Rethinking Data Curation in LLM Training: Online Reweighting Offers Better Generalization than Offline Methods

Anonymous authors
Paper under double-blind review

Abstract

Data curation is a critical yet underexplored component in large language model (LLM) training. Existing approaches (such as data selection and data mixing) operate in an offline paradigm, decoupled from the training process. This separation introduces extra engineering overhead and makes curated subsets brittle: once the model or task changes, the entire pipeline must be re-run. Moreover, offline methods alter dataset size through hard filtering or resampling, often discarding data diversity, and thus face the generalization issue.

We propose to rethink data curation as an online reweighting problem, where sample importance is dynamically adjusted during training via loss weighting rather than static preprocessing. This view preserves data diversity, adapts continuously to evolving model states, and yields a better performance–FLOPs tradeoff.

Thus, we introduce **ADAPT** (**A**daptive **D**ata reweighting for **P**retraining and Fine**T**uning), a dynamic online framework that reweights training samples with adaptive per-sample learning rates guided by similarity-based quality signals, without changing the number of training samples. ADAPT integrates reweighting directly into the optimization loop with negligible overhead. Experiments on both instruction tuning and large-scale pretraining show that ADAPT consistently outperforms offline selection/mixing and prior online methods, achieving stronger cross-benchmark generalization under equal FLOPs.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across diverse tasks (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023a), yet their performance hinges critically on the quality and proper mixture of training data (Hoffmann et al., 2022; Kaplan et al., 2020a). Data curation—deciding what data to keep and how to weight it—has thus become a cornerstone of model development in both pretraining and finetuning. In practice, however, data curation has often remained a "secret sauce": pipelines rely on opaque heuristics, ad-hoc engineering choices, or costly trial-and-error, rather than principled frameworks.

Current approaches to data curation largely follow an *offline paradigm*, falling into two camps: data selection, which keeps a subset of "valuable" examples, and data mixing, which adjusts sampling frequencies to rebalance distributions. Despite their differences, both approaches typically follow a multi-stage *offline* pipeline: (1) train or extract features with a proxy model, (2) compute quality signals as selection or weighting criteria on a validation set, and (3) retrain the main model on the curated data. While effective in some cases, this paradigm suffers from fundamental drawbacks (Sec. 4): offline subsets often overfit to a specific benchmark, and once the architecture or corpus changes, previously curated data quickly becomes suboptimal, forcing practitioners to restart the pipeline.

In contrast, we propose a shift to *online reweighting*, which dynamically adjusts sample contributions during training by weighting losses, without modifying the underlying dataset. This preserves data diversity compared to offline's "hard cuts" intelligently controlling each sample's "step size" in parameter updates to adapt to evolving model states.

To solve the limitation of offline data curation, we propose a unified online reweighting framework, **ADAPT** (**A**daptive **D**ata reweighting for **P**retraining and Fine**T**uning), which aligns training samples with a validation set via similarity-based weighting. Our method incurs nearly zero additional overhead yet achieves superior cross-benchmark generalization. Under a unified evaluation protocol, it consistently outperforms both offline selection/mixing strategies and prior online reweighting methods, across large-scale pretraining and instruction tuning setups.

Our contributions. Our paper makes three contributions:

1. **Formalization and unification of dataset curation pipelines.** We present a unified view of data curation, showing that data selection, mixing, and balancing can all be expressed as special cases of *data reweighting*. This perspective shifts the focus from offline pre-processing to dynamic, in-training adjustment of data weights. Which offers a fair protocol we propose to view and evaluate them through a unified lens of *online data reweighting*.
2. **Online sample-level reweighting method.** We introduce a new online global reweighting algorithm that is effective under this unified formalization and protocol. It employs an adaptive per-sample learning rate guided by sample quality signals, without explicitly altering the dataset size, while dynamically adjusting sample weights during training.
3. **Better performance and generalization across tasks.** On both instruction tuning and large-scale pretraining experiments, our framework achieves up to higher accuracy per FLOP than offline baselines including LESS (Xia et al., 2024), DoReMi (Xie et al., 2023b), and RegMix (Liu et al., 2024), and other online reweighting method (Sow et al., 2025), and demonstrates stronger out-of-domain generalization across benchmarks.

2 Background and Related Work

2.1 Data curation for large language models

Researchers and Engineers usually rely on web crawls to gather large datasets for training large language models (Brown, 2020; Computer, 2023; Penedo et al., 2024; Tang et al., 2024; Bai et al., 2023; Kandpal et al., 2025). However, these crawls often include a substantial amount of low-quality or irrelevant content, which makes data curation necessary to build high-quality training sets. Most data curation efforts focus on methods for improving model performance (Raffel et al., 2020; Brown, 2020; Rae et al., 2021; Penedo et al., 2023; Soldaini et al., 2024), including filtering by language (Raffel et al., 2020; Xue et al., 2020), heuristic-based filtering (Gao et al., 2020; Rae et al., 2021; Penedo et al., 2023; Soldaini et al., 2024), quality filtering (Du et al., 2022; Xie et al., 2023c), data deduplication (Lee et al., 2021) and mixing (Xie et al., 2023a; Soboleva et al., 2023; Albalak et al., 2023). However, current approaches to curating such datasets are generally ad-hoc. We aim to develop a principled and automated method for data curation that can also unify different processing stages.

Pretraining data curation. Several recent studies (Xie et al., 2023a; Chen et al., 2023b; Fan et al., 2023; Thakkar et al., 2023) have explored various reweighting techniques to enhance the generalization and efficiency of language models pretraining. For instance, Xie et al. (2023a) and Fan et al. (2023) are aiming at finding the optimal mixture of pretraining corpora to enhance performance across domains. Chen et al. (2023b) propose an ordered skill learning method for data selection measuring how effectively it teaches interdependent skills for continual pretraining and fine-tuning. Although effective, these works are aiming at the group level, whereas our work explores reweighting at the sample level, offering fine-grained control during model training dynamics. Instance-level reweighting has been used in pretraining settings of LLMs (Chen et al., 2024; Jiang et al., 2024), where each sample per mini-batch is weighted over how individual samples are treated based on their loss values. In contrast, our work studies the effects of various adaptive learning rate with different quality signals considering the model states to enhance both performance and generalization for LLMs pretraining and fine-tuning regimes.

Instruction data curation. Research has demonstrated that prioritizing data quality and diversity over quantity is more helpful for instruction-following capabilities (Cao et al., 2023; Chen et al., 2023a; Bukharin & Zhao, 2023; Du et al., 2023; Liu et al., 2023; Li et al., 2023). Instruction tuning data includes task-based datasets curated from traditional NLP tasks (Wang et al., 2022; Sanh et al., 2022; Wei et al., 2022a; Longpre et al., 2023), and open-ended datasets (Taori et al., 2023; Conover et al., 2023; Köpf et al., 2023; Xu et al., 2023; Mukherjee et al., 2023; Zhou et al., 2023; Ding et al., 2023). In our work, we are focusing on adaptively adjust the data weight during instruction tuning without explicitly reduce the number of instruction data samples and obtain model with generalization.

2.2 Commonly used quality signal for data curation

- **Lexical Similarity (BM25)** (Silva & Barbosa, 2024) quantifies term-based overlap between training and validation data through sparse retrieval scoring:
$$s_{\text{BM25}}(x) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{v \in \mathcal{D}_{\text{val}}} \text{BM25}(x, v). \tag{1}$$
This metric captures surface-level textual similarity without semantic understanding.
- **Semantic Similarity (Embedding)** (Rubin et al., 2021) measures dense representation alignment using pretrained encoders:
$$s_{\text{Embed}}(x) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{v \in \mathcal{D}_{\text{val}}} \cos\big(\phi(x), \phi(v)\big), \tag{2}$$
where $\phi(\cdot)$ denotes a frozen embedding model. This approach captures semantic proximity beyond lexical overlap.
- **Distributional Alignment (Perplexity)** (Antonello et al., 2020) evaluates likelihood under a reference language model θ_0:
$$s_{\text{PPL}}(x) = -\log P_{\theta_0}(x). \tag{3}$$
Lower perplexity indicates stronger distributional alignment with the reference corpus.
- **Gradient-Based Influence** (Xia et al., 2024) estimates training utility through first-order approximation of validation loss reduction:
$$s_{\text{Grad}}(x) = \big\langle \nabla_\theta \ell(x; \theta_0), \nabla_\theta \ell(\mathcal{D}_{\text{val}}; \theta_0) \big\rangle, \tag{4}$$
where θ_0 denotes a proxy model. This metric directly quantifies how training on x influences performance on the validation set $\mathcal{D}_{\text{val}}$.

3 Unification of Fomalizing Different Data Curation

Setup. Given two datasets: the *train* set $\mathcal{D}_{\text{train}}$ and the *validation* set $\mathcal{D}_{\text{val}}$. Usually, the size of the validation set is much smaller than the train set. When $\mathcal{D}_{\text{val}}$ is sampled from the distribution of downstream test data, the validation dataset $\mathcal{D}_{\text{val}}$ is considered as *in-domain*. The train dataset $\mathcal{D}_{\text{train}}$ on the other hand consists of both in-domain and *out-of-domain* samples. This is the case when web crawling is used to collect training data from the whole internet. Our goal is to subsample or weight the train set $\mathcal{D}_{\text{train}}$ under the guidance of the validation set $\mathcal{D}_{\text{val}}$, so that the model training is less affected by the out-of-domain samples.

3.1 Design Space of Data Quality Signal

In Sec. 2, we present a unified framework for data quality assessment where *scoring function* takes the form $s(x) \equiv s(x; \theta, \mathcal{D}_{\text{val}})$ that assigns a quality signal for each data example x, which optionally depends on a (proxy) model θ and the validation set $\mathcal{D}_{\text{val}}$.

3.2 Offline Data Curation

The scores are employed once before training (e.g., with a proxy model θ_0 and fixed validation set $\mathcal{D}_{\text{val}}$). The resulting weights $\{w(x)\}_{x \in \mathcal{D}_{\text{train}}}$ are fixed and used to make a decision to pass which training examples to the real training stage.

- **Data Selection (sample-level binary weights)** Data selection (or filtering) removes part of the training corpus before pretraining begins. It keeps data examples with quality signal above a certain threshold τ:
$$\mathcal{D}'_{\text{train}} = \{x \in \mathcal{D}_{\text{train}} \mid s(x) \geq \tau\}.$$

From a data reweighting perspective, filtering data points is equivalent to assigning binary weights of the form $w(x) = \mathbf{1}[s(x) \geq \tau] \in \{0, 1\}$.

- **Data Mixing (domain-level fractional weights)** Data mixing can be seen as a coarse-grained data reweighting method that operates at the domain level. In other words, data points within the same domain receive the same quality score. Pulling all domain scores $\{s_d\}_{d \in \text{domains}}$ and normalize them to obtain data mixing probability for each domain d

$$w_d = \frac{g(s_d)}{\sum_{d'} g(s_{d'})},$$

where g transform the score to be non-negative with a common choice being $g\colon s \mapsto \exp(s)$. As such, w_d determine how much of a total training budget B is allocated to each domain: $B_d = w_d B$. Alternatively, training can be implemented by sampling domains with probability w_d (probability mixing) or by assigning a fixed quota B_d (quota mixing). Since w_d is decided before training, mixing is an *offline* operation that alters the effective number of examples each domain contributes.

3.3 Online Data Curation

In this setting, the scoring function depends on the evolving model state θ_t and, in some cases, on a dynamic validation set. The resulting weights $\{w_t(x)\}_{x \in \mathcal{D}_{\text{train}}}$ evolve throughout training, adjusting the *gradient contributions* of examples while preserving the full volume of the training set $\mathcal{D}_{\text{train}}$.

Data Reweighting (sample-level fractional weights) Data reweighting assigns normalized weights to examples while keeping the training set size unchanged. Similar to Sec. 3.2, the quality score for example x is transformed to a non-negative weight: $w(x) = g(s(x))$, which scales their contribution to the loss:

$$\mathcal{L}^*(\theta) = \frac{1}{Z} \sum_{x \in \mathcal{D}} w(x)\, \mathcal{L}(\theta; x), \qquad Z = \sum_{x \in \mathcal{D}} w(x).$$

Equivalently, from a stochastic gradient descent perspective, the stochastic gradient w.r.t. x is scaled by $\frac{w(x)}{Z}$, which plays a role of *per-sample* learning rate in addition to the *global* learning rate η:

$$\theta_{t+1} = \theta_t - \eta \frac{w(x)}{Z} \nabla_\theta \ell(\theta_t; x).$$

Since the total number of training examples remains unchanged, only their relative contribution to parameter updates is modified, data reweighting is inherently an *online* method.

3.4 Unified Evaluation of Total FLOPs (Offline vs. Online)

The unification enables us to compare different methods under the same FLOPS calculation framework. We evaluate curation methods under a *cost-aware* metric that accounts for all computation spent to obtain and use the curated data. Let B denote a training budget (tokens or update steps).

Total FLOPs for offline data curation Offline curation modifies the *amount of data that participates in training* before training starts. Its total compute is

$$\mathsf{F}_{\text{total}}^{\text{off}} = \underbrace{\mathsf{F}_{\text{prep}}^{\text{off}}(\mathcal{D}; f)}_{\text{data scoring as preprocessing}} + \underbrace{\mathsf{F}_{\text{train}}\big(\mathcal{D}',\ B'\big)}_{\text{model training on the sampled subset}}, \tag{5}$$

where (i) $\mathsf{F}_{\text{prep}}^{\text{off}}$ includes any corpus-wide scoring, proxy-model passes, retrieval or filtering necessary to construct $\mathcal{D}'$; (ii) $\mathcal{D}' \subseteq \mathcal{D}$ is the retained subset (for selection), or an *effective* subset induced for data mixing; (iii) B' is the effective training budget after curation. By construction, offline selection uses *binary* example weights (keep/drop), and offline domain mixing changes per-domain sampling rates, thereby reducing or reallocating the volume of data seen during training.

Total FLOPs for online data curation Online curation *does not change the amount of training data*; instead, it modulates each example's contribution *during training*. Its total compute is

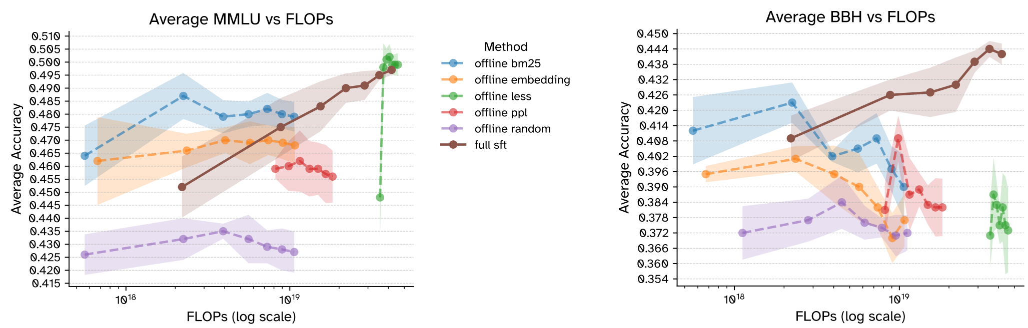
$$\mathsf{F}_{\text{total}}^{\text{on}} = \underbrace{\mathsf{F}_{\text{train}}\big(\mathcal{D},\ B\big)}_{\text{model training on the full corpus}} + \underbrace{\mathsf{F}_{\text{metrics}}^{\text{on}}}_{\text{on-the-fly features}}, \tag{6}$$

where $\mathsf{F}^{\mathrm{on}}_{\mathrm{metrics}}$ denotes lightweight, in-training computations needed to obtain scores (e.g., using current logits, per-example loss, or gradient norms). Because scoring is amortized inside the training loop and no data are removed, $\mathsf{F}^{\mathrm{on}}_{\mathrm{metrics}}$ is typically modest relative to $\mathsf{F}_{\mathrm{train}}$.

Conversely, online reweighting maintains the full training signal while adding only minimal in-loop metrics computation, providing a clearer accuracy–compute trade-off when measured by *total* FLOPs.

4 Revisiting of Offline Data Selection

In this section, based on the evaluation protocol derived from the unification above, we revisit the accuracy–FLOPs trade-off of existing offline data selection methods.



(a) In-domain evaluation: MMLU validation set → MMLU benchmark.

(b) Out-of-domain evaluation: MMLU validation set → BBH benchmark.

Figure 1. Efficiency–accuracy tradeoff of different data selection methods and full finetune method under our proposed **total FLOPs** metric. Selection methods (e.g., `LESS`) appear competitive under in-domain evaluation, but their advantage vanishes or even reverses in out-of-domain settings. In contrast, (`full sft`) remains consistently strong across domains.

Observation. Figure 1 illustrates the limitation of offline data selection methods: **Overfitting to the validation benchmark.** Model trained on offline selected data with MMLU (Hendrycks et al., 2021) as validation set performs well on the MMLU benchmark (Figure 1a), but the same model generalize poorly to BBH benchmark (Suzgun et al., 2022) (Figure 1b). This issue is especially severe when using LESS (Xia et al., 2024). This reveals that offline methods often *overfit to the chosen validation task*, lacking a true generalization. However, we observed that vanilla full dataset training demonstrate more stable performance in both benchmark. One possible reason is that official data curation which we directly change the number of data involved in the training by repetition would cause the model to replace generalization ability with memorization (Hernandez et al., 2022).

In summary, the offline paradigm has severe limitations, being neither cost-effective nor generalizable. This motivates us to explore an online alternative, which may alleviate memorization while retaining more data to enhance generalization. In the next section, we leverage these insights to design an *online reweighting method*.

5 Adaptive Learning Rate for Online Data Reweighting

The limitations of offline data selection suggest that a different paradigm is needed: one that is generalizable, and adaptive to the evolving state of the model. To this end, we propose an **online data reweighting framework** that is also suitable for data selection, data mixing with a unified formalization. Unlike offline methods that commit to a fixed subset before training begins, our framework dynamically adjusts data weights as the model learns, so it naturally adapts to new model states without re-running the data preprocessing and training pipeline from scratch.

5.1 Per-Sample Learning Rate Update

We cast data selection and mixing as *online data reweighting*. At training step t, given a minibatch $B_t \subset \mathcal{D}$, we update model parameters as

$$\theta_{t+1} = \theta_t - \eta \sum_{i \in B_t} w_t(i)\, \nabla_\theta \ell\big(f_\theta(x_i), y_i\big), \tag{7}$$

where $w_t(i) \geq 0$ denotes a dynamic weight assigned to sample i. Here, $w_t(i)$ directly scales the gradient contribution of sample i, and thus acts as a *per-sample learning rate multiplier*: larger weights amplify the effective step size on informative examples, while smaller weights downweight less useful ones.

In our framework, weights are derived from scoring functions $s_t(v, i)$ that compare a training example i with validation/query points $v \in \mathcal{D}_{\text{val}}$. These scores are aggregated across v, normalized within the current batch or pool, and transformed by a smooth gating function to produce the final weights $\widehat{w}_t(i)$. To ensure stability, weights are clipped to prevent excessively large effective learning rates and avoid gradient explosion. We now describe our score functions.

5.2 Model-Agnostic Per-Sample Learning Rate Update

ADAPT-BM25 quantifies term-based overlap between training and validation data through sparse retrieval scoring:

$$s_{\text{BM25}}(x) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{v \in \mathcal{D}_{\text{val}}} \text{BM25}(x, v). \tag{8}$$

This metric captures surface-level textual similarity without semantic understanding. We use the standard BM25 score $s_{\text{BM25}}(v, i)$ over sparse token matches between query v and example i. Aggregated scores are normalized to obtain target weights.

5.3 Per-Sample Learning Rate Update with Model States

ADAPT measures alignment using the model's own dense representations rather than a frozen encoder (Ivison et al., 2025). For an input x with last-layer hidden states $\{h_i\}_{i=1}^{L}$, we compute a position-weighted mean pooling:

$$w_i = \frac{i}{\sum_{j=1}^{L} j}, \qquad \phi(x) = \sum_{i=1}^{L} w_i h_i, \tag{9}$$

where later tokens receive higher weights to counteract the causal mask bias of decoder-only models. We then define the similarity score:

$$s_{\text{ADAPT}}(x) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{v \in \mathcal{D}_{\text{val}}} \cos\big(\phi(x), \phi(v)\big). \tag{10}$$

Equivalently, we can instantiate a representation-based scorer $s_{\text{ADAPT}}(v, i) = \cos\big(\phi(v), \phi(i)\big)$, where $\phi(\cdot)$ is the weighted hidden representation. Let $z_t(i)$ denote the per-example aggregated score at step t, optionally normalized across the current batch/pool. We then produce a *target* weight $\widehat{w}_t(i)$ via smooth gating of these scores. This metric captures semantic proximity in the representation space of the model being trained, while accounting for position-dependent contextualization.

6 Evaluation

6.1 Experimental Design

Instruction Tuning Models and Data. We use LoRA (Hu et al., 2021) to fine-tune the base model Llama-2-7B (Touvron et al., 2023c). Following the experimental setup in Wang et al. (2023), we use the instruction tuning datasets including Flan V2 (Longpre et al., 2023), CoT (Wei et al., 2022b), Dolly (Conover et al., 2023) and Open Assistant 1 (Köpf et al., 2023). The datasets do not contain any obvious in-domain data for the target queries. We evaluate our method on MMLU (Hendrycks et al., 2020), multiple-choice dataset spanning 57 tasks and BBH (Suzgun et al., 2023) from BIG-Bench selected to evaluate reasoning capabilities. For MMLU, we report 5-shot accuracy; for BBH, we report 3-shot exact match score. Appendix G contains more details on the training hyperparameter, Appendix D and E contain more dataset details.

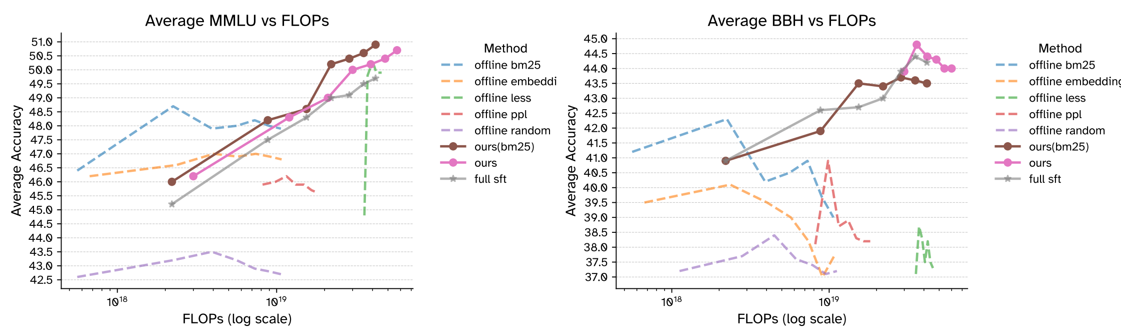
Instruction Tuning Baselines. We introduce instruction finetuning baselines in Sec. 2.2.

Pretraining Models and Data. We adopt **Tinyllama** architecture (Zhang et al., 2024) with 120M parameters, with FlashAttention (Dao et al., 2022) and Lit-GPT (LightningAI, 2023). More details about the training settings can be found at Appendix F. Following prior works (Touvron et al., 2023b;

Zhang et al., 2024; Wettig et al., 2024; Xie et al., 2023b), we employ **SlimPajama** (Touvron et al., 2023b; Computer, 2023) as the text corpus, which is specifically curated for pre-training LLMs. All selections are performed on about 590M training files of SlimPajama, processed with Llama tokenizer (Touvron et al., 2023b). We evaluate our method on a diverse set of 15 downstream benchmarks, following common practice in prior work such as RegMix (Liu et al., 2024). These tasks span a wide range of realistic settings, including: ARC-E (Clark et al., 2018), ARC-C (Clark et al., 2018), COPA (Sarlin et al., 2020), HellaSwag Zellers et al. (2019), Lambada-S (Paperno et al., 2016), Lambada-O (Radford et al., 2019), LogiQA (Liu et al., 2020), MultiRC (Khashabi et al., 2018), OpenBookQA (Mihaylov et al., 2018), PiQA (Bisk et al., 2020), QQP (Wang, 2018), RACE (Lai et al., 2017), SciQ (Welbl et al., 2017), Social IQA (Sap et al., 2019), WinoGrande (Sakaguchi et al., 2021). The reported accuracy in table 2 is measured in the 0-shot setting scored using the lm-eval-harness evaluation framework (Gao et al., 2024).

Pretraining Baselines. We compare ADAPT with **Uniform** selection and existing file selection methods for LLM pre-training, including **Doremi** (Xie et al., 2023b) and **RegMix** (Liu et al., 2024). ADAPT requires an anchor set typically consists of examples in the evaluation distribution. To construct this set, we sample 50 validation examples from each of eight evaluation benchmarks: ARC-C, COPA, Lambada, MultiRC, PiQA, RACE, SciQ, and Social IQA. For Doremi and Regmix we use the domain weights in (Lu et al., 2023) as the selection ratio of text samples in different domains in our experiment. For Uniform we use the same ratio to sample from each domain. We also compare **LinUpper** (Sow et al., 2025): an online sample reweighting strategy where the sample weight is proportional to the normalized loss but is capped at a predefined α value, ensuring that outliers do not dominate the training process. Due to the large cost of pretraining, for each method we sample 9B unique tokens from SlimPajama, and train for a total budget of 50B tokens, i.e., train for approximately 5.6 epochs.

6.2 Results for Instruction Tuning



(a) Subset distribution of selected examples.

(b) Out-of-domain evaluation: MMLU validation set → BBH benchmark.

Figure 2. Efficiency–accuracy tradeoff of different offline selection and online reweighting methods under our proposed **total FLOPs** metric. Offline methods (e.g., `LESS`) appear competitive under in-domain evaluation, but still weaker than online methods `ours(bm25)`, `ADAPT`. Also, their advantage vanishes or even reverses in out-of-domain settings. In contrast, online reweighting remains consistently strong across domains.

As shown in Figure 2, our proposed online data reweighting method consistently outperforms existing approaches under comparable computational budgets:

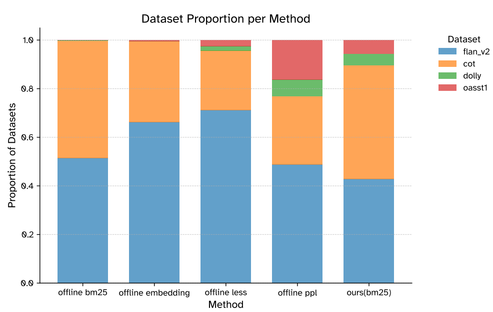
Comparison with offline data selection methods. Across all baselines in both Fig. 2a and Fig. 2b, our approach consistently lies on the Pareto frontier, demonstrating the best trade-off between computational efficiency and final accuracy. At the same FLOPs, our method achieves substantially higher accuracy than the offline data selection SoTA method LESS. To reach the same performance level, our approach requires significantly fewer FLOPs than LESS. Moreover, after reaching this performance, further training with LESS leads to overfitting and accuracy degradation, whereas online data reweighting continues to exhibit steady performance gains. Notably, our online reweighting method using BM25 as quality signal (ADAPT-BM25) has better generalization than its offline selection method (BM25) with the other configuration keep the same, which demonstrates the advantage of the online data curation.

Table 1. Generalization results under different validation-test configurations.

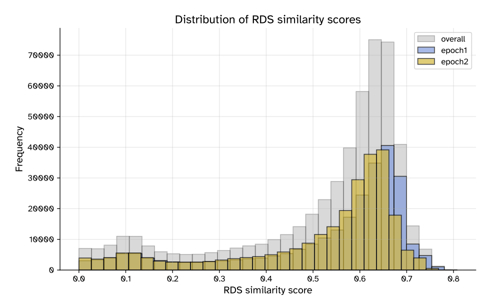
Method	MMLU*(val)* - MMLU*(test)*	MMLU*(val)* - BBH*(test)*
BM25	48.7 ± 0.9	42.3 ± 0.8
Embedding	47.0 ± 0.6	40.1 ± 0.5
LESS	50.2 ± 0.5	38.7 ± 1.5
PPL	46.2 ± 1.1	40.9 ± 0.9
Random	43.5 ± 0.3	38.4 ± 1.0
Full Dataset SFT	49.7 ± 0.2	44.4 ± 0.3
ADAPT-bm25	**50.9 ± 0.6**	**43.7 ± 1.2**
ADAPT	**50.7 ± 0.7**	**44.8 ± 1.3**

Comparison with full-data fine-tuning. In Fig. 2a and Table. 1, our method also outperforms full-data fine-tuning under equal FLOPs, with the performance gap widening as training progresses. In Fig. 2b and Table. 1, our reweighting strategy that incorporates model state information (ADAPT) outperforms our fixed-state online reweighting (ADAPT-BM25), highlighting the importance of adapting to the evolving model state to better generalisation.

Generalization cross benchmark. As shown in Figure 2b, we further validate the generalization capability of models trained with , where we use MMLU (Hendrycks et al., 2021) as the validation set to select the data, and evaluate the model trained on selected data on the BBH (Suzgun et al., 2022) task. The results, also reported in Table 1, show that ADAPT achieves a better generalization performance between benchmark compared to other offline data selection baselines and demonstrates comparable generalization with vanilla full dataset instruction tuning. This highlights a key strength of ADAPT: when new downstream tasks or benchmarks emerge, there is no need to repeat the entire data selection and model training pipeline. Over the long term, this substantially reduces FLOPs costs, making highly practical for real-world applications.



(a) Distribution of effective selected examples



(b) Similarity distribution for different epochs

Figure 3. A summary of the distribution of effective selected samples and similarity scores. Both differences and changes show the proposed method's ability on capturing data features and adaptive learning.

Data mixing and curriculum. We present the distribution of the effective selected examples for different methods in Fig. 3a. Different approaches end up selecting very different amounts of data from each training subset. It demonstrats that online reweighting solution can also help with deciding proper data mixture. We also calculate the *effective proportion of used data*, by summing up weights of all data together, with the result of 0.501 in our experimental setting. In practice, this number would automatically adapt according to the overall quality of the training corpus we are selecting from.

We also presented Fig. 3b. The shift in similarity distributions from epoch 1 to epoch 2 reflects a transition from collapsed, overly homogeneous representations toward more diverse and fine-grained embeddings. This diversification enhances generalisation by improving the model's ability to capture subtle distinctions and reducing over-reliance on coarse features. Notably, this process can be viewed as an instance of implicit curriculum learning: the model first clusters samples based on coarse, "easier" patterns and progressively moves toward harder, fine-grained discrimination, thereby mirroring the principles of curriculum learning without explicit scheduling.

Table 2. **Benchmark performance of *TinyLlama-120M* trained on 50B tokens using Uniform, LinUpper, DoReMi, RegMix, and ADAPT.**

Tasks	Uniform	LinUpper	DoReMi	RegMix	ADAPT
ARC-C	17.75	18.77	18.00	18.60	**19.11**
ARC-E	39.60	38.85	40.70	**41.04**	39.06
COPA	63.00	60.00	**66.00**	61.00	64.00
HellaSwag	**28.39**	27.90	28.03	27.63	28.11
Lambada-O	**24.68**	23.64	22.38	24.82	24.63
Lambada-S	16.98	16.79	16.30	18.30	**18.07**
LogiQA	20.43	21.35	21.51	20.89	**21.66**
MultiRC	56.68	50.74	48.99	**56.70**	55.67
OpenBookQA	15.80	14.00	15.20	**18.20**	14.20
PIQA	60.55	60.83	59.90	58.98	**61.48**
QQP	36.88	**37.85**	36.84	36.83	36.81
RACE	**27.85**	26.41	26.41	**27.85**	26.60
SciQ	71.00	70.70	**72.90**	71.20	72.50
SocialIQA	36.90	36.85	36.95	**37.41**	37.05
WinoGrande	50.75	50.83	49.64	50.12	**50.99**
Average (↑)	37.81	37.03	37.32	37.97	**38.00**

6.3 Pretraining results

As shown in Table 2, ADAPT outperforms all baselines in terms of average performance, achieving a 0.19% improvement over Uniform, 0.68% over DoReMi, and a 0.97% over LinUpper. RegMix reaches comparable but slightly lower accuracy on average. Interestingly, LinUpper performs worse than Uniform sampling, suggesting that naive sample-level adjustments may not generalize well. Compared to LinUpper, which conducts normalization and calculates sample-level weights for each batch, our global sample-level reweighting solution demonstrates superior performance in handling unknown ratios of mixed-quality corpora. To examine the effect of larger training budgets, we extend training to 100B tokens. As shown in Table 3, ADAPT achieves higher benchmark performance than Uniform on 11 out of 15 downstream tasks, with an average improvement of 0.38%. Table 6 further shows that ADAPT consistently yields lower validation perplexity than Uniform at both 50B and 100B training budgets. These findings demonstrate that ADAPT improves both validation perplexity and downstream task performance consistently across compute scales.

7 Discussion and Conclusion

Our work unified data selection, mixing, and reweighting under a FLOPs-aware framework. We showed that offline pipelines often incur significant cost via data preprocessing and faces issues with generalization. In contrast, online reweighting integrates seamlessly into training, adapts to model state, and improves efficiency without altering the dataset size.

Empirically, we demonstrated that ADAPT consistently outperforms both offline and online baselines across instruction tuning and pretraining setups. In instruction tuning, ADAPT not only achieves higher accuracy under equal FLOPs but also exhibits stronger cross-benchmark generalization (e.g., from MMLU to BBH), mitigating the brittleness of offline methods. In pretraining, ADAPT improves both downstream task accuracy and validation perplexity under 50B and 100B token budgets, underscoring its robustness across compute scales.

Table 3. **Benchmark performance of *TinyLlama-120M* trained on 100B tokens using Uniform and ADAPT.**

Tasks	Uniform	ADAPT
ARC-C	17.75	**18.86**
ARC-E	**40.66**	40.45
COPA	**63.00**	61.00
HellaSwag	28.35	**28.75**
Lambada-O	25.13	**25.91**
Lambada-S	18.47	**20.14**
LogiQA	20.28	**21.66**
MultiRC	55.98	**56.44**
OpenBookQA	**16.00**	15.00
PIQA	61.48	**61.97**
QQP	36.81	**36.82**
RACE	27.37	**27.46**
SciQ	71.50	**72.40**
SocialIQA	**35.77**	37.05
WinoGrande	51.14	**51.54**
Average (↑)	37.98	**38.36**

Ethics and Reproducibility Statements

We adhere to the ICLR Code of Ethics. This paper focuses on comparing methodologies for data curation. To ensure fairness, we evaluate online data reweighting and offline data mixing methods under a FLOPs-equivalent setting. We also rigorously test their performance in both instruction fine-tuning and pretraining scenarios to provide a comprehensive and balanced comparison.

During the preparation of this manuscript, we utilized large language models (LLMs) to assist with grammar correction and refinement of the writing.

We provide all necessary details to ensure reproducibility of our work. The theoretical justification for our FLOPs-equivalent data curation formalization is presented in Section 3, with detailed FLOP calculations in Appendix C. Implementation details and training protocols are provided in Section 6.1, Appendix B, Appendix F, and Appendix G, while descriptions of the training data are given in Appendix D.

References

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training, 2023.

Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L. Leavitt, and Mansheej Paul. Perplexed by perplexity: Perplexity-based data pruning with small reference models, 2024. URL https://arxiv.org/abs/2405.20541.

Richard Antonello, Nicole Beckage, Javier Turek, and Alexander Huth. Selecting informative contexts improves language model finetuning. *arXiv preprint arXiv:2005.00175*, 2020.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

Alexander Bukharin and Tuo Zhao. Data diversity matters for robust instruction tuning. *arXiv preprint arXiv:2311.14736*, 2023.

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

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

Mayee F Chen, Nicholas Roberts, Kush Bhatia, Jue WANG, Ce Zhang, Frederic Sala, and Christopher Re. Skill-it! a data-driven skills framework for understanding and training language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=IoizwO1NLf.

Xuxi Chen, Zhendong Wang, Daouda Sow, Junjie Yang, Tianlong Chen, Yingbin Liang, Mingyuan Zhou, and Zhangyang Wang. Take the bull by the horns: Hard sample-reweighted continual training improves llm generalization. *arXiv preprint arXiv:2402.14270*, 2024.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

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

Together Computer. Redpajama: an open dataset for training large language models, 2023. URL https://github.com/togethercomputer/RedPajama-Data.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free Dolly: Introducing the world's first truly open instruction-tuned LLM, 2023.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning (ICML)*, pp. 5547–5569. PMLR, 2022.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. Mods: Model-oriented data selection for instruction tuning, 2023.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. DOGE: Domain reweighting with generalization estimation. In *Second Agent Learning in Open-Endedness Workshop*, 2023. URL https://openreview.net/forum?id=qiKqsqwYXm.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. 2020.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, et al. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. Large-scale data selection for instruction tuning. *arXiv preprint arXiv:2503.01807*, 2025.

Chunyang Jiang, Chi-min Chan, Wei Xue, Qifeng Liu, and Yike Guo. Importance weighting can help large language models self-improve. *arXiv preprint arXiv:2408.09849*, 2024.

Nikhil Kandpal, Brian Lester, Colin Raffel, Sebastian Majstorovic, Stella Biderman, Baber Abbasi, Luca Soldaini, Enrico Shippole, A Feder Cooper, Aviya Skowron, et al. The common pile v0. 1: An 8tb dataset of public domain and openly licensed text. *arXiv preprint arXiv:2506.05209*, 2025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020a.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020b. URL https://arxiv.org/abs/2001.08361.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*, 2023.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Annual Meeting of the Association for Computational Linguistics*, 2021. URL https://api.semanticscholar.org/CorpusID:235829052.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Junhao Liu, Tongliang Liu, Fei Huang, et al. One shot learning as instruction data prospector for large language models. *arXiv preprint arXiv:2312.10302*, 2023.

LightningAI. Litgpt. https://github.com/Lightning-AI/litgpt, 2023.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*, 2020.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. *arXiv preprint arXiv:2407.01492*, 2024.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*, 2023.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. In *Proceedings of Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HtqnVSCj3q.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale, 2023. URL https://arxiv.org/abs/2309.04564.

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

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*, 2023.

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

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

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 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. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*, 2021.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Levy Silva and Luciano Barbosa. Improving dense retrieval models with llm augmented data for dataset search. *Knowledge-Based Systems*, 294:111740, 2024.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://cerebras.ai/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, June 2023. URL https://huggingface.co/datasets/cerebras/SlimPajama-627B.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2402.00159.

Daouda Sow, Herbert Woisetschläger, Saikiran Bulusu, Shiqiang Wang, Hans Arno Jacobsen, and Yingbin Liang. Dynamic loss-based sample reweighting for improved large language model pretraining. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gU4ZgQNsOC.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Liping Tang, Nikhil Ranjan, Omkar Pangarkar, Xuezhi Liang, Zhen Wang, Li An, Bhaskar Rao, Linghao Jin, Huijuan Wang, Zhoujun Cheng, et al. Txt360: A top-quality llm pre-training dataset requires the perfect blend, 2024.

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

Megh Thakkar, Tolga Bolukbasi, Sriram Ganapathy, Shikhar Vashishth, Sarath Chandar, and Partha Talukdar. Self-influence guided data reweighting for language model pre-training. *arXiv preprint arXiv:2311.00913*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023b.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023c.

Alex Wang. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109, 2022.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. In *Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*, 2023.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=gEZrGCozdqR.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022b.

Johannes Welbl, Nelson F Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*, 2017.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. In *Forty-first International Conference on Machine Learning*, 2024.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*, 2024.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=lXuByUeHhd.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36, 2023b.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023c.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

Junjie Oscar Yin and Alexander M. Rush. Compute-constrained data selection, 2024. URL https://arxiv.org/abs/2410.16208.

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

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KBMOKmX2he.

APPENDIX

Contents

A Extended Discussion. 17

B Instruction Tuning Baselines 17

C Data-Selection FLOPs 17

D Training Datasets in Instruction Tuning 17

E Evaluation Datasets in Instruction Tuning 17

F Pretraining Details 18

G Instruction Tuning Details 18

H Pretraining Perplexity 18

A Extended Discussion.

Implication. ADAPT could be effectively used for getting better control under constrained budgets (both *data budget* and *compute budget*). Our empirical observation provides a principled view and reference for automatically allocating computational resources to the most valuable samples while maintaining end-to-end efficiency. Additionally, we expect our method to perform even better where the data quality is often low, which is the common case in practical setup. where data is crude and is not ideal for direct use for LLM training, sources of generally low quality, and biases inherent to the distribution of content on the web.

Future Work. We didn't explicitly discuss and evaluate deduplication, which is a key step for data preprocessing. For example, we will treat deduplication as reweighting related documents in accordance with their frequency. Based on this framework, we will explore different trade-offs inherent in each stage, such as quality-aware deduplication where high-quality documents are allowed to be duplicated more than once.

B Instruction Tuning Baselines

We compare ADAPT with the following baselines: **1) Random Selection**: We randomly sample data from the instruction tuning dataset. **2) BM25** (Robertson et al., 2009): We assign weights to training samples based on textual statistical features (i.e., TF-IDF), and select the top k data points with the highest scores. **4) PPL** (Yin & Rush, 2024; Marion et al., 2023; Ankner et al., 2024): We compute the loss of each training sample on our original base model and use it as its score. **6) LESS** Xia et al. (2024): We train LoRAs on a random subset of the data, and then selecting data by computing the gradient-based influence of each training sample to validation samples.

C Data-Selection FLOPs

To estimate computational costs throughout our paper, we adopt the methodology of Kaplan et al. (2020b), which approximates the training step computation as approximately $6N$ FLOPs per processed token, where N represents the model's parameter count (approximately 7B). According to Kaplan et al. (2020b), the forward pass consumes roughly half the computational resources of the backward pass, yielding an estimate of $2N$ FLOPs per token during sample processing. We employ an approximation of 2,048 tokens per sample, as we limit all samples to this maximum length during both the training and selection phases. Note that in all experimental configurations, we conduct full fine-tuning of models over two complete epochs. If we define N as model size, P as the data pool magnitude (measured in sample quantity), and D as the number of samples chosen for training, we can calculate the computational expense for each methodology as follows:

1. **Random Selection**: $2k \times 6N \times D \times E$
2. **BM25**: $2k \times 6N \times D \times E$
3. **Embedding**: $2k \times 2N' \times P + 2k * 6N \times D \times E$ (embed model N')
3. **PPL**: $2k \times 2NP + 2k \times 6N \times D \times E$
4. **LESS**: $1.53 \times 2k \times 6N \times P + 2k \times 6N \times D \times E$ (LESS computes gradients for three checkpoints over the entire pool.)
5. **ADAPT-BM25**: $2k \times 6N \times P \times E$
6. **ADAPT**: $2k \times 8N \times P \times E$

D Training Datasets in Instruction Tuning

Table 4 contains information about the training sets used in instruction tuning.

E Evaluation Datasets in Instruction Tuning

Table 5 contains detailed statistics of the evaluation datasets used in instruction finetuning.

Table 4. Datails of training dataset from Wang et al. (2023). Len. is short for token length.

Dataset	# Instance	Sourced from	# Rounds	Prompt Len.	Completion Len.
Flan V2	100,000	NLP datasets and human-written instructions	1	355.7	31.2
CoT	100,000	NLP datasets and human-written CoTs	1	266	53.2
Dolly	15,011	Human-written from scratch	1	118.1	91.3
Open Assistant 1	55,668	Human-written from scratch	1.6	34.8	212.5

Table 5. Statistics of evaluation datasets. The selection of evaluation tasks cover different kinds of answer types.

Dataset	# Shot	# Tasks	$\lvert\mathcal{D}_{\text{val}}\rvert$	$\lvert\mathcal{D}_{\text{test}}\rvert$	Answer Type
MMLU	5	57	285	18,721	Letter options
BBH	3	23	69	920	COT and answer

F Pretraining Details

We follow all settings in TinyLlama (Zhang et al., 2024). The optimizer is AdamW (Loshchilov & Hutter, 2019), setting parameters β_1 at 0.9 and β_2 at 0.95. We adopt the cosine learning rate schedule with a maximum learning rate of 4e-4 and the minimum of 4e-5, the batch size of 2M tokens, the weight decay of 0.1, and the gradient clipping threshold of 1.

G Instruction Tuning Details

All experiments were conducted with parameter-efficient finetuning method LoRA (Hu et al., 2021). For the LoRA adapter, we specified a rank of 128, an α value of 512, and a dropout rate of 0.1 and applied it across all attention matrices. Adding the LoRA adapter introduce minimal FLOPs overhead during training—having no impact on our FLOPS analysis—and mainly reduce memory requirements for more accessible training.

H Pretraining Perplexity

In Table 6, we evaluate validation perplexity on SlimPajama for TinyLlama-120M under two compute budgets (50B and 100B tokens). Across both settings, ADAPT achieves consistently lower perplexity than the Uniform baseline, highlighting its effectiveness in improving data efficiency.

Table 6. **Validation perplexity of TinyLlama-120M on SlimPajama under different training budgets.**

Method	50B Train Budget	100B Train Budget
Uniform	17.15	15.94
ADAPT	16.55	15.36