Entropy Law: The Story Behind Data Compression and LLM Performance

Mingjia Yin[†], **Chuhan Wu**[*†], **Yufei Wang**[†], **Hao Wang**[*†],
Wei Guo, Yasheng Wang, Yong Liu, Ruiming Tang[*], **Defu Lian, Enhong Chen**[*]

State Key Laboratory of Cognitive Intelligence & University of Science and Technology of China
Noah's Ark Lab, Huawei
{mingjia-yin, wanghao3, cheneh}@ustc.edu.cn
{wuchuhan1, tangruiming}@huawei.com

Abstract

Data is the cornerstone of large language models (LLMs), but not all data is useful for model learning. Carefully selected data can better elicit the capabilities of LLMs with much less computational overhead. Most methods concentrate on evaluating the quality of individual samples in data selection, while the combinatorial effects among samples are neglected. Even if each sample is of perfect quality, their combinations may be suboptimal in teaching LLMs due to their intrinsic homogeneity or contradiction. In this paper, we aim to uncover the underlying relationships between LLM performance and data selection. Inspired by the information compression nature of LLMs, we uncover an "entropy law" that connects LLM performance with data compression ratio and first-epoch training loss, which reflect the information redundancy of a dataset and the mastery of inherent knowledge encoded in this dataset, respectively. Through both theoretical deduction and empirical evaluation, we find that model performance is negatively correlated to the compression ratio of training data, which usually yields a lower training loss. Based on the findings of the entropy law, we propose a quite efficient and universal data selection method named **ZIP** for training LLMs, which aim to prioritize data subsets exhibiting a low compression ratio. Based on a multi-stage algorithm that selects diverse data in a greedy manner, we can obtain a good data subset with satisfactory diversity. Extensive experiments have been conducted to validate the entropy law and the superiority of ZIP across different LLM backbones and alignment stages. We also present an interesting application of entropy law that can detect potential performance risks at the beginning of model training.

1 Introduction

In recent years, Large Language Models (LLMs) have gained significant attention from both academia and industry, applied in various domains, such as chatbots (Ouyang et al., 2022; Achiam et al., 2023), chemistry tools (M. Bran et al., 2024), and programming assistants (GitHub, 2020). The great success of LLMs depends on their general intelligence obtained from a vast amount of data collected from various sources (Albalak et al., 2024; Wang et al., 2023c). Through pretraining on trillions of tokens to master diverse knowledge and tuning on smaller instruction data to align models with human preference, LLMs can effectively utilize their knowledge to follow user instructions, do commonsense reasoning, and solve real-world problems (Zhao et al., 2023).

However, not all data are useful for teaching LLMs, especially when computational resources are limited (Albalak et al., 2024). For example, we can better elicit the capability of LLMs by fine-tuning

---

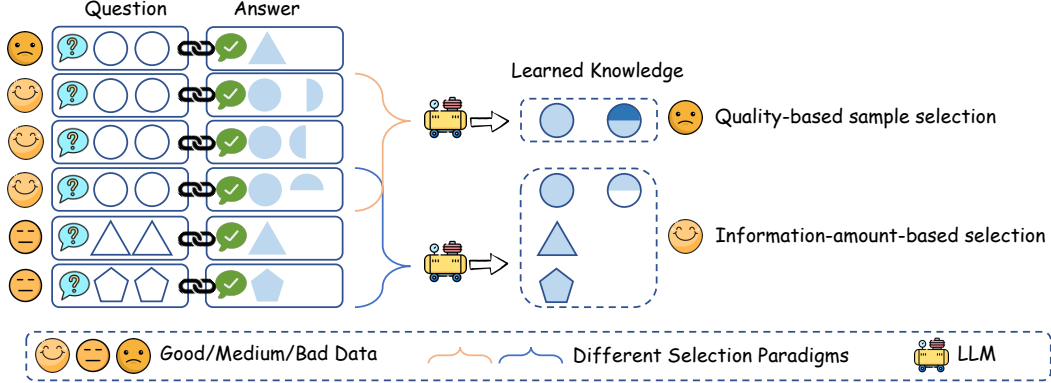*Corresponding authors. † Equal contribution.

Working in Progress.

Figure 1: An illustrative example describing different data selection paradigms. Quality-based data selection relies on sample-level data quality measurements while overlooking combinatorial effects among samples. Information-amount-based selection aims to select samples maximizing the overall information amount.

them on carefully curated samples rather than a large but noisy data collection (Ouyang et al., 2022; Chowdhery et al., 2023; Meta, 2020; Zhou et al., 2023). However, selecting the proper data for LLM training is quite complicated and abstruse, since the space of data preprocessing and combination is almost unlimited. Due to the huge computational overhead of LLM training, manual or empirical data selection based on trial-and-error feedback is rather cumbersome and even impractical. Therefore, automatic data selection methods are necessary for LLM development under limited computational budgets.

Intuitively, high-quality samples are expected to have better efficiency in teaching LLMs. For example, the successful practice of LIMA (Zhou et al., 2023) shows the powerful effect of data quality on LLM performance that can surpass the amount of data. Therefore, existing methods usually focus on quality-oriented data selection, based either on heuristic rules (Raffel et al., 2020; Rae et al., 2021; Xie et al., 2023; Chowdhery et al., 2023; Li et al., 2023) or evaluation models (Wettig et al., 2024; Chen et al., 2023; Lu et al., 2023; Liu et al., 2023; Cui et al., 2023). Heuristic methods typically involve hand-crafted rules (e.g., sentence number (Raffel et al., 2020), word count (Rae et al., 2021), length (Shen, 2024)) to evaluate data across multiple dimensions. Model-based approaches, on the contrary, rely on well-established LLMs such as GPT-4 (Achiam et al., 2023) to provide quality assessments of training samples in different views, such as direct scoring (Chen et al., 2023), task tagging (Lu et al., 2023), and pairwise scoring (Liu et al., 2023). However, most of these approaches evaluate different data samples independently, which neglects the intricate combinatorial effects among samples. As illustrated in Figure 1, even if each sample is in perfect quality, their combinations may still be suboptimal due to their mutual information redundancy or inconsistency. Although the quality-based subset is composed of all three good samples, the knowledge they encode is actually redundant and conflicting. In contrast, another data subset composed of several relatively lower-quality but diverse samples may convey more information than the above subset in the teaching of LLMs. Therefore, quality-based data selection does not fully align with the goal of maximizing the knowledge mastery of LLMs.

In many recent studies, researchers have shown that the basic mechanism of auto-regressive language modeling in LLMs is information compression (Delétang et al., 2023; Huang et al., 2024). Thus, the knowledge condensed by LLMs actually depends on the effective information encoded by training data. This intuition opens another direction of data selection, i.e., based on the effective information amount of data. In this paper, we uncover the underlying relations between LLM performance and data homogeneity, which can be measured by various canonical lossless compression algorithms (e.g., DEFLATE in ZIP). Through both theoretical analysis and empirical experiments, we formulate the "*entropy law*", which shows that the compression ratio of training data is a decisive factor affecting model performance, if the overall quality and consistency of selected samples remain unchanged. Motivated by the entropy law, we propose an effective and efficient data selection algorithm called **ZIP** to select heterogeneous data with low compression ratio, which aims to maximize the effective information amount of information for LLM learning. Specifically, we devise a multi-stage greedy

strategy to find an approximate solution that guarantees a low compression ratio without exhausting all possible combinations, and it iterates continuously until we obtain a predetermined number of samples. In each iteration, ZIP performs preliminary filtering to choose a smaller pool of candidates, and then selects a few samples from the reduced pool that minimizes the compression ratio of the selected dataset through a cascaded manner. By learning LLMs on a collection of diverse samples that encode heterogeneous and complementary information, the capabilities of LLMs can be better elicited. Extensive experiments on different LLM backbones at different stages of LLM alignment demonstrate the superiority of ZIP over various quality-based baselines. We also present an interesting application of the entropy law that can detect potential performance risks at the beginning of model training, which can effectively reduce the computational overhead in LLM development.

2 Related Works

2.1 Large Modeling and Information Compression

The relationship between language modeling and data compression has long intrigued researchers (Shannon, 1948, 1951). Recent empirical studies have confirmed that language models can act as versatile data compressors (Delétang et al., 2023), and the intelligence of LLMs can be quantified by their capacity for text compression (Huang et al., 2024). Let a text corpus be generated from an underlying distribution ρ. A lossless compression algorithm $\mathcal{C}$ is then expected to encode a text sequence $x_{1:n}$ into a bitstream $\mathcal{C}(x_{1:n})$ of minimal length, ensuring that $x_{1:n}$ can be perfectly recovered from $\mathcal{C}(x_{1:n})$. The expected number of bits of an optimal $\mathcal{C}$ is equal to $\mathbb{E}_{x \sim \rho}[-\log_2 \rho(x)] = \mathbb{E}_{x \sim \rho}[-\sum_{i=1}^{n} \log_2 \rho(x_i|x_{1:i-1})]$ (Shannon, 1948). The underlying distribution ρ is usually unknown in reality, but it can be estimated by a language model ρ_{model}. Then the expected number of bits of an optimal $\mathcal{C}$ can be updated:

$$\mathbb{E}_{x \sim \rho}[-\sum_{i=1}^{n} \log_2 \rho_{\text{model}}(x_i|x_{1:i-1})]. \tag{1}$$

Equation 1 is the cross-entropy loss employed in training LLMs, thereby establishing a coherent relationship between LLMs and information compression. This foundational insight paves the way for this work.

2.2 Alignment of Large Language Models

Large Language Models (LLMs) have recently gained significant attention from academia and industry. LLM alignment, which includes supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF), has emerged as a crucial technique for adapting LLMs to end tasks using natural language instructions (Zhao et al., 2023; Wang et al., 2023c). Alignment is performed using instruction datasets consisting of multiple (Instruction, Output) pairs, which require LLMs to follow the instructions and generate corresponding outputs. Early explorations have focused on constructing or expanding instruction datasets through methods such as crowd-sourcing (Wang et al., 2022; Köpf et al., 2024), self-instruction (Taori et al., 2023; Peng et al., 2023; Wang et al., 2023b), or the combination of existing datasets (Wang et al., 2023a; Ivison et al., 2023). Fine-tuned LLMs on these datasets have demonstrated promising capabilities to adhere to instructions across various contexts and align with human expectations.

2.3 Data selection for LLM alignment

A growing body of research has emphasized the importance of selecting appropriate data for LLM alignment, which can prevent potential quality issues and optimize computational resource allocation. As a prominent example, Lima (Zhou et al., 2023) has demonstrated superior performance by carefully crafting only 1,000 high-quality samples for SFT, highlighting the crucial importance of data quality. The current literature on selecting alignment data has focused on selecting samples according to individual sample quality, which can be categorized into heuristic methods (Shen, 2024) and model-based methods (Chen et al., 2023; Lu et al., 2023; Liu et al., 2023; Li et al., 2023, 2024; Du et al., 2023). Heuristic methods typically employ specific criteria, such as response length (Shen, 2024), to guide data selection. On the other hand, model-based methods adopt various strategies to leverage the capabilities of established language models for evaluating sample quality. For example,

IFD (Li et al., 2023) measures the change in response loss when instructions are removed, and selects those with the most significant changes. Building upon IFD, SuperFiltering (Li et al., 2024) introduces a lightweight proxy model for a more efficient calculation of the IFD score. In addition, other model-based methods employ proprietary LLMs to assess data quality. In a pioneering work, AlpaGasus (Chen et al., 2023) uses ChatGPT directly to assign data quality scores to samples, while #InsTag (Lu et al., 2023) proposes assigning tags to each sample using ChatGPT and evaluates sample quality based on the number of tags. DEITA (Liu et al., 2023) uses ChatGPT-generated data to train two Llama-based scorers, assigning complexity and quality scores to each sample, and ultimately selecting samples with the highest hybrid scores. However, existing methods are mainly designed to pick data based on sample-wise quality measurements, which are usually weak in reflecting the overall dataset quality. In this paper, we focus on the relation between performance and dataset quality, which can be efficiently measured by data compression metrics.

3 Entropy Law: Connecting Model Performance with Data Compression

In this section, we provide some theoretical analysis of the relations between data compression and LLM performance. Intuitively, the correctness and diversity of the training data would affect the performance of the final model. Meanwhile, the performance of LLM may be suboptimal if the data have severe intrinsic conflicts or the model has poor mastery of the information encoded by the data, which can be indicated by the training loss. Based on these assumptions, we denote the performance of an LLM as Z, which is expected to be influenced by the following factors:

- Data compression ratio R: This metric can be derived by dividing the pre-compression data size by the post-compression size, which can be computed by various off-the-shelf compression algorithms. Intuitively, a dataset with a lower compression ratio indicates a higher information density.

- Training loss L: Indicates whether the data are hard for the model to memorize. Given the same base model, a high training loss is usually due to noisy or inconsistent information in the dataset. In practice, the average loss of a small number of training steps in the first training epoch is sufficient to produce an indicative L value so that the model does not overfit the data.

- Data consistency C: The consistency of data is reflected by the entropy of the probability of the next token given the previous contexts. Higher data consistency usually yields a lower training loss. The performance of LLMs is usually suboptimal if the dataset has poor consistency.[2]

- Average data quality Q: This reflects the average sample-level quality of the data, which can be measured through various objective and subjective aspects.

Given a certain amount of training data, the model performance can be estimated by the above factors:

$$Z \propto f(R, L, C, Q), \tag{2}$$

where f is a hidden function. Given a specific base model, the scale of L usually depends on R and C, which can be formulated as:

$$L \propto g(R, C). \tag{3}$$

L is expected to be monotonous on R and C, since a dataset with higher homogeneity or better data consistency is easier for a model to learn. Thus, we can rewrite the above formula as follows:

$$C \propto g'(R, L), \tag{4}$$

where g' is an inverse function. By combining the three above equations, we have:

$$Z \propto f(R, L, g'(R, L), Q) \propto h(R, L, Q), \tag{5}$$

where h is another hidden function. If a data selection method does not substantially change the average data quality Q, we can approximately regard the variable Q as a constant. Therefore, the final performance can be roughly formulated as follows:

$$Z \propto h'(R, L), \tag{6}$$

---

[2]Assume we have two question-answer pairs (q_1, a_1) and (q_2, a_2). When an LLM updates on each QA pair, it learns the mutual information (MI) between the question and answer, i.e., $I(q_1; a_1)$ and $I(q_2; a_2)$. When it updates on both, it learns the joint MI $I(q_1 q_2; a_1 a_2)$. If q_1 and q_2 are not independent, we have $I(q_1 q_2; a_1 a_2) < I(q_1; a_1) + I(q_2; a_2)$, whose detailed derivation can be found in Appendix A. This implies that the total knowledge learned by LLMs is narrowed if the answers to similar questions are highly inconsistent.

which means that the model performance is correlated with the data compression ratio and training loss. We name this relationship as "**Entropy Law**".

We can raise two deductions based on the entropy law:

- If we further regard the data consistency as a constant, the training loss is directly influenced by the compression ratio (Eq. 3). Thus, the model performance is controlled by the compression ratio: Z is usually worse if the data compression ratio R is higher, which will be validated by our experiments.

- Given the same compression ratio R, a higher training loss means a lower data consistency. Thus, the effective knowledge learned by the model may be more limited. This can be used to predict the performance of LLM on different data with similar compression ratios and sample qualities. We will show later the application of this deduction in our practice.

4 ZIP: Lightweight Data Selection for LLM Alignment

Guided by the findings of the entropy law, we propose an effective and efficient method named **ZIP** to select data samples based on data compression ratios, which aims to maximize the amount of effective information given a limited training data budget. Although there exists a subset with the lowest compression ratio, it is impractical to find it due to the huge combination space of data samples. Thus, we propose an iterative multi-stage greedy algorithm to efficiently obtain an approximate solution with a relatively low compression ratio. In each iteration, we first use a global selection stage to choose a pool of candidate samples that have low compression ratios, which aims to find samples with high information density. We then employ a coarse-grained local selection stage incorporating a smaller set of samples with the lowest redundancy with already selected samples. Finally, we use a fine-grained local selection stage that minimizes the similarity between samples to add. The above process is conducted until we obtain a sufficient size of data. The workflow of our method is summarized in Algorithm 1, whose details are introduced as follows.

4.1 Global Selection

In general, we maintain an information redundancy state $\pi_{\mathcal{D}}$ that evaluates the "information gain" of each sample. Intuitively, data with high intra-sample information redundancy are unlikely to have good global diversity. For example, a sample with repeated patterns or echoed conversation turns usually has low education value in LLM training. Thus, we initialize this state by calculating the sample-level compression ratio for the entire dataset $\mathcal{D}$. In each iteration, We select K_1 samples with the lowest scores in $\pi_{\mathcal{D}}$ to form an initial candidate pool $\mathcal{D}_{K_1}$, which provides a good set for subsequent local selection.

4.2 Local Coarse-grained Selection

Since the global selection does not well consider the mutual relations among samples, we further conduct local selection to pick diverse samples. To ensure good computational efficiency, we introduce a coarse-grained selection phase to narrow the candidate pool into a smaller one with K_2 samples. We first compute the compression ratio of a merged set that adds each sample in $\mathcal{D}_{K_1}$ to the selected set $\mathcal{D}'$. We use this score to update the information redundancy state $\pi_{\mathcal{D}}$ to better indicate the current information gain of these samples. Based on the scores of the samples in $\mathcal{D}_{K_1}$, we select K_2 samples with the lowest scores. These samples form a small subset for final fine-grained selection, where each sample has good diversity with the selected dataset $\mathcal{D}'$.

4.3 Local Fine-grained Selection

Although the above stage ensures that the candidate pool has distinct information from the selected set, the information redundancy among the samples within this pool is not measured. Thus, we aim to pick further samples from this subset that are diverse from each other. Concretely, we initialize a local selected set $\mathcal{D}_{K_3}$, and compute the compression ratio of the union of $\mathcal{D}_{K_3}$ and each sample in $\mathcal{D}_{K_2}$. We add the sample with the lowest compression ratio into $\mathcal{D}_{K_3}$, and remove it from $\mathcal{D}_{K_2}$. By repeating this process, we obtain a small subset that contains samples not only different from the selected set $\mathcal{D}'$ but also distinct from each other. We conduct the three stages above until the size of

---

Algorithm 1 Pseudo code of ZIP

Input: The original dataset $\mathcal{D}$ of size N. A data compressor $\mathcal{C}$. The number of selected samples m. The number of compression ratios to be updated in the global selection K_1. The data pool size of local selection K_2. The data budget of local selection K_3. Compression ratio calculation function $g(\mathcal{C}(D)) = \frac{\text{Bits}(D)}{\text{Bits}(\mathcal{C}(D))}$, which measures compression ratio by the number of bits.

Output: A data subset $\mathcal{D}'$ of size m with a relatively low compression ratio.

1: Init $\mathcal{D}'$ as an empty set Φ
2: Init $\pi_\mathcal{D} = \{g(d_0), g(d_1), \ldots, g(d_N)\}$ by calculating the compression ratio of each sample in $\mathcal{D}$
3: **while** $|\mathcal{D}'| < m$ **do**
4:     **// Stage 1: Global Selection**
5:     $\mathcal{D}_{K_1}$ = Bottom-K($\mathcal{D} \setminus \mathcal{D}'$, $\pi_{\mathcal{D} \setminus \mathcal{D}'}$, K_1) // Select K_1 samples with the lowest scores
6:     **// Stage 2: Local Coarse-grained Selection**
7:     $\pi_{K_1} = g(\mathcal{D}' \cup \{d\})$, where $d \in \mathcal{D}_{K_1}$ //Compression ratios of the union of $\mathcal{D}'$ and d
8:     Update corresponding values in $\pi_\mathcal{D}$ with π_{K_1}
9:     $\mathcal{D}_{K_2}$ = Bottom-K($\mathcal{D}_{K_1}$, π_{K_1}, K_2) // Select K_2 samples with the lowest compression ratios
10:    **// Stage 3: Local Fine-grained Selection**
11:    Init $\mathcal{D}_{K_3}$ as an empty set Φ
12:    **while** $|\mathcal{D}_{K_3}| < K_3$ **do**
13:       $\pi_{K_2} = \{g(\mathcal{D}_{K_3} \cup \{d\}) | d \in \mathcal{D}_{K_2}\}$
14:       d = $\arg\min_d \pi_{K_2}$
15:       $\mathcal{D}_{K_3} = \mathcal{D}_{K_3} \cup \{d\}$
16:       $\mathcal{D}_{K_2} = \mathcal{D}_{K_2} \setminus \{d\}$
17:    **end while**
18:    **// Update the selected dataset**
19:    $\mathcal{D}' = \mathcal{D}' \cup \mathcal{D}_{K_3}$
20: **end while**

$\mathcal{D}'$ reaches a predefined data budget m. In our method, the entire selection process is quite easy to implement since it is model-free, and can be accelerated using multiple threads. It can select data efficiently and effectively from a large candidate pool for high-quality LLM training.

5 Experiments

ZIP is content-agnostic and model-free, making it suitable for various stages of LLM alignment. We systematically evaluate the effectiveness of ZIP through experiments conducted in the SFT and RLHF stages, as described in Sections 5.1 and 5.2, respectively. Subsequently, Section 5.3 presents an in-depth analysis to empirically support the proposed entropy law, including a practical application guided by this law.

5.1 Data Selection for SFT

5.1.1 Setup

Data Pool & Data Selection We follow DEITA (Liu et al., 2023) to establish a large-scale data pool comprising 300K high-quality samples obtained from WizardLM (Xu et al., 2023), ShareGPT (Chiang et al., 2023), and UltraChat (Ding et al., 2023). Subsequently, various data selection techniques are employed to extract a subset of this pool for LLM instruction tuning. Notably, previous studies controlled the data budget by limiting the number of instances, whereas we managed the total token count to ensure a fair allocation of the compute budget among all methods. To achieve this, we initially select 10,000 samples using ZIP and calculate the corresponding token count. Then, we apply other methods to continue data selection until the required token count is reached.

Training & Evaluation We fine-tune Mistral-7B (Jiang et al., 2023) and LLama-3-8B (Meta, 2020) on the selected dataset. Other training details can be found in Appendix B. As for evaluation, we adopt MT-bench(Zheng et al., 2023) as our benchmark. Specifically, MT-bench is a challenging multi-turn question set with LLM judgements to evaluate model responses, which exhibits a high-level human preferences alignment.

Table 1: Model performance comparison between different data selection baselines based on Mistral-7B and Llama-3-8B on the SFT stage. We also provide the computational cost, average token length of selected data, and the average data quality produced by Alpagasus Chen et al. (2023). The best results are bolded, and the second-best numbers are underlined. † means CPU-only methods.

Model	MT-bench ↑	Cost ↓	Avg.length	Quality
Mistral-7B-based models with SFT				
Random †	6.85	**10s**	976	4.08
Cluster	<u>6.91</u>	15h	970	4.05
Perplexity	6.89	8h	981	4.09
SuperFiltering	6.12	14h	1579	4.10
DEITA	6.82	21h	2048	4.03
ZIP †	**7.08**	<u>4.5h</u>	543	4.00
Llama-3-8B-based models with SFT				
Random †	7.16	**10s**	892	4.08
Cluster	<u>7.18</u>	16h	886	3.95
Perplexity	7.09	9h	895	3.96
SuperFiltering	6.59	14h	1481	3.99
DEITA	7.11	21h	2048	4.09
ZIP †	**7.28**	<u>4.5h</u>	470	4.00

Baselines We select the baseline from two aspects. The first group includes heuristic methods: (1) *Random*, which randomly selects instances from the data pool to verify the fundamental effectiveness of other methods; (2) *Cluster*, which adopts K-means clustering based on the sample representations and select cluster centroids; (3) *Perplexity*, which selects the samples with highest training loss. The second group of baselines includes model-based methods: (1) *DEITA* (Liu et al., 2023), which employs ChatGPT-generated data to train a Llama-based data complexity evaluator and a quality evaluator, and selects samples with the highest hybrid scores; (2) *SuperFiltering* (Li et al., 2024), which assesses each sample by calculating the change in response loss upon instruction removal and introduce a lightweight proxy model to calculate the score more efficiently.

5.1.2 Results

Main comparison We compare ZIP with various data selection methods based on Mistral-7B and Llama-3-8B, and the results are presented in Table 1. ZIP outperforms other data selection approaches on all backbones, which can be attributed to ZIP's ability to model the complex combinatorial effects among samples. Furthermore, our observations indicate that model-based data selection methods often fail to produce satisfactory outcomes when a fixed token number is given. This is because the sample-level evaluations are not updated correspondingly after selecting some samples, leading to biased evaluations for the remaining samples. Additionally, some of these methods adopt strategies to enhance data diversity, such as *DEITA*, which controls the representation distances of selected samples. However, these strategies only provide a rough assessment of the combinatorial effects within the representation space, since semantic distances do not necessarily reflect information redundancy.

Selection bias in sample length across different strategies We also provide the average length of tokenized samples in Table 1. The average token length of *Random* provides an estimation for the entire data pool, which is used to analyze other methods. From the tables, we can observe that *Cluster* and *Perplexity* exhibit similar selection preferences as *Random*. Additionally, *Deita* and *SuperFiltering* predominantly select lengthy data samples. This bias may stem from the LLMs' inclination toward generating longer responses (Saito et al., 2023). However, given the limited budget of selected tokens, choosing excessively lengthy data will reduce the information density and degrade the capabilities of models trained on such data. In contrast, ZIP tends to select shorter samples. Furthermore, we plot the token length distribution of these methods, as depicted in Figure 2 and Figure 6. Consistent with the previous results, we can observe similar distributions for *Random*, *Cluster*, and *Perplexity*. The token length distributions of *DEITA* and *SuperFiltering* are severely skewed,
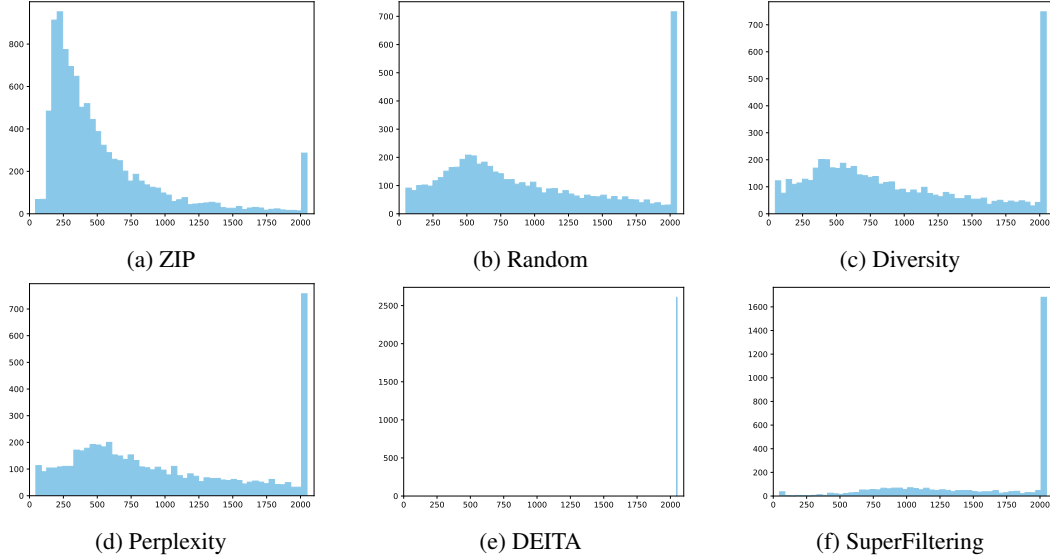
Figure 2: The distribution of average token number across datasets selected by different algorithms for Mistral-7B.

deviating greatly from the original data distribution. In contrast to these model-based approaches, ZIP exhibits no bias toward selecting lengthy samples.

Cost comparison of different strategies We provide a detailed cost analysis of each method in Table 1. Except for the *Random* method, ZIP required the least time to complete the data selection process, demonstrating greater efficiency than other methods. Notably, ZIP's computations are entirely executed on CPUs, resulting in significant cost savings. Furthermore, ZIP is independent of proprietary LLMs used by *DEITA* or the proxy model employed by *Cluster*, *Perplexity*, and *SuperFiltering*. This model-free characteristic endows ZIP with notable efficiency and versatility.

Selected data quality of different strategies We have followed Alpagasus (Chen et al., 2023) to evaluate the quality of each data sample in the selected datasets by prompting ChatGPT, with the quality scores ranging from 0 to 5. The quality scores of multi-turn samples are the average scores of each turn. The results have been presented in Table 1. Surprisingly, the quality scores of selected datasets are highly similar, even with significant differences in selection mechanisms. This may suggest that the average quality distribution remains relatively uniform in the original data pool. Notably, even the SOTA model-based methods like *DEITA* (Liu et al., 2023) and *SuperFiltering* (Li et al., 2024) select data with similar quality scores, potentially contradicting their original conclusions. We posit that this discrepancy stems from the setting of the data budget, which is controlled by the number of samples in prior studies. Considering the selection bias discussed above, these methods tend to select lengthy samples, resulting in a significantly higher token count compared with baselines. For instance, under this setting, data selected by *DEITA* will possess 2.7 times the number of tokens compared to *ZIP*. However, we argue it is fairer to control the data budget by the token count since it guarantees a similar compute budget among all methods[3].

## 5.2 Data Selection for RLHF

5.2.1 Setup

Data Pool & Data Selection The data pool used for preference alignment is a cleaned version of UltraFeedback (Cui et al., 2023; Bartolome et al., 2023), which consists of around 60k samples in the form of a "chosen-rejected" pair. Similarly to the SFT stage, we ensure each data selection method selects data with an approximately equal token count. Since a "chosen-rejected" data pair

---

[3]In practical implementation, the training steps of all methods are almost equal by employing the packing technique detailed in Axolotl.

Table 2: Model performance comparison between different data selection baselines based on Llama-3-8B on the RLHF stage. We also provide the computational cost of each method.

Model	MT-bench ↑	Cost	Avg.length
Base	7.18	NA	NA
Random †	7.33	5s	464
Score	7.30	NA	489
ZIP †	**7.42**	1.1h	357
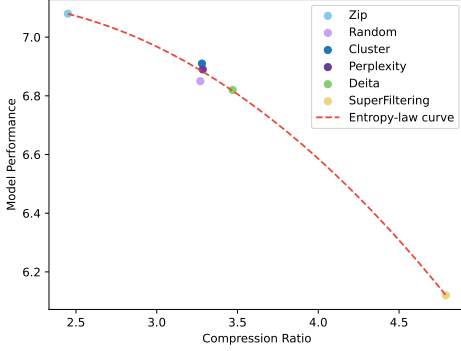
(a) Entropy law w.r.t. compression ratio
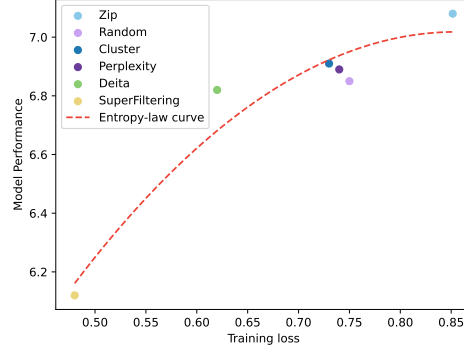
(b) Entropy law w.r.t. training loss

Figure 3: Entropy law demonstration of Mistral-7B. The Entropy law curve is fitted with the results of different methods.

encompasses two data points, we select 5,000 data pairs with ZIP and then apply other methods to select data with the corresponding token budget.

Training & Evaluation Building upon the model previously fine-tuned with SFT, we further refine it using RLHF. In particular, we employ Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024) for preference alignment, a novel method that shows promising potential in aligning preferences. Additional training details can be found in Appendix B. For evaluation, we continue to utilize MT-bench (Zheng et al., 2023) as our benchmark to assess the capabilities of LLMs fine-tuned with data selected using diverse data selection strategies.

Baselines We compare ZIP with the following baselines: (1) *Random*, which randomly samples some "chosen-rejected" pairs from the data pool. (2) *Score*, which selects the "chosen-rejected" pairs with the highest "chosen-scores". These scores are obtained through LLM evaluation of the response quality (Cui et al., 2023; Bartolome et al., 2023).

5.2.2 Main results

Table 2 presents the results of different data selection strategies on the preference alignment stage of LLMs. Similar to the SFT stage, models aligned with data selected by ZIP can yield the best downstream performance, demonstrating the necessity for modeling the combinatorial effects. Besides, we find *Score* and *Random* are on par with each other, even though the selection process of *Score* is far more expensive than *Random*. This is unsurprising, as *Score* does not consider the combinatorial effects, which may limit the knowledge amount of the selected dataset.

5.3 Empirical Validation of Entropy Law

In this section, we aim to demonstrate the proposed entropy law. Specifically, we have plotted the model performance of Mistral-7B and Llama-3-8B concerning data compression ratio and training loss in Figure 3 and 4, respectively. Besides, we plot entropy-law curves by fitting the results. From the two figures, we can draw the following analysis:

Relationship between model performance, data compression ratio, and training loss In Figure 3a and Figure 4a, LLMs trained on data with a lower compression ratio typically exhibit enhanced

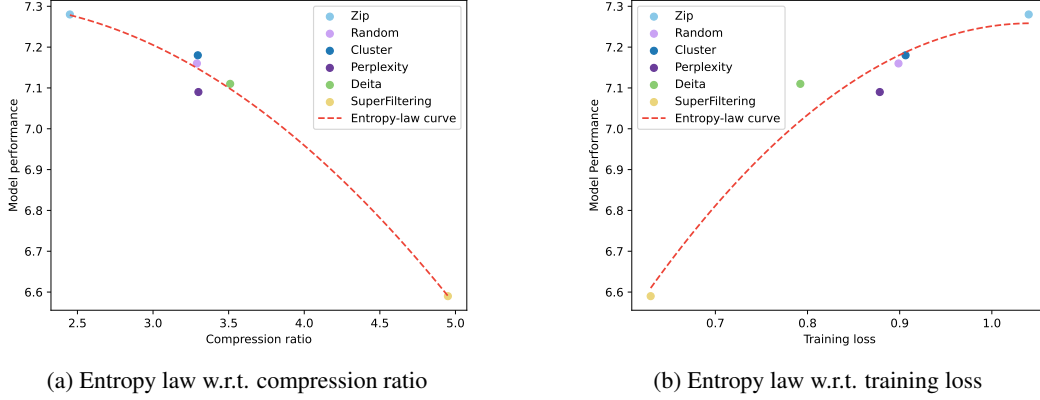(a) Entropy law w.r.t. compression ratio   (b) Entropy law w.r.t. training loss

Figure 4: Entropy law curve of Llama-3-8B. The Entropy law curve is fitted with the results of different methods.
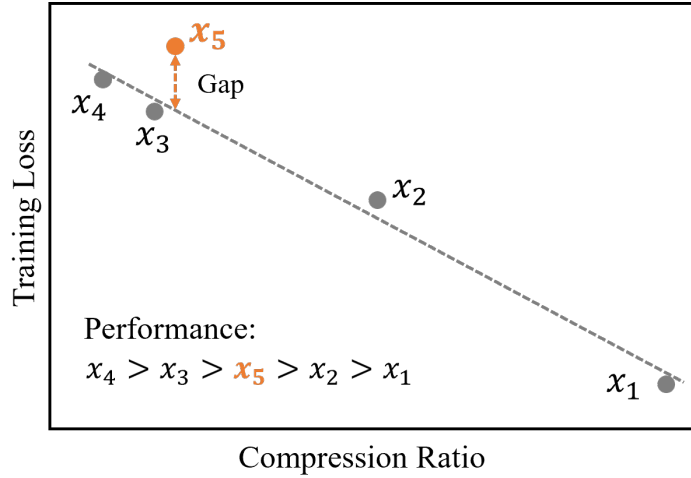

Figure 5: Practical application of Entropy law in incremental training data update, where x_1, x_2, x_3, x_4, x_5 are five data versions.

performance. Since the learning process of LLMs is highly relevant to information compression, we can regard LLMs as data compressors. Then the data with a lower compression ratio means a higher knowledge amount, which is more beneficial to the compressors. Besides, a higher compression ratio usually indicates a higher training loss, as illustrated in Figures 3b and 4b. This is because data resistant to compression carries more knowledge, posing a greater challenge for LLMs to absorb the encapsulated knowledge.

Model performance interpretation with entropy law Considering the three methods with comparable compression ratios and training loss, namely *Random*, *Cluster*, and *Perplexity*, the corresponding model performances are close. This phenomenon may seem counter-intuitive, given the distinct criteria used for data selection. However, it aligns with the predictions of our proposed entropy law: when the average data quality, training loss, and data compression ratio are similar, the model performance is expected to be comparable as well. Thus, the entropy law has the potential to serve as a criterion for predicting the model's performance on data, thereby guiding the training of LLMs.

Practical application of entropy law Incremental version update of training data is a common setting in practical LLM development. Usually, the training data amount remains relatively stable, with only a minor portion undergoing modification. We have conducted incremental training data update experiments in real scenarios, with results depicted in Figure 5. Due to confidentiality, only the relative order of the results is provided. Guided by entropy law, assuming the data quality Q does not significantly decay after each incremental update, we can expect a model performance

10

improvement with a decreased data compression ratio. This expectation is supported by the results from data versions x_1 to x_4 in Figure 5 since their compression ratios are decreased after each incremental update. However, the data version x_5 exhibits an abnormal increase in the loss and data compression ratio, which serves as an early indicator of potential model performance degradation due to a decline in training data consistency. This prediction is further confirmed by subsequent post-training model performance evaluations, as illustrated in Figure 5. Thus, the entropy Law can be utilized as a guideline for LLM training to identify potential risks of experimental failure without training the model on the full dataset until convergence. This is particularly significant given the substantial costs associated with training an LLM.

6   Conclusion

In this paper, we delve deeply into the data selection problem from a data compression perspective. Inspired by the insight that language modeling is performing information compression, we propose an entropy law delineating the coherent relationship between model performance, data compression ratio, and training loss. Theoretically guided by the entropy law, we propose a new data selection algorithm, **ZIP**, to select data with the nearly lowest compression ratio, which is model-free and content-agnostic. rendering it significantly lightweight and versatile. Experimental results have demonstrated the effectiveness and efficiency of ZIP, based on various LLM backbones, during the SFT and RLHF stages. Further in-depth analysis provided empirical evidence of Entropy law, which could serve as a criterion for LLM performance prediction on specific data.

References

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

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A Survey on Data Selection for Language Models. arXiv:2402.16827 [cs.CL]

Alvaro Bartolome, Gabriel Martin, and Daniel Vila. 2023. Notus. `https://github.com/argilla-io/notus`.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701* (2023).

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)* 2, 3 (2023), 6.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research* 24, 240 (2023), 1–113.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377* (2023).

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. 2023. Language Modeling Is Compression. *CoRR* abs/2309.10668 (2023).

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing Chat Language Models by Scaling High-quality Instructional Conversations. In *EMNLP*. Association for Computational Linguistics, 3029–3051.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. 2023. Mods: Model-oriented data selection for instruction tuning. *arXiv preprint arXiv:2311.15653* (2023).

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306* (2024).

GitHub. 2020. GitHub Copilot. `https://github.com/features/copilot/`

Yuzhen Huang, Jinghan Zhang, Zifei Shan, and Junxian He. 2024. Compression Represents Intelligence Linearly. *arXiv preprint arXiv:2404.09937* (2024).

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew E. Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a Changing Climate: Enhancing LM Adaptation with Tulu 2. *CoRR* abs/2311.10702 (2023).

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *CoRR* abs/2310.06825 (2023).

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. 2024. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems* 36 (2024).

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024. Superfiltering: Weak-to-Strong Data Filtering for Fast Instruction-Tuning. *CoRR* abs/2402.00530 (2024).

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023. From Quantity to Quality: Boosting LLM Performance with Self-Guided Data Selection for Instruction Tuning. *CoRR* abs/2308.12032 (2023).

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023. What Makes Good Data for Alignment? A Comprehensive Study of Automatic Data Selection in Instruction Tuning. *CoRR* abs/2312.15685 (2023).

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. # InsTag: Instruction Tagging for Analyzing Supervised Fine-tuning of Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence* (2024), 1–11.

Meta. 2020. Llama3. `https://ai.meta.com/blog/meta-llama-3/`

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction Tuning with GPT-4. *CoRR* abs/2304.03277 (2023).

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446* (2021).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. Verbosity bias in preference labeling by large language models. *arXiv preprint arXiv:2310.10076* (2023).

Claude E. Shannon. 1948. A mathematical theory of communication. *Bell Syst. Tech. J.* 27, 3 (1948), 379–423.

Claude E Shannon. 1951. Prediction and entropy of printed English. *Bell system technical journal* 30, 1 (1951), 50–64.

Ming Shen. 2024. Rethinking Data Selection for Supervised Fine-Tuning. *CoRR* abs/2402.06094 (2024).

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023a. How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources. In *NeurIPS*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *ACL (1)*. Association for Computational Linguistics, 13484–13508.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705* (2022).

Zige Wang, Wanjun Zhong, Yufei Wang, Qi Zhu, Fei Mi, Baojun Wang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023c. Data management for large language models: A survey. *arXiv preprint arXiv:2312.01700* (2023).

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. QuRating: Selecting High-Quality Data for Training Language Models. *arXiv preprint arXiv:2402.09739* (2024).

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2023. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems* 36 (2023), 34201–34227.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. WizardLM: Empowering Large Language Models to Follow Complex Instructions. *CoRR* abs/2304.12244 (2023).

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. *CoRR* abs/2303.18223 (2023).

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *NeurIPS*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less Is More for Alignment. In *NeurIPS*.
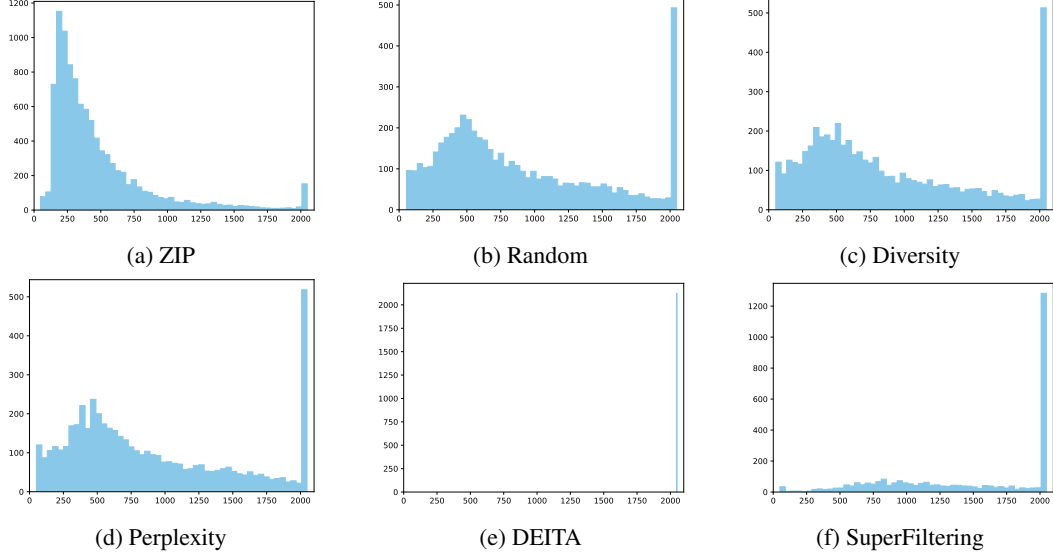
Figure 6: The distribution of average token number across datasets selected by different algorithms for Llama-3-8B.

A Derivations of joint mutual information of two QA pairs

$$
\begin{aligned}
I(q_1q_2; a_1a_2) &= H(a_1a_2) - H(a_1a_2|q_1q_2) \\
&= H(a_1a_2) - H(a_1a_2|q_1q_2) \\
&= H(a_1a_2) - H(a_1|q_1q_2) - H(a_2|a_1q_1q_2) \\
&= H(a_1a_2) - H(a_1|q_1) - H(a_2|q_2) \\
&\leq H(a_1) + H(a_2) - H(a_1|q_1) - H(a_2|q_2) \\
&= I(q_1; a_1) + I(q_2; a_2).
\end{aligned}
\tag{7}
$$

The equality is achieved when a_1 is independent on a_2 (similarly, q_1 needs to be independent on q_2).

B Training Details

Platform All experiments were finished on a platform with 64 Intel Xeon Gold 6326 CPU cores @ 2.90GHz, two main-stream high-performance GPUs, and 500GB memories. The training code is based on a popular open-source framework Axolotl[4].

Data preprocessing To format the multi-turn conversation data, we adopt the Vicuna-style template for Mistral-7B and the Llama-3 template for Llama-3-8B. Samples longer than the maximum input sequence length will be truncated. Besides, the data will be packed to speed up training for SFT.

Hyper-parameters For ZIP, the selection numbers K_1, K_2, and K_3 are set to 10000, 200, and 100, respectively. As for SFT, we share these hyper-parameters for all backbones: training batch size is 128, training epochs is 4, input sequence length is 2048, and the warm-up ratio is 0.1. We adopt different learning rates for each backbone: the learning rate of Mistral-7B is set to 4e-6, and the learning rate of Llama-3-8B is set to 1e-5. As for RLHF, the learning rate for KTO is set to 1e-6, and the batch size is set to 128.

C Token length distribution of more backbones

The token length distribution of data selected for Llama-3-8B is depicted in Figure 6, similar to the ones of Mistral-7B.
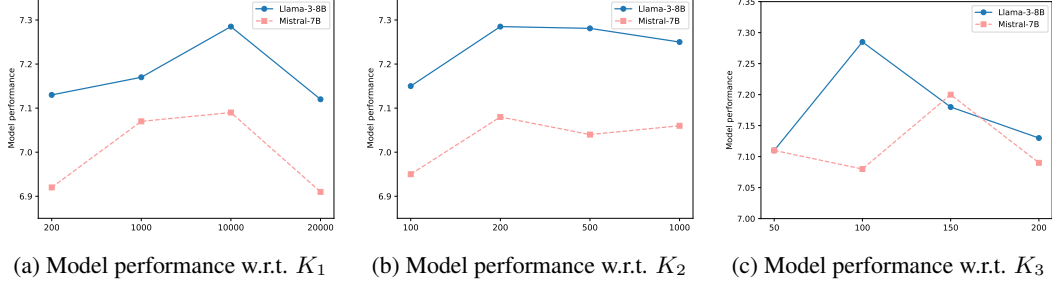
---

[4]https://github.com/OpenAccess-AI-Collective/axolotl

(a) Model performance w.r.t. K_1 (b) Model performance w.r.t. K_2 (c) Model performance w.r.t. K_3

Figure 7: Model performance w.r.t. different hyper-parameters.

D Hyper-parameter sensitivity

ZIP involves three hyper-parameters K_1, K_2, and K_3 for improved efficiency. We aim to investigate the impact of these hyper-parameters on the model performance, with results depicted in Figure 7.

Perceived sample number in global selection K_1 K_1 decides the number of samples to be updated in the global selection stage. We set K_1 among range [200, 1000, 10000, 20000], and the results are presented in Figure 7a. In the figure, the model performance exhibits an increasing trend when K_1 increases. When a smaller K_1 is specified, ZIP is only exposed to a limited set of samples. This can lead ZIP to degenerate into a variant that consistently selects samples based on individual compression ratios, neglecting the modeling of combinatorial effects. Furthermore, the compression ratio associated with the currently selected dataset typically increases with each update, whereas the compression ratios of other samples remain unchanged. Consequently, a large K_1 may result in the compression ratio of the un-updated samples being underestimated, leading to inferior samples' selection. As a result, a model performance degradation can be found when K_1 is set to 20,000.

Data pool size of local selection K_2 K_2 decides the number of samples selected from the previous K_1 samples. We set K_2 among range [100, 200, 500, 1000], and the results are presented in Figure 7b. The model performance increases with an increased K_2, which aligns with intuition since the algorithm can consider the combinatorial effects of more samples. But when K_2 exceeds a threshold, the model performance reaches a saturated phase, which indicates similar local selection results even with increased local data budget.

Data budget of local selection K_3 K_3 decides the number of samples selected from the previous K_2 samples. We set K_3 among range [50, 100, 150, 200], and the results are presented in Figure 7b. The results exhibit a similar trend as the results of K_1, yet the underlying causes are inverse. A large K_3 will make ZIP degenerate into a trivial variant that consistently selects samples based on individual compression ratios. On the other hand, a small K_3 will lead to more frequent compression ratio updates, which can also lead to underestimated compression ratios of some inferior samples.