrix $\bar{\boldsymbol{S}}$, we introduce the quantity called *rule correlation*:

$$\rho(\bar{\boldsymbol{S}}) \stackrel{\text{def}}{=} \frac{1}{r}\left\|\widehat{\boldsymbol{C}}(\bar{\boldsymbol{S}})-\boldsymbol{I}_r\right\|_F = \frac{1}{r}\sqrt{\sum_{i\neq j}\left(\widehat{\boldsymbol{C}}_{i,j}(\bar{\boldsymbol{S}})\right)^2}. \quad (1)$$

Here, $\boldsymbol{I}_r$ is the identity matrix and $\|\cdot\|_F$ is the Frobenius norm. Intuitively, $\widehat{\boldsymbol{C}}(\bar{\boldsymbol{S}}) \approx \boldsymbol{I}_r$ means the columns of $\bar{\boldsymbol{S}}$ have low pairwise correlations, making $\rho(\bar{\boldsymbol{S}})$ small. This metric quantifies how much the columns of $\bar{\boldsymbol{S}}$ deviate from orthogonality, by measuring the deviation of its correlation matrix from the identity matrix. The second equality in (1) provides another intuitive understanding: $\rho(\bar{\boldsymbol{S}})$ essentially aggregates the correlations of all pairwise correlations of rules (i,j) for $i\neq j$.

To validate our approach of using $\mathcal{B}$, batch of random n samples, to generate rating scores and evaluate rule correlations based on these score vectors, we present the following theorem, which characterizes the concentration of the sample rule correlation around the true rule correlation.

Theorem 1. *Let $\boldsymbol{C} \in \mathbb{R}^{r\times r}$ be the* true *correlation matrix among the r rules, i.e.,*

$$\boldsymbol{C}_{j,\ell} = \frac{\Sigma_{j,\ell}}{\sqrt{\Sigma_{j,j}\Sigma_{\ell,\ell}}}, \quad (2)$$

and assume each candidate rule has nontrivial variance ($\Sigma_{j,j} \ge \sigma_{\min}^2 > 0$ for some constant $\sigma_{\min}$). We draw n i.i.d. samples $\{\boldsymbol{x}^{(k)}\}_{1\le k\le n}$ where each $\boldsymbol{x}^{(k)} \in [0,1]^r$ represents ratings for the k-th sample based on r rules. From these we form the empirical correlation matrix $\widehat{\boldsymbol{C}}$. Then there exists a universal constant $c>0$ such that for any $\delta>0$ and sufficiently large n,

$$\mathbb{P}\left(\left|\rho(\bar{\boldsymbol{S}}) - \frac{1}{r}\left\|\boldsymbol{C}-\mathbf{I}_r\right\|_F\right| \le c\sqrt{\frac{\ln\left(\frac{r^2}{\delta}\right)}{n}}\right) > 1-\delta.$$

In particular, if $\boldsymbol{C}$ is close to $\mathbf{I}_r$ in Frobenius norm (i.e., if the rules are nearly uncorrelated "in truth"), then the observed rule correlation $\rho(\bar{\boldsymbol{S}})$ also remains close to zero for sufficiently large n.

Proof. See Appendix A. □

3.2 Determinantal point process (DPP)

The optimal solution to this mathematical problem of selecting the most orthogonal subset of a set of vectors is NP-hard (Civril and Magdon-Ismail 2007; Kulesza, Taskar et al. 2012) but we use DPP sampling to provide a relatively good solution. The determinant point process (DPP) is a probabilistic model that describes the likelihood of selecting diverse subsets from a larger set (Macchi 1975; Borodin and Olshanski 2000). Mathematically, a DPP is defined by a kernel matrix that describes the similarities between elements in a set. The probability of selecting a particular subset is proportional to the determinant of the corresponding submatrix of this kernel matrix. Intuitively, subsets with highly similar items (leading to higher correlation in the submatrix) have smaller determinants and are thus less likely to be chosen.

DPP Definitions. Given a discrete ground set $\mathcal{Y}$, without loss of generality we let $\mathcal{Y} = \{1,2,\ldots,R\}$, a (discrete) DPP defines a probability measure over $2^{\mathcal{Y}}$, the power set of $\mathcal{Y}$. Let Y be a randomly chosen subset. Then for any subset $A \subseteq \mathcal{Y}$, the probability of A being chosen by a DPP is given by:

$$\mathbb{P}(A \subseteq Y) = \det(\boldsymbol{K}_A)$$

where $\boldsymbol{K} \in \mathbb{R}^{R\times R}$ is a real positive-semidefinite matrix called the *kernel matrix* and $\boldsymbol{K}_A \stackrel{\text{def}}{=} [\boldsymbol{K}]_{i,j\in A}$ is the submatrix of $\boldsymbol{K}$ indexed by elements in A.

Kernel Matrix. Each entry K_{ij} in the kernel matrix $\boldsymbol{K}$ describes the similarity between elements i and j in $\mathcal{Y}$. For our purpose of selecting orthogonal rules, we will define $\boldsymbol{K}$ as the Gram matrix of the score vectors: $\boldsymbol{K} \stackrel{\text{def}}{=} \boldsymbol{S}^\top\boldsymbol{S}$.

DPP Sampling. To sample a diverse subset using DPP, there are several existing algorithms (Hough et al. 2006; Kulesza, Taskar et al. 2012; Tremblay, Barthelme, and Amblard 2018) and the Python library `DPPy` (Gautier et al. 2019) implements some of them. The computation of the DPP sampling primarily hinges on the overhead of computing the inner product kernel matrix $\boldsymbol{K}$ and its eigendecomposition. In our case, $\boldsymbol{K} \in \mathbb{R}^{R\times R}$ and hence it requires $O(R^3)$ time, where R is the number of all rules. Nonetheless, we set $R = 50$ in our experiments, therefore our DPP rule selection algorithm is extremely fast (typically within 0.1 seconds). Further details about DPP sampling algorithms and their time complexities can be found in Appendix E.

3.3 DPP rule-based rating algorithm

Our rule-based data selection framework proceeds in five steps (Figure 1):

Step 1. Rule generation. We prompt GPT-4 with a task description and source data context to generate R candidate rules. These rules are then manually refined for clarity and relevance.

Step 2. Rule-based rating: We employ Llama3-8B-Instruct (AI@Meta 2024), to rate the batch data $\mathcal{B}$ according to R rules, resulting in the score matrix $\boldsymbol{S} \in \mathbb{R}^{n\times R}$.

Step 3. Rule selection using DPP: From $\boldsymbol{S}$, we aim to select r relatively independent columns using a DPP, forming the submatrix $\tilde{\boldsymbol{S}} \in \mathbb{R}^{n\times r}$. We define the kernel matrix of

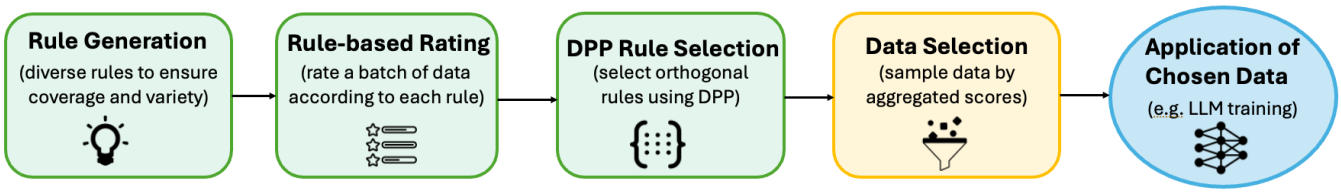

Figure 1: Pipeline for rule-based data rating and selection in five steps (detailed in Section 3.3)

DPP as follows:

$$\boldsymbol{K} \stackrel{\text{def}}{=} \boldsymbol{S}^\top \boldsymbol{S} \in \mathbb{R}^{R \times R}, \tag{3}$$

where each entry $K_{i,j} = \langle S_i, S_j \rangle$ (each S_i is the i-th column of $\boldsymbol{S}$), representing the similarity between rule i and rule j. We then employ the DPP sampling algorithm to select r indices from $\{1, 2, \ldots, R\}$, corresponding to the r chosen rules.

Note that the cost of generating R rules is negligible, requiring just a single GPT-4 query, and the cost of obtaining the rating matrix $\boldsymbol{S}$ can be managed by adjusting the batch size n. The motivation to select a fixed small number of r rules is driven by the computational costs associated with using LLMs for data rating and the need to maintain a consistent dimensionality for explaining data quality. These practical considerations lead us to treat r as a hyperparameter. Discussions on the optimal choices of r are explored in Section 4 and Appendix I.5.

Another important remark is that, even with the same set of rules, they could have different correlations conditioned on a specific task or dataset. Therefore during DPP selection, instead of employing fixed representations such as semantic encodings—which result in static rule representations and selections across all tasks—we use *task-aware* score vectors to adaptively represent the rules. These vectors allow the entire pipeline to be customized for a particular downstream task.

Step 4. Stochastic data selection: We extend the rating process to cover all data samples using the selected r rules, expanding the rating matrix $\bar{\boldsymbol{S}}$ from $n \times r$ to $N \times r$. We then aggregate these fine-grained ratings by averaging across the r columns of $\bar{\boldsymbol{S}}$, resulting in a score vector $\boldsymbol{v} = [v_1, v_2, \ldots, v_N]^\top$ that assigns a quality score to each of the N samples.

Given the N scores and a fixed budget of selecting k samples for training ($k = 20000$ in Section 5), rather than choose the traditional top-k approach, (selecting the highest scored samples), we adopt a stochastic sampling strategy, where we sample k data points according to the distribution:

$$p(\boldsymbol{x}_i) = \frac{e^{v_i/\tau}}{\sum_{j=1}^{N} e^{v_i/\tau}} \tag{4}$$

for each data point $\boldsymbol{x}_i \in \mathcal{D}$, and τ is the "temperature" balancing between top-k and uniform sampling (we use $\tau = 1$ by default). This stochastic data selection mechanism introduces greater diversity into the sampling process and is used in several other works (Wettig et al. 2024; Sachdeva et al. 2024).

Step 5. Apply the selected data on given downstream tasks, such as for LLM domain fine-tuning.

4 Preliminary Experiments: Evaluating Against Ground Truth Ratings

We evaluate our method through two complementary approaches: (A) comparing rule-based ratings **against ground truth human ratings**, where smaller deviations from human scores indicate better performance, and (B) training LLMs (Pythia-1B and Llama3-8B) on the selected data and *evaluating their downstream performance on domain-specific benchmarks*. This section focuses on the first evaluation setup (A), while the full-scale LLM training experiments (B) are presented in Section 5. Because Evaluation A avoids expensive LLM training, it enables a broader exploration of factors such as the effect of rule set size and the influence of model scale (Llama3-8B vs. Llama3-70B). These experiments offer deeper insights into the robustness and effectiveness of our approach.

4.1 Experiments Setup

Datasets and ground-truth ratings: We use four datasets spanning different domains: StanfordNLP-IMDB (Maas et al. 2011) (IMDB), MedQA (Jin et al. 2021) (Medical), GSM8K (Cobbe et al. 2021) (Math), and MBPP (Austin et al. 2021) (Code). For each dataset, we randomly sample 300 examples and ask five human annotators to assign quality scores, with the average scores treated as ground truth (GT). In this section, we present results on IMDB as a representative example; results on the remaining domains are provided in Appendix H.2.

Rule-based rating: We apply our rule-based approach to rate the same data. For each of the $R = 50$ rules, we obtain ratings from two LLMs: Llama3-8B-Instruct and Llama3-70B-Instruct, to evaluate the effect of rater capability. Each rule i produces a score vector $S_i \in \mathbb{R}^n$ over $n = 300$ samples, forming a full rating matrix $\boldsymbol{S} \in \mathbb{R}^{n \times R}$. We then select a subset of r rules, forming a submatrix $\bar{\boldsymbol{S}} \in \mathbb{R}^{n \times r}$. To evaluate how well this rule subset aligns with ground-truth scores $S_{\text{GT}} \in \mathbb{R}^n$, we compute the mean squared error (MSE):

$$\epsilon(\bar{\boldsymbol{S}}) \stackrel{\text{def}}{=} \frac{1}{n} \left\| \frac{1}{r} \sum_{j=1}^{r} \bar{S}_j - S_{GT} \right\|_2^2 \tag{5}$$

where $\bar{S}_j$ is the j-th column of $\bar{\boldsymbol{S}}$. As baselines, we include: (1) the four human-designed rules in QuRating (Wettig et al. 2024), and (2) a rule-free method where Llama3 directly produces an overall quality score ("NoRule").

Our experiments in this section aim to explore the following research questions:

- **(Q1)** Does greater rule diversity lead to more accurate ratings?
- **(Q2)** Does rule-based selection generally outperform rule-free methods?
- **(Q3)** Is the rating quality based on our DPP-selected rules better than that based on human-created rules and NoRule setting?
- **(Q4)** Does DPP select better rules than randomly chosen ones?
- **(Q5)** Can the method possibly generalize well to pre-train datasets?

4.2 Results

Correlation of $\rho(\bar{\boldsymbol{S}})$ and the MSE $\epsilon(\bar{\boldsymbol{S}})$ (answer to Q1). For each rule subset size $r \in \{1, 2, \ldots, 50\}$, we sample $\min\{10000, \binom{50}{r}\}$ sets of indices of size r, which are used to choose rules and form $\bar{\boldsymbol{S}}$. We then calculate its rule correlation $\rho(\bar{\boldsymbol{S}})$ and MSE $\epsilon(\bar{\boldsymbol{S}})$, as well as the Pearson correlation between them. The Pearson correlation is surprisingly strong across both rating model (Llama3 8B and 70B) (Figures 2a and 2b). This confirms that higher rule diversity is positively correlated with the accuracy of rating results. In other words, the highly correlated or redundant rules can potentially lead to non-robust ratings.

Rule-based v.s. Rule-free (answer to Q2): We sample 10^6 random rule subsets of size r and compare their MSEs against that of the NoRule baseline. Figures 2c and 2d show that rule-based methods consistently achieve lower MSE. Interestingly, even randomly chosen domain-specific rules outperform the fixed QuRating rules, illustrating the limitations of static, general-purpose rule sets.

DPP v.s. QuRating v.s. NoRule (answer to Q3). Using DPP, we select r rules and repeat this for 100 trials. We report both the average MSE and the winning rate compared to QuRating and NoRule (Figures 3a, 3d). Results show that once r exceeds a modest threshold, DPP-selected rules almost always outperform both baselines.

DPP rules v.s. Randomly selected rules (answer to Q4). We compare DPP-selected rule subsets to randomly sampled ones of the same size, evaluating both the rule correlation $\rho(\bar{\boldsymbol{S}})$ and MSE $\epsilon(\bar{\boldsymbol{S}})$ for their corresponding score submatrices $\bar{\boldsymbol{S}}$. As shown in Figures 3b and 3c, DPP consistently selects rules with lower correlation and MSE, regardless of the value r. Notably, MSE increases when r is too small or too large, which matches our intuition: when r is too small, there are too few rules to cover all important data quality aspects for rating, and when r is too large, rule redundancy can negatively affect the rating outcomes. Therefore, we use $r = 10$ in subsequent experiments.

Apply to pre-training datasets (answer to Q5). So far, our experiments have focused on domain-specific datasets. We also applied the rule-based method to the CommonCrawl dataset (Common Crawl 2024), a general-purpose web corpus commonly used for LLM pre-training. We find that the plots of rule correlation show similar patterns, and the Pearson correlation between ρ and ϵ is consistently positive. However, the trends and Pearson values are less pronounced than in domain-specific settings. This may be because domain-specific datasets support more concrete and well-defined quality rules, whereas quality assessment in general pretraining data tends to be more ambiguous and less structured. Detailed results and analysis are provided in Appendix H.3.

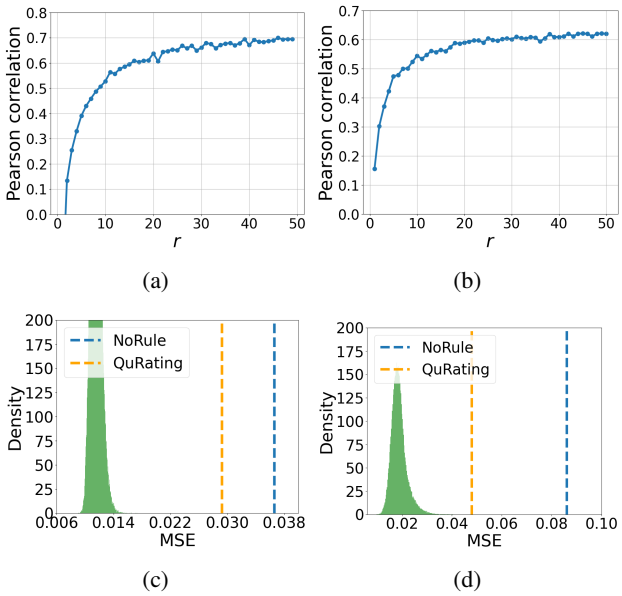

Figure 2: (a) and (b): Pearson correlation of the rule correlation $\rho(\bar{\boldsymbol{S}})$ and the MSE $\epsilon(\bar{\boldsymbol{S}})$, using Llama3 8B and 70B raters respectively. (c) and (d): Distribution of MSE from 10^6 possible rule subsets with size r, using Llama3 8B and 70B raters respectively, where two vertical lines represent the MSE values of QuRating and NoRule.

5 Experiments: Data Selection for LLM Fine-tuning

We now evaluate our full framework in a realistic setting by following the pipeline in Section 3.3 to fine-tune LLMs (Pythia-1B and Llama3-8B) on selected data, then measure their downstream performance. This setup reflects practical applications of LLM data selection in domain-specific fine-tuning.

5.1 Experiments Setup

Evaluation Benchmarks. We assess performance across four domains using standard benchmarks: **IMDB**: Sentiment classification using the IMDB dataset (Maas et al. 2011), **Code**: Code generation evaluated on HumanEval (Chen et al. 2021), MBPP (Austin et al. 2021), and Multiple-py/cpp (Cassano et al. 2022), **Math/Medical**: Relevant subsets of MMLU (Hendrycks et al. 2020). We set the inference temperature to 0, unless the the benchmark protocol itself requires top-k sampling. Full benchmark details are provided in Appendix I.4.

Data Source. We use SlimPajama (Cerebras Systems 2023), a large, deduplicated multi-source corpus for LLM

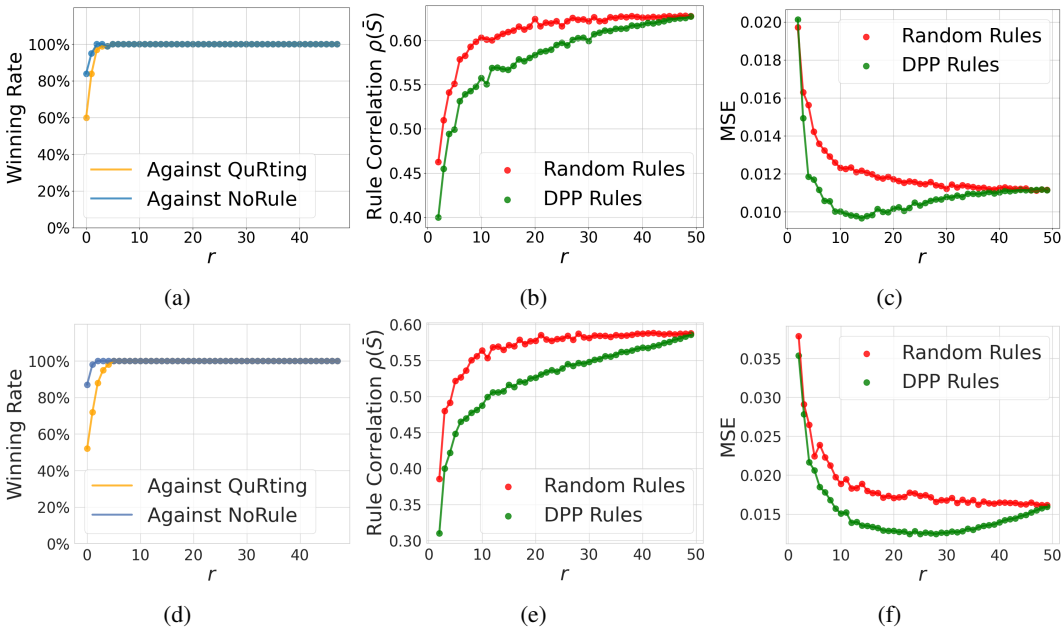

Figure 3: (a) Winning rate of DPP-selected rules compared to QuRating's four rules and the NoRule setting respectively, based on MSE across 100 DPP trials. (b) Comparison of rule correlation ρ between DPP-selected and randomly selected rules, averaged across 100 trials. (c) Comparison of MSE between DPP-selected and randomly selected rules, averaged across 100 trials. Plots (a), (b), and (c) display results using Llama3-8B rater, while (d), (e), and (f) for the Llama3-70B rater.

training. From it, we randomly sample 1M examples (around 1B tokens) as our raw dataset $\mathcal{D}$, from which we select a high-quality subset of $k = 20{,}000$ samples for model training.

Models. We intentionally choose Pythia-1B (Biderman et al. 2023) as the base model for the IMDB and Medical tasks, as it was pre-trained on the Pile dataset (Gao et al. 2020), offering a more controlled comparison than using models that may have been trained on SlimPajama. To validate the generalization ability of our framework across different LLMs, we train Llama3-8B (AI@Meta 2024) with LoRA (Hu et al. 2021) for the Math and Code domains.

Baselines. We compare our method against the following baselines, including both rule-free and rule-based data selection methods:

- **Rule-Free Methods**
  - *Uniform Sampling*: Randomly select 20K samples from the dataset.
  - *No Rule*: Use Llama3-8B to assign a quality score to each sample without any rule prompts, then apply the same sampling procedure as in Section 3.3.
  - *DSIR* (Xie et al. 2024): Importance resampling based on similarity to a high-quality reference dataset (we use test sets from evaluation benchmarks).
  - *LESS* (Xia et al. 2024): Estimate data influence on downstream performance and select samples with the highest impact.
- **Rule-Based Methods**
  - *QuRating* (Wettig et al. 2024): Rate and select data using a fixed set of four manually designed rules.
  - *GPT-Uncorrelated*: Prompt GPT-4 to directly generate 10 "uncorrelated" rules, then use them to rate and select data.
- **All Data**
  - *AllData*: Use the entire 1M SlimPajama samples without applying any selection.

For our automated rule-based selection algorithm, we set $R = 50$ and $r = 10$. The choice of r as a hyperparameter is based on experimental observations from Section 4. As inferences of LLM are a lot less computing-consuming than model training, we set $n = 10^4$ in all our experiments (the size of a randomly selected batch for generating score vectors and selecting orthogonal rules).

5.2 Fine-tuning Results

For each domain, we select 20K relevant training samples using different selection strategies. Results are shown in Table 1. We first note that training with all 1M data gives underperforming results, aligning with findings from prior works such as Zhou et al. (2023). *Uniform Sampling* generally fails to yield significant improvements. While *DSIR* performs well on certain tasks, it relies on access to a high-quality reference dataset or even the test dataset, which may not always be feasible. Among all rule-based methods, *QuRating* underperforms. As previously noted, its limited hand-crafted rules may reflect subjective preferences or omit task-dependent correlations between rules. The *GPT-Uncorrelated* rules face a similar issue where the rule selection process is entirely

Method	IMDB	Medical				Math				Code				
	SA	CM	PM	MG	Med Avg	ES	HS	CMath	Math Avg	HE	mbpp	MPy	MCpp	Code Avg
Backbone	44.5	21.4	34.2	23.0	26.2	41	39.6	34	38.2	46.3	42.9	44	48.4	45.4
Uniform Sampling	43.9	23.0	42.1	22.5	28.9	40.5	39.2	35	38.2	38.7	38.2	38.2	39.7	38.7
No Rule	51.1	23.1	42.6	22.0	29.2	42.3	37.4	37	38.9	45.1	43.9	42.8	52.1	45.9
DSIR	50.2	22.5	32.4	17.0	23.9	41.5	**41.1**	34	38.9	45.1	43.6	**49.1**	52.2	47.5
LESS	46.6	23.6	40.4	24	29.3	41.5	40.4	33	38.3	41.4	43.5	43.9	45.3	43.5
QuRating	47.7	21.3	42.2	22	28.5	41.5	38.1	35	38.2	43.2	43.4	40.5	45.6	43.1
GPT-Uncorrelated	50.9	23.7	42	22.7	29.4	41.4	39	37.3	39.2	41.2	43.5	39.6	48.6	43.2
AllData	45.5	21.9	37.8	24.0	27.9	41.8	37.7	33.0	37.5	45.1	43.8	42.2	52.1	45.8
DPP 10 Rules (ours)	**53.5**	**24.6**	**43.3**	**26.8**	**31.6**	**43.7**	40.6	**38**	**40.8**	**50.5**	**44.2**	46.9	**52.7**	**48.6**

Table 1: Fine-tuning on IMDB, Medical, Math and Code domains, each using 20K selected data samples from SlimPajama. The first row shows the performance of the original backbone model (Pythia-1B for IMDB and Medical and Llama3-8B for Math and Code). Abbreviations for subtasks: SA = Sentimental Analysis accuracy, CM = College Medicine, PM = Professional Medicine, MG = Medical Genetics, ES = Elementary School Math, HS = High School Math, CMath = College Math, HE = HumanEval, MPy = Multiple-py, MCpp = Multiple-cpp. For *Uniform Sampling* and methods involving randomness in rule selections, we conducted 3 independent trials and averaged the results. The standard deviations are reported in Appendix I.3.

independent of the data. In contrast, our method first generates a large rule pool and then selects a subset based on rule orthogonality derived from data-specific score vectors. This task-aware design, grounded in a rigorous metric, leads to improved fine-tuning outcomes across all domains.

5.3 Ablation Study

To demonstrate the effectiveness of DPP in rule selection, we conduct an ablation study and evaluate the performance of the following variations developed within our framework:

- *GPT-selected 10 Rules*: Query GPT to select 10 "uncorrelated" rules from the rule pool.
- *All 50 Rules*: Average score vectors from all 50 rules to rate and select data.
- *Random 10 Rules*: Randomly choose 10 rules and average the score vectors to rate and select data.

Results are provided in Appendix B. Key findings include: *DPP 10 Rules* consistently outperforms both *All 50 Rules* and *Random 10 Rules*. This aligns with our argument of the importance of rule orthogonality, as well as the intuition that the optimal r is not near the boundaries (both validated by experiments in Section 4). *GPT-selected 10 Rules* provides a strong baseline, but since it relies solely on semantic features and not task-specific scores, it is generally outperformed by our DPP-based selection. These results confirm that our DPP-based selection mechanism effectively identifies a compact, high-quality, and diverse rule subset tailored to the target task, resulting in superior data selection and model performance.

6 Conclusion

We presented an automated, rule-based framework for selecting high-quality training data for LLMs, combining LLM-generated rules with a DPP-based selection mechanism to promote rule diversity and reduce redundancy. Our work is the first to introduce a principled, automated metric for rule evaluation, integrated into a fully data-driven pipeline that generalizes effectively across diverse domains and tasks. We first validated our approach on a dataset with ground truth quality scores, demonstrating improved rating accuracy. We then fine-tuned LLMs using data selected by our method and showed consistent improvements over strong baselines across four domains. These results confirm that selecting a compact set of diverse, task-aware rules leads to higher-quality data and ultimately better model performance.

Our framework is broadly applicable to various scenarios, including general LLM pre-training, domain-specific fine-tuning, and RLHF data annotation. Moreover, it supports natural extensions such as rule re-weighting, enabled by the orthogonality of selected rules. Future work includes exploring these extensions and adapting the framework for settings requiring dynamic or task-conditioned rule weighting.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 Model Card.

Albalak, A.; Elazar, Y.; Xie, S. M.; Longpre, S.; Lambert, N.; Wang, X.; Muennighoff, N.; Hou, B.; Pan, L.; Jeong, H.; et al. 2024. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*.

Allamanis, M. 2019. The adverse effects of code duplication in machine learning models of code. In *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, 143–153.

Austin, J.; Odena, A.; Nye, M.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C.; Terry, M.; Le, Q.; et al. 2021. Program Synthesis with Large Language Models. *arXiv preprint arXiv:2108.07732*.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022. Constitutional AI: harmlessness from AI feedback. 2022. *arXiv preprint arXiv:2212.08073*.

Biderman, S.; Schoelkopf, H.; Anthony, Q. G.; Bradley, H.; O'Brien, K.; Hallahan, E.; Khan, M. A.; Purohit, S.; Prashanth, U. S.; Raff, E.; et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2397–2430. PMLR.

Borodin, A.; and Olshanski, G. 2000. Distributions on Partitions, Point Processes and the Hypergeometric Kernel. *Communications in Mathematical Physics*, 211: 335–358.

Borsos, Z.; Mutny, M.; and Krause, A. 2020. Coresets via bilevel optimization for continual learning and streaming. *Advances in neural information processing systems*, 33: 14879–14890.

Brown, T. B. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Cao, Y.; Kang, Y.; and Sun, L. 2023. Instruction mining: High-quality instruction data selection for large language models. *arXiv preprint arXiv:2307.06290*.

Cassano, F.; Gouwar, J.; Nguyen, D.; Nguyen, S.; Phipps-Costin, L.; Pinckney, D.; Yee, M.-H.; Zi, Y.; Anderson, C. J.; Feldman, M. Q.; et al. 2022. Multipl-e: A scalable and extensible approach to benchmarking neural code generation. *arXiv preprint arXiv:2208.08227*.

Cerebras Systems. 2023. SlimPajama: A 627B Token Cleaned and Deduplicated Version of RedPajama. Blog post on Cerebras Systems. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama.

Chen, L.; Li, S.; Yan, J.; Wang, H.; Gunaratna, K.; Yadav, V.; Tang, Z.; Srinivasan, V.; Zhou, T.; Huang, H.; et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; de Oliveira Pinto, H. P.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; Ray, A.; Puri, R.; Krueger, G.; Petrov, M.; Khlaaf, H.; Sastry, G.; Mishkin, P.; Chan, B.; Gray, S.; Ryder, N.; Pavlov, M.; Power, A.; Kaiser, L.; Bavarian, M.; Winter, C.; Tillet, P.; Such, F. P.; Cummings, D.; Plappert, M.; Chantzis, F.; Barnes, E.; Herbert-Voss, A.; Guss, W. H.; Nichol, A.; Paino, A.; Tezak, N.; Tang, J.; Babuschkin, I.; Balaji, S.; Jain, S.; Saunders, W.; Hesse, C.; Carr, A. N.; Leike, J.; Achiam, J.; Misra, V.; Morikawa, E.; Radford, A.; Knight, M.; Brundage, M.; Murati, M.; Mayer, K.; Welinder, P.; McGrew, B.; Amodei, D.; McCandlish, S.; Sutskever, I.; and Zaremba, W. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113.

Civril, A.; and Magdon-Ismail, M. 2007. Finding maximum Volume sub-matrices of a matrix. *RPI Comp Sci Dept TR*, 07–08.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Common Crawl. 2024. Common Crawl Dataset. https://commoncrawl.org.

Du, N.; Huang, Y.; Dai, A. M.; Tong, S.; Lepikhin, D.; Xu, Y.; Krikun, M.; Zhou, Y.; Yu, A. W.; Firat, O.; et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, 5547–5569. PMLR.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dubois, Y.; Li, C. X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P. S.; and Hashimoto, T. B. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.

Gao, L.; Biderman, S.; Black, S.; Golding, L.; Hoppe, T.; Foster, C.; Phang, J.; He, H.; Thite, A.; Nabeshima, N.; et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Gautier, G.; Polito, G.; Bardenet, R.; and Valko, M. 2019. DPPy: DPP Sampling with Python. *Journal of Machine Learning Research - Machine Learning Open Source Software (JMLR-MLOSS)*. Code at http://github.com/guilgautier/DPPy/ Documentation at http://dppy.readthedocs.io/.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Hough, J. B.; Krishnapur, M.; Peres, Y.; and Virág, B. 2006. Determinantal processes and independence.

Hsieh, C.-Y.; Li, C.-L.; Yeh, C.-K.; Nakhost, H.; Fujii, Y.; Ratner, A.; Krishna, R.; Lee, C.-Y.; and Pfister, T. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, S.; Siddarth, D.; Lovitt, L.; Liao, T. I.; Durmus, E.; Tamkin, A.; and Ganguli, D. 2024. Collective Constitutional AI: Aligning a Language Model with Public Input. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, 1395–1417.

Jiang, T.; Yuan, X.; Chen, Y.; Cheng, K.; Wang, L.; Chen, X.; and Ma, J. 2022. FuzzyDedup: Secure fuzzy deduplication for cloud storage. *IEEE Transactions on Dependable and Secure Computing*.

Jin, D.; Pan, E.; Oufattole, N.; Weng, W.-H.; Fang, H.; and Szolovits, P. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14): 6421.

Kendall, M. G. 1938. A new measure of rank correlation. *Biometrika*, 30(1-2): 81–93.

Kool, W.; Van Hoof, H.; and Welling, M. 2019. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, 3499–3508. PMLR.

Kulesza, A.; Taskar, B.; et al. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3): 123–286.

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

Li, M.; Zhang, Y.; Li, Z.; Chen, J.; Chen, L.; Cheng, N.; Wang, J.; Zhou, T.; and Xiao, J. 2023. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

Macchi, O. 1975. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1): 83–122.

Mu, T.; Helyar, A.; Heidecke, J.; Achiam, J.; Vallone, A.; Kivlichan, I.; Lin, M.; Beutel, A.; Schulman, J.; and Weng, L. 2024. Rule based rewards for language model safety. *arXiv preprint arXiv:2411.01111*.

Penedo, G.; Malartic, Q.; Hesslow, D.; Cojocaru, R.; Cappelli, A.; Alobeidli, H.; Pannier, B.; Almazrouei, E.; and Launay, J. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Reimers, N. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*.

Sachdeva, N.; Coleman, B.; Kang, W.-C.; Ni, J.; Hong, L.; Chi, E. H.; Caverlee, J.; McAuley, J.; and Cheng, D. Z. 2024. How to Train Data-Efficient LLMs. *arXiv preprint arXiv:2402.09668*.

Soldaini, L.; Kinney, R.; Bhagia, A.; Schwenk, D.; Atkinson, D.; Authur, R.; Bogin, B.; Chandu, K.; Dumas, J.; Elazar, Y.; et al. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*.

Sun, Z.; Shen, Y.; Zhou, Q.; Zhang, H.; Chen, Z.; Cox, D.; Yang, Y.; and Gan, C. 2024. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36.

Together AI. 2023. Red Pajama. Blog post on Together AI. https://www.together.ai/blog/redpajama.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Tremblay, N.; Barthelme, S.; and Amblard, P.-O. 2018. Optimized algorithms to sample determinantal point processes. *arXiv preprint arXiv:1802.08471*.

Wang, Z.; Dong, Y.; Delalleau, O.; Zeng, J.; Shen, G.; Egert, D.; Zhang, J. J.; Sreedhar, M. N.; and Kuchaiev, O. 2024. HelpSteer2: Open-source dataset for training top-performing reward models. *arXiv preprint arXiv:2406.08673*.

Wenzek, G.; Lachaux, M.-A.; Conneau, A.; Chaudhary, V.; Guzmán, F.; Joulin, A.; and Grave, E. 2019. CCNet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Wettig, A.; Gupta, A.; Malik, S.; and Chen, D. 2024. Qurating: Selecting high-quality data for training language models. *arXiv preprint arXiv:2402.09739*.

Xia, M.; Malladi, S.; Gururangan, S.; Arora, S.; and Chen, D. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Xia, X.; Liu, J.; Yu, J.; Shen, X.; Han, B.; and Liu, T. 2022. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations*.

Xie, S. M.; Santurkar, S.; Ma, T.; and Liang, P. S. 2024. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36.

Yang, Y.; Wang, H.; Wen, M.; and Zhang, W. 2024. P3: A Policy-Driven, Pace-Adaptive, and Diversity-Promoted Framework for Optimizing LLM Training. *arXiv preprint arXiv:2408.05541*.

Yuan, W.; Pang, R. Y.; Cho, K.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2023. Opt: Open pre-trained transformer language models, 2022. *URL https://arxiv. org/abs/2205.01068*, 3: 19–0.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; et al. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36: 55006–55021.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.