

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 pre-processing. Unlike offline methods that enforce a static data distribution, ADAPT acts as an implicit curriculum learner, progressively shifting focus from coarse-grained patterns to fine-grained semantic distinctions as the model’s representations evolve. This view preserves data diversity, adapts continuously to evolving model states, and yields a better performance–FLOPs tradeoff. Thus, we introduce **ADAPT** (Adaptive Data reweighting for Pretraining and FineTuning), 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 minimal 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) due to its static and subtractive nature. First, it ignores training dynamics: the ‘value’ of a data sample is not static but changes as the model learns. Offline selection freezes this value based on a proxy model, creating a misalignment with the training model’s evolving needs. Second, it compromises diversity: by relying on hard filtering to select a fixed subset, offline methods risk discarding the broad data distribution essential for robust generalization. Consequently, the resulting subsets often

054 overfit to specific benchmarks and are brittle, which requires the entire pipeline to be re-run whenever
 055 the model architecture or corpus changes.

056
 057 In contrast, we propose a shift to *online reweighting*, which fundamentally differs from offline selection
 058 not just in *when* the signal is computed, but also in *how* the data is used. Instead of hard-filtering
 059 a subset based on static scores, online reweighting retains the full dataset to preserve diversity, but
 060 dynamically modulates each sample’s contribution via loss weighting. This preserves data diversity
 061 compared to offline’s ‘hard cuts’ while intelligently controlling each sample’s ‘step size’ in parameter
 062 updates to adapt to evolving model states.

063 To solve the limitation of offline data curation, we propose a unified online reweighting framework,
 064 **ADAPT** (Adaptive Data reweighting for Pretraining and FineTuning), which aligns training samples
 065 with a validation set via similarity-based weighting. Our method incurs nearly zero additional
 066 overhead yet achieves superior cross-benchmark generalization. Under a unified evaluation protocol,
 067 it consistently outperforms both offline selection/mixing strategies and prior online reweighting
 068 methods, across large-scale pretraining and instruction tuning setups.

069 **Our contributions.** Our paper makes three contributions:

- 071 1.  **Formalization and unification of dataset curation pipelines.** We present a unified view of
 072 data curation, showing that data selection, mixing, and balancing can all be expressed as special
 073 cases of *data reweighting*. This perspective shifts the focus from offline pre-processing to dynamic,
 074 in-training adjustment of data weights. Which offers a fair protocol we propose to view and
 075 evaluate them through a unified lens of *online data reweighting*.
- 076 2.  **Online sample-level reweighting method.** We introduce a new online global reweighting
 077 algorithm that is effective under this unified formalization and protocol. It employs an adaptive
 078 per-sample learning rate guided by sample quality signals, without explicitly altering the dataset
 079 size, while dynamically adjusting sample weights during training.
- 080 3.  **Better performance and generalization across tasks.** On both instruction tuning and large-
 081 scale pretraining experiments, our framework achieves up to up to 7.2% higher accuracy per FLOP
 082 than offline baselines including LESS (Xia et al., 2024), DoReMi (Xie et al., 2023b), and Reg-
 083 Mix (Liu et al., 2024), and other online reweighting method (Sow et al., 2025), and demonstrates
 084 6.1% stronger out-of-domain generalization across benchmarks.

086 2 BACKGROUND AND RELATED WORK

088 2.1 DATA CURATION FOR LARGE LANGUAGE MODELS

089
 090 Researchers and Engineers usually rely on web crawls to gather large datasets for training large language
 091 models (Brown, 2020; Computer, 2023; Penedo et al., 2024; Tang et al., 2024; Bai et al., 2023;
 092 Kandpal et al., 2025). However, these crawls often include a substantial amount of low-quality or
 093 irrelevant content, which makes data curation necessary to build high-quality training sets. Most
 094 data curation efforts focus on methods for improving model performance (Raffel et al., 2020; Brown,
 095 2020; Rae et al., 2021; Penedo et al., 2023; Soldaini et al., 2024), including filtering by language (Raf-
 096 fel et al., 2020; Xue et al., 2020), heuristic-based filtering (Gao et al., 2020; Rae et al., 2021; Penedo
 097 et al., 2023; Soldaini et al., 2024), quality filtering (Du et al., 2022; Xie et al., 2023c), data dedupli-
 098 cation (Lee et al., 2021) and mixing (Xie et al., 2023a; Soboleva et al., 2023; Albalak et al., 2023).
 099 However, current approaches to curating such datasets are generally ad-hoc. We aim to develop a
 100 principled and automated method for data curation that can also unify different processing stages.

101 **Pretraining data curation.** Several recent studies (Xie et al., 2023a; Chen et al., 2023b; Fan et al.,
 102 2023; Thakkar et al., 2023) have explored various reweighting techniques to enhance the generaliza-
 103 tion and efficiency of language models pretraining. For instance, Xie et al. (2023a) and Fan et al.
 104 (2023) are aiming at finding the optimal mixture of pretraining corpora to enhance performance
 105 across domains. Chen et al. (2023b) propose an ordered skill learning method for data selection
 106 measuring how effectively it teaches interdependent skills for continual pretraining and fine-tuning.
 107 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:

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

This metric captures surface-level textual similarity without semantic understanding.

- **Semantic Similarity (Embedding)** (Rubin et al., 2021) measures dense representation alignment using pretrained encoders:

$$s_{\text{Embed}}(x) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{v \in \mathcal{D}_{\text{val}}} \cos(\phi(x), \phi(v)), \quad (2)$$

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 :

$$s_{\text{PPL}}(x) = -\log P_{\theta_0}(x). \quad (3)$$

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:

$$s_{\text{Grad}}(x) = \langle \nabla_{\theta} \ell(x; \theta_0), \nabla_{\theta} \ell(\mathcal{D}_{\text{val}}; \theta_0) \rangle, \quad (4)$$

where θ_0 denotes a proxy model. This metric directly quantifies how training on x influences performance on the validation set \mathcal{D}_{val} .

3 UNIFIED FORMALIZATION OF DATA CURATION

Setup. Given two datasets: the *train* set $\mathcal{D}_{\text{train}}$ and the *validation* set \mathcal{D}_{val} . Usually, the size of the validation set is much smaller than the train set. When \mathcal{D}_{val} is sampled from the distribution of downstream test data, the validation dataset \mathcal{D}_{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}_{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}_{val} .

162 3.2 OFFLINE DATA CURATION
163

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

- 167
- 168 • **Data Selection (sample-level binary weights)** Data selection (or filtering) removes part of the
169 training corpus before pretraining begins. It keeps data examples with quality signal above a certain
170 threshold τ :

$$171 \mathcal{D}'_{\text{train}} = \{x \in \mathcal{D}_{\text{train}} \mid s(x) \geq \tau\}.$$

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

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

$$178 w_d = \frac{g(s_d)}{\sum_{d'} g(s_{d'})},$$

179 where g transform the score to be non-negative with a common choice being $g: s \mapsto \exp(s)$. As
180 such, w_d determine how much of a total training budget B is allocated to each domain: $B_d = w_d B$.
181 Alternatively, training can be implemented by sampling domains with probability w_d (probability
182 mixing) or by assigning a fixed quota B_d (quota mixing). Since w_d is decided before training,
183 mixing is an *offline* operation that alters the effective number of examples each domain contributes.
184
185

186 3.3 ONLINE DATA CURATION
187

188 In this setting, the scoring function depends on the evolving model state θ_t and, in some cases, on a
189 dynamic validation set. The resulting weights $\{w_t(x)\}_{x \in \mathcal{D}_{\text{train}}}$ evolve throughout training, adjusting
190 the *gradient contributions* of examples while preserving the full volume of the training set $\mathcal{D}_{\text{train}}$.
191

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

$$196 \mathcal{L}^*(\theta) = \frac{1}{Z} \sum_{x \in \mathcal{D}} w(x) \mathcal{L}(\theta; x), \quad Z = \sum_{x \in \mathcal{D}} w(x).$$

199 Equivalently, from a stochastic gradient descent perspective, the stochastic gradient w.r.t. x is scaled
200 by $\frac{w(x)}{Z}$, which plays a role of *per-sample* learning rate in addition to the *global* learning rate η :
201

$$202 \theta_{t+1} = \theta_t - \eta \frac{w(x)}{Z} \nabla_{\theta} \ell(\theta_t; x).$$

204 Since the total number of training examples remains unchanged, only their relative contribution to
205 parameter updates is modified, data reweighting is inherently an *online* method.
206

207 3.4 UNIFIED EVALUATION OF TOTAL FLOPS (OFFLINE VS. ONLINE)

208 The unification enables us to compare different methods under the same FLOPS calculation frame-
209 work. We evaluate curation methods under a *cost-aware* metric that accounts for all computation
210 spent to obtain and use the curated data. Let B denote a training budget (tokens or update steps).
211

212 **Total FLOPs for offline data curation** Offline curation modifies the *amount of data that partici-*
213 *pates in training* before training starts. Its total compute is

$$214 F_{\text{total}}^{\text{off}} = \underbrace{F_{\text{prep}}^{\text{off}}(\mathcal{D}; f)}_{\text{data scoring as preprocessing}} + \underbrace{F_{\text{train}}(\mathcal{D}', B')}_{\text{model training on the sampled subset}}, \quad (5)$$

215

where (i) $F_{\text{prep}}^{\text{off}}$ 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

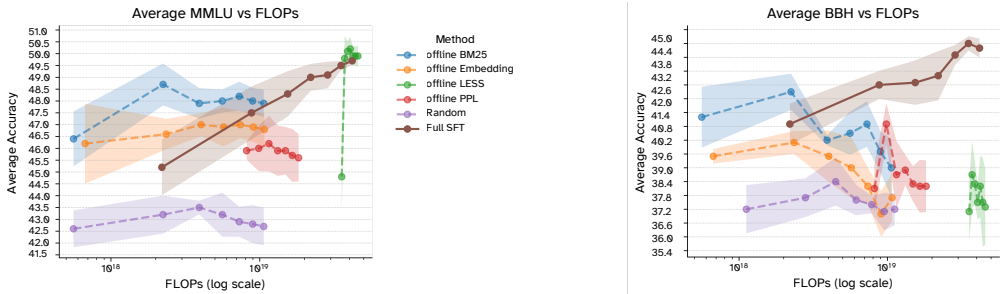
$$F_{\text{total}}^{\text{on}} = \underbrace{F_{\text{train}}(\mathcal{D}, B)}_{\text{model training on the full corpus}} + \underbrace{F_{\text{metrics}}^{\text{on}}}_{\text{on-the-fly features}}, \tag{6}$$

where $F_{\text{metrics}}^{\text{on}}$ 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, $F_{\text{metrics}}^{\text{on}}$ is typically modest relative to F_{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

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

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 $\hat{w}_t(i)$. 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:

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

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 (Iverson et al., 2025). For an input x with last-layer hidden states $\{h_i\}_{i=1}^L$, we compute a position-weighted mean pooling:

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

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

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

Equivalently, we can instantiate a representation-based scorer $s_{\text{ADAPT}}(v, i) = \cos(\phi(v), \phi(i))$, where $\phi(\cdot)$ is the weighted hidden representation.

We use a sigmoid function with temperature scaling to produce *absolute* weights that are independent of the batch composition:

$$w_t(i) = \sigma\left(\frac{s_{\text{ADAPT}}(x_i)}{\max(\tau, \epsilon)}\right) = \frac{1}{1 + \exp(-s_{\text{ADAPT}}(x_i)/\max(\tau, \epsilon))} \quad (11)$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\tau > 0$ is a temperature hyperparameter (default $\tau = 1.0$), and $\epsilon > 0$ is a small constant (e.g., $\epsilon = 10^{-8}$) used for numerical stability. The temperature parameter

τ controls the steepness of the sigmoid: larger values produce flatter weight distributions, while smaller values create sharper distinctions between high and low similarity samples.

Global Weighting via Sigmoid: The sigmoid transformation maps similarity scores to absolute weights in the interval $[0, 1]$ without requiring normalization across samples in the batch. This design choice ensures that the weight assigned to each sample x_i depends solely on its similarity score $s_{ADAPT}(x_i)$ relative to the anchor set, rather than its rank within the current mini-batch. Consequently, a sample with a given similarity score receives the same weight regardless of whether it appears in a high-quality or low-quality batch, making the weighting mechanism robust to batch-level variations in data quality. This contrasts with normalized weighting schemes (e.g., softmax normalization) where weights are relative to other samples in the same batch, potentially amplifying or dampening effects based on batch composition.

Embedding Normalization: We apply L2 normalization to embeddings $\phi(x)$ and $\phi(v)$ before computing cosine similarities to ensure scale-invariant similarity measurements. Specifically, for each embedding vector, we compute $\phi(x) \leftarrow \phi(x) / \max(\|\phi(x)\|_2, \epsilon)$ where ϵ prevents division by zero. This normalization step is distinct from weight normalization and serves to standardize the embedding space for similarity computation.

Online Validation Embedding Updates: To maintain alignment with the evolving model representations, anchor embeddings $\{\phi(v)\}_{v \in \mathcal{D}_{val}}$ are refreshed every R training steps via forward passes on the current model parameters θ_t . This online update mechanism ensures that similarity scores reflect the model’s current representation space rather than outdated embeddings from earlier training stages. The refresh interval R is a hyperparameter that balances computational efficiency with representation freshness. The complete algorithm is summarized in Algorithm 1.

While gradient-based methods (e.g., LESS) directly measure influence, gradients significantly fluctuate during early training and are computationally expensive to compute frequently. In contrast, semantic embeddings provide a smoother, more stable signal of relevance that evolves consistently with the model’s representation space, offering a robust guide for reweighting without the noise of instantaneous gradient estimates.

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.

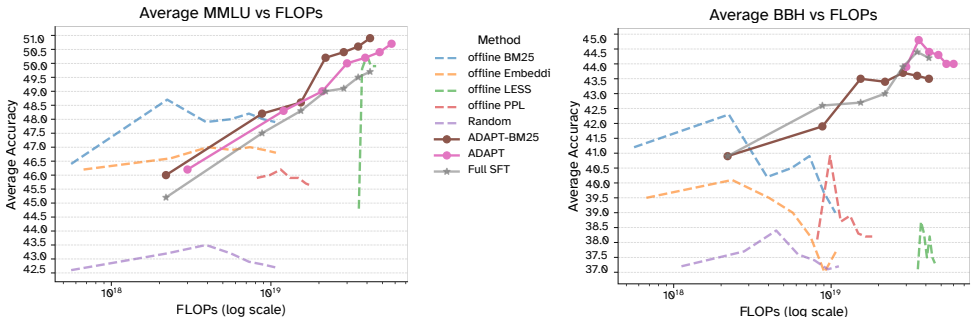
Instruction Tuning Baselines. We introduce instruction finetuning baselines in Sec. 2.2.

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) In-domain evaluation: MMLU validation set \rightarrow MMLU benchmark. (b) Out-of-domain evaluation: MMLU validation set \rightarrow BBH benchmark.

Figure 2. Efficiency-accuracy tradeoff. (a) While offline BM25 is efficient at low FLOPs, it saturates early. ADAPT dominates in the high-performance regime (higher FLOPs), avoiding the early plateauing of heuristics. (b) Crucially, LESS overfits to the in-domain task, showing regression on OOD tasks (BBH), whereas ADAPT consistently improves generalization on the Pareto frontier.

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

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

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

Comparison with offline data selection methods. Across all baselines in both Figure 2a and Figure 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

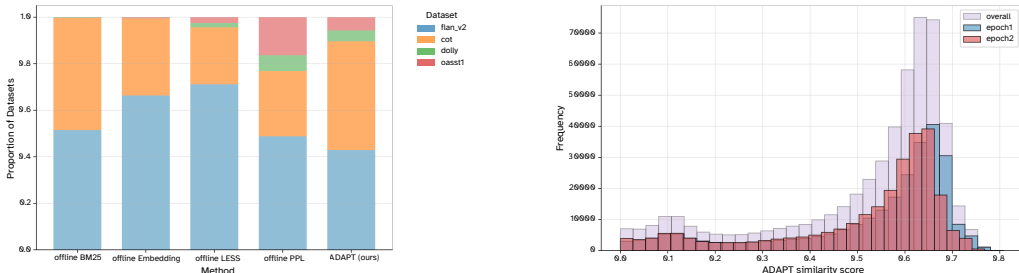
Table 2. Benchmark performance of *TinyLlama-120M* trained on 50B tokens using Uniform, LinUpper, DoReMi, RegMix, and ADAPT.

Tasks	Uniform	LinUpper	DoReMi	RegMix	ADAPT-BM25	ADAPT
ARC-C	17.75 \pm 0.00	18.77 \pm 0.24	18.00 \pm 0.84	18.60 \pm 0.06	18.31 \pm 0.55	19.11 \pm 0.60
ARC-E	39.60 \pm 0.10	38.85 \pm 0.12	40.70 \pm 0.87	41.04 \pm 0.21	39.44 \pm 0.60	39.06 \pm 0.54
COPA	63.00 \pm 0.00	60.00 \pm 1.72	66.00 \pm 1.95	61.00 \pm 1.5	63.20 \pm 0.40	64.00 \pm 1.32
HellaSwag	28.39 \pm 0.00	27.90 \pm 0.13	28.03 \pm 0.03	27.63 \pm 0.03	27.96 \pm 0.07	28.11 \pm 0.00
Lambada-O	24.68 \pm 0.64	23.64 \pm 0.28	22.38 \pm 0.73	24.82 \pm 0.94	26.19 \pm 0.30	24.63 \pm 0.53
Lambada-S	16.98 \pm 0.81	16.79 \pm 0.91	16.30 \pm 1.05	18.30 \pm 0.14	18.43 \pm 0.33	18.07 \pm 0.79
LogiQA	20.43 \pm 0.54	21.35 \pm 0.03	21.51 \pm 0.11	20.89 \pm 0.07	20.52 \pm 0.69	21.66 \pm 0.69
MultiRC	56.68 \pm 0.00	50.74 \pm 0.29	48.99 \pm 2.57	56.70 \pm 0.93	56.53 \pm 0.44	55.67 \pm 1.11
OpenBookQA	15.80 \pm 0.40	14.00 \pm 0.18	15.20 \pm 0.60	18.20 \pm 0.50	15.40 \pm 0.72	14.20 \pm 0.43
PIQA	60.55 \pm 0.29	60.83 \pm 0.45	59.90 \pm 0.39	58.98 \pm 0.29	61.00 \pm 0.19	61.48 \pm 0.23
QQP	36.88 \pm 0.00	37.85 \pm 1.21	36.84 \pm 0.04	36.83 \pm 0.00	36.82 \pm 0.00	36.81 \pm 0.00
RACE	27.85 \pm 0.14	26.41 \pm 0.18	26.41 \pm 0.03	27.85 \pm 0.00	27.39 \pm 0.35	26.60 \pm 0.49
SciQ	71.00 \pm 1.00	70.70 \pm 1.34	72.90 \pm 0.23	71.20 \pm 0.50	72.48 \pm 0.44	72.50 \pm 1.37
SocialIQA	36.90 \pm 0.03	36.85 \pm 0.16	36.95 \pm 0.07	37.41 \pm 0.46	36.52 \pm 0.47	37.05 \pm 0.30
WinoGrande	50.75 \pm 0.17	50.83 \pm 0.09	49.64 \pm 0.79	50.12 \pm 0.72	50.61 \pm 1.25	50.99 \pm 0.00
Average (All) (\uparrow)	37.81 \pm 0.13	37.03 \pm 0.12	37.32 \pm 0.15	37.97 \pm 0.02	38.05 \pm 0.21	38.00 \pm 0.22
Average (Unseen) (\uparrow)	31.98 \pm 0.09	30.56 \pm 0.28	31.95 \pm 0.17	32.46 \pm 0.39	33.49 \pm 0.37	33.73 \pm 0.39
FLOPs overhead (\downarrow)	0	0	4.92×10^{19}	3.072×10^{18}	$\ll 1.0 \times 10^{14}$	$\ll 1.1 \times 10^{15}$

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.

Comparison with full-data fine-tuning. In Figure 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 ADAPT, 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 ADAPT 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 Figure 3a. Different approaches end up selecting very different amounts of data

486 from each training subset. It demonstrates that online reweighting solution can also help with deciding
 487 proper data mixture. We also calculate the *effective proportion of used data*, by summing up weights
 488 of all data together, with the result of 0.501 in our experimental setting. In practice, this number
 489 would automatically adapt according to the overall quality of the training corpus we are selecting
 490 from.

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

498 6.3 PRETRAINING RESULTS

501 As shown in Table 2, ADAPT outperforms
 502 all baselines in terms of average perform-
 503 ance, achieving a 0.19% improvement
 504 over Uniform, 0.68% over DoReMi, and
 505 a 0.97% over LinUpper. RegMix reaches
 506 comparable but slightly lower accuracy on
 507 average. Interestingly, LinUpper performs
 508 worse than Uniform sampling, suggesting
 509 that naive sample-level adjustments may
 510 not generalize well. Compared to Lin-
 511 Upper, which conducts normalization and
 512 calculates sample-level weights for each
 513 batch, our global sample-level reweight-
 514 ing solution demonstrates superior per-
 515 formance in handling unknown ratios of
 mixed-quality corpora.

516 To examine the effect of larger training bud-
 517 gets, we extend training to 100B tokens.
 518 As shown in Table 3, ADAPT achieves
 519 higher benchmark performance than Uni-
 520 form on 11 out of 15 downstream tasks,
 521 with an average improvement of 0.38%.
 522 Table 6 further shows that ADAPT con-
 523 sistentlly yields lower validation perplexity
 524 than Uniform at both 50B and 100B train-
 525 ing budgets. These findings demonstrate that ADAPT improves both validation perplexity and down-
 526 stream task performance consistently across compute scales.

527 7 DISCUSSION AND CONCLUSION

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

532 Empirically, we demonstrated that ADAPT consistently outperforms both offline and online baselines
 533 across instruction tuning and pretraining setups. In instruction tuning, ADAPT not only achieves
 534 higher accuracy under equal FLOPs but also exhibits stronger cross-benchmark generalization (e.g.,
 535 from MMLU to BBH), mitigating the brittleness of offline methods. In pretraining, ADAPT im-
 536 proves both downstream task accuracy and validation perplexity under 50B and 100B token budgets,
 537 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 (All) (↑)	37.98	38.36
Average (Unseen) (↑)	32.51	32.67

540 REFERENCES

- 541
- 542 Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing
543 for language model pre-training, 2023.
- 544 Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L. Leavitt, and Man-
545 sheej Paul. Perplexed by perplexity: Perplexity-based data pruning with small reference models,
546 2024. URL <https://arxiv.org/abs/2405.20541>.
- 547
- 548 Richard Antonello, Nicole Beckage, Javier Turek, and Alexander Huth. Selecting informative con-
549 texts improves language model finetuning. *arXiv preprint arXiv:2005.00175*, 2020.
- 550
- 551 Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge,
552 Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.
- 553
- 554 Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical
555 commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*,
556 volume 34, pp. 7432–7439, 2020.
- 557
- 558 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal,
559 Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are
560 few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- 561
- 562 Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- 563
- 564 Alexander Bukharin and Tuo Zhao. Data diversity matters for robust instruction tuning. *arXiv*
565 *preprint arXiv:2311.14736*, 2023.
- 566
- 567 Yihan Cao, Yanbin Kang, and Lichao Sun. Instruction mining: High-quality instruction data selec-
568 tion for large language models. *arXiv preprint arXiv:2307.06290*, 2023.
- 569
- 570 Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay
571 Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpapasus: Training a better alpaca with fewer data.
572 *arXiv preprint arXiv:2307.08701*, 2023a.
- 573
- 574 Mayee F Chen, Nicholas Roberts, Kush Bhatia, Jue WANG, Ce Zhang, Frederic Sala, and Christo-
575 pher Re. Skill-it! a data-driven skills framework for understanding and training language mod-
576 els. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL
577 <https://openreview.net/forum?id=Ioizw01Nlf>.
- 578
- 579 Xuxi Chen, Zhendong Wang, Daouda Sow, Junjie Yang, Tianlong Chen, Yingbin Liang, Mingyuan
580 Zhou, and Zhangyang Wang. Take the bull by the horns: Hard sample-reweighted continual train-
581 ing improves llm generalization. *arXiv preprint arXiv:2402.14270*, 2024.
- 582
- 583 Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam
584 Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM:
585 Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.
- 586
- 587 Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and
588 Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge.
589 *arXiv preprint arXiv:1803.05457*, 2018.
- 590
- 591 Together Computer. Redpajama: an open dataset for training large language models, 2023. URL
592 <https://github.com/togethercomputer/RedPajama-Data>.
- 593
- 588 Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick
589 Wendell, Matei Zaharia, and Reynold Xin. Free Dolly: Introducing the world’s first truly open
590 instruction-tuned LLM, 2023.
- 591
- 592 Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-
593 efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*,
35:16344–16359, 2022.

- 594 Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong
595 Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional
596 conversations. *arXiv preprint arXiv:2305.14233*, 2023.
- 597
- 598 Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim
599 Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language
600 models with mixture-of-experts. In *International Conference on Machine Learning (ICML)*, pp.
601 5547–5569. PMLR, 2022.
- 602 Qianlong Du, Chengqing Zong, and Jiajun Zhang. Mods: Model-oriented data selection for instruc-
603 tion tuning, 2023.
- 604
- 605 Simin Fan, Matteo Pagliardini, and Martin Jaggi. DOGE: Domain reweighting with generalization
606 estimation. In *Second Agent Learning in Open-Endedness Workshop*, 2023. URL [https://](https://openreview.net/forum?id=qiKqsqwYXm)
607 openreview.net/forum?id=qiKqsqwYXm.
- 608 Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang,
609 Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for
610 language modeling. *arXiv preprint arXiv:2101.00027*, 2020.
- 611
- 612 Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Fos-
613 ter, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muen-
614 nighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang
615 Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for
616 few-shot language model evaluation, 2024.
- 617 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob
618 Steinhardt. Measuring Massive Multitask Language Understanding. 2020.
- 619
- 620 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob
621 Steinhardt. Measuring massive multitask language understanding. In *Proceedings of the Interna-*
622 *tional Conference on Learning Representations (ICLR)*, 2021.
- 623 Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nel-
624 son Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, et al. Scaling laws and inter-
625 pretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022.
- 626
- 627 Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Ruther-
628 ford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training
629 compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.
- 630 Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,
631 and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint*
632 *arXiv:2106.09685*, 2021.
- 633
- 634 Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. Large-scale data
635 selection for instruction tuning. *arXiv preprint arXiv:2503.01807*, 2025.
- 636
- 637 Chunyang Jiang, Chi-min Chan, Wei Xue, Qifeng Liu, and Yike Guo. Importance weighting can
638 help large language models self-improve. *arXiv preprint arXiv:2408.09849*, 2024.
- 639 Nikhil Kandpal, Brian Lester, Colin Raffel, Sebastian Majstorovic, Stella Biderman, Baber Abbasi,
640 Luca Soldaini, Enrico Shippole, A Feder Cooper, Aviya Skowron, et al. The common pile v0. 1:
641 An 8tb dataset of public domain and openly licensed text. *arXiv preprint arXiv:2506.05209*, 2025.
- 642
- 643 Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott
644 Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models.
645 *arXiv preprint arXiv:2001.08361*, 2020a.
- 646
- 647 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>.

- 648 Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking be-
649 yond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceed-*
650 *ings of the 2018 Conference of the North American Chapter of the Association for Computational*
651 *Linguistics: Human Language Technologies*, 2018.
- 652 Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens,
653 Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. Openassistant
654 conversations—democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*,
655 2023.
- 656 Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading
657 comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.
- 658 Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-
659 Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In
660 *Annual Meeting of the Association for Computational Linguistics*, 2021. URL [https://api.](https://api.semanticscholar.org/CorpusID:235829052)
661 [semanticscholar.org/CorpusID:235829052](https://api.semanticscholar.org/CorpusID:235829052).
- 662 Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiayi Yang, Min Yang, Lei Zhang, Shuzheng Si, Junhao Liu,
663 Tongliang Liu, Fei Huang, et al. One shot learning as instruction data prospector for large language
664 models. *arXiv preprint arXiv:2312.10302*, 2023.
- 665 LightningAI. Litgpt. <https://github.com/Lightning-AI/litgpt>, 2023.
- 666 Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A
667 challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint*
668 *arXiv:2007.08124*, 2020.
- 669 Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing
670 Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. *arXiv*
671 *preprint arXiv:2407.01492*, 2024.
- 672 Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for align-
673 ment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint*
674 *arXiv:2312.15685*, 2023.
- 675 Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V
676 Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective
677 instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.
- 678 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International*
679 *Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.
680 OpenReview.net, 2019.
- 681 Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu,
682 and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language mod-
683 els. In *Proceedings of Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
684 URL <https://openreview.net/forum?id=HtqnVSCj3q>.
- 685 Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When
686 less is more: Investigating data pruning for pretraining llms at scale, 2023. URL [https://arxiv.](https://arxiv.org/abs/2309.04564)
687 [org/abs/2309.04564](https://arxiv.org/abs/2309.04564).
- 688 Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct
689 electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*,
690 2018.
- 691 Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed
692 Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint*
693 *arXiv:2306.02707*, 2023.
- 694 Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi,
695 Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset:
696 Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

- 702 Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli,
703 Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb
704 dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv*
705 *preprint arXiv:2306.01116*, 2023.
- 706 Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin
707 Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the
708 finest text data at scale, 2024. URL <https://arxiv.org/abs/2406.17557>.
- 709 Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language
710 models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- 711 Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John
712 Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models:
713 Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.
- 714 Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi
715 Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text
716 transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.
- 717 Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond.
718 *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.
- 719 Ohad Rubín, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context
720 learning. *arXiv preprint arXiv:2112.08633*, 2021.
- 721 Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adver-
722 sarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.
- 723 Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine
724 Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker,
725 Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, De-
726 bajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen,
727 Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen,
728 Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao,
729 Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training
730 enables zero-shot task generalization. In *International Conference on Learning Representations*,
731 2022. URL <https://openreview.net/forum?id=9Vrb9D0WI4>.
- 732 Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialliqa: Common-
733 sense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.
- 734 Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue:
735 Learning feature matching with graph neural networks. In *Conference on Computer Vision and*
736 *Pattern Recognition (CVPR)*, 2020.
- 737 Levy Silva and Luciano Barbosa. Improving dense retrieval models with llm augmented data for
738 dataset search. *Knowledge-Based Systems*, 294:111740, 2024.
- 739 Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel
740 Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and
741 deduplicated version of RedPajama. [https://cerebras.ai/blog/
742 slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama](https://cerebras.ai/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama),
743 June 2023. URL <https://huggingface.co/datasets/cerebras/SlimPajama-627B>.
- 744 Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur,
745 Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha,
746 Sachin Kumar, Li Lucy, Xinxu Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas
747 Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle
748 Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke
749 Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and
750 Kyle Lo. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining
751 Research. *arXiv preprint*, 2024. URL <https://arxiv.org/abs/2402.00159>.

- 756 Daouda Sow, Herbert Woiseschläger, Saikiran Bulusu, Shiqiang Wang, Hans Arno Jacobsen, and
757 Yingbin Liang. Dynamic loss-based sample reweighting for improved large language model pre-
758 training. In *The Thirteenth International Conference on Learning Representations*, 2025. URL
759 <https://openreview.net/forum?id=gU4ZgQNs0C>.
- 760 Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung,
761 Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks
762 and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.
- 763 Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung,
764 Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. Challenging big-bench tasks and
765 whether chain-of-thought can solve them. In *Findings of the Association for Computational Lin-*
766 *guistics: ACL 2023*, pp. 13003–13051, 2023.
- 767 Liping Tang, Nikhil Ranjan, Omkar Pangarkar, Xuezhi Liang, Zhen Wang, Li An, Bhaskar Rao,
768 Linghao Jin, Huijuan Wang, Zhoujun Cheng, et al. Txt360: A top-quality llm pre-training dataset
769 requires the perfect blend, 2024.
- 770 Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy
771 Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model.
772 https://github.com/tatsu-lab/stanford_alpaca, 2023.
- 773 Megh Thakkar, Tolga Bolukbasi, Sriram Ganapathy, Shikhar Vashishth, Sarath Chandar, and Partha
774 Talukdar. Self-influence guided data reweighting for language model pre-training. *arXiv preprint*
775 *arXiv:2311.00913*, 2023.
- 776 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée
777 Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and
778 efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.
- 779 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée
780 Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Ar-
781 mand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation
782 language models, 2023b.
- 783 Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay
784 Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation
785 and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023c.
- 786 Alex Wang. Glue: A multi-task benchmark and analysis platform for natural language understanding.
787 *arXiv preprint arXiv:1804.07461*, 2018.
- 788 Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei,
789 Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al.
790 Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In
791 *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp.
792 5085–5109, 2022.
- 793 Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu,
794 David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go?
795 exploring the state of instruction tuning on open resources. In *Neural Information Processing*
796 *Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*, 2023.
- 797 Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, An-
798 drew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International*
799 *Conference on Learning Representations*, 2022a. URL <https://openreview.net/forum?id=gEzrGCozdqR>.
- 800 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny
801 Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint*
802 *arXiv:2201.11903*, 2022b.
- 803 Johannes Welbl, Nelson F Liu, and Matt Gardner. Crowdsourcing multiple choice science questions.
804 *arXiv preprint arXiv:1707.06209*, 2017.

- 810 Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality
811 data for training language models. In *Forty-first International Conference on Machine Learning*,
812 2024.
- 813 Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Se-
814 lecting influential data for targeted instruction tuning. In *International Conference on Machine*
815 *Learning (ICML)*, 2024.
- 816
- 817 Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang,
818 Quoc V Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up
819 language model pretraining. In *Thirty-seventh Conference on Neural Information Processing Sys-*
820 *tems*, 2023a. URL <https://openreview.net/forum?id=1XuByUeHhd>.
- 821 Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang,
822 Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up lan-
823 guage model pretraining. *Advances in Neural Information Processing Systems*, 36, 2023b.
- 824
- 825 Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language mod-
826 els via importance resampling. In *Advances in Neural Information Processing Systems (NeurIPS)*,
827 2023c.
- 828 Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin
829 Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv*
830 *preprint arXiv:2304.12244*, 2023.
- 831
- 832 Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya
833 Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv*
834 *preprint arXiv:2010.11934*, 2020.
- 835 Junjie Oscar Yin and Alexander M. Rush. Compute-constrained data selection, 2024. URL <https://arxiv.org/abs/2410.16208>.
- 836
- 837 Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine
838 really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.
- 839
- 840 Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small
841 language model. *arXiv preprint arXiv:2401.02385*, 2024.
- 842
- 843 Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat,
844 Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy.
845 LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Process-*
846 *ing Systems*, 2023. URL <https://openreview.net/forum?id=KBMOkmX2he>.
- 847
- 848
- 849
- 850
- 851
- 852
- 853
- 854
- 855
- 856
- 857
- 858
- 859
- 860
- 861
- 862
- 863

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

APPENDIX

Contents

A	Extended Discussion.	18
B	Instruction Tuning Baselines	18
C	Data-Selection FLOPs	18
D	Training Datasets in Instruction Tuning	18
E	Evaluation Datasets in Instruction Tuning	18
F	Pretraining Details	19
G	Instruction Tuning Details	19
H	Pretraining Perplexity	19
I	Pseudo Code	20
J	Quantitative Analysis	20

918 A EXTENDED DISCUSSION.

919
920 **Implication.** ADAPT could be effectively used for getting better control under constrained budgets
921 (both *data budget* and *compute budget*). Our empirical observation provides a principled view and
922 reference for automatically allocating computational resources to the most valuable samples while
923 maintaining end-to-end efficiency. Additionally, we expect our method to perform even better where
924 the data quality is often low, which is the common case in practical setup. where data is crude and is
925 not ideal for direct use for LLM training, sources of generally low quality, and biases inherent to the
926 distribution of content on the web.

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

933 B INSTRUCTION TUNING BASELINES

934
935 We compare ADAPT with the following baselines: **1) Random Selection:** We randomly sample data
936 from the instruction tuning dataset. **2) BM25** (Robertson et al., 2009): We assign weights to training
937 samples based on textual statistical features (i.e., TF-IDF), and select the top k data points with the
938 highest scores. **4) PPL** (Yin & Rush, 2024; Marion et al., 2023; Ankner et al., 2024): We compute
939 the loss of each training sample on our original base model and use it as its score. **6) LESS** Xia et al.
940 (2024): We train LoRAs on a random subset of the data, and then selecting data by computing the
941 gradient-based influence of each training sample to validation samples.

943 C DATA-SELECTION FLOPS

944
945 To estimate computational costs throughout our paper, we adopt the methodology of Kaplan et al.
946 (2020b), which approximates the training step computation as approximately $6N$ FLOPs per pro-
947 cessed token, where N represents the model's parameter count (approximately 7B). According to
948 Kaplan et al. (2020b), the forward pass consumes roughly half the computational resources of the
949 backward pass, yielding an estimate of $2N$ FLOPs per token during sample processing. We employ
950 an approximation of 2,048 tokens per sample, as we limit all samples to this maximum length during
951 both the training and selection phases. Note that in all experimental configurations, we conduct full
952 fine-tuning of models over two complete epochs. If we define N as model size, P as the data pool
953 magnitude (measured in sample quantity), and D as the number of samples chosen for training, we
954 can calculate the computational expense for each methodology as follows:

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

964 D TRAINING DATASETS IN INSTRUCTION TUNING

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

969 E EVALUATION DATASETS IN INSTRUCTION TUNING

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

Table 4. Details 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	$ \mathcal{D}_{\text{val}} $	$ \mathcal{D}_{\text{test}} $	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

I PSEUDO CODE

Algorithm 1: Online Data Reweighting via Embedding Similarity

```

1026 I PSEUDO CODE
1027
1028 Algorithm 1: Online Data Reweighting via Embedding Similarity
1029
1030 Input : Training set  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ , validation set  $\mathcal{D}_{val} = \{v_j\}_{j=1}^{|\mathcal{D}_{val}|}$ , model  $f_\theta$ , temperature  $\tau$ ,
1031 learning rate  $\eta$ , refresh interval  $R$ , training steps  $T$ , small constant  $\epsilon > 0$ 
1032 Output : Updated parameters  $\theta_T$ 
1033
1034 1 Initialize model parameters  $\theta_0$ 
1035 2 for  $t = 1$  to  $T$  do
1036 3   Sample mini-batch  $B_t \subset \mathcal{D}$ 
1037    $\triangleright$  Refresh anchor embeddings every  $R$  steps via forward pass on current model  $\theta_t$ 
1038   (online update)
1039 4   if  $t \bmod R = 1$  then
1040 5     foreach  $v \in \mathcal{D}_{val}$  do
1041 6       Forward pass:  $(h_1^{(v)}, \dots, h_{L_v}^{(v)}) \leftarrow f_{\theta_t}(v)$ 
1042 7       Position weights:  $w_\ell \leftarrow \ell / \sum_{j=1}^{L_v} j$  for  $\ell = 1, \dots, L_v$ 
1043 8        $\phi(v) \leftarrow \sum_{\ell=1}^{L_v} w_\ell h_\ell^{(v)}$ 
1044 9        $\phi(v) \leftarrow \phi(v) / \max(\|\phi(v)\|_2, \epsilon)$ 
1045 10    end
1046 11  end
1047    $\triangleright$  Forward propagation for training samples
1048 12  foreach  $(x_i, y_i) \in B_t$  do
1049 13    Forward pass:  $(\hat{y}_i, h_1^{(i)}, \dots, h_{L_i}^{(i)}) \leftarrow f_{\theta_t}(x_i)$ 
1050 14    Position weights:  $w_\ell \leftarrow \ell / \sum_{j=1}^{L_i} j$  for  $\ell = 1, \dots, L_i$ 
1051 15     $\phi(x_i) \leftarrow \sum_{\ell=1}^{L_i} w_\ell h_\ell^{(i)}$ 
1052 16     $\phi(x_i) \leftarrow \phi(x_i) / \max(\|\phi(x_i)\|_2, \epsilon)$ 
1053 17  end
1054    $\triangleright$  Compute quality scores
1055 18  foreach  $i \in B_t$  do
1056 19     $s_{ADAPT}(x_i) \leftarrow \frac{1}{|\mathcal{D}_{val}|} \sum_{v \in \mathcal{D}_{val}} \cos(\phi(x_i), \phi(v))$ 
1057 20  end
1058    $\triangleright$  Compute per-sample weights
1059 21  foreach  $i \in B_t$  do
1060 22     $w_t(i) \leftarrow \sigma(s_{ADAPT}(x_i) / \max(\tau, \epsilon))$ 
1061 23  end
1062    $\triangleright$  Compute weighted loss
1063 24   $\mathcal{L}_t \leftarrow \sum_{i \in B_t} w_t(i) \ell(\hat{y}_i, y_i)$ 
1064    $\triangleright$  Backward propagation and parameter update
1065 25   $\theta_{t+1} \leftarrow \theta_t - \eta \sum_{i \in B_t} w_t(i) \nabla_{\theta_t} \ell(\hat{y}_i, y_i)$ 
1066 26 end
1067 27 return  $\theta_T$ 

```

J QUANTITATIVE ANALYSIS

We provide a detailed qualitative comparison in Table 7, Table 8 and Table 9.

1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

Table 7. Examples with highest and lowest scores retrieved by BM25 for the MMLU validation set.

MMLU Validation Set Examples	
User: Which expression is equivalent to 5×9 ? A. $(5 \times 4) \times (6 \times 5)$ B. $(5 \times 5) + (5 \times 4)$ C. $(5 \times 5) + (5 \times 9)$ D. $(5 \times 9) \times (6 \times 9)$. Answer: B	
Assistant: B	
User: Let A be the set of all ordered pairs of integers (m, n) such that $7m + 12n = 22$. What is the greatest negative number in the set $B = \{m + n : (m, n) \in A\}$? A. -5 B. -4 C. -3 D. -2 Answer: B	
Assistant: B	
User: Ten students take a biology test and receive the following scores: 45, 55, 50, 70, 65, 80, 40, 90, 70, 85. What is the mean of the students' test scores? A. 55 B. 60 C. 62 D. 65 Answer: D	
Assistant: D	
Retrieved by BM25 (Highest similarity)	Retrieved by BM25 (Lowest similarity)
Sample Index: cot 60736	Sample Index: oasst1_41616
Score: 0.9403	Score: 0.0
User: Question: A multiple choice test consists of 4 questions, and each question has 5 answer choices. In how many ways can the test be completed if every question is unanswered? Options: (A) 24 (B) 120 (C) 625 (D) 720 (E) 1024 Answer: 5 choices for each of the 4 questions, thus total of $5 \times 5 \times 5 \times 5 = 5^4 = 625$ ways.	User: ทำไมภาษาลาวคล้ายกับภาษาไทย Assistant: ภาษาลาวและภาษาไทยมีความคล้ายคลึงกันเนื่องจากเป็นภาษาในกลุ่มภาษาไทย-ลาว ซึ่งเป็นกลุ่มภาษาในภูมิภาคอินโดจีน ซึ่งประกอบด้วยภาษาที่มีลักษณะทางการเขียนเหมือนกันและมีวามคล้ายคลึงกันในระบบไวยากรณ์และการออกเสียง
Assistant: The answer is (C).	ารที่ภาษาลาวและภาษาไทยมีความคล้ายคลึงกันเป็นเรื่องธรรมชาติเนื่องจากมีประวัติศาสตร์และวัฒนธรรมที่คล้ายคลึงกัน
Sample Index: cot 60736 (same entry, continued)	โดยทั้งสองภาษามีผลมาจากการได้รับอิทธิพลจากภูมิภาคและประวัติศาสตร์เดียวกัน
Score: 0.9403	นอกจากนี้
User: Question: In how many different ways can the letters of the word "CLAIM" be rearranged? Options: (A) 110 (B) 120 (C) 125 (D) 130 (E) 140 Answer: The total number of arrangements is $5! = 120$.	การมีการติดต่อกันและการแลกเปลี่ยนทางวัฒนธรรมในประวัติศาสตร์ระหว่างประเทศลาวและไทยก็ส่งผลให้ภาษาลาวและภาษาไทยมีความคล้ายคลึงกันในบางด้านของภาษาด้วย
Assistant: The answer is (B).	
Sample Index: cot 60736 (same entry, continued)	Sample Index: oasst1_4573
Score: 0.9403	Score: 0.0
User: Question: A pipe can fill a cistern in 20 minutes whereas the cistern when full can be emptied by a leak in 70 minutes. When both are opened, find when the cistern will be full? Options: (A) 17 minutes (B) 87 minutes (C) 70 minutes (D) 28 minutes (E) 11 minutes Answer: $1/20 - 1/70 = (7 - 2)/140 = 5/140 = 1/28$ 28 minutes.	Question: Buenos días Answer: Hola buenos días. ¿En qué te puedo ayudar?
Assistant: The answer is (D).	
Sample Index: cot 60736 (same entry, continued)	Sample Index: oasst1_42983
Score: 0.9403	Score: 0.0
User: Question: What is the area of a rectangle whose length is twice its width and whose perimeter equals that of a square with area 1? Options: (A) $1/9$ sq units. (B) $4/9$ sq units. (C) $8/9$ sq units. (D) $7/9$ sq units. (E) $6/9$ sq units. Answer: Perimeter of square = 4, so perimeter of rectangle = 4 $\rightarrow 2(l + w) = 4 \rightarrow l + w = 2$. Since $l = 2w \rightarrow 3w = 2 \rightarrow w = 2/3, l = 4/3$. Area = $(4/3) \times (2/3) = 8/9$.	Question: 第一次去深圳旅游, 给我安排一下行程, 大概3天的时间。 Answer: 深圳是中国广东省的一座现代化大城市, 以其高科技、美丽的海滨风光和丰富的购物娱乐场所而闻名。这里为您安排一个3天的深圳旅游行程: [...] (完整3天行程)
Assistant: The answer is (C).	
Sample Index: oasst1_43212	Sample Index: oasst1_43212
Score: 0.0	Score: 0.0
User: 私は猫である名前はまだない	Question: 私に猫である名前はまだない
Assistant: 夏目漱石の「吾輩は猫である」という作品の最初の一節だね。	Answer: 夏目漱石の「吾輩は猫である」という作品の最初の一節だね。

1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187

Table 8. Examples with highest and lowest scores retrieved by LESS (gradient-based) for the MMLU validation set.

MMLU Validation Set Examples	
<p>User: Which expression is equivalent to 5×9? A. $(5 \times 4) \times (6 \times 5)$ B. $(5 \times 5) + (5 \times 4)$ C. $(5 \times 5) + (5 \times 9)$ D. $(5 \times 9) \times (6 \times 9)$. Answer: Assistant: B</p> <p>User: Let A be the set of all ordered pairs of integers (m, n) such that $7m + 12n = 22$. What is the greatest negative number in the set $B = \{m + n : (m, n) \in A\}$? A. -5 B. -4 C. -3 D. -2 Answer: Assistant: B</p> <p>User: Ten students take a biology test and receive the following scores: 45, 55, 50, 70, 65, 80, 40, 90, 70, 85. What is the mean of the students' test scores? A. 55 B. 60 C. 62 D. 65 Answer: Assistant: D</p>	
Retrieved by LESS (Highest similarity)	Retrieved by LESS (Lowest similarity)
<p>Sample Index: flan_v2_78815 Score: 0.9922</p> <p>User: the compound ratio of $2/3$, $6/7$, $4/3$ and $1/8$ is given by? Options: a) $2/21$, b) $1/48$, c) $1/98$, d) $1/96$, e) $1/68$ Assistant: a</p> <p>Sample Index: flan_v2_40013 Score: 1.0</p> <p>User: Look, this is a photo of Mr. White's family. The man in the photo is Mr. White, the father. The woman is the mother. They have two daughters. The girl behind Mr. White is Mary. She's 14. The girl behind Mrs. White is Jane. She is 11. Mary and Jane are in the same school, but not the same grade. Mary is in Grade 2. Jane is in Grade 1. They are good students. How old are Mary and Jane? Options: A) Eleven and thirteen B) Fourteen and eleven C) Eleven D) Thirteen Assistant: B</p> <p>Sample Index: flan_v2_40832 Score: 0.946</p> <p>User: Please answer the following question: Dogs have an understanding of fair play and become angry if they feel that another dog is getting a better deal, a new study has found. The study looked at how dogs react when a companion is rewarded for the same trick in an unequal way. Friederike Range, a researcher at the University of Vienna in Austria, and her colleagues did a series of experiments with dogs who knew how to respond to the command "give the paw". The dogs were normally happy to repeatedly give the paw, whether they got a reward or not. Range found that, compared with monkeys, dogs</p>	<p>Sample Index: flan_v2_81227 Score: 0.0</p> <p>User: Personality: I only sleep 3 hours a day. I have no parents or siblings. I'm incredibly smart. [...] I don't blame you. What is your favorite food? I like well done steak. Candidates 1) Hi, how are you doing? 2) Hi, I am Kathie! I don't like eating fast food. 4) I like Mac N cheese. Assistant: I like Mac N cheese.</p> <p>Sample Index: flan_v2_40281 Score: 0.0187</p> <p>User: Where will a book being read by one person be kept? Options: school room, university, bedside table, pantry, discount store I believe the correct choice is "bedside table", here's why: Assistant: a book being read by one person be kept bedside table.</p> <p>Sample Index: flan_v2_78179 Score: 0.0153</p> <p>User: sbjerg forenede Boldklubber (Efb) is a professional Danish football club from the port city of Esbjerg in West Jutland. The club was formed in 1924, as a merger between Esbjerg Boldklub af 1898 and Esbjerg Amatørklub af 1911. The club has training facilities and stadium at Gl. Vardevej in Esbjerg, and plays in blue and white striped shirts. [...] Subject: morten karlsen, Relation: place_of_birth. Options: A 1955 B central C copenhagen D denmark E esbjerg F farum G island (H) jutland (I) of (J) once (K) randers (L) time (M) uerdingen (N) union (O) viborg (P) zealand Assistant: copenhagen</p>
<p>Options: A care more about whether they are rewarded. B care less about what they are rewarded with. C care more about what they are ordered to do. D care less about who gives them orders. Assistant: B</p>	

1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241

Table 9. Examples with highest and lowest scores retrieved by ADAPT for the MMLU validation set.

MMLU Validation Set Examples	
<p>User: Which expression is equivalent to 5×9? A. $(5 \times 4) \times (6 \times 5)$ B. $(5 \times 5) + (5 \times 4)$ C. $(5 \times 5) + (5 \times 9)$ D. $(5 \times 9) \times (6 \times 9)$. Answer: Assistant: B</p> <p>User: Let A be the set of all ordered pairs of integers (m, n) such that $7m + 12n = 22$. What is the greatest negative number in the set $B = \{m + n : (m, n) \in A\}$? A. -5 B. -4 C. -3 D. -2 Answer: Assistant: B</p> <p>User: Ten students take a biology test and receive the following scores: 45, 55, 50, 70, 65, 80, 40, 90, 70, 85. What is the mean of the students' test scores? A. 55 B. 60 C. 62 D. 65 Answer: Assistant: D</p>	<p style="text-align: center;">Retrieved by ADAPT (Highest similarity)</p> <p>User: Write something based on this summary: canada foreign minister regrets handling of visa denial for haitian prime minister [...] Assistant: the maximum disturbing and unraveling word for an author 's agent to hear is "no." so in the ceremony of book publishing , the relationship that is most culturally challenging and polarizing is the one between writers ' agents and publishers. [...]</p> <p>User: Generate a short summary: an airplane en route to algiers from paris made an unscheduled stop thursday in central france af- ter a passenger reported mistakenly bringing on board a suitcase that belonged to someone else [...] (several summarization tasks) Assistant: paris-algiers flight makes unexpected stop in central france [...] (several summarization answers)</p>
	<p style="text-align: center;">Retrieved by ADAPT (Lowest similarity)</p> <p>User: Vale Assistant: Dime Tio, que necesitas?</p> <p>User: Write a sentence not in English. Assistant: MDN DOM 195 Chef - Service d'examen Vérification Description:</p> <p>User: Kiiltoraunioinen Translate to English Assistant: Asplenium adiantum-nigrum</p>