The Lexical Imprint of Pre-training Data

Jeonghoon Kim[*]
NAVER Cloud & KAIST[†]

Woojin Chung[*‡]
NAVER Cloud

Woomin Song
KAIST

Cheonbok Park
NAVER Cloud & KAIST

Nako Sung
NAVER Cloud

Jinwoo Shin
KAIST

Abstract

How much of zero-shot benchmark accuracy is captured by word-level alignment between a pre-training corpus and the benchmark target? We study this question through word-level unigram cross-entropy (UCE), a tokenizer-agnostic, model-free measure of lexical alignment between a reference corpus and a target benchmark. Likelihood training shapes the model's predictive distribution from the pre-training corpus, and zero-shot scores under likelihood-based evaluation depend in part on the lexical alignment between that corpus and the benchmark target. Cleanly tracing how the lexical imprint of pre-training data scales with corpus–benchmark alignment requires controlled from-scratch pre-training that varies corpus identity while holding the training recipe fixed. Across this controlled sweep, spanning 10 zero-shot benchmarks, 4 pre-training corpora, and 3 model scales, we find that corpus–benchmark lexical alignment, UCE, consistently tracks zero-shot accuracy on every benchmark, with corpora more aligned with the benchmark (lower UCE) achieving higher accuracy. The same signal extends from measurement to control at finer scopes. At the document level, strengthening the imprint by selecting training documents to maximize lexical alignment with the benchmark produces the accuracy gains predicted by the mechanism. At the prompt level, matching the imprint per prompt by routing each prompt to the drafter whose domain corpus is most lexically aligned with it produces measurable throughput gains over a generalist-drafter baseline. Taken together, pre-training corpora leave a measurable lexical imprint on base-model zero-shot behavior, and UCE provides a model-free probe of lexical alignment between reference and target data.

1 Introduction

Modern language models are typically assessed on zero-shot benchmarks, which are constructed around representative problem formulations and treated as quantitative measurements of model capability [1–6]. A quantitative account of how distinct the benchmark distribution is from the pre-training corpus is a natural input to interpreting such accuracy. This relationship has been studied at the level of individual benchmark items and pieces of the lexicon, where term-level frequency of benchmark vocabulary tracks few-shot accuracy [7], test-instance lexical popularity correlates with task performance [8], and corpus-side perplexity signatures predict performance across panels of pretrained models [9]. These post-hoc analyses operate on pretrained checkpoints, and the corpus identity that drives the lexical signal is entangled with training recipe, scale, and training history. Cleanly tracing how the lexical imprint of pre-training data scales with corpus–benchmark

[*]Equal contribution
[†]Korea Advanced Institute of Science and Technology (KAIST)
[‡]Work done during an internship at NAVER Cloud
Correspondence to Jinwoo Shin (jinwoo@kaist.ac.kr) and Jeonghoon Kim (jeonghoon.samuel@gmail.com).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

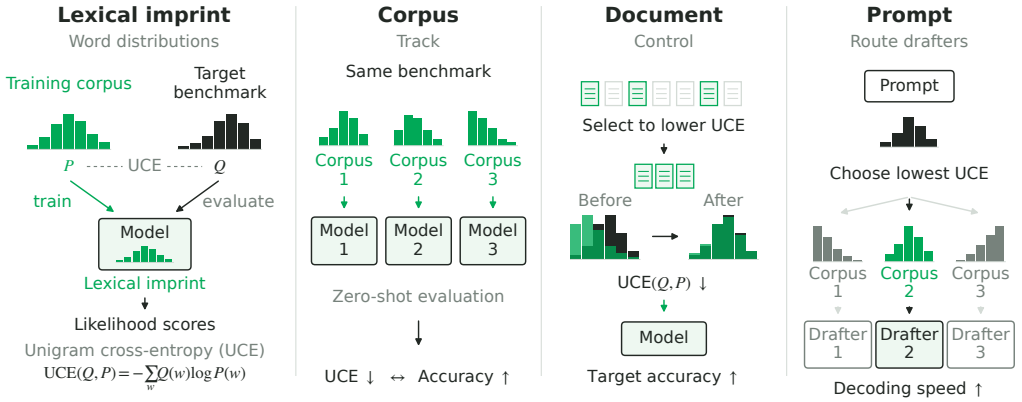


Figure 1: Training on a corpus with word distribution P leaves a lexical imprint on the model, partly visible in likelihood scores on a target benchmark with word distribution Q. Word-level unigram cross-entropy (UCE), a tokenizer-agnostic measure of P–Q alignment, is a proxy for this imprint. Across training corpora, lower UCE goes with higher zero-shot accuracy on the same benchmark. Selecting training documents that lower UCE strengthens the imprint and raises target accuracy. Routing each prompt to the drafter whose corpus has the lowest UCE against it raises decoding speed.

alignment therefore calls for controlled from-scratch pre-training across corpora, a setting in which this alignment remains under-characterized at scale.

Empirical evidence suggests that models achieving lower training loss on high-quality pre-training data tend to attain stronger downstream performance [10–12]. This relationship is consistent with the compression view of language modeling [13–15], under which likelihood-based predictions on benchmark data are expected to be more reliable when training and benchmark distributions are close. Concretely, how much of zero-shot benchmark performance can the corpus–benchmark distributional alignment alone explain? We approach this question information-theoretically, by summarizing both the pre-training corpus and a benchmark target through their word-level unigram distributions and treating their cross-entropy (UCE) as a measure of distributional alignment. We adopt the word as the fundamental unit since words are discrete, semantically meaningful elements of language[4].

Across corpora, benchmark vocabulary is fully covered during pre-training, so zero-shot performance differences are driven by how probability mass is allocated over benchmark-relevant words. The lower unigram cross-entropy reflects a closer alignment of corpus-benchmark word-distribution and predicts higher zero-shot accuracy in all benchmarks we test. This admits a simple interpretation. Likelihood-based pre-training learns a conditional distribution from the training corpus, while zero-shot evaluation performs probabilistic inference using the learned distribution by scoring the conditional likelihoods of benchmark candidates. Corpus-level word-distribution alignment may therefore provide a low-order proxy connecting training data to benchmark behavior.

Contributions. We propose unigram cross-entropy (UCE) as a single corpus-side lens for tracing the lexical imprint of pre-training data on zero-shot behavior, applied at three points of the language-model lifecycle (Figure 1). *(i) At the corpus level.* Across the controlled from-scratch sweep, UCE matches the zero-shot accuracy ordering on every benchmark, with cross-corpus differences captured by how probability mass is allocated to benchmark-relevant words. *(ii) At the document level.* Treating UCE as a per-document objective gives the unigram cross-entropy filter (UCE-filter), which controls document-level lexical alignment under a fixed token budget. Under the same from-scratch grid, alignment-driven selection reproduces the corpus-level UCE–accuracy relationship at the document scope. *(iii) At the prompt level.* In speculative decoding with a pool of corpus-specialized drafters, evaluating UCE between each prompt and each drafter's training corpus produces a third instance of the same functional acting at a finer scope, with measurable throughput gains over a generalist-drafter baseline. Together, through this exploration, lexical alignment between a pre-training corpus and a benchmark target, measured by word-level unigram cross-entropy, provides a measurable trace of the lexical imprint that pre-training data leaves on zero-shot behavior.

[4]We follow the standard lexicographic notion of a word as the smallest meaningful unit of speech that can stand by itself (cf. the Oxford English Dictionary entry for "word").

2 Motivation: From Likelihood Training to Lexical Imprints

The downstream performance of modern pre-trained language models is typically assessed by selecting the least surprising candidate under the learned distribution, that is, choosing the option with the highest conditional likelihood given the input [16][5]. By maximizing next-token likelihood over the training corpus, language-model pre-training estimates empirical conditional patterns, producing a parametric approximation to the data-generating distribution through empirical cross-entropy minimization [13–18]. Accordingly, neural language models can be viewed as parametric conditional estimators analogous to classical n-gram models, as both aim to approximate the underlying conditional distribution despite differing parameterizations.

This view motivates the working hypothesis that the model's predicted conditional distribution is, at least approximately, aligned with that statistically estimated from its training data. This hypothesis is supported by empirical and theoretical findings showing that neural language models often capture low-order n-gram statistics, particularly in early training, and that such n-gram statistics provide a meaningful and relatively tight lower bound on model expressivity [19–21]. Moreover, higher-order n-gram models trained on sufficient data can improve the performance of language models when combined with them [22]. From this perspective, likelihood-based pre-training learns a conditional distribution from the training corpus, while zero-shot evaluation performs probabilistic inference under that distribution by selecting the most likely candidate. Although this structure is infeasible to estimate directly at corpus scale, the corpus-level word-frequency distribution may provide a tractable low-order proxy for the shared dependence between pre-training data and downstream behavior. We refer to this trace as the *lexical imprint* of pre-training data on zero-shot behavior, made measurable through corpus–benchmark word-distribution alignment. This leads to a central research question.

How much of zero-shot benchmark performance is captured by the alignment between corpus and benchmark word distributions?

A quantitative answer would distinguish accuracy gains attributable to corpus alignment from gains reflecting generalization beyond it. Direct estimation of conditional alignment at corpus scale is intractable, so Section 3 operationalizes this question through a tractable word-distribution proxy that bridges the pre-training corpus and the benchmark target.

3 Measuring Lexical Alignment with Unigram Cross-Entropy

Our goal is to empirically evaluate how alignment between the conditional distributions of the training and benchmark data affects language model performance. We first introduce word-level unigram cross-entropy (UCE) as a stable and efficient proxy for measuring this alignment (Section 3.1), and then motivate the use of word-level unigram cross-entropy as the primary metric (Section 3.2).

3.1 Definition of Unigram Cross-Entropy

Conditional distributions and entropy rates are difficult to estimate directly at corpus scale because of finite-sample bias, data sparsity, smoothing sensitivity, and the limited reliability of long-context statistics, so we proxy them with the word-level unigram distribution (Figure 1). The proxy is coarse but computationally efficient, robust to sparsity, and implicitly reflects character-level n-gram structure since individual words contain rich non-i.i.d. character patterns [23].

Let P and Q denote the empirical word-frequency distributions of the pre-training corpus and the benchmark dataset, which we call the *reference distribution* and the *target distribution*, respectively. Word-level unigram cross-entropy (UCE) compares the two datasets solely through these marginals,

$$UCE(Q, P) = -\sum_{w \in \mathcal{V}_b} Q(w)\, \log P(w) = H(Q) + D_{\mathrm{KL}}(Q \,\|\, P)\,,$$

where $\mathcal{V}_b$ is the benchmark vocabulary. Up to the constant $H(Q)$, UCE equals the KL divergence between Q and P, so it provides a clean measure of their overlap. Words are obtained by Unicode NFKC

[5] Zero-shot multiple-choice evaluation in lm-evaluation-harness is fundamentally likelihood-based. Given a question, the model concatenates each candidate answer and selects the one with the lowest language modeling loss.

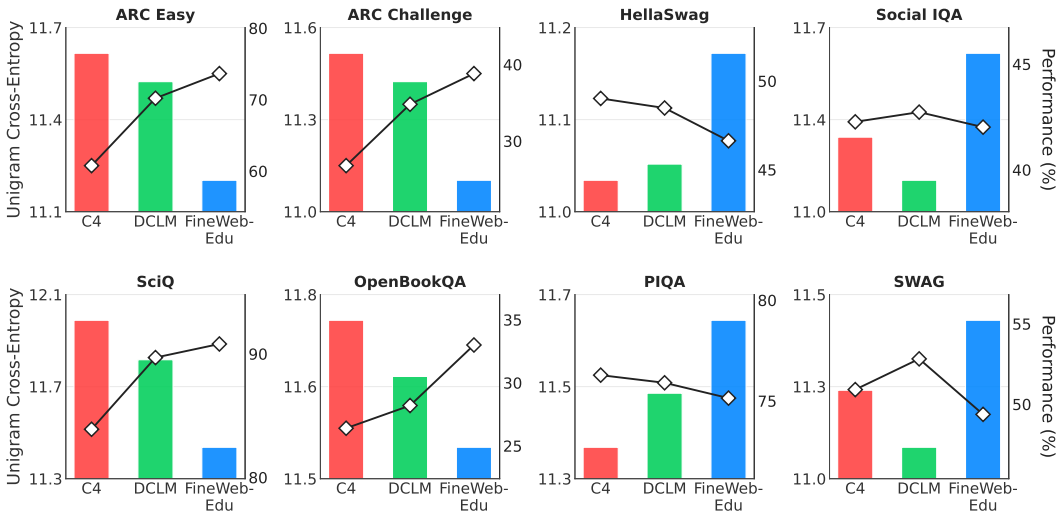


Figure 2: Word-level unigram cross-entropy (left axis, bars, bits ↓) and zero-shot accuracy (right axis, diamonds) across 8 benchmarks for a 3.36B model trained on 60B tokens of **C4**, **DCLM**, or **FineWeb-Edu**. Lower cross-entropy tracks higher accuracy in every panel. Full per-task results across all scales are in Appendix D.4.

normalization, punctuation stripping, and whitespace splitting (no lowercasing), with full implementation details in Appendix A.1 and preprocessing-robustness ablations in Appendix D.3. Appendix D collects further corpus-level analyses, including a comparison against alternative lexical alignment measures (Appendix D.5), the pairwise separation of the corpora themselves (Appendix D.6), and capability families outside the statistic's scope (Appendix D.7).

3.2 Why Words and Unigrams?

We adopt word-level unigram cross-entropy for two reasons. First, subword tokenizers are many-to-one mappings, so token-level cross-entropy conflates genuine distributional mismatch with segmentation artefacts induced by token boundaries that do not align with the true data-generating process [24, 25]. Word-level granularity removes this confound by operating at a unit common to the corpus and the benchmark. Second, although higher-order n-gram cross-entropy for fixed n decomposes into a target-dependent constant and a conditional KL divergence between the two datasets, reliable estimation at corpus scale is hindered by sparsity in the context space, making the estimate sensitive to finite-sample bias and smoothing choices. Word-level unigram cross-entropy instead decomposes exactly into the marginal entropy of the target distribution and the KL divergence between the corresponding word distributions. It therefore avoids the estimation difficulties associated with higher-order conditional statistics. The unigram decomposition is provided in Appendix A.

4 Corpus-Level Lexical Alignment Tracks Zero-Shot Behavior

This section examines whether the word-level unigram cross-entropy (UCE) of Section 3 tracks zero-shot benchmark accuracy at the corpus level, and whether the trend is stable across pre-training corpora, model scales, and random subsamples.

4.1 From-Scratch Pre-training and Evaluation Protocol

For the training setup, we pre-train decoder-only Transformer models [26] from scratch at three scales (400M, 1.33B, 3.36B) on four pre-training corpora spanning a range of data quality, namely the high-quality FineWeb-Edu [27] and DCLM [28], the earlier high-quality C4 [29], and the comparatively lower-quality OpenWebText [30]. From C4, DCLM, and FineWeb-Edu we draw three random sub-samples of 8.5B, 26B, and 60B tokens. The matched (model, subset) pairings $(400\text{M}, 8.5\text{B})$, $(1.33\text{B}, 26\text{B})$, and $(3.36\text{B}, 60\text{B})$ each sit at or near the Chinchilla compute-optimal ratio of 20 tokens

per parameter [31]. OpenWebText is only ~ 9B tokens in total and is therefore evaluated at the 8.5B scale only. All from-scratch runs in this section and in Section 5.2 use AdamW with warmup and cosine learning-rate decay, a sequence length of 2048, and a global batch size of 256, and count tokens with the GPT-2 tokenizer [32]. The two studies differ in peak learning rate, warmup length, β_2, and weight decay (Appendix C, Tables 6–9).

For the evaluation setup, we measure zero-shot performance on a pool of 12 benchmarks (ARC Easy [33], ARC Challenge [33], HellaSwag [34], MMLU [3], SciQ [35], OpenBookQA [36], PIQA [37], LAMBADA [38], SocialIQA [39], SWAG [40], CommonsenseQA [41], and BoolQ [42]). This section uses the 10 benchmarks other than CommonsenseQA and BoolQ, and the UCE-filter study of Section 5.2 uses the 11 other than MMLU. Per-task metrics differ between the two studies and are listed in Appendix C.1. Each corpus is used as released, and Section 7.2 audits all four with a 13-gram index over the 12 benchmarks and tests whether removing the overlapping benchmark items or training documents changes the UCE-filter results.

4.2 Lexical Alignment Matches Corpus-Level Benchmark Rankings

To assess whether corpus–benchmark alignment in word-frequency distributions predicts downstream performance and ranks pre-training corpora accordingly, we measure the UCE between every training subset of Section 4.1 and each of the 10 benchmarks, and compare the corpus ranking by UCE with the ranking by downstream score at each (model, subset) pairing.

For a 3.36B model trained on 60B tokens, Figure 2 shows that UCE predicts the accuracy order of the three corpora on all eight benchmarks. The newer DCLM and FineWeb-Edu, commonly regarded as higher-quality data [27, 28], lead with the lowest UCE on six of them, FineWeb-Edu on the knowledge-heavy ARC Easy/Challenge, SciQ, and OpenBookQA, and DCLM on SocialIQA and SWAG. Their UCE is higher than C4's wherever they trail C4, on HellaSwag and PIQA for both and on SocialIQA and SWAG for FineWeb-Edu. UCE thus tracks the task-dependent overlap that quality reputation misses, and this overlap plays a central role in shaping downstream performance. Because curation and genre also vary across these corpora, the corpus-level UCE–accuracy relationship is associational, and Section 5 tests it within each source corpus at a fixed token budget.

Decomposing the corpus–benchmark overlap shows that all four corpora attain near-saturated coverage of the benchmark vocabulary at our scales and differ in how they allocate probability mass over that vocabulary. On every benchmark in Figure 2, the higher a corpus scores, the smaller the share of benchmark word mass that falls on its long tail of rare words. Appendix D.1 discusses this decomposition, and Appendix D.4 shows that the inverse relationship between performance and UCE holds across the 10 benchmarks, dataset sizes, and model scales.

4.3 The Imprint Is Stable Across Random Subsets

To test whether the inverse relationship depends on the specific subset chosen, we draw five different 8.5B-token subsets each from C4, DCLM, and FineWeb-Edu and one from OpenWebText, then pre-train a 400M model on each.

The inverse relationship holds across the 16 runs in Figure 3 ($R^2 = 0.928$ on ARC Easy and 0.805 on PIQA). Within each corpus the subsets have nearly the same UCE and differ little in accuracy. The fluctuations are consistent with run-to-run noise rather than systematic sensitivity to subset composition. Appendix D.4 gives the per-task results for each subset.

5 Controlling Document-Level Lexical Alignment via Data Selection

Section 4 established that lower unigram cross-entropy (UCE) tracks higher zero-shot accuracy at the corpus level. This section moves one step beyond that descriptive observation and asks what happens when the pre-training corpus is made more lexically aligned with the benchmark target at a fixed token budget. To do so, we propose the UCE-filter, a data-selection tool that controls corpus–benchmark lexical alignment through row-local cross-entropy filtering[6]. Our aim is to use it

[6]A *row* here denotes a single training example, namely one document drawn from the source corpus. We adopt this terminology from the row-oriented format of Common Crawl-derived pre-training corpora such as DCLM [28] and FineWeb-Edu [27], which are distributed with one document per row. The UCE-filter scores and selects at this granularity.

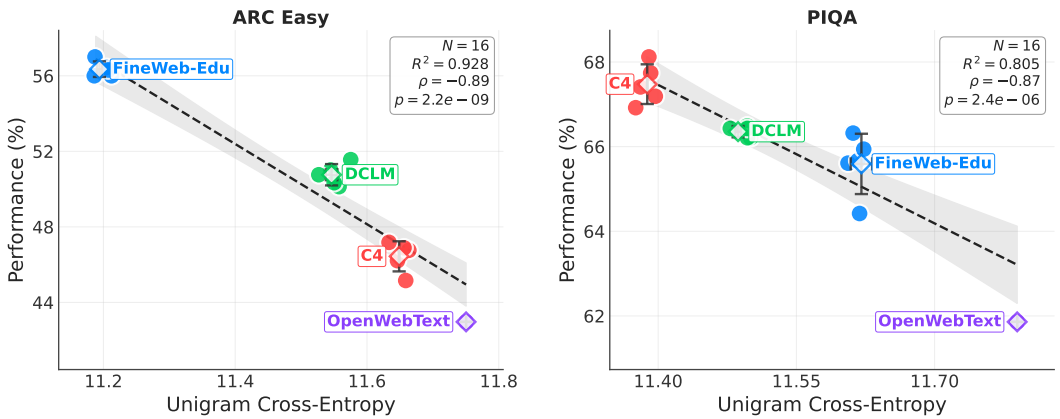


Figure 3: Cross-entropy (bits ↓) versus zero-shot accuracy for N=16 random 8.5B-token subsets at the 400M scale (5 each from **C4**, **DCLM**, and **FineWeb-Edu**, and 1 from **OpenWebText**). Solid dots mark per-subset points, open diamonds with whiskers mark per-corpus mean $\pm 1\sigma$, and the dashed line is the OLS fit with its 95% confidence band.

Algorithm 1 One-shot row selection in the UCE filter

Input: reference distribution P, target distribution Q, training rows $\mathcal{T} = \{r_i\}_{i=1}^{N}$ removal budget K, pre-training corpus counts $c(w)$

Set $B \leftarrow \sum_{w \in \mathcal{V}_b} \mathbf{1}[c(w) = 0]$, Define $P(w) \leftarrow \frac{\max\{c(w),1\}}{C+B}$ for all $w \in \mathcal{V}_b$

for each row $r_i \in \mathcal{T}$ **do**

  Compute $C_i \leftarrow \sum_{w \in \mathcal{V}_b} c_i(w)$ and row-local counts $c_i(w)$

  Define $P^{\setminus r_i}(w) \leftarrow \frac{\max\{c(w)-c_i(w),1\}}{C-C_i+B^{\setminus r_i}}$ where $B^{\setminus r_i} \leftarrow \sum_{w \in \mathcal{V}_b} \mathbf{1}[c(w) - c_i(w) \leq 0]$

  Analytically compute

$$\Delta H_i \leftarrow H(Q, P^{\setminus r_i}) - H(Q, P)$$

end for

Select

$$\mathcal{R} \leftarrow \left\{ r_i \in \mathcal{T} : i \in \arg\operatorname{topK}_{j \in \{1,\ldots,N\}} (-\Delta H_j) \right\}, \quad |\mathcal{R}| = K.$$

Output: $\mathcal{T} \setminus \mathcal{R}$

as a controlled probe that turns the descriptive corpus-level signal of Section 4.2 into a prescriptive selection criterion under a known target, so that the analytical relationship between reference corpus, benchmark target, and language-model behavior can be measured rather than only described. We do not advance the UCE-filter as a stand-alone data-selection method.

5.1 Document-Level Unigram Cross-Entropy Filtering

We define the UCE-filter, a row-local rule that retains the subset of training rows lowering the corpus–benchmark cross-entropy $H(Q, P)$. The row-local update ΔH_r is analytically tractable and rankable in a single linear pass, which is what a controlled probe requires at multi-billion-row scale. For efficiency, we use one-shot ranking which ignores interactions among removed rows. Importance resampling based on hashed n-gram features [43] and the cross-entropy difference filter of Moore and Lewis [44] are the closest existing rules. For each training row r, the UCE-filter measures corpus–benchmark cross-entropy by

$$\Delta H_r = H(Q, P^{\setminus r_i}) - H(Q, P),$$

where Q is the benchmark unigram distribution and P is the reference unigram distribution. The UCE-filter discards the rows with the smallest ΔH_r, namely those whose presence most increases $H(Q, P)$, and retains the rest. Because removing a row only changes the counts of words appearing in that row, ΔH_r admits an analytic row-local update from row-local counts and benchmark-aware corpus counts, without re-estimating the full training distribution. Algorithm 1 summarizes the

Table 1: Corpus–benchmark UCE (bits ↓) of each source corpus (Full) and of the subsets selected from it. Targeted values are measured against the word distribution pooled over the 7 tasks used for selection (ARC-Easy, ARC-Challenge, HellaSwag, SciQ, OpenBookQA, PIQA, SWAG), and Out-of-target values against that of the 4 other tasks (LAMBADA, CommonsenseQA, BoolQ, SocialIQA). The Out-of-target block is measured on a fixed-seed sample of each subset, with Full repeating Random, and its values are comparable only within the block.

	Targeted (7-task)				**Out-of-target (4-task)**			
Source	**Full**	**Random**	**DSIR** [43]	**UCE-Filter**	**Full**	**Random**	**DSIR**	**UCE-Filter**
DCLM	11.17	11.17	11.08	10.92	11.39	11.39	11.43	11.41
FineWeb-Edu	11.32	11.32	11.20	10.93	11.62	11.62	11.65	11.58

procedure. The sign convention, the row-local efficiency argument, and the parallel implementation we use at multi-billion-row scale are deferred to Appendix E.1.

5.2 Raising Lexical Alignment Shifts Accuracy Toward the Target

To test whether a training subset selected for lower UCE against a target raises zero-shot accuracy on that target at a fixed token budget, we evaluate the UCE-filter under the protocol of Section 4.1 on a 2×2 grid of source (DCLM, FineWeb-Edu) and parameter scale (1.34B, 3.33B, counting the tied embedding). The models follow the DataComp-LM architecture patterns [28] and train on 70B tokens from each selected subset, at or above the Chinchilla compute-optimal ratio for both scales [31]. The UCE-filter retains the 20% of rows most aligned with the targeted suite Q aggregated over 7 commonsense tasks, denoted CSR7 (ARC-Easy, ARC-Challenge, HellaSwag, SciQ, OpenBookQA, PIQA, SWAG). A random control alone is insufficient to attribute an accuracy gain specifically to alignment, since any rule that selects toward the benchmark Q could raise accuracy on the targeted suite. We therefore compare against two same-source controls matched to the 70B-token budget, a row-count-matched uniform random sample and a faithful reproduction of the DSIR importance-resampling rule of Xie et al. [43], which targets the same Q through hashed n-gram features. Each (Source, Scale, Method) cell is repeated under three independent training seeds (Appendix E.8). Four out-of-target tasks, denoted OOT-4 (LAMBADA, CommonsenseQA, BoolQ, SocialIQA), test whether a gain is specific to the target, and the Overall average covers all 11 tasks. Appendices C.2 and E.2 give the full setup and the DSIR implementation.

Table 1 shows that the UCE-filter lowers Targeted UCE well below both controls, and DSIR lowers it by less. These Targeted gaps far exceed selection noise, since the UCE-filter is deterministic and the Targeted UCE of Random and DSIR varies by less than 0.001 bits across selection seeds (Appendix E.8). The UCE-filter and DSIR subsets have out-of-target UCE values close to those of the random control, suggesting that selection mainly lowers UCE for the specified target. The relationship of Section 4.2 also holds at the document scope, where within each source the lowest-UCE subset has the highest Targeted accuracy (Table 2). Like the alignment, the UCE-filter's accuracy gain concentrates on the targeted suite, with the Targeted average above the random control in every (Source, Scale) cell and the out-of-target average close to it. Given the same Q, DSIR gains less on the targeted suite and falls below the random control on the out-of-target average in every cell. Selecting documents for lower UCE against a target thus raises zero-shot accuracy mainly on that target, so the lexical imprint that Section 4 observes across corpora also appears when alignment is changed within each source corpus at a fixed token budget. Because Q is estimated from the target tasks, the selection gains should be interpreted as an upper bound on the alignment effect under a known target rather than as zero-shot generalization to unseen targets. Appendix E.2 and Table 32 attribute much of DSIR's Targeted shortfall against the UCE-filter, and of its LAMBADA loss, to a length bias in its document scores. BETR [45], an embedding-similarity selector given the same CSR7 targets, raises Targeted accuracy at the cost of the out-of-target average while its Targeted UCE stays at the DSIR level, and it trails the UCE-filter on the Targeted average except at 1.34B on DCLM (Appendix E.3). Appendix E.6 shows that the UCE gap between the UCE-filter and random subsets is unchanged when measured against a Q built from the train splits alone, without the test splits it otherwise pools. The UCE-filter's Targeted gain over the random control survives contamination removal at both ends. At a matched training seed, dropping the evaluation items flagged by the 13-gram scan moves it by at most 0.05 pp in every cell, and removing flagged DCLM

Table 2: Zero-shot accuracy of 1.34B and 3.33B models trained from scratch on 70B tokens of each selected subset, as mean ± standard deviation over three training seeds. Targeted averages the 7 tasks used for selection (ARC-Easy, ARC-Challenge, HellaSwag, SciQ, OpenBookQA, PIQA, SWAG), and Out-of-target the 4 other tasks (LAMBADA, CommonsenseQA, BoolQ, SocialIQA).

		Targeted (7-task) avg. (%)			**Out-of-target (4-task) avg. (%)**		
Source	**Model**	**Random**	**DSIR** [43]	**UCE-Filter**	**Random**	**DSIR**	**UCE-Filter**
DCLM	1.34B	58.85 ± 0.56	59.85 ± 0.53	62.49 ± 0.26	46.57 ± 0.53	43.79 ± 0.23	46.22 ± 0.94
	3.33B	62.95 ± 0.03	63.01 ± 0.33	66.26 ± 0.20	48.75 ± 0.53	46.23 ± 0.38	49.56 ± 0.09
FineWeb-Edu	1.34B	59.12 ± 0.41	60.15 ± 0.08	61.85 ± 0.15	42.60 ± 1.06	41.14 ± 1.37	43.40 ± 0.29
	3.33B	62.80 ± 0.09	62.77 ± 0.18	64.64 ± 0.21	44.85 ± 0.49	43.39 ± 0.33	45.61 ± 0.27

documents before re-selection and retraining at 1.34B keeps 3.40 pp of the same-seed 4.13 pp gain (Appendix F). On the Targeted average, selection-seed and training-seed accuracy variation are of the same order, and the method ordering of Table 2 holds at every selection seed (Appendix E.9). Under 5-shot prompting the UCE-filter has the highest Targeted and out-of-target means in every cell (Appendix G.1). Appendix E.4 gives results for a conditional higher-order variant of the filter, and Appendix G gives the per-task zero-shot accuracies.

5.3 30B Stage-2 Production-Scale Case Study

To test whether the alignment signal isolates a subset of a heavily curated pre-training mixture that accounts for most of its gain, we apply the UCE-filter in the pre-training of an in-house 30B-parameter language model that largely follows the multi-stage pre-training recipe and architecture of Nemotron-3 Nano [46]. The model's Stage 2 continues from the Stage 1 endpoint on a 2T-token production mixture, and the filtered variant replaces the web component of that mixture with its UCE-filtered subset, keeps the non-web components and the production hyperparameters, and trains on 200B tokens in total. This is a single-point case study, with architecture, mixture, and caveats in Appendix E.7.

In Table 3, the UCE-filtered web variant keeps the bulk of the Stage 2 gain at roughly $10\times$ fewer tokens, with the production mixture's MMLU-Pro matched and most of the Stage $1 \to 2$ MMLU gap closed. The MMLU shortfall is consistent with the web-only nature of the filter, since the shorter run also sees less of the code, math, and supervised data, which the filter leaves untouched. The pattern is one of isolation rather than improvement, and we do not claim general improvement over a heavily curated full mix.

6 Inference-Time Routing via Prompt-Level Lexical Alignment

To examine whether lexical alignment continues to act at the prompt scope, we apply the same functional per prompt in speculative decoding, which drafts candidate tokens with a fast drafter model and verifies them in parallel against a large target model [47, 48]. Given a verifier M_v (DeepSeek-R1-Distill-Llama-8B [49]) and a pool of K corpus-specialized drafters $\{M_d^{(k)}\}_{k=1}^{K}$ with corpus word distributions $\{P_d^{(k)}\}$, we route each prompt x (with empirical word distribution Q_x) to the drafter $M_d^{(k^\star)}$ where $k^\star = \arg\min_k H(Q_x, P_d^{(k)})$, reusing the unigram cross-entropy (UCE). The $P_d^{(k)}$ are precomputed once at the corpus level, so per-prompt routing reduces to a sparse dot product with negligible latency. Specifically, we construct three specialist drafters using the OpenThoughts-114k [50] dataset, each corresponding to a sub-domain (math, code, and science). As a baseline, we also construct a generalist drafter, using equal-sized subsets from each specialist drafter's training corpus. We use SAM [51] as the drafter architecture, and all drafters are constructed using the same corpus size for fair comparison. We evaluate the speculative decoding performance on four benchmarks spanning math (AIME [52]), code (LiveCodeBench [53]), general knowledge (MMLU [3]), and medical question answering (MedQA [54]).

In Table 4, UCE-based routing adds 5.3 pp of average speedup over the generalist drafter and raises both τ and speedup on every benchmark, including MMLU and MedQA, which no specialist targets.

Table 3: Zero-shot MMLU and MMLU-Pro accuracy of a 30B-parameter model after Stage 1 and after Stage 2, where Stage 2 uses either the production mixture or the same mixture with the web component replaced by its UCE-filtered subset. Tokens is the number of Stage 2 training tokens.

Stage	Strategy	Tokens	MMLU	MMLU-Pro
Stage 1	Endpoint (= Stage 2 init)	init.	71.17	44.46
Stage 2	Production Mix (baseline)	$\sim$2T	75.61	48.64
	UCE-Filter Web Subset	**200**B	74.28	48.81

Table 4: Mean acceptance length τ and per-token decoding speedup of speculative decoding with DeepSeek-R1-Distill-Llama-8B as the verifier, using a generalist drafter or UCE-based routing among specialist drafters. Speedup is relative to the verifier alone (Base Model).

	AIME		LiveCodeBench		MMLU		MedQA		Average	
	τ	Speedup	τ	Speedup	τ	Speedup	τ	Speedup	τ	Speedup
Base Model	1.000	-	1.000	-	1.000	-	1.000	-	1.000	-
Generalist Drafter	1.702	55.6%	1.587	53.0%	1.436	45.7%	1.409	45.7%	1.534	50.0%
UCE-Based Routing	**1.707**	**61.6%**	**1.615**	**54.7%**	**1.451**	**55.3%**	**1.426**	**49.6%**	**1.550**	**55.3%**

The same alignment functional that controls corpus identity at the document scope therefore also acts at the prompt scope, making the lexical imprint visible.

7 Discussion and Limitations

7.1 One Alignment Functional Across Corpora, Documents, and Prompts

Maximizing next-token likelihood minimizes KL divergence to the empirical training distribution. To first order, this aligns the model's word predictions on $\mathcal{V}_b$ with the corpus word distribution, while higher-order conditional structure is learned later and only partially, especially on the long tail [19, 21, 55]. The descriptive result of Section 4.2 is therefore a structural consequence of this ordering and reflects the optimization geometry shared by likelihood training and likelihood scoring rather than properties specific to the scales or corpora we examine. These results suggest that likelihood-based zero-shot performance partly reflects corpus–benchmark lexical alignment, alongside the capabilities each benchmark is intended to assess.

The three uses of UCE in this paper are three instantiations of the same $D_{\mathrm{KL}}(Q \,\|\, P)$ functional at different scopes. Corpus-level ranking (Section 4) evaluates D_{KL} once per (corpus, benchmark) pair to compare web-scale training corpora. Row-local data selection (Section 5) evaluates the marginal contribution of each row to D_{KL} and removes the rows whose removal most lowers it. Per-prompt drafter routing (Section 6) evaluates D_{KL} between each prompt's word distribution and each drafter's corpus word distribution, and assigns the prompt to the drafter with the lowest D_{KL}.

7.2 Contamination as an Alternative Explanation

Benchmark text inside a pre-training corpus could raise lexical alignment and accuracy together. We audit samples of all four corpora with a GPT-3-style word-level 13-gram index [56] (Table 5). Nine of the 12 benchmarks have under 0.3% of items matched in every sample, and the three exceptions, BoolQ, SciQ, and MMLU, peak on FineWeb-Edu. On SWAG, SocialIQA, and LAMBADA, where at most 0.01% of items match on C4, DCLM, or FineWeb-Edu, a corpus with lower UCE still scores better, so contamination is unlikely to explain the relationship there. Among the three exceptions, BoolQ supplies its passage at evaluation time, so a match on that passage cannot carry the yes or no label. The audit does not rule out a contribution on SciQ and MMLU, where FineWeb-Edu has both the most matched items and the highest score.

In the selection study, the UCE-filter's gain over the random control survives contamination removal at both ends (Appendix F). The training-end removal drops 0.13% of DCLM rows, a re-scan of one-million-document samples of each re-selected subset finds no remaining match, and each subset's UCE moves by at most 0.005 bits. After retraining, the UCE-filter retains a 3.40 pp CSR7 advantage

Table 5: Share of benchmark items (%, all splits) with at least one word-level 13-gram found in a sample of up to one million documents per corpus.

	ARC-Challenge	ARC-Easy	BoolQ	CommonsenseQA	HellaSwag	LAMBADA	MMLU	OpenBookQA	PIQA	SciQ	SocialIQA	SWAG
DCLM	0.08	0.02	3.07	0.00	0.11	0.00	0.64	0.00	0.12	0.64	0.00	0.00
FineWeb-Edu	0.04	0.06	11.25	0.00	0.29	0.00	1.39	0.02	0.12	4.91	0.00	0.00
C4	0.00	0.02	3.14	0.00	0.12	0.00	0.37	0.02	0.07	0.25	0.01	0.00
OpenWebText	0.00	0.02	5.35	0.00	0.02	0.06	0.58	0.00	0.02	0.15	0.00	0.00

over the random control, compared with 4.13 pp before removal (Table 31). At the evaluation end, 161 and 163 of $36{,}934$ CSR7 items are flagged for 13-gram overlap with DCLM and FineWeb-Edu, respectively. Benchmark memorization alone is therefore unlikely to explain the gain, although the retraining check covers only DCLM at 1.34B.

7.3 What Unigram Cross-Entropy Captures and Where It Fails

UCE measures only the alignment of word distributions. It is by construction blind to conditional structure such as syntax, multi-token composition, and factual co-occurrence, so UCE is best understood as a lower-bound proxy for distributional alignment rather than a sufficient summary of corpus quality or a fundamental property of the models. UCE does not order grammar (BLiMP [57]), arithmetic (MathQA [58]), or non-English (Global-PIQA [59]) benchmarks (Appendix D.7). Whitespace tokenization is a second boundary, since UCE applies most directly to whitespace-segmented languages and leaves non-segmented scripts (e.g., Chinese, Japanese) and highly agglutinative morphologies for future work.

Apart from the routing study, the 30B case study, and 5-shot checks, every claim concerns base-model zero-shot behavior under a controlled from-scratch protocol, and the selection gains are an upper bound under a known target (Section 5.2). Two directions follow directly from our design, a paraphrase protocol that tests how far the observed relationship depends on exact lexical realization, and a lexical decomposition that restricts UCE to a benchmark-characteristic vocabulary. Whether the lexical imprint persists through SFT and RLHF, which differ from pre-training in data volume, training objective, and what is measured, requires a controlled study built for those objectives. A general account of what makes a model generalize would need a design that varies the other properties of pre-training data along with this lexical one.

8 Conclusion

Pre-trained language-model behavior is shaped by many intertwined factors, including higher-order conditional structure, scale, and post-training reshaping. This paper takes up one of them, the lexical imprint that the pre-training corpus leaves on zero-shot behavior, as captured by word-level unigram cross-entropy. Its descriptive force comes from a single mechanism. Likelihood training allocates probability mass according to the pre-training corpus, and likelihood-based zero-shot evaluation partly reflects the lexical alignment between the pre-training corpus and the benchmark target.

This descriptive signal becomes the same statistic at finer scopes. Applied as document-level data selection, it controls the lexical imprint, and this control shows it more clearly than comparing corpora can, since accuracy rises mainly on the target whose alignment is raised. Applied as prompt-level drafter routing, it raises speculative-decoding throughput. Across these uses, the analytic relationship between pre-training corpus, benchmark target, and language-model behavior is measured rather than only described. The same logic marks the limits of the account. UCE ignores syntax, multi-token composition and the effects of post-training. It makes one part of the imprint measurable, namely how the lexical statistics of pre-training data align with those of the target data.

We close with:

> *Pre-training pulls a model's predictive distribution toward its corpus's empirical conditional patterns, leaving a measurable lexical imprint on zero-shot behavior.*

Acknowledgments

This work was partially supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Program(KAIST) and No. RS-2022-II220713, Meta-learning Applicable to Real-world Problems).

References

[1] Micah Goldblum, Marc Anton Finzi, Keefer Rowan, and Andrew Gordon Wilson. Position: The no free lunch theorem, Kolmogorov complexity, and the role of inductive biases in machine learning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=EaJ7nqJ2Fa.

[2] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=BH8TYy0r6u.

[3] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

[4] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

[5] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Daron Anderson, Tung Nguyen, Mobeen Mahmood, Fiona Feng, Steven Y. Feng, Haoran Zhao, Michael Yu, Varun Gangal, Chelsea Zou, Zihan Wang, Jessica P. Wang, Pawan Kumar, Oleksandr Pokutnyi, Robert Gerbicz, Serguei Popov, John-Clark Levin, Mstyslav Kazakov, Johannes Schmitt, Geoff Galgon, Alvaro Sanchez, Yongki Lee, Will Yeadon, Scott Sauers, Marc Roth, Chidozie Agu, Søren Riis, Fabian Giska, Saiteja Utpala, Zachary Giboney, Gashaw M. Goshu, Joan of Arc Xavier, Sarah-Jane Crowson, Mohinder Maheshbhai Naiya, Noah Burns, Lennart Finke, Zerui Cheng, Hyunwoo Park, Francesco Fournier-Facio, John Wydallis, Mark Nandor, Ankit Singh, Tim Gehrunger, Jiaqi Cai, Ben McCarty, Darling Duclosel, Jungbae Nam, Jennifer Zampese, Ryan G. Hoerr, Aras Bacho, Gautier Abou Loume, Abdallah Galal, Hangrui Cao, Alexis C. Garretson, Damien Sileo, Qiuyu Ren, Doru Cojoc, Pavel Arkhipov, Usman Qazi, Lianghui Li, Sumeet Motwani, Christian Schröder de Witt, Edwin Taylor, Johannes Veith, Eric Singer, Taylor D. Hartman, Paolo Rissone, Jaehyeok Jin, Jack Wei Lun Shi, Chris G. Willcocks, Joshua Robinson, Aleksandar Mikov, Ameya Prabhu, Longke Tang, Xavier Alapont, Justine Leon Uro, Kevin Zhou, Emily de Oliveira Santos, Andrey Pupasov Maksimov, Edward Vendrow, Kengo Zenitani, Julien Guillod, Yuqi Li, Joshua Vendrow, Vladyslav Kuchkin, and Ng Ze-An. Humanity's last exam. *CoRR*, abs/2501.14249, 2025. doi: 10.48550/ARXIV.2501.14249. URL https://doi.org/10.48550/arXiv.2501.14249.

[6] Dan Hendrycks, Dawn Song, Christian Szegedy, Honglak Lee, Yarin Gal, Erik Brynjolfsson, Sharon Li, Andy Zou, Lionel Levine, Bo Han, Jie Fu, Ziwei Liu, Jinwoo Shin, Kimin Lee, Mantas Mazeika, Long Phan, George Ingebretsen, Adam Khoja, Cihang Xie, Olawale Salaudeen, Matthias Hein, Kevin Zhao, Alexander Pan, David Duvenaud, Bo Li, Steve Omohundro, Gabriel Alfour, Max Tegmark, Kevin McGrew, Gary Marcus, Jaan Tallinn, Eric Schmidt, and Yoshua Bengio. A definition of AGI. *CoRR*, abs/2510.18212, 2025. doi: 10.48550/ARXIV.2510.18212. URL https://doi.org/10.48550/arXiv.2510.18212.

[7] Yasaman Razeghi, Robert L. Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot numerical reasoning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Findings of ACL, pages 840–854. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-EMNLP.59. URL https://doi.org/10.18653/v1/2022.findings-emnlp.59.

[8] Elena Sofia Ruzzetti, Fabio Massimo Zanzotto, and Tommaso Caselli. Lexical popularity: Quantifying the impact of pre-training for LLM performance. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), EACL 2026*, pages 1209–1230. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.55. URL https://aclanthology.org/2026.eacl-long.55/.

[9] Siyang Wu, Honglin Bao, Sida Li, Ari Holtzman, and James Evans. Mapping overlaps in benchmarks through perplexity in the wild. In *The Fourteenth International Conference on Learning Representations, ICLR 2026*. OpenReview.net, 2026. URL https://arxiv.org/abs/2509.23488.

[10] Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Luca Soldaini, Jenia Jitsev, Alex Dimakis, Gabriel Ilharco, Pang Wei Koh, Shuran Song, Thomas Kollar, and et al. Language models scale reliably with overtraining and on downstream tasks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=iZeQBqJamf.

[11] Jeonghoon Kim, Byeongchan Lee, Cheonbok Park, Yeontaek Oh, Beomjun Kim, Taehwan Yoo, Seongjin Shin, Dongyoon Han, Jinwoo Shin, and Kang Min Yoo. Peri-LN: Revisiting normalization layer in the transformer architecture. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=ci1S6wmXfO.

[12] Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. Spike no more: Stabilizing the pre-training of large language models. *CoRR*, abs/2312.16903, 2023. doi: 10.48550/ARXIV.2312.16903. URL https://doi.org/10.48550/arXiv.2312.16903.

[13] Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=jznbgiynus.

[14] Yuzhen Huang, Jinghan Zhang, Zifei Shan, and Junxian He. Compression represents intelligence linearly. *CoRR*, abs/2404.09937, 2024. doi: 10.48550/ARXIV.2404.09937. URL https://doi.org/10.48550/arXiv.2404.09937.

[15] David Heurtel-Depeiges, Anian Ruoss, Joel Veness, and Tim Genewein. Compression via pre-trained transformers: A study on byte-level multimodal data. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=c5R0lKLIr3.

[16] Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models. *CoRR*, abs/2405.14782, 2024. doi: 10.48550/ARXIV.2405.14782. URL https://doi.org/10.48550/arXiv.2405.14782.

[17] Woojin Chung and Jeonghoon Kim. Exploiting vocabulary frequency imbalance in language model pre-training. In *Advances in Neural Information Processing Systems 39: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2 - 7, 2025*, 2025. doi: 10.48550/ARXIV.2508.15390. URL https://doi.org/10.48550/arXiv.2508.15390.

[18] Ori Yoran, Kunhao Zheng, Fabian Gloeckle, Jonas Gehring, Gabriel Synnaeve, and Taco Cohen. The kolmogorov test: Compression by code generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=C45YqeBDUM.

[19] Timothy Nguyen. Understanding transformers via n-gram statistics. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/b1c446eebd9a317dd0e96b16908c821a-Abstract-Conference.html.

[20] Anej Svete and Ryan Cotterell. Transformers can represent n-gram language models. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 6845–6881. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.381. URL https://doi.org/10.18653/v1/2024.naacl-long.381.

[21] Xuyang Ge, Wentao Shu, Jiaxing Wu, Yunhua Zhou, Zhengfu He, and Xipeng Qiu. Evolution of concepts in language model pre-training. *CoRR*, abs/2509.17196, 2025. doi: 10.48550/ARXIV.2509.17196. URL https://doi.org/10.48550/arXiv.2509.17196.

[22] Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. *CoRR*, abs/2401.17377, 2024. doi: 10.48550/ARXIV.2401.17377. URL https://doi.org/10.48550/arXiv.2401.17377.

[23] Nived Rajaraman, Jiantao Jiao, and Kannan Ramchandran. An analysis of tokenization: Transformers under Markov data. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/724afcaae4ae92a9220a077ffe80088d-Abstract-Conference.html.

[24] Buu Phan, Marton Havasi, Matthew J. Muckley, and Karen Ullrich. Understanding and mitigating tokenization bias in language models. *CoRR*, abs/2406.16829, 2024. doi: 10.48550/ARXIV.2406.16829. URL https://doi.org/10.48550/arXiv.2406.16829.

[25] Pietro Lesci, Clara Meister, Thomas Hofmann, Andreas Vlachos, and Tiago Pimentel. Causal estimation of tokenisation bias. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 28325–28340. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.1374/.

[26] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta,

Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

[27] Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin A. Raffel, Leandro von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

[28] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Kumar Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee F. Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah M. Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Raghavi Chandu, Thao Nguyen, Igor Vasiljevic, Sham M. Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models. *CoRR*, abs/2406.11794, 2024. doi: 10.48550/ARXIV.2406.11794. URL https://doi.org/10.48550/arXiv.2406.11794.

[29] Jesse Dodge, Maarten Sap, Ana Marasovic, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1286–1305. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.98. URL https://doi.org/10.18653/v1/2021.emnlp-main.98.

[30] Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

[31] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022. URL https://arxiv.org/abs/2203.15556.

[32] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. URL https://api.semanticscholar.org/CorpusID:160025533. OpenAI technical report.

[33] 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. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

[34] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics,*

ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

[35] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin, editors, *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 94–106. Association for Computational Linguistics, 2017. doi: 10.18653/V1/W17-4413. URL https://doi.org/10.18653/v1/w17-4413.

[36] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1260. URL https://doi.org/10.18653/v1/d18-1260.

[37] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6239. URL https://doi.org/10.1609/aaai.v34i05.6239.

[38] 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. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/V1/P16-1144. URL https://doi.org/10.18653/v1/p16-1144.

[39] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4462–4472. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1454. URL https://doi.org/10.18653/v1/D19-1454.

[40] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 93–104. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1009. URL https://doi.org/10.18653/v1/d18-1009.

[41] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1421. URL https://doi.org/10.18653/v1/n19-1421.

[42] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, 2019. URL https://arxiv.org/abs/1905.10044.

[43] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6b9aa8f418bde2840d5f4ab7a02f663b-Abstract-Conference.html.

[44] Robert C. Moore and William D. Lewis. Intelligent selection of language model training data. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden, Short Papers*, pages 220–224. The Association for Computer Linguistics, 2010. URL https://aclanthology.org/P10-2041/.

[45] David Mizrahi, Anders Boesen Lindbo Larsen, Jesse Allardice, Suzie Petryk, Yuri Gorokhov, Jeffrey Li, Alex Fang, Josh Gardner, and Tom Gunter. Language models improve when pretraining data matches target tasks. *CoRR*, abs/2507.12466, 2025. URL https://arxiv.org/abs/2507.12466.

[46] NVIDIA. Nemotron 3 Nano: Open, efficient mixture-of-experts hybrid Mamba-Transformer model for Agentic reasoning, 2025. URL https://arxiv.org/abs/2512.20848. Technical report.

[47] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

[48] Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *CoRR*, abs/2302.01318, 2023. doi: 10.48550/ARXIV.2302.01318. URL https://doi.org/10.48550/arXiv.2302.01318.

[49] Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

[50] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models, 2025. URL https://arxiv.org/abs/2506.04178.

[51] Yuxuan Hu, Ke Wang, Xiaokang Zhang, Fanjin Zhang, Cuiping Li, Hong Chen, and Jing Zhang. SAM decoding: Speculative decoding via suffix automaton. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12187–12204, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.595. URL https://aclanthology.org/2025.acl-long.595/.

[52] Mathematical Association of America. AIME: American Invitational Mathematics Examination, 2024. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

[53] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=chfJJYC3iL.

[54] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421. URL https://www.mdpi.com/2076-3417/11/14/6421.

[55] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 15696–15707. PMLR, 2023. URL https://proceedings.mlr.press/v202/kandpal23a.html.

[56] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[57] Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. Blimp: The benchmark of linguistic minimal pairs for english. *Trans. Assoc. Comput. Linguistics*, 8:377–392, 2020. doi: 10.1162/TACL_A_00321. URL https://doi.org/10.1162/tacl_a_00321.

[58] Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms, 2019. URL https://arxiv.org/abs/1905.13319.

[59] Tyler A. Chang and Catherine Arnett. Global PIQA: evaluating physical commonsense reasoning across 100+ languages and cultures. *CoRR*, abs/2510.24081, 2025. doi: 10.48550/ARXIV.2510.24081. URL https://doi.org/10.48550/arXiv.2510.24081.

[60] Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. Generalization v.s. memorization: Tracing language models' capabilities back to pretraining data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=IQxBDLmVpT.

[61] Dietrich Klakow. Selecting articles from the language model training corpus. In *IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP 2000, 5-9 June, 2000, Hilton Hotel and Convention Center, Istanbul, Turkey*, pages 1695–1698. IEEE, 2000. doi: 10.1109/ICASSP.2000.862077. URL https://doi.org/10.1109/ICASSP.2000.862077.

[62] Amittai Axelrod. Cynical selection of language model training data. *CoRR*, abs/1709.02279, 2017. URL http://arxiv.org/abs/1709.02279.

[63] Yukun Feng, Patrick Xia, Benjamin Van Durme, and João Sedoc. Automatic document selection for efficient encoder pretraining. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 9522–9530. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.647. URL https://doi.org/10.18653/v1/2022.emnlp-main.647.

[64] Tristan Thrush, Christopher Potts, and Tatsunori Hashimoto. Improving pretraining data using perplexity correlations. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=huuKoVQnB0.

[65] Ian Magnusson, Nguyen Tai, Ben Bogin, David Heineman, Jena D. Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, Noah A. Smith, Pang Wei Koh, and Jesse Dodge. Datadecide: How to predict best pretraining data with small experiments. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=p9YlQPF8fE.

[66] Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Findings of ACL, pages 10776–10787. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.722. URL https://doi.org/10.18653/v1/2023.findings-emnlp.722.

[67] Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Evan Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hannaneh Hajishirzi, Noah A. Smith, and Jesse Dodge. What's in my big data? In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=RvfPnOkPV4.

[68] Yunlong Hou, Fengzhuo Zhang, Cunxiao Du, Xuan Zhang, Jiachun Pan, Tianyu Pang, Chao Du, Vincent Y. F. Tan, and Zhuoran Yang. BanditSpec: Adaptive speculative decoding via bandit algorithms. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/hou25h.html.

[69] Hang Wu, Jianian Zhu, Yinghui Li, Haojie Wang, Biao Hou, and Jidong Zhai. SpecRouter: Adaptive routing for multi-level speculative decoding in large language models. *CoRR*, abs/2505.07680, 2025. doi: 10.48550/ARXIV.2505.07680. URL https://doi.org/10.48550/arXiv.2505.07680.

[70] Fenglu Hong, Ravi Raju, Jonathan Lingjie Li, Bo Li, Urmish Thakker, Avinash Ravichandran, Swayambhoo Jain, and Changran Hu. Training domain draft models for speculative decoding: Best practices and insights. *CoRR*, abs/2503.07807, 2025. URL https://arxiv.org/abs/2503.07807.

[71] Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D. Lee, and Di He. REST: Retrieval-based speculative decoding. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1582–1595. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.88. URL https://doi.org/10.18653/v1/2024.naacl-long.88.

[72] Woomin Song, Saket Dingliwal, Sai Muralidhar Jayanthi, Bhavana Ganesh, Jinwoo Shin, Aram Galstyan, and Sravan Babu Bodapati. Accelerated test-time scaling with model-free speculative sampling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025*, 2025. URL https://arxiv.org/abs/2506.04708. Oral presentation.

[73] Bettina Messmer, Vinko Sabolcec, and Martin Jaggi. Enhancing multilingual llm pretraining with model-based data selection. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025.

[74] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Appendix

A Unigram Cross-Entropy Measures Marginal Mismatch

Let A and B denote two text corpora modeled as realizations of stationary and ergodic sequence distributions P_A and P_B on $\mathcal{V}^{\mathbb{N}}$, where $\mathcal{V}$ is the vocabulary. Define the stationary single-token marginals

$$p_A(x) := P_A(X_k = x), \qquad p_B(x) := P_B(X_k = x), \qquad x \in \mathcal{V}. \tag{1}$$

By stationarity, these marginals do not depend on k. When $\mathcal{V}$ is finite, p_A and p_B are categorical distributions over the vocabulary.

The unigram cross-entropy of the target marginal p_B relative to the reference marginal p_A is

$$\begin{aligned} H(p_B, p_A) &:= \mathbb{E}_{X \sim p_B}\left[-\log p_A(X)\right] \\ &= -\sum_{x \in \mathcal{V}} p_B(x) \log p_A(x). \end{aligned} \tag{2}$$

We assume that p_B is absolutely continuous with respect to p_A, i.e.,

$$p_B(x) > 0 \quad \Longrightarrow \quad p_A(x) > 0. \tag{3}$$

Otherwise, both the cross-entropy and the corresponding KL divergence are infinite. In finite-sample computation, this support condition can be enforced through smoothing.

Adding and subtracting $-\sum_{x \in \mathcal{V}} p_B(x) \log p_B(x)$ yields

$$H(p_B, p_A) = H(p_B) + D_{\mathrm{KL}}(p_B \| p_A), \tag{4}$$

where

$$H(p_B) := -\sum_{x \in \mathcal{V}} p_B(x) \log p_B(x) \tag{5}$$

is the marginal entropy of the target distribution, and

$$D_{\mathrm{KL}}(p_B \| p_A) := \sum_{x \in \mathcal{V}} p_B(x) \log \frac{p_B(x)}{p_A(x)} \tag{6}$$

is the KL divergence from the target marginal to the reference marginal. For a fixed target corpus B, the marginal entropy $H(p_B)$ is constant across reference corpora A. Ranking reference corpora by unigram cross-entropy is therefore equivalent to ranking them by $D_{\mathrm{KL}}(p_B \| p_A)$. Across the eleven evaluation benchmarks, however, $H(p_B)$ itself spans 1.86 bits, so absolute UCE values are not comparable between benchmarks and every comparison in this paper is within a benchmark across corpora.

A.1 UCE computation details

All cross-entropy and KL divergence values in the paper, including the UCE defined in Section 3.1, are reported in bits, with logarithms taken base 2.

Tokenization. Words are obtained by Unicode NFKC normalization of the raw text, followed by punctuation stripping and whitespace splitting. We do not lowercase, since case-bearing words such as proper nouns carry distributional information relevant to many benchmarks. The benchmark vocabulary $\mathcal{V}_b$ is the set of distinct words obtained from the benchmark dataset under the same procedure, and word-frequency distributions P_{pre} and P_{eval} are computed as relative frequencies of $w \in \mathcal{V}_b$ in the pre-training corpus and the benchmark dataset respectively.

Zero count smoothing. For words $w \in \mathcal{V}_b$ that appear in the benchmark but not in the pre-training corpus, the empirical probability $P_{\text{pre}}(w)$ would be zero and the cross-entropy in Section 3.1 would diverge. We therefore apply zero-count smoothing to P_{pre} over $\mathcal{V}_b$, assigning a unit count only to w absent from the pre-training corpus:

$$\tilde{c}(w) = \max\{c(w), 1\}, \qquad P_{\text{pre}}(w) = \frac{\tilde{c}(w)}{T + B}, \tag{7}$$

where $c(w)$ is the raw count of w in the pre-training corpus, T is the total word-token count of the pre-training corpus, and $B = \sum_{w \in \mathcal{V}_b} \mathbf{1}[c(w) = 0]$ is the number of unseen benchmark-vocabulary types. Unlike Laplace smoothing, this leaves all positive counts unchanged. Because nearly all benchmark-vocabulary types are observed in the pre-training corpus, flooring only zero counts perturbs the empirical training distribution less than Laplace smoothing. The very small probability mass of $w \in B$ shown in Appendix D.4 further suggests that zero-count smoothing is more appropriate than Laplace smoothing for measuring empirical distributional alignment between the pre-training corpus and benchmark corpora. The same smoothing is applied to all corpora and scales reported in Sections 4.2–6.

B Related Work

Lexical traces of pre-training data. A line of work has linked lexical signals to language-model behavior at increasingly coarse granularities. Razeghi et al. [7] link item-level term frequency to few-shot numerical reasoning, Kandpal et al. [55] extend this to entity-level long-tail frequency in QA, Wang et al. [60] formalize distributional memorization between LLM output probabilities and per-task n-gram counts, and Ruzzetti et al. [8] relate test-instance lexical popularity to performance. Closest in spirit, Wu et al. [9] mine salient tokens from in-the-wild web corpora whose token-level perplexity, taken over a panel of 32 pretrained models, predicts performance on 89 benchmarks. Across all of these, the lexical signal is measured on pretrained checkpoints, and the corpus identity that drives it is entangled with training recipe, scale, tokenizer, and training history. We instead vary corpus identity under a fixed from-scratch recipe (Section 4.1) and intervene on document-level alignment (Section 5).

Compression and Markov approximations. Section 2 discusses work that frames language modeling as compression of the training distribution [13, 15] and likelihood training as estimation of the empirical data distribution [17], together with work relating neural language models to n-gram statistics [19–21].

Data selection. Aligning training data with a downstream target has a long history. Klakow [61] trains a unigram language model on the full non-domain corpus and scores each candidate segment by how removing it would change the in-domain data log-likelihood under that model. Moore and Lewis [44] introduce the cross-entropy difference filter, while Axelrod [62] and Feng et al. [63] develop cynical data selection, a greedy method that chooses sentences by how much adding each one improves the language-model likelihood of the target-domain data. More recently, Xie et al. [43] scale importance resampling using hashed n-gram features, and Thrush et al. [64] use the correlation of per-domain LLM perplexity with downstream accuracy across 90 LMs as a selection signal. Magnusson et al. [65] use small-scale proxy models to predict which pre-training data yields the best models at larger scale across downstream benchmarks. The UCE-filter shares this goal but operationalizes alignment as a single information-theoretic functional requiring only word counts, with the same statistic that drives our descriptive analysis (Section 4.2) reused as the row-local selection signal (Section 5.1). Appendix E.3 compares the UCE-filter with the embedding-based selector BETR [45] under our from-scratch grid.

Benchmark contamination. Another line of work treats the corpus–evaluation interface as a contamination problem at the level of exact or near-exact overlap [66], supported by large-scale corpus auditing [67]. UCE complements contamination detection by summarizing the interface at the coarser level of probability mass on benchmark words rather than exact-match overlap. Appendix F reports our own 13-gram audit of the four corpora, with contaminated items and documents removed at the evaluation and training ends.

Speculative decoding. Section 6 routes between corpus-specialized drafters using unigram cross-entropy. The base setup follows Leviathan et al. [47] and Chen et al. [48]. Among learned routing policies, Hou et al. [68] use bandit-based drafter selection and Wu et al. [69] adapt across multiple drafter levels, while Hong et al. [70] report best practices for training domain-specialized drafters. A separate model-free direction retrieves candidates from non-parametric text statistics [71, 72]. UCE-routing keeps the parametric drafter pool fixed and selects among them via a precomputed corpus-level unigram distribution, with no per-prompt training, online learning, or retrieval at inference.

C Setup

The corpus-ranking study of Section 4.2 (§C.1) and the UCE-filter data-selection study of Section 5.2 (§C.2) use different model sizes, learning-rate schedules, and token budgets, so we report their configurations separately.

C.1 Corpus-ranking experiments (Section 4.2)

Table 6 gives the training recipe and Table 7 the three LLaMA-architecture [26] model sizes.

Table 6: Training configurations for the corpus-ranking experiments (Section 4.2). All runs use this recipe across the three model scales of Table 7.

Optimizer	β_1	β_2	ϵ	**Weight Decay**
AdamW	0.9	0.999	10^{-8}	0.1

Peak LR	**Schedule**	**Warmup**	**Weight Init.**
3×10^{-4}	Cosine	700 steps	0.02

Global Batch	**Per-device Batch**	**Devices**	**Sequence Length**	**Seed**
256	32	8 GPUs	2048	42

Table 7: Model configurations for the corpus-ranking experiments (Section 4.2). Sizes count non-embedding parameters.

Size	n_{layers}	n_{heads}	d_{model}	d_{ffn}	d_{head}
402M	24	8	1024	4096	128
1.33B	30	30	1920	5120	64
3.36B	32	20	2560	10240	128

Evaluation metric. For LAMBADA, the corpus-ranking tables in Appendix D.4 report perplexity (lower better), while the UCE-filter experiment of Section 5.2 uses last-word accuracy (higher better, see Table 32). Last-word accuracy is too noisy to separate corpora reliably on the 400M–3.36B corpus-ranking grid but informative on the 1.34B–3.33B UCE-filter grid trained on 70B tokens. All other benchmarks use accuracy throughout. For the selection experiments of Section 5.2 each task is scored with its primary lm-evaluation-harness metric, length-normalized accuracy for ARC-Easy/Challenge, HellaSwag, OpenBookQA, and PIQA, and plain accuracy for SciQ, SWAG, LAMBADA, CommonsenseQA, BoolQ, and SocialIQA.

C.2 UCE-filter experiments (Section 5.2)

We follow the architectural patterns of DataComp-LM [28]. The two scales (Table 9) are 1.34B parameters (1.23B non-embedding), following the DCLM "1B-1x" specification, and 3.33B parameters (3.17B non-embedding), following the DCLM 7B pattern scaled to 28 layers. The four selection methods (the UCE-filter, Random, and the DSIR-paper and DSIR-LN variants of Appendix E.2) train on 70B GPT-2 tokens under the recipe of Table 8, with gradient clipping at 1.0 and a distributed optimizer, and each model is evaluated zero-shot with lm-evaluation-harness using each task's primary metric.

Out-of-target tasks and MMLU omission. Compared with the 10 corpus-ranking tasks of Section 4.1, the UCE-filter evaluation suite omits MMLU and adds CommonsenseQA and BoolQ. We omit MMLU because all selection methods cluster within ±1 pp of chance (around 25%) at the 1.34B and 3.33B scales with a 70B-token budget, so the task does not separate methods (raw MMLU scores are reported in the supplementary material). We add BoolQ, a yes/no calibration task plausibly orthogonal to the targeted-suite vocabulary, as a stronger test of whether UCE-filtering's gains reflect

targeted vocabulary alignment rather than a generic improvement. The four out-of-target tasks (LAMBADA, CommonsenseQA, BoolQ, SocialIQA) are reported alongside the targeted-suite average in every (Source, Scale, Method) cell. SocialIQA is evaluated via the `lighteval/siqa` parquet mirror, which has the same schema as the canonical `allenai/social_i_qa` release.

Selection-pool sizes and training-exposure ratios. On DCLM, the UCE-filter selection contains roughly 227B tokens, the row-count-matched random sample 155B, DSIR-LN 105B, and DSIR-paper 106B. On FineWeb-Edu, the four pools are roughly 90B, 70B, 86B, and 101B respectively. All four methods then train for the same 70B-token budget, with the data loader sampling uniformly from each pool, so the exposure ratio (training tokens divided by pool tokens) varies across methods. On DCLM the UCE-filter trains on roughly 31% of its selected pool while DSIR-LN trains on roughly 66% of its own, and on FineWeb-Edu the corresponding ratios are 78% and 81%. The downstream-accuracy lead of the UCE-filter (Section 5.2, Table 2) is therefore not an artifact of a larger selection pool.

Noise calibration. Training seeds are 1234, 42, and 7777 for the UCE-filter, Random, and DSIR-paper arms and 1234 for the DSIR-LN ablation, with per-method standard deviations and Welch tests in Appendix E.8.

Table 8: Training configurations for the UCE-filter experiments (Section 5.2). All sixteen runs (Source $\times$ Scale $\times$ {UCE-filter, Random, DSIR-paper, DSIR-LN}) use this recipe.

Optimizer	β_1	β_2	ϵ	**Weight Decay**
AdamW	0.9	0.95	10^{-8}	0.033

Peak LR	**Min LR**	**Schedule**	**Warmup**
1×10^{-3}	1×10^{-5}	Cosine	5000 iters

Global Batch	**Sequence Length**	**Tokens**	**Precision**	**Hardware**
256	2048	70B	BF16	16 B200 (2 nodes)

Table 9: Model architectures for the UCE-filter experiments (Section 5.2). DataComp-LM patterns [28]. All use SwiGLU activations, RMSNorm, RoPE positions, and tied input/output embeddings.

Size	n_{layers}	n_{heads}	d_{model}	d_{ffn}	d_{head}	**Pattern**
1.34B	24	16	2048	5632	128	DCLM 1B-1x
3.33B	28	24	3072	8192	128	DCLM 7B (scaled to 28 layers)

D Corpus-level UCE analyses

Section 4.2 ranks corpora by aggregate UCE and ties this ranking to downstream accuracy. The analyses below test that corpus-level finding and mark where the statistic's scope ends. The ranking is robust along the axes they examine, and Appendix D.2 shows in particular that the in-vocabulary distribution rather than the smoothing tail carries the cross-corpus signal.

D.1 Coverage and mass allocation across benchmarks

To probe the mechanism behind the inverse cross-entropy/accuracy relation in Section 4.2 (Figure 2), we decompose the corpus–benchmark overlap into *vocabulary coverage* (the fraction of the benchmark vocabulary that the corpus has observed at all) and *probability-mass allocation* (the share of benchmark word mass attributable to top-percentile training words versus the long tail). Table 13 reports both at the 1.33B/26B-token operating point for every task. Coverage is essentially saturated on every task ($\sim 99.9\%$ Seen on every corpus), so cross-corpus differences cannot be explained by coverage alone. Mass allocation, in contrast, varies systematically with cross-entropy and tracks the per-task ordering of zero-shot accuracy on ARC Easy, HellaSwag, and MMLU.

D.2 Out-of-vocabulary contribution to UCE

The coverage analysis above shows that DCLM and FineWeb-Edu both place virtually all of the benchmark vocabulary inside their support (Seen $\approx 99.9\%$), so cross-corpus differences in UCE cannot be attributed to which words are observed at all. This raises the question of whether the residual differences come from the in-vocabulary distribution or from the zero-count smoothing tail introduced in Appendix A.1, which assigns a unit count to benchmark words that the pre-training corpus never produced. Because these words still enter UCE through their $\log P_{\text{pre}}(w)$ term, we isolate their contribution by splitting the benchmark vocabulary into the covered subset $\mathcal{V}_b^{\text{cov}} = \mathcal{V}_b \cap \{w : c(w) > 0\}$ and its complement $\mathcal{V}_b^{\text{unc}} = \mathcal{V}_b \setminus \mathcal{V}_b^{\text{cov}}$, so that

$$H(P_{\text{eval}}, P_{\text{pre}}) \;=\; \underbrace{-\sum_{w \,\in\, \mathcal{V}_b^{\text{cov}}} P_{\text{eval}}(w)\, \log P_{\text{pre}}(w)}_{\text{covered contribution}} \;+\; \underbrace{-\sum_{w \,\in\, \mathcal{V}_b^{\text{unc}}} P_{\text{eval}}(w)\, \log P_{\text{pre}}(w)}_{\text{uncovered contribution}}. \tag{8}$$

Under zero-count smoothing, the uncovered contribution collapses to $m_{\text{unc}} \cdot \log(T + B)$, where $m_{\text{unc}} = \sum_{w \in \mathcal{V}_b^{\text{unc}}} P_{\text{eval}}(w)$ is the benchmark mass on uncovered words, T is the total word-token count of the pre-training corpus, and $B = \sum_{w \in \mathcal{V}_b} \mathbf{1}[c(w) = 0] = |\mathcal{V}_b^{\text{unc}}|$ is the number of unseen benchmark vocabulary types.

Table 10 reports the two components for DCLM and FineWeb-Edu against the CSR7 and ten-task corpus-ranking groupings, measured on a uniform five-percent fragment sample of each source corpus (roughly 30.5B DCLM tokens and 13.7B FineWeb-Edu tokens). In every cell the covered subset captures more than 99.95% of the benchmark mass, leaving m_{unc} below 5×10^{-4} and an uncovered contribution between 0.0095 and 0.0151 bits, which is less than 0.14% of the UCE value. Subtracting the uncovered term from each cell preserves both orderings of the main results, with DCLM below FineWeb-Edu and CSR7 below the corpus-ranking grouping. The cross-corpus and cross-benchmark structure of UCE is therefore a property of the in-vocabulary distribution rather than an artifact of the smoothing tail.

Table 10: UCE decomposition by coverage (bits ↓) for DCLM and FineWeb-Edu against the CSR7 and ten-task corpus-ranking benchmark groupings, measured on a five-percent fragment sample of each source corpus. The benchmark mass on uncovered words m_{unc} and its contribution to UCE are computed from Equation (8).

Source	Benchmark	UCE	Type cov.	Mass cov.	m_{unc}	Uncov. contrib.	Share of UCE
DCLM	CSR7	11.1665	95.54%	99.973%	2.73×10^{-4}	0.0095	0.085%
DCLM	Corpus-ranking	11.1843	95.83%	99.972%	2.80×10^{-4}	0.0098	0.087%
FineWeb-Edu	CSR7	11.3256	93.88%	99.957%	4.32×10^{-4}	0.0146	0.128%
FineWeb-Edu	Corpus-ranking	11.3572	94.12%	99.955%	4.50×10^{-4}	0.0151	0.133%

D.3 Robustness of UCE to preprocessing choices

The implementation specified in Section 3.1 and Appendix A.1 combines NFKC normalization, punctuation stripping, whitespace tokenization without case folding, and zero-count smoothing of P_{pre} over $\mathcal{V}_b^{\text{unc}}$, where each word $w \in \mathcal{V}_b^{\text{unc}}$ is assigned a unit count. To check that the DCLM-versus-FineWeb-Edu ranking reported in Section 4.2 is not driven by these defaults, each component is perturbed in turn while the rest of the implementation is held fixed. UCE is recomputed on a uniform one-percent fragment sample of the two pre-training corpora (roughly 6.2B DCLM tokens and 2.9B FineWeb-Edu tokens) and evaluated against both the seven-task targeted grouping (CSR7) of Section 5.2 and the ten-task corpus-ranking grouping of Section 4.2. The perturbations cover case folding before counting, Lidstone smoothing at concentration $1/2$, unsmoothed raw vocabulary counts, and an inverse-document-frequency reweighting of the per-word contribution.[7]

Table 11 collects UCE for the four corpus-by-grouping combinations under each variant, alongside the within-grouping gap $\Delta_{\text{Fw}-\text{DCLM}}$. FineWeb-Edu sits above DCLM in every cell of every variant. Under the default pipeline and its smoothing perturbations the gap lies between 0.16 and 0.17, with the two alternative smoothers (concentration $1/2$ and no smoothing) moving absolute UCE by less than 0.003 in any cell. Case folding shifts all four UCE values downward by roughly 0.32 as case variants collapse into shared types, leaving the gap intact. Inverse-document-frequency reweighting raises absolute UCE by about 3.2 and widens the within-grouping gap to roughly 0.24. The default configuration is used throughout the paper.

Table 11: UCE (bits ↓) under five preprocessing variants on the DCLM and FineWeb-Edu corpora (one-percent fragment sample), evaluated against the seven-task targeted grouping (CSR7) of Section 5.2 and the ten-task corpus-ranking grouping of Section 4.2. $\Delta_{\text{Fw}-\text{DCLM}}$ is the within-grouping FineWeb-Edu minus DCLM UCE.

Variant	**DCLM**		**FineWeb-Edu**		$\Delta_{\text{Fw}-\text{DCLM}}$	
	CSR7	Corpus-rank	CSR7	Corpus-rank	CSR7	Corpus-rank
Baseline	11.1650	11.1832	11.3215	11.3539	+0.1565	+0.1707
Lowercase	10.8486	10.8652	10.9952	11.0246	+0.1466	+0.1594
Lidstone ($\alpha = \frac{1}{2}$)	11.1666	11.1848	11.3243	11.3566	+0.1577	+0.1718
No smoothing	11.1656	11.1838	11.3234	11.3556	+0.1578	+0.1718
IDF weighting	14.3153	14.3110	14.5502	14.5539	+0.2349	+0.2429

[7] Lemmatization and stopword removal would be informative additional axes but require lexical resources external to the measurement pipeline and are not considered here.

D.4 Per-task benchmark accuracies and word overlap correlations

The tables below report per-task zero-shot accuracy at the three matched (model, subset) operating points of the corpus-level analysis (3.36B / 60B, 1.33B / 26B, 402M / 8.5B), so that the per-corpus entropy-vs-accuracy tracking of Section 4.2 can be inspected task by task. Table 13 also reports the 402M model trained on the same 26B-token subsets. OpenWebText, a single 8.5B-token set, is listed only with subset 1 (Table 14).

Table 12: Per-task word-frequency coverage, unigram cross-entropy (bits ↓), and zero-shot score for a 3.36B model on 60B tokens. LAMBADA is reported in perplexity (lower better) and other tasks in accuracy (higher better). The ∼80% and ∼95% columns report cumulative shares of benchmark word mass attributable to top-percentile training words, the 95–100% column the share on the remaining tail words, and Seen is the fraction of benchmark vocabulary observed at least once. Source for Figure 2.

Task	Dataset	∼80%	∼95%	95–100%	Seen	Entropy	Score
ARC-Easy	FineWeb-Edu	83.97% (150,599)	98.15% (176,031)	1.84% (3,298)	99.99% (179,329)	11.1946	73.65
	DCLM	78.54% (140,870)	97.24% (174,411)	2.75% (4,926)	99.99% (179,337)	11.5477	70.20
	C4	77.74% (139,428)	96.63% (173,311)	3.35% (6,003)	99.98% (179,314)	11.6493	60.77
ARC-Challenge	FineWeb-Edu	85.04% (93,417)	98.16% (107,823)	1.82% (1,997)	99.97% (109,820)	11.0831	38.82
	DCLM	80.32% (88,236)	97.45% (107,051)	2.54% (2,791)	99.99% (109,842)	11.3847	34.81
	C4	79.88% (87,745)	97.05% (106,610)	2.89% (3,180)	99.95% (109,790)	11.4714	26.79
HellaSwag	FineWeb-Edu	82.70% (6,900,957)	96.31% (8,035,886)	3.64% (304,022)	99.95% (8,339,908)	11.2117	46.63
	DCLM	82.74% (6,903,940)	96.90% (8,085,503)	3.06% (255,401)	99.96% (8,340,904)	11.0590	48.48
	C4	83.68% (6,981,914)	96.99% (8,093,304)	2.96% (246,704)	99.95% (8,340,008)	11.0377	49.02
MMLU	FineWeb-Edu	80.37% (953,768)	95.99% (1,139,104)	3.97% (47,072)	99.96% (1,186,176)	11.3635	25.23
	DCLM	78.64% (933,162)	95.95% (1,138,626)	4.01% (47,596)	99.96% (1,186,222)	11.4071	24.58
	C4	78.45% (930,923)	95.09% (1,128,357)	4.85% (57,534)	99.94% (1,185,891)	11.5237	23.77
SciQ	FineWeb-Edu	80.28% (960,226)	95.74% (1,145,183)	4.22% (50,508)	99.97% (1,195,691)	11.4675	90.80
	DCLM	74.95% (896,531)	94.13% (1,125,843)	5.83% (69,791)	99.96% (1,195,634)	11.8577	89.70
	C4	73.69% (881,360)	92.98% (1,112,130)	6.93% (82,930)	99.91% (1,195,060)	12.0335	83.90
OBQA	FineWeb-Edu	79.76% (105,348)	96.69% (127,702)	3.30% (4,356)	99.99% (132,058)	11.5424	33.00
	DCLM	77.40% (102,225)	96.51% (127,467)	3.48% (4,596)	99.99% (132,063)	11.6525	28.20
	C4	77.21% (101,974)	95.97% (126,746)	4.01% (5,302)	99.98% (132,048)	11.7398	26.40
PIQA	FineWeb-Edu	74.89% (707,350)	95.56% (902,571)	4.42% (41,744)	99.98% (944,315)	11.6208	75.14
	DCLM	75.43% (712,414)	96.67% (912,988)	3.32% (31,403)	99.99% (944,391)	11.4875	75.90
	C4	77.37% (730,709)	97.25% (918,478)	2.73% (25,819)	99.98% (944,297)	11.3887	76.28
LAMBADA	FineWeb-Edu	81.12% (256,695)	94.54% (299,132)	5.32% (16,840)	99.86% (315,972)	11.5117	11.72
	DCLM	83.40% (263,900)	96.60% (305,663)	3.31% (10,468)	99.91% (316,131)	10.9253	5.90
	C4	82.45% (260,885)	95.76% (303,016)	4.14% (13,091)	99.90% (316,107)	11.1855	9.17
Social IQA	FineWeb-Edu	79.06% (877,613)	96.62% (1,072,651)	3.37% (37,381)	99.99% (1,110,032)	11.6220	42.02
	DCLM	81.70% (906,937)	98.46% (1,093,035)	1.54% (17,050)	100.00% (1,110,085)	11.1622	42.73
	C4	81.47% (904,368)	97.47% (1,082,087)	2.52% (27,959)	99.99% (1,110,046)	11.3183	42.27
SWAG	FineWeb-Edu	78.28% (4,363,250)	95.46% (5,320,574)	4.54% (252,910)	99.99% (5,573,484)	11.4145	49.37
	DCLM	79.74% (4,444,545)	97.28% (5,421,994)	2.72% (151,704)	100.00% (5,573,698)	11.1036	52.82
	C4	80.07% (4,463,191)	96.63% (5,385,917)	3.36% (187,069)	99.98% (5,572,986)	11.2432	50.91

Table 13: 1.33B and 402M models trained on the same 26B-token subsets. Conventions as in Table 12. The coverage and entropy columns apply to both models.

Task	Dataset	∼80%	∼95%	95–100%	Seen	Entropy	Score (1.33B)	Score (402M)
ARC-Easy	FineWeb-Edu	83.87% (150,418)	98.09% (175,920)	1.90% (3,407)	99.98% (179,327)	11.1921	65.87	62.42
	DCLM	78.55% (140,887)	97.24% (174,412)	2.74% (4,923)	99.99% (179,335)	11.5461	63.17	56.78
	C4	77.74% (139,429)	96.63% (173,309)	3.35% (6,002)	99.98% (179,311)	11.6491	54.55	50.38
ARC-Challenge	FineWeb-Edu	84.93% (93,296)	98.11% (107,769)	1.87% (2,049)	99.97% (109,818)	11.0807	31.83	28.24
	DCLM	80.32% (88,230)	97.45% (107,050)	2.53% (2,780)	99.98% (109,830)	11.3833	27.65	26.19
	C4	79.88% (87,745)	97.05% (106,607)	2.89% (3,178)	99.94% (109,785)	11.4708	24.49	20.99
HellaSwag	FineWeb-Edu	82.72% (6,901,956)	96.28% (8,033,455)	3.66% (305,784)	99.94% (8,339,239)	11.2045	40.21	36.82
	DCLM	82.75% (6,905,130)	96.91% (8,086,092)	3.05% (254,461)	99.96% (8,340,553)	11.0568	40.75	37.17
	C4	83.68% (6,982,092)	97.00% (8,093,366)	2.95% (245,912)	99.94% (8,339,278)	11.0367	41.29	37.45
MMLU	FineWeb-Edu	80.34% (953,323)	95.93% (1,138,375)	4.02% (47,671)	99.95% (1,186,046)	11.3649	25.89	25.07
	DCLM	78.64% (933,166)	95.95% (1,138,586)	4.01% (47,564)	99.96% (1,186,150)	11.4063	24.96	23.84
	C4	78.45% (930,919)	95.09% (1,128,353)	4.83% (57,372)	99.92% (1,185,725)	11.5230	23.39	23.44
SciQ	FineWeb-Edu	80.23% (959,649)	95.72% (1,144,872)	4.24% (50,670)	99.95% (1,195,542)	11.4639	86.10	83.80
	DCLM	74.97% (896,735)	94.13% (1,125,856)	5.83% (69,679)	99.95% (1,195,535)	11.8554	85.00	82.40
	C4	73.69% (881,384)	92.98% (1,112,195)	6.91% (82,688)	99.90% (1,194,883)	12.0327	79.30	75.20
OBQA	FineWeb-Edu	79.67% (105,223)	96.66% (127,660)	3.33% (4,394)	99.98% (132,054)	11.5388	26.60	24.20
	DCLM	77.39% (102,219)	96.50% (127,449)	3.49% (4,611)	99.99% (132,060)	11.6511	22.60	20.20
	C4	77.21% (101,974)	95.97% (126,750)	4.00% (5,285)	99.97% (132,035)	11.7396	20.00	19.80
PIQA	FineWeb-Edu	74.90% (707,409)	95.54% (902,398)	4.43% (41,869)	99.98% (944,267)	11.6130	70.89	69.97
	DCLM	75.46% (712,664)	96.66% (912,971)	3.33% (31,405)	99.99% (944,376)	11.4864	71.87	70.24
	C4	77.37% (730,757)	97.24% (918,454)	2.73% (25,774)	99.97% (944,228)	11.3888	73.07	70.62
LAMBADA	FineWeb-Edu	81.08% (256,542)	94.48% (298,938)	5.36% (16,969)	99.84% (315,907)	11.5117	20.74	28.51
	DCLM	83.40% (263,882)	96.60% (305,658)	3.30% (10,439)	99.90% (316,097)	10.9257	9.82	13.08
	C4	82.45% (260,886)	95.76% (303,010)	4.13% (13,055)	99.89% (316,065)	11.1845	15.32	21.86
Social IQA	FineWeb-Edu	79.06% (877,630)	96.58% (1,072,202)	3.40% (37,789)	99.99% (1,109,991)	11.6236	40.74	38.84
	DCLM	81.70% (907,011)	98.46% (1,093,021)	1.54% (17,050)	100.00% (1,110,071)	11.1627	41.81	40.99
	C4	81.47% (904,368)	97.83% (1,086,047)	2.16% (23,975)	99.99% (1,110,022)	11.3181	41.50	40.28
SWAG	FineWeb-Edu	78.25% (4,361,339)	95.38% (5,316,131)	4.62% (257,257)	99.99% (5,573,388)	11.4153	45.53	43.74
	DCLM	79.74% (4,444,837)	97.30% (5,423,101)	2.70% (150,564)	100.00% (5,573,665)	11.1041	48.92	47.07
	C4	80.08% (4,463,291)	96.63% (5,385,961)	3.35% (186,544)	99.98% (5,572,505)	11.2428	47.21	44.83

Table 14: 402M model, 8.5B-token subset 1. Conventions as in Table 12. Source for the 8.5B subsets in Figure 3.

Task	Dataset	∼80%	∼95%	95–100%	Seen	Entropy	Score
ARC-Easy	FineWeb-Edu	83.88% (150,436)	98.03% (175,827)	1.94% (3,484)	99.98% (179,311)	11.1923	57.00
	DCLM	78.33% (140,481)	97.18% (174,302)	2.80% (5,029)	99.99% (179,331)	11.5566	51.56
	C4	77.74% (139,425)	96.63% (173,317)	3.33% (5,978)	99.97% (179,295)	11.6480	45.16
	OpenWebText	76.05% (136,394)	96.09% (172,350)	3.88% (6,962)	99.98% (179,312)	11.7504	42.97
ARC-Challenge	FineWeb-Edu	84.93% (93,296)	98.11% (107,769)	1.87% (2,049)	99.97% (109,818)	11.0808	23.81
	DCLM	80.12% (88,016)	97.40% (106,994)	2.58% (2,835)	99.98% (109,829)	11.3923	21.59
	C4	79.90% (87,771)	97.06% (106,615)	2.86% (3,144)	99.92% (109,759)	11.4691	20.65
	OpenWebText	78.15% (85,844)	96.54% (106,048)	3.41% (3,751)	99.95% (109,799)	11.5597	17.83
HellaSwag	FineWeb-Edu	82.66% (6,896,872)	96.26% (8,031,936)	3.67% (305,970)	99.93% (8,337,906)	11.2040	31.71
	DCLM	82.66% (6,897,550)	96.89% (8,084,471)	3.06% (255,340)	99.95% (8,339,811)	11.0629	31.92
	C4	83.67% (6,981,505)	97.00% (8,093,738)	2.92% (244,031)	99.92% (8,337,769)	11.0380	32.31
	OpenWebText	81.02% (6,760,740)	95.86% (7,998,935)	4.06% (339,052)	99.93% (8,337,987)	11.2910	28.94
MMLU	FineWeb-Edu	80.28% (952,654)	95.88% (1,137,793)	4.05% (48,043)	99.93% (1,185,836)	11.3675	23.52
	DCLM	78.55% (932,136)	95.90% (1,138,087)	4.04% (47,896)	99.94% (1,185,983)	11.4105	23.47
	C4	78.45% (930,884)	95.27% (1,130,478)	4.62% (54,851)	99.89% (1,185,329)	11.5219	22.95
	OpenWebText	78.72% (934,124)	95.20% (1,129,739)	4.69% (55,672)	99.89% (1,185,411)	11.4556	22.99
SciQ	FineWeb-Edu	80.19% (959,123)	95.64% (1,143,922)	4.30% (51,384)	99.93% (1,195,306)	11.4668	77.90
	DCLM	74.72% (893,732)	94.04% (1,124,873)	5.89% (70,414)	99.93% (1,195,287)	11.8684	75.50
	C4	73.69% (881,363)	92.99% (1,112,225)	6.86% (82,089)	99.85% (1,194,314)	12.0307	72.20
	OpenWebText	72.24% (864,088)	92.09% (1,101,431)	7.78% (93,010)	99.86% (1,194,441)	12.1509	63.60
OBQA	FineWeb-Edu	79.68% (105,241)	96.61% (127,599)	3.36% (4,439)	99.97% (132,038)	11.5375	21.40
	DCLM	77.32% (102,119)	96.50% (127,446)	3.49% (4,607)	99.98% (132,053)	11.6558	19.60
	C4	77.20% (101,958)	95.97% (126,757)	3.99% (5,264)	99.96% (132,021)	11.7388	18.00
	OpenWebText	74.83% (98,835)	95.14% (125,652)	4.83% (6,385)	99.97% (132,037)	11.8795	16.20
PIQA	FineWeb-Edu	74.86% (707,036)	95.48% (901,747)	4.49% (42,418)	99.97% (944,165)	11.6146	65.61
	DCLM	75.38% (711,941)	96.64% (912,711)	3.35% (31,609)	99.98% (944,320)	11.4894	66.49
	C4	77.33% (730,403)	97.24% (918,421)	2.72% (25,729)	99.97% (944,150)	11.3885	67.41
	OpenWebText	71.96% (679,645)	94.86% (895,943)	5.10% (48,198)	99.96% (944,141)	11.7896	61.86
LAMBADA	FineWeb-Edu	81.04% (256,423)	94.43% (298,786)	5.36% (16,964)	99.79% (315,750)	11.5149	77.79
	DCLM	83.42% (263,968)	96.62% (305,710)	3.24% (10,242)	99.85% (315,952)	10.9105	25.27
	C4	82.70% (261,669)	95.88% (303,386)	3.92% (12,414)	99.80% (315,800)	11.0636	43.89
	OpenWebText	82.44% (260,870)	95.76% (303,009)	4.08% (12,924)	99.85% (315,933)	11.1827	55.97
Social IQA	FineWeb-Edu	79.01% (877,094)	96.56% (1,071,970)	3.42% (37,963)	99.98% (1,109,933)	11.6292	38.38
	DCLM	81.71% (907,113)	98.46% (1,093,033)	1.53% (16,998)	99.99% (1,110,031)	11.1542	38.84
	C4	81.46% (904,300)	97.83% (1,086,091)	2.15% (23,815)	99.98% (1,109,906)	11.3178	38.43
	OpenWebText	82.24% (912,925)	97.75% (1,085,165)	2.23% (24,764)	99.98% (1,109,929)	11.2691	38.56
SWAG	FineWeb-Edu	78.03% (4,349,313)	95.36% (5,315,178)	4.63% (258,007)	99.99% (5,573,185)	11.4158	40.17
	DCLM	79.70% (4,442,453)	97.29% (5,423,029)	2.70% (150,521)	99.99% (5,573,550)	11.0974	43.16
	C4	80.07% (4,462,853)	96.63% (5,386,222)	3.33% (185,788)	99.97% (5,572,010)	11.2423	41.47
	OpenWebText	78.46% (4,373,230)	96.46% (5,376,797)	3.52% (196,380)	99.99% (5,573,177)	11.2255	40.41

Table 15: 402M model, 8.5B-token subset 2. Conventions as in Table 12.

Task	Dataset	∼80%	∼95%	95–100%	Seen	Entropy	Score
ARC-Easy	FineWeb-Edu	83.93% (150,539)	98.10% (175,941)	1.88% (3,380)	99.98% (179,321)	11.1955	56.00
	DCLM	78.63% (141,023)	97.27% (174,461)	2.72% (4,870)	99.99% (179,331)	11.5345	51.01
	C4	77.74% (139,427)	96.63% (173,312)	3.34% (5,995)	99.97% (179,307)	11.6487	46.21
ARC-Challenge	FineWeb-Edu	85.01% (93,382)	98.11% (107,775)	1.84% (2,017)	99.95% (109,792)	11.0829	23.38
	DCLM	80.41% (88,332)	97.49% (107,090)	2.48% (2,729)	99.97% (109,819)	11.3728	21.16
	C4	79.88% (87,745)	97.05% (106,604)	2.89% (3,170)	99.93% (109,774)	11.4701	20.73
HellaSwag	FineWeb-Edu	82.72% (6,901,878)	96.30% (8,034,960)	3.63% (303,121)	99.93% (8,338,081)	11.2078	31.65
	DCLM	82.81% (6,909,890)	96.92% (8,087,440)	3.02% (252,232)	99.95% (8,339,672)	11.0474	31.90
	C4	83.68% (6,981,979)	96.99% (8,092,739)	2.94% (245,508)	99.93% (8,338,247)	11.0335	32.02
MMLU	FineWeb-Edu	80.36% (953,600)	95.97% (1,138,888)	3.96% (46,981)	99.93% (1,185,869)	11.3618	24.18
	DCLM	78.73% (934,314)	95.96% (1,138,864)	3.97% (47,098)	99.94% (1,185,962)	11.4009	23.84
	C4	78.45% (930,913)	95.09% (1,128,372)	4.81% (57,072)	99.90% (1,185,444)	11.5219	22.92
SciQ	FineWeb-Edu	80.24% (959,729)	95.70% (1,144,637)	4.24% (50,667)	99.93% (1,195,304)	11.4689	78.00
	DCLM	75.09% (898,207)	94.17% (1,126,320)	5.76% (68,945)	99.93% (1,195,265)	11.8404	75.50
	C4	73.69% (881,384)	92.99% (1,112,297)	6.87% (82,181)	99.86% (1,194,478)	12.0307	71.40
OBQA	FineWeb-Edu	79.75% (105,336)	96.69% (127,697)	3.29% (4,343)	99.97% (132,040)	11.5407	22.20
	DCLM	77.44% (102,277)	96.49% (127,441)	3.49% (4,608)	99.98% (132,049)	11.6460	20.60
	C4	77.20% (101,958)	95.97% (126,757)	3.99% (5,264)	99.96% (132,021)	11.7388	18.80
PIQA	FineWeb-Edu	74.91% (707,492)	95.54% (902,377)	4.42% (41,788)	99.97% (944,165)	11.6176	65.67
	DCLM	75.45% (712,608)	96.66% (912,938)	3.32% (31,355)	99.98% (944,293)	11.4824	66.43
	C4	77.37% (730,720)	97.23% (918,316)	2.74% (25,846)	99.97% (944,162)	11.3884	68.12
LAMBADA	FineWeb-Edu	81.16% (256,800)	94.53% (299,097)	5.27% (16,661)	99.79% (315,758)	11.4997	77.77
	DCLM	83.27% (263,497)	96.57% (305,562)	3.28% (10,381)	99.85% (315,943)	10.9398	27.20
	C4	82.57% (261,269)	95.88% (303,386)	3.92% (12,414)	99.80% (315,800)	11.0776	44.22
Social IQA	FineWeb-Edu	79.06% (877,649)	96.62% (1,072,595)	3.36% (37,302)	99.98% (1,109,897)	11.6201	37.46
	DCLM	81.69% (906,890)	98.06% (1,088,553)	1.93% (21,473)	99.99% (1,110,026)	11.1720	38.56
	C4	81.46% (904,306)	97.46% (1,081,944)	2.52% (28,017)	99.99% (1,109,961)	11.3177	38.43
SWAG	FineWeb-Edu	78.29% (4,364,031)	95.45% (5,319,967)	4.54% (253,144)	99.99% (5,573,111)	11.4134	40.12
	DCLM	79.68% (4,441,257)	97.27% (5,421,703)	2.72% (151,826)	99.99% (5,573,529)	11.1120	42.77
	C4	80.07% (4,463,259)	96.63% (5,385,879)	3.34% (186,280)	99.97% (5,572,159)	11.2426	41.47

Table 16: 402M model, 8.5B-token subset 3. Conventions as in Table 12.

Task	Dataset	∼80%	∼95%	95–100%	Seen	Entropy	Score
ARC-Easy	FineWeb-Edu	84.15% (150,929)	98.21% (176,149)	1.77% (3,166)	99.98% (179,315)	11.1988	56.44
	DCLM	78.55% (140,881)	97.24% (174,412)	2.74% (4,908)	99.98% (179,320)	11.5464	50.13
	C4	77.74% (139,430)	96.63% (173,307)	3.34% (5,988)	99.97% (179,295)	11.6495	46.76
ARC-Challenge	FineWeb-Edu	85.24% (93,635)	98.22% (107,895)	1.72% (1,894)	99.95% (109,789)	11.0862	24.40
	DCLM	80.32% (88,233)	97.45% (107,052)	2.51% (2,760)	99.97% (109,812)	11.3837	20.99
	C4	79.89% (87,760)	97.05% (106,612)	2.87% (3,151)	99.92% (109,763)	11.4708	19.88
HellaSwag	FineWeb-Edu	82.75% (6,904,886)	96.46% (8,042,889)	3.54% (295,358)	99.93% (8,338,247)	11.2212	31.42
	DCLM	82.75% (6,904,401)	96.92% (8,086,616)	3.03% (253,093)	99.95% (8,339,709)	11.0586	31.62
	C4	83.67% (6,981,546)	97.00% (8,093,233)	2.93% (244,769)	99.93% (8,338,002)	11.0391	31.92
MMLU	FineWeb-Edu	80.51% (955,378)	96.10% (1,140,328)	3.84% (45,518)	99.93% (1,185,846)	11.3586	24.73
	DCLM	78.63% (933,113)	95.96% (1,138,658)	3.99% (47,289)	99.94% (1,185,947)	11.4060	23.22
	C4	78.45% (930,922)	95.27% (1,130,473)	4.62% (54,865)	99.89% (1,185,338)	11.5220	22.93
SciQ	FineWeb-Edu	80.39% (961,541)	95.87% (1,146,649)	4.07% (48,688)	99.94% (1,195,337)	11.4719	77.50
	DCLM	75.08% (898,086)	94.18% (1,126,518)	5.75% (68,742)	99.93% (1,195,260)	11.8468	73.90
	C4	73.70% (881,558)	92.97% (1,112,026)	6.88% (82,302)	99.85% (1,194,328)	12.0300	70.90
OBQA	FineWeb-Edu	79.93% (105,568)	96.77% (127,807)	3.20% (4,233)	99.97% (132,040)	11.5499	20.60
	DCLM	77.39% (102,210)	96.51% (127,459)	3.47% (4,589)	99.98% (132,048)	11.6511	18.40
	C4	77.21% (101,972)	95.96% (126,741)	4.00% (5,288)	99.97% (132,029)	11.7394	18.00
PIQA	FineWeb-Edu	74.89% (707,303)	95.59% (902,817)	4.38% (41,351)	99.97% (944,168)	11.6359	65.94
	DCLM	75.45% (712,622)	96.66% (912,925)	3.32% (31,381)	99.98% (944,306)	11.4871	66.21
	C4	77.34% (730,414)	97.27% (918,653)	2.70% (25,516)	99.97% (944,169)	11.3879	66.92
LAMBADA	FineWeb-Edu	81.20% (256,925)	94.69% (299,603)	5.09% (16,095)	99.77% (315,698)	11.5040	79.56
	DCLM	83.39% (263,869)	96.60% (305,671)	3.23% (10,212)	99.83% (315,883)	10.9298	28.50
	C4	82.54% (261,187)	95.86% (303,318)	3.94% (12,482)	99.80% (315,800)	11.0797	44.45
Social IQA	FineWeb-Edu	79.11% (878,186)	96.70% (1,073,476)	3.28% (36,439)	99.98% (1,109,915)	11.6180	38.54
	DCLM	81.69% (906,883)	98.06% (1,088,616)	1.93% (21,410)	99.99% (1,110,026)	11.1630	38.89
	C4	81.42% (903,876)	97.48% (1,082,107)	2.51% (27,819)	99.98% (1,109,926)	11.3174	38.56
SWAG	FineWeb-Edu	78.31% (4,365,081)	95.58% (5,327,285)	4.41% (245,889)	99.99% (5,573,174)	11.4139	39.93
	DCLM	79.74% (4,444,536)	97.30% (5,423,119)	2.70% (150,404)	99.99% (5,573,523)	11.1035	42.71
	C4	80.07% (4,462,896)	96.63% (5,386,154)	3.34% (186,109)	99.97% (5,572,263)	11.2428	41.36

Table 17: 402M model, 8.5B-token subset 4. Conventions as in Table 12.

Task	Dataset	∼80%	∼95%	95–100%	Seen	Entropy	Score
ARC-Easy	FineWeb-Edu	84.07% (150,785)	98.16% (176,062)	1.81% (3,253)	99.98% (179,315)	11.1948	56.31
	DCLM	78.62% (141,009)	97.27% (174,461)	2.71% (4,863)	99.98% (179,324)	11.5415	50.76
	C4	77.74% (139,431)	96.63% (173,318)	3.34% (5,986)	99.97% (179,304)	11.6484	46.89
ARC-Challenge	FineWeb-Edu	85.15% (93,533)	98.16% (107,827)	1.79% (1,965)	99.95% (109,792)	11.0826	23.63
	DCLM	80.41% (88,332)	97.47% (107,074)	2.50% (2,746)	99.97% (109,820)	11.3788	22.18
	C4	79.88% (87,742)	97.05% (106,610)	2.87% (3,157)	99.93% (109,767)	11.4695	20.14
HellaSwag	FineWeb-Edu	82.74% (6,904,066)	96.30% (8,035,511)	3.63% (302,637)	99.93% (8,338,148)	11.2218	31.44
	DCLM	82.82% (6,910,870)	96.94% (8,089,041)	3.00% (250,631)	99.95% (8,339,672)	11.0520	31.63
	C4	83.68% (6,982,173)	97.00% (8,093,434)	2.93% (244,862)	99.93% (8,338,296)	11.0344	32.04
MMLU	FineWeb-Edu	80.41% (954,215)	96.04% (1,139,663)	3.89% (46,175)	99.93% (1,185,838)	11.3603	24.42
	DCLM	78.73% (934,224)	95.99% (1,139,058)	3.95% (46,901)	99.94% (1,185,959)	11.4031	23.21
	C4	78.45% (930,980)	95.08% (1,128,306)	4.81% (57,132)	99.90% (1,185,438)	11.5220	22.93
SciQ	FineWeb-Edu	80.42% (961,873)	95.81% (1,145,980)	4.13% (49,377)	99.94% (1,195,357)	11.4662	79.50
	DCLM	75.08% (898,086)	94.18% (1,126,518)	5.75% (68,742)	99.93% (1,195,260)	11.8468	77.70
	C4	73.69% (881,411)	92.99% (1,112,222)	6.88% (82,257)	99.86% (1,194,479)	12.0306	70.70
OBQA	FineWeb-Edu	79.85% (105,459)	96.71% (127,729)	3.26% (4,309)	99.97% (132,038)	11.5459	20.20
	DCLM	77.47% (102,314)	96.52% (127,479)	3.46% (4,566)	99.98% (132,045)	11.6497	19.60
	C4	77.21% (101,972)	95.97% (126,747)	4.00% (5,280)	99.96% (132,027)	11.7389	19.00
PIQA	FineWeb-Edu	74.97% (708,091)	95.56% (902,547)	4.40% (41,602)	99.97% (944,149)	11.6283	64.42
	DCLM	75.46% (712,687)	96.69% (913,170)	3.30% (31,125)	99.98% (944,295)	11.4858	66.21
	C4	77.37% (730,753)	97.25% (918,464)	2.72% (25,690)	99.97% (944,154)	11.3877	67.19
LAMBADA	FineWeb-Edu	81.19% (256,887)	94.60% (299,317)	5.18% (16,405)	99.78% (315,722)	11.5068	79.11
	DCLM	83.40% (263,897)	96.60% (305,647)	3.25% (10,289)	99.85% (315,936)	10.9178	25.28
	C4	82.55% (261,209)	95.86% (303,326)	3.94% (12,474)	99.80% (315,800)	11.0748	44.05
Social IQA	FineWeb-Edu	79.08% (877,869)	96.64% (1,072,859)	3.34% (37,045)	99.98% (1,109,904)	11.6230	38.54
	DCLM	81.73% (907,341)	98.47% (1,093,097)	1.52% (16,923)	99.99% (1,110,020)	11.1652	38.74
	C4	81.46% (904,356)	97.47% (1,081,988)	2.52% (27,954)	99.98% (1,109,942)	11.3185	38.56
SWAG	FineWeb-Edu	78.33% (4,365,802)	95.48% (5,321,835)	4.51% (251,304)	99.99% (5,573,139)	11.4149	39.93
	DCLM	79.76% (4,445,550)	97.31% (5,423,857)	2.69% (149,703)	99.99% (5,573,560)	11.1052	43.05
	C4	80.08% (4,463,317)	96.63% (5,385,779)	3.34% (186,327)	99.97% (5,572,106)	11.2427	41.31

Table 18: 402M model, 8.5B-token subset 5. Conventions as in Table 12.

Task	Dataset	∼80%	∼95%	95–100%	Seen	Entropy	Score
ARC-Easy	FineWeb-Edu	83.85% (150,383)	98.11% (175,963)	1.87% (3,353)	99.98% (179,316)	11.1883	56.00
	DCLM	78.50% (140,792)	97.19% (174,316)	2.79% (5,009)	99.98% (179,325)	11.5521	50.34
	C4	77.74% (139,430)	96.63% (173,307)	3.34% (5,988)	99.97% (179,295)	11.6494	47.18
ARC-Challenge	FineWeb-Edu	84.91% (93,273)	98.13% (107,792)	1.83% (2,006)	99.95% (109,798)	11.0771	23.38
	DCLM	80.27% (88,171)	97.41% (107,004)	2.54% (2,795)	99.95% (109,799)	11.3884	21.16
	C4	79.89% (87,754)	97.05% (106,603)	2.88% (3,162)	99.92% (109,765)	11.4704	20.73
HellaSwag	FineWeb-Edu	82.72% (6,902,594)	96.27% (8,033,174)	3.66% (304,978)	99.93% (8,338,152)	11.1998	31.79
	DCLM	82.72% (6,901,953)	96.91% (8,086,108)	3.04% (253,697)	99.95% (8,339,805)	11.0597	31.95
	C4	83.68% (6,982,062)	97.00% (8,093,623)	2.93% (244,707)	99.93% (8,338,330)	11.0361	32.35
MMLU	FineWeb-Edu	80.27% (952,532)	95.93% (1,138,403)	4.00% (47,427)	99.93% (1,185,830)	11.3638	24.29
	DCLM	78.64% (933,176)	95.95% (1,138,561)	4.00% (47,418)	99.94% (1,185,979)	11.4081	23.72
	C4	78.46% (931,051)	95.08% (1,128,318)	4.82% (57,151)	99.90% (1,185,469)	11.5215	22.95
SciQ	FineWeb-Edu	80.35% (961,072)	95.79% (1,145,712)	4.15% (49,618)	99.94% (1,195,330)	11.4544	78.20
	DCLM	74.90% (895,940)	94.07% (1,125,194)	5.86% (70,066)	99.93% (1,195,260)	11.8623	77.00
	C4	73.69% (881,372)	92.98% (1,112,163)	6.88% (82,347)	99.87% (1,194,510)	12.0320	70.30
OBQA	FineWeb-Edu	79.59% (105,119)	96.62% (127,615)	3.35% (4,426)	99.97% (132,041)	11.5377	22.40
	DCLM	77.38% (102,203)	96.48% (127,430)	3.49% (4,614)	99.98% (132,044)	11.6540	19.80
	C4	77.21% (101,976)	95.97% (126,747)	3.99% (5,276)	99.96% (132,023)	11.7396	18.00
PIQA	FineWeb-Edu	74.91% (707,530)	95.55% (902,477)	4.41% (41,692)	99.97% (944,169)	11.6064	66.32
	DCLM	75.42% (712,317)	96.67% (913,018)	3.31% (31,287)	99.98% (944,305)	11.4889	66.43
	C4	77.37% (730,737)	97.26% (918,583)	2.71% (25,560)	99.96% (944,143)	11.3882	67.74
LAMBADA	FineWeb-Edu	81.03% (256,387)	94.45% (298,855)	5.33% (16,863)	99.78% (315,718)	11.5124	80.26
	DCLM	83.39% (263,876)	96.59% (305,630)	3.26% (10,306)	99.85% (315,936)	10.9242	26.06
	C4	82.76% (261,869)	96.07% (303,986)	3.92% (12,414)	99.80% (315,800)	11.0568	42.44
Social IQA	FineWeb-Edu	79.05% (877,575)	96.58% (1,072,198)	3.40% (37,717)	99.98% (1,109,915)	11.6217	38.28
	DCLM	81.73% (907,279)	98.46% (1,093,055)	1.53% (16,981)	99.99% (1,110,036)	11.1589	39.15
	C4	81.47% (904,381)	97.84% (1,086,151)	2.14% (23,796)	99.98% (1,109,947)	11.3182	38.43
SWAG	FineWeb-Edu	78.21% (4,359,356)	95.37% (5,315,578)	4.62% (257,541)	99.99% (5,573,119)	11.4170	40.03
	DCLM	79.73% (4,444,270)	97.30% (5,423,387)	2.69% (150,193)	100.00% (5,573,580)	11.1005	42.98
	C4	80.07% (4,463,112)	96.63% (5,385,922)	3.34% (186,345)	99.97% (5,572,267)	11.2422	41.31

D.5 Alternative lexical alignment measures

The corpus-ranking result of Section 4.2 would be less informative if any lexical-overlap score reproduced it. We therefore repeat the per-task corpus-ranking test with four alternatives, bigram cross-entropy (the same estimator at bigram order), unigram cosine and TF–IDF cosine similarity over the same frequency vectors (the latter weighted by corpus document-frequency IDF), and mean perplexity under a trigram LM trained on the benchmark text (the perplexity scorer is evaluated on its own corpus sample). All count-based measures are computed from matched corpus samples (six files per corpus, approximately $300{,}000$ rows per file, fixed seed). A measure agrees on a task when it orders DCLM, C4, and FineWeb-Edu as the trained models' accuracies do at the $3.36\text{B}/60\text{B}$ operating point.

Table 19: Corpus-ranking agreement per task (a ✓ means the measure reproduces the trained-accuracy ordering of the three corpora).

Task	Accuracy order	UCE	Bigram CE	Ppl	Cosine	TF–IDF
ARC-E	FwEdu > DCLM > C4	✓	✓	✓	✓	×
ARC-C	FwEdu > DCLM > C4	✓	✓	✓	✓	×
HellaSwag	C4 > DCLM > FwEdu	✓	×	✓	✓	×
MMLU	FwEdu > DCLM > C4	✓	×	✓	✓	×
SciQ	FwEdu > DCLM > C4	✓	✓	✓	✓	×
OBQA	FwEdu > DCLM > C4	✓	✓	×	×	×
PIQA	C4 > DCLM > FwEdu	✓	✓	×	✓	×
SIQA	DCLM > C4 > FwEdu	✓	✓	✓	✓	×
SWAG	DCLM > C4 > FwEdu	✓	✓	✓	×	✓
Total		$9/9$	$7/9$	$7/9$	$7/9$	$1/9$

The accuracy orderings reverse across tasks, so no single quality axis can pass this test, and UCE is the only measure that reproduces all nine orderings. TF–IDF reweights the same word counts UCE uses yet reproduces only one ordering, so the signal lies in the probabilistic form of UCE rather than in shared vocabulary. The alternatives also select substantially different documents. Each measure's top-20% selection overlaps the UCE-filter selection at a Jaccard of 0.170 and 0.191 for unigram cosine (DCLM and FineWeb-Edu), 0.143 and 0.104 for bigram cross-entropy, and 0.266 and 0.412 for TF–IDF, against 0.111 for two independent random selections of the same size, with rank correlations between -0.11 and $+0.46$. The benchmark-trigram perplexity scorer overlaps at 0.26 to 0.33 and, unlike UCE, has to be trained.

D.6 How far apart are the corpora themselves

UCE is directional by construction, since replacing the benchmark Q with a corpus changes the entropy term across comparisons. For a symmetric view of the corpus space we report pairwise Jensen–Shannon divergence between the corpus unigram distributions, computed without smoothing on 8.5B-token samples per corpus (OpenWebText is supply-limited and contributes 1.27B tokens). The diagonal holds the divergence between two disjoint halves of the same corpus, a sampling-noise floor that is larger for the smaller OpenWebText sample. The smallest cross-corpus divergence is 15 times the largest floor, so the separations are not sampling noise. DCLM and OpenWebText are the closest pair, and FineWeb-Edu is the most distinct, every pair involving it exceeding every pair that does not.

Table 20: Pairwise Jensen–Shannon divergence between corpus unigram distributions (bits). Diagonals (italic) are split-half noise floors.

	FineWeb-Edu	DCLM	C4	OpenWebText
FineWeb-Edu	*0.000196*	0.0506	0.0447	0.0557
DCLM	0.0506	*0.000282*	0.0325	0.0228
C4	0.0447	0.0325	*0.000204*	0.0380
OpenWebText	0.0557	0.0228	0.0380	*0.001490*

D.7 Where the scope of the statistic ends

UCE measures one property of the data, the word distribution, and nothing above it. Grammar is determined by word order, arithmetic by multi-step structure, and a non-English benchmark shares little vocabulary with an English corpus, so a unigram statistic should not predict any of the three. We report the measurement because the boundary is what makes the positive results interpretable. A benchmark within the scope of UCE would show falling scores as UCE rises, and none of the three families does so on the corpus-trained grid of Section 4.2. Every BLiMP word is seen in every corpus and the lowest-UCE corpus scores worst. On MathQA the UCE values are near identical and every score sits at the 20% chance floor. On Global-PIQA the ordering runs the wrong way, with the most aligned language scoring lowest.

Table 21: Capability families outside the scope of UCE. Entries within each row are ordered by rising UCE.

Benchmark	Compared across (rising UCE)	UCE (bits)	Score
BLiMP (grammar, 1.33B)	DCLM / C4 / FineWeb-Edu	12.42 / 12.66 / 12.87	61.0 / 80.5 / 80.9
MathQA (arithmetic, 1.33B)	DCLM / FineWeb-Edu / C4	11.34 / 11.34 / 11.41	23.3 / 23.3 / 22.1
Global-PIQA (non-English, 3.36B)	Fr / Es / De / Zh / Ar	19.8 / 20.9 / 22.5 / 27.2 / 28.8	46 / 50 / 58 / 49 / 52

E UCE-filter (document-level): algorithm, baselines, and result details

E.1 UCE-filter algorithm details (Section 5.1)

Sign convention and selection rule. A row whose word distribution is closer to Q contributes more to lowering $H(Q, P)$, so removing it raises the corpus–benchmark cross-entropy ($\Delta H_r > 0$). A row whose word distribution is farther from Q contributes more to raising $H(Q, P)$, and removing it lowers the cross-entropy ($\Delta H_r < 0$). The UCE-filter discards the rows with the smallest ΔH_r, those whose presence most increases $H(Q, P)$, and retains the rest.

Row-local efficiency. Removing a row changes only the counts of words in that row, so ΔH_r can be computed analytically from row-local counts and the global benchmark-aware corpus counts, without an expensive re-estimation of the full training distribution. The per-row cost depends only on the number of unique benchmark-overlapping words in the row, which is typically small relative to $|\mathcal{V}_b|$. The procedure therefore reduces to two passes over the source corpus, the first to compute the benchmark-aware training counts $c(w)$ over $w \in \mathcal{V}_b$ and the baseline $H(Q, P)$, and the second to evaluate ΔH_r for every row from its sparse local counts and the global counts.

Parallel implementation. Both passes are independent across rows. The counting pass reduces per-shard counts into a global $c(w)$ vector restricted to $\mathcal{V}_b$, and the scoring pass runs in batches across multiple workers. Our implementation scans the data in parallel and scales linearly with the number of CPU cores until shared-filesystem bandwidth becomes the bottleneck at our throughput.

E.2 DSIR baseline details (Section 5.2)

We compare the UCE-filter against two variants of the DSIR importance-resampling baseline [43], evaluated under the same source corpora, target distribution Q, and 70B-token training budget as the UCE-filter and the random control.

Reproducing the DSIR algorithm (DSIR-paper). We follow Algorithm 1 of Xie et al. [43][8] exactly. Each document x is mapped to a vector of hashed unigram and bigram counts using MurmurHash3 into $B = 10{,}000$ buckets. Bigram distributions P_t and P_s are estimated on the concatenated CSR7 benchmark text (the same target text that defines Q for our UCE-filter) and on a uniform sample of the source corpus, with add-one smoothing. The per-document log-importance is

$$\log w(x) \;=\; \sum_{n\text{-gram}\in x} \log \tfrac{P_t(n)}{P_s(n)},$$

[8] Reference implementation released by the authors at `https://github.com/p-lambda/dsir`.

where the sum ranges over all n-gram occurrences in x rather than over unique n-grams. Selection uses the Gumbel top-K trick. We add i.i.d. $\mathrm{Gumbel}(0, 1)$ noise to each $\log w(x)$ and retain the top K, which is equivalent to importance resampling without replacement. We grow K until the selected subset contains at least 70B tokens, matching the budget used by all other selection methods (this required $K \approx 300$M docs on DCLM and $K \approx 220$M on FineWeb-Edu).

Length-normalized ablation (DSIR-LN). DSIR-LN scores each document by the mean rather than the sum log-importance,

$$\bar{w}(x) \;=\; \tfrac{1}{|x|_{n\text{-gram}}} \sum_{n\text{-gram}\in x} \log \tfrac{P_t(n)}{P_s(n)}.$$

We use deterministic top-K since length normalization reduces the dependence on $|x|$.

Why both variants. The sum-based DSIR-paper variant has a length bias. Its score adds one log-ratio per n-gram occurrence, and these ratios are negative on average over source text, so the score of a typical source document falls as the document grows and the selection favors short documents. This is why K had to reach about 300M documents on DCLM and 220M on FineWeb-Edu to fill the budget. We report DSIR-paper as the primary baseline because it matches the published algorithm verbatim, and DSIR-LN, which reduces this confound, as an ablation that shows what DSIR's content selection achieves with less dependence on length.

Empirical effect of length normalization. Length normalization recovers most of the gap between DSIR-paper and the UCE-filter on the Targeted suite. The numbers below compare DSIR-LN to the UCE-filter at a matched single training seed (seed 1234, the only seed available for DSIR-LN), so that the contrast isolates the length-normalization effect from seed averaging. The UCE-filter's three-seed means (Tables 2 and 32) lie within 0.2 pp of these single-seed values on the Targeted suite, 0.3 pp on the Overall average, and 0.9 pp on the out-of-target average. On Targeted (CSR7), DSIR-LN reaches 62.88% on 1.34B DCLM and 60.97% on 1.34B FineWeb-Edu, against the UCE-filter's 62.39% and 61.96%, and at 3.33B DSIR-LN reaches 65.54% on DCLM and 64.24% on FineWeb-Edu, against the UCE-filter's 66.07% and 64.76%. The Targeted gap between DSIR-LN and the UCE-filter falls within ± 1 pp in three of the four (Source, Scale) cells, with DSIR-LN slightly ahead on 1.34B DCLM ($+0.48$ pp) and the UCE-filter ahead in the other three cells by $+0.52$ to $+0.99$ pp (Table 32). On the 11-task Overall, however, the UCE-filter leads in every cell by $+0.85$ to $+1.00$ pp (56.84 vs. 55.98 on 1.34B DCLM, 55.15 vs. 54.15 on 1.34B FineWeb-Edu, 60.09 vs. 59.24 on 3.33B DCLM, and 57.68 vs. 56.79 on 3.33B FineWeb-Edu), since it does not pay the same out-of-target cost on LAMBADA that DSIR-LN inherits from the hashed-bigram importance score. On the 4-task out-of-target average, the UCE-filter leads by $+1.00$ to $+3.20$ pp in the same cell order (47.11 vs. 43.91, 43.22 vs. 42.22, 49.63 vs. 48.21, and 45.31 vs. 43.74). The DSIR-paper σ_{select} on the Targeted suite is at most 0.56 pp in every n_{select}=3 cell (Appendix E.9), smaller than the DSIR-LN versus UCE-filter differences quoted above. Unlike the hashed-bigram importance score, the UCE-filter objective has an information-theoretic interpretation as D_{KL} between corpus and benchmark word distributions.

E.3 Embedding-based selection (BETR)

BETR [45] ranks documents by embedding similarity to benchmark items and distills the ranking into a fastText-style classifier. With the CSR7 items as its targets, BETR selects for the same benchmarks as the UCE-filter and scores alignment by embedding similarity, where the UCE-filter uses a corpus-to-benchmark cross-entropy. We run BETR with three training seeds through the same 2×2 grid of source and scale and the same 70B-token from-scratch recipe as the UCE-filter (Appendix C). The Random, DSIR-paper, and UCE-filter columns of Table 22 repeat Table 32, which also lists BETR task by task.

BETR raises Targeted accuracy over the random control in all four cells, but at 3.33B it trails the UCE-filter by 0.7 to 1.5 pp and falls below the random control on the 11-task Overall average (57.57 versus 57.78 on DCLM and 55.15 versus 56.27 on FineWeb-Edu). The UCE-filter exceeds the random control on both Targeted and Overall in every cell. At 1.34B the UCE-filter leads BETR on FineWeb-Edu on both averages (61.85 versus 60.63 on Targeted and 55.14 versus 52.94 on Overall), while on DCLM BETR scores 62.96 on Targeted against 62.49 for the UCE-filter yet still trails it

by 1.5 pp on Overall. On the out-of-target average, BETR falls below the random control in every cell, by 5.10 and 5.22 pp on DCLM and by 3.78 and 3.11 pp on FineWeb-Edu at 3.33B and 1.34B respectively, while the UCE-filter stays within 0.9 pp of the control. Most of this loss comes from LAMBADA, where BETR scores 10.3 to 19.2 pp below the random control, while the other three out-of-target tasks stay within 2.3 pp of it (Table 32).

Against CSR7, the BETR subsets reach a corpus-to-benchmark UCE of 11.08 bits on DCLM and 11.25 bits on FineWeb-Edu (Table 23), at the level of DSIR and 0.16 to 0.32 bits above the UCE-filter. BETR attains its Targeted gains while lowering CSR7 UCE by only 0.07 to 0.09 bits relative to the random control, against 0.25 to 0.39 bits for the UCE-filter. On OOT-4 the UCE-filter stays within 0.04 bits of the random control, while BETR sits 0.3 to 0.4 bits above it.

Table 22: Zero-shot accuracy (%) of the UCE-filter and of embedding-based selection (BETR) with Random and DSIR-paper, across sources and scales. Mean ± standard deviation over 3 training seeds (1234, 42, 7777), under the per-task metric convention of Table 32. Targeted averages the 7 tasks used for selection (ARC-Easy, ARC-Challenge, HellaSwag, SciQ, OpenBookQA, PIQA, SWAG), Out-of-target the 4 other tasks (LAMBADA, CommonsenseQA, BoolQ, SocialIQA), and Overall all 11. **Bold** marks the best mean in each row. The UCE-filter column is shaded.

Source	Model	Metric	Random	DSIR-paper	UCE-Filter	BETR
DCLM	3.33B	Targeted	62.95±0.03	63.01±0.33	**66.26**±0.20	65.53±0.34
		Out-of-target	48.75±0.53	46.23±0.38	**49.56**±0.09	43.65±0.25
		Overall	57.78±0.17	56.91±0.35	**60.19**±0.14	57.57±0.15
	1.34B	Targeted	58.85±0.56	59.85±0.53	62.49±0.26	**62.96**±0.43
		Out-of-target	**46.57**±0.53	43.79±0.23	46.22±0.94	41.35±0.46
		Overall	54.38±0.54	54.01±0.28	**56.58**±0.23	55.10±0.25
FineWeb-Edu	3.33B	Targeted	62.80±0.09	62.77±0.18	**64.64**±0.21	63.19±0.14
		Out-of-target	44.85±0.49	43.39±0.33	**45.61**±0.27	41.07±0.42
		Overall	56.27±0.22	55.72±0.24	**57.72**±0.14	55.15±0.10
	1.34B	Targeted	59.12±0.41	60.15±0.08	**61.85**±0.15	60.63±0.49
		Out-of-target	42.60±1.06	41.14±1.37	**43.40**±0.29	39.49±0.36
		Overall	53.11±0.64	53.24±0.45	**55.14**±0.02	52.94±0.33

Table 23: Corpus-to-benchmark UCE (bits ↓) on the selected subsets against the 7-task targeted suite (CSR7) and the 4 out-of-target tasks (OOT-4), with the BETR selection added to the methods of Table 1. OOT-4 values are measured with the pipeline of Appendix A.1 on a fixed-seed sample of each subset, so absolutes are not comparable across the CSR7 and OOT-4 blocks. **Bold** marks the lowest CSR7 value in each row. The UCE-filter column is shaded.

Target	Source	Random	DSIR [43]	BETR [45]	UCE-Filter
CSR7	DCLM	11.17	11.08	11.08	**10.92**
	FineWeb-Edu	11.32	11.20	11.25	**10.93**
OOT-4	DCLM	11.39	11.43	11.79	11.41
	FineWeb-Edu	11.62	11.65	11.93	11.58

E.4 Higher-order conditional filters

Section 3.2 argues that higher-order conditional estimates are limited by context sparsity and finite-sample bias. The document scope allows a direct test. We apply a conditional 3-gram cross-entropy filter to the same 630M-document DCLM pool at the same 20% keep rate, with unseen contexts smoothed toward the unigram distribution by a Dirichlet prior of strength β, so that $P(w \mid c) = (N(c, w) + \beta\,\pi(w))/(N(c) + \beta)$ tends to the unigram $\pi(w)$ as β grows. Sweeping β therefore moves the filter continuously from the third order toward the unigram filter. Every subset is measured against the random control in one pass under the fixed-seed sampling convention of the OOT-4 block of Table 1. The alignment recovered by the order-3 filter rises monotonically with β yet reaches only three quarters of the unigram filter's at $\beta = 10^4$, and the out-of-target UCE stays at the control level throughout (Table 24). Since the shortfall persists across the sweep, the third order loses

selection-level alignment through sparse conditional counts rather than through any one smoothing choice.

Table 24: Selected-subset UCE relative to the random control (bits), for the conditional 3-gram filter across smoothing strengths β and for the unigram UCE-filter.

Selection rule	**vs CSR7**	**vs OOT-4**
order-3, $\beta = 10$	-0.183	-0.004
order-3, $\beta = 10^2$	-0.194	$+0.006$
order-3, $\beta = 10^3$	-0.202	$+0.009$
order-3, $\beta = 10^4$	-0.209	$+0.013$
UCE-filter (unigram)	$\mathbf{-0.274}$	$+0.016$

The selection-level gap does not translate into a downstream gap at this scale. We train the order-3 ($\beta = 10$) selection from scratch at 1.34B on DCLM under the paper's recipe and evaluate it against the same-seed runs of the other rules (Table 25). Its targeted average is identical to the UCE-filter's to two decimals, and both sit far above the random and DSIR controls, while the UCE-filter keeps a small out-of-target edge (0.85 pp, within the 0.94 pp seed noise of that column). The subset-UCE gap of Table 24 therefore overstates the downstream difference between filter orders, and the downstream gains appear to be a property of the shared cross-entropy objective rather than of the unigram estimator specifically. The unigram retains the practical advantage of needing only marginal counts, so it avoids conditional sparsity and is cheaper to fit at corpus scale.

Table 25: Zero-shot accuracy (%) of the trained order-3 ($\beta = 10$) selection and of the paper's rules (1.34B, DCLM, one matched training seed). The three-seed standard deviations of these arms in Table 2 reach 0.56 pp on the targeted average and 0.94 pp on the out-of-target average. Targeted averages the seven selection targets, Out-of-target the four held-out tasks, Overall all eleven.

Selection rule	**Targeted**	**Out-of-target**	**Overall**
order-3, $\beta = 10$	62.39	46.26	56.53
UCE-filter (unigram)	62.39	47.11	56.84
Random	58.26	45.97	53.79
DSIR-paper	59.42	44.06	53.84

E.5 Held-out target transfer

Because the target distribution Q is estimated from the targeted tasks, the UCE-filter gains of Section 5.2 are an upper bound under a known target rather than zero-shot generalization to unseen targets. We test whether the gain reflects target-directed alignment or merely the selection of benchmark-adjacent text. We define a held-out target T_B of BoolQ, SocialIQA, and LAMBADA, disjoint from the CSR7 suite, apply the UCE-filter with T_B as the target, train the selection from scratch at 3.33B on 70B tokens under the recipe of Appendix C, and evaluate on both CSR7 and T_B (Table 26). These are the reliably measurable non-CSR7 benchmarks at this scale, since a 3.33B model trained on 70B tokens scores above chance mostly on English commonsense benchmarks. Retargeting also changes the selection itself materially, replacing 55% of the documents the filter keeps on the same FineWeb-Edu pool at the same 20% budget, against a 20% chance overlap for two independent same-size selections.

Retargeting the selection moves accuracy toward the new target at the expense of the original one. On T_B, the T_B-targeted selection beats the random control by 1.20 pp on DCLM and 1.85 pp on FineWeb-Edu, and the CSR7-targeted selection lifts T_B by comparable amounts, so on that suite the choice of target barely registers. On CSR7 the same choice decides the sign, with the T_B selection 1.68 pp and 1.67 pp below random and the CSR7-targeted selection 3.31 pp and 1.84 pp above it. A memorization-based explanation, in which benchmark-adjacent text benefits all benchmarks, would instead predict a positive gain on CSR7 from the T_B selection.

The relation between the two benchmark groups can be measured directly at the word level, with no training corpus involved. Table 27 reports the cross-UCE between the CSR7 and OOT-4 group distri-

Table 26: Held-out target transfer (3.33B). Each row is a from-scratch model trained on a UCE-filter selection targeting either the CSR7 suite or the held-out target T_B (BoolQ, SocialIQA, LAMBADA), or a row-count-matched random control, evaluated on both targets under the per-task metric convention of Table 32. CSR7 averages its 7 tasks and T_B its 3. All cells are means over three training seeds (1234, 42, 7777), evaluated in one pass, and the T_B rows additionally report σ.

Source	Selection target	**CSR7** (7)	T_B (3)
DCLM	CSR7	66.26	59.44
	T_B	61.27 $\pm$ 0.13	59.62 $\pm$ 0.44
	Random	62.95	58.42
FineWeb-Edu	CSR7	64.64	53.91
	T_B	61.13 $\pm$ 0.20	54.46 $\pm$ 0.25
	Random	62.80	52.61

butions in both directions, together with the $H(Q)$ and $D_{\mathrm{KL}}(Q\|P)$ decomposition of Appendix A and the share of target mass covered by the reference vocabulary. Each group covers 96 to 98 percent of the other group's probability mass, so the two groups share most of their commonsense English vocabulary. The divergence is nevertheless asymmetric, and encoding the out-of-target words under the CSR7 distribution costs 1.51 bits of divergence against 1.01 bits in the reverse direction. The two groups therefore stand in partial overlap with residual heterogeneity rather than inclusion or independence. This structure is consistent with the transfer pattern of Table 26. The residual divergence is the component that targeted selection redistributes, which is why the UCE-filter lowers UCE against its own target while leaving OOT-4 UCE at the random control (Table 1).

Table 28 repeats the measurement per task. Because each group marginal contains the task itself, closeness to the own group is partly built in (HellaSwag and SWAG alone carry 84.4% of the CSR7 word mass, and BoolQ and SocialIQA carry 81.0% of OOT-4). Under each of two controls that remove this bias (a leave-one-out marginal and a size-matched leave-one-out marginal), six of the seven tasks holding under 11% of their own group's mass (ARC-Challenge, ARC-Easy, OpenBookQA, PIQA, CommonsenseQA, LAMBADA) remain 0.13 to 0.77 bits closer to their own group, SciQ crosses to the OOT-4 side, and the ordering for the mass-dominant tasks is not separately identifiable. The separation from the CSR7 marginal is widest for SocialIQA and BoolQ.

Table 27: Benchmark-group cross-UCE (bits ↓). Word-level unigram cross-entropy between the two benchmark groups, with no training corpus involved. The target Q sets the vocabulary and the marginal, and the reference P supplies counts over that vocabulary under the zero-count smoothing of Appendix A.1. Coverage is the share of the target's probability mass whose words occur in the reference. Diagonal rows reduce to the marginal entropy $H(Q)$.

Target Q	**Reference** P	**Coverage (%)**	$D_{\mathrm{KL}}(Q\|P)$	**UCE** $H(Q,P)$
CSR7	CSR7 (self)	100.00	0.000	10.21
CSR7	OOT-4	97.77	1.008	11.21
OOT-4	OOT-4 (self)	100.00	0.000	10.73
OOT-4	CSR7	96.02	1.509	12.24

E.6 Robustness of the target distribution and the retention rate

Rebuilding Q from train splits alone. Q pools every released split of the targeted datasets, including test, since manipulating lexical alignment requires a fully specified target distribution and Q carries only word counts, no items and no answer keys. Rebuilding Q from the train splits alone and re-measuring every selection subset in one pass moves no subset's UCE by more than 0.013 bits and leaves the contrast that defines the experiment unchanged, with a UCE-filter minus Random gap of -0.274 against -0.267 bits on DCLM and -0.361 against -0.350 on FineWeb-Edu (all-splits against train-only, both under the fixed-seed sampling convention, whose gaps differ from the full-scan 0.25 and 0.39 bits of Table 1 by at most 0.03 bits). The alignment effect therefore comes from word statistics that train and test splits share, not from the presence of test text.

Table 28: Per-task cross-UCE (bits ↓). Each task's word distribution measured against the two group marginals, which include the task itself, so closeness to the own group is partly built in for the mass-dominant tasks (see text).

Group	Task	UCE vs CSR7	UCE vs OOT-4
CSR7	ARC-Challenge	10.87	11.16
	ARC-Easy	11.01	11.36
	HellaSwag	10.24	11.13
	OpenBookQA	11.18	11.36
	PIQA	10.45	11.39
	SciQ	11.46	11.82
	SWAG	9.63	10.82
OOT-4	LAMBADA	11.63	10.72
	CommonsenseQA	11.67	10.97
	SocialIQA	11.82	9.55
	BoolQ	12.69	11.45

Retention rate. Alignment strengthens smoothly and with diminishing returns as retention tightens (Table 29), and the paper's 20% operating point already captures most of the available UCE reduction. At 20% the kept pools span roughly 227B GPT-2 tokens on DCLM and 90B on FineWeb-Edu. At 5% they fall below the 70B-token training budget on both sources, so more aggressive settings force data repetition.

Table 29: Selected-subset UCE against CSR7 (bits ↓) as a function of the retention rate, measured on fixed-seed samples of each subset, the convention of the OOT-4 block of Table 1. Absolute values sit slightly below that table's full-scan CSR7 column, so within-column deltas are the quantity of interest. The 100% row is the unfiltered corpus.

Retention	DCLM	FineWeb-Edu
5%	10.769	10.814
10%	10.779	10.841
20% (paper)	10.801	10.883
40%	10.836	10.944
60%	10.871	10.997
80%	10.929	11.069
100%	11.075	11.245

E.7 30B-scale Stage-2 pre-training case study: setup and caveats (Section 5.3)

Setup. The model is an in-house 30B-parameter hybrid Mamba–Transformer mixture-of-experts language model in the spirit of Nemotron 3 Nano [46]. Pre-training proceeds in two stages. Stage 1 is a from-scratch run on a generic web mixture, and its endpoint initializes Stage 2, which continues pre-training on a production-grade mixture of web, code, math, and supervised data. The UCE-filter variant replaces the web component of the Stage 2 mixture with its UCE-filtered subset and leaves the code, math, and supervised components unchanged.

Caveats. The case study reports a single (model, mixture) operating point with a single seed, so the MMLU-Pro margin (under 0.2 percentage points) should be interpreted as evidence of a token-efficient match rather than of a meaningful absolute improvement. The Stage 2 continuation is replay-only with no fresh data injection. Stage 2 hyperparameters and infrastructure are inherited from the production pipeline rather than re-tuned for the UCE-filtered regime. The model is in-house, and we do not audit the Stage 2 mixture for benchmark contamination beyond standard production controls.

E.8 A three-seed replication and 5-shot evaluation (Section 5.2)

To establish that the means in Table 2 are not driven by a single training seed, every cell of the 2×2 Source $\times$ Scale grid was retrained under two additional training seeds (42 and 7777) alongside the original 1234, for each of the three methods (UCE-filter, Random, DSIR-paper). The full sweep is 36 from-scratch runs (12 cells $\times$ 3 seeds). The tokenized subsets are held fixed across seeds, so the spread reflects training-seed effects only (initialization, dropout, and data shuffle). DSIR-LN is a single-seed length-normalization ablation (Appendix E.2). Targeted UCE is selection-deterministic for the UCE-filter, and for Random and DSIR-paper it varies by less than 0.001 bits across three independent data-selection seeds (42, 100, and 200) on the same source corpora. The Targeted UCE values in Table 1 therefore measure an intrinsic property of each selection method rather than a sampling artifact.

Table 2 (zero-shot) and Table 34 (5-shot) report the Targeted (7-task) average as mean $\pm\sigma$ across the three training seeds. In every cell the UCE-filter mean is the highest of the three methods, the same ordering holds for every individual seed, and the per-method standard deviation is small (UCE-filter $\sigma \leq 0.36$ pp, Random and DSIR-paper $\sigma \leq 0.66$ pp). The UCE-filter's mean lead over both baselines exceeds the 95% confidence-interval half-width in every (Source, Scale) cell, in both the zero-shot and 5-shot settings, so the gap is not absorbed by training-seed noise and is not specific to a particular scale or prompt format. With $n=3$ training seeds we report these as descriptive observations rather than as a formal hypothesis test, but Welch's two-sample t-test on the same data yields $p < 0.005$ for the UCE-filter vs. Random and UCE-filter vs. DSIR-paper comparisons in every (Source, Scale, setting) cell.

E.9 Selection-seed sweep on the stochastic baselines (Section 5.2)

We vary the data-selection seed of the two stochastic methods (Random and DSIR-paper) over $\{42, 100, 200\}$ at a fixed training seed 42. The UCE-filter is selection-deterministic (Section 5.1), so $\sigma_{\text{select}}=0$ for every cell. The DCLM $\times$ Random cell at each scale was not extended beyond the original selection seed and is reported at $n_{\text{select}}=1$, and the six other (Source, Scale, Method) cells at $n_{\text{select}}=3$.

Table 30 reports the Targeted-suite mean and σ_{select} per cell, alongside σ_{train} from Table 2. On the six $n_{\text{select}}=3$ cells, σ_{select} is at most 0.56 pp on the Targeted suite, of the same order as σ_{train}. The method ordering of Table 2 is recovered at every individual selection seed.

Table 30: Targeted-suite mean and σ_{select} from selection seeds $\{42, 100, 200\}$ at training seed 42, alongside σ_{train} reproduced from Table 2 (training seeds $\{1234, 42, 7777\}$ at the original selection seed). The UCE-filter is selection-deterministic and the column reports its value at training seed 42.

Scale	Source $\times$ Method	Mean (Avg)	σ_{select} (pp)	σ_{train} (pp)
1.34B	DCLM $\times$ UCE-filter	62.30	0 (det.)	0.26
	DCLM $\times$ DSIR-paper	60.20	0.56	0.53
	DCLM $\times$ Random	59.36†	n/a (n=1)	0.56
	FineWeb-Edu $\times$ UCE-filter	61.68	0 (det.)	0.15
	FineWeb-Edu $\times$ DSIR-paper	59.84	0.48	0.08
	FineWeb-Edu $\times$ Random	58.95	0.26	0.41
3.33B	DCLM $\times$ UCE-filter	66.23	0 (det.)	0.20
	DCLM $\times$ DSIR-paper	63.26	0.12	0.33
	DCLM $\times$ Random	62.91†	n/a (n=1)	0.03
	FineWeb-Edu $\times$ UCE-filter	64.39	0 (det.)	0.21
	FineWeb-Edu $\times$ DSIR-paper	62.58	0.23	0.18
	FineWeb-Edu $\times$ Random	62.85	0.45	0.09

† Single selection seed (42). Selection seeds 100 and 200 at this cell were not run.

F Contamination audit and removal

Contamination could inflate both lexical alignment and benchmark accuracy at once, so we audit it independently rather than relying on the corpus curators' procedures, and then remove it at both the evaluation end and the training end.

Protocol. The audit follows the word-level 13-gram protocol of GPT-3 [56], with the scan repeated under the protocol variants of Brown et al. [56], Messmer et al. [73], Chowdhery et al. [74] and the 80% near-duplicate criterion of Llama 2 [26]. The n-grams are built with the filter's own normalizer (NFKC, punctuation stripped, whitespace split, case preserved), so the audit measures the same text the metric measures. Each item enters as its full evaluation text (passage, question, and every option, with no answer field), the same text that defines the target distribution Q. The index covers 12 benchmarks and 303,139 items, and the item-share scan covers spread samples of up to one million documents from each of the four corpora. Nine of the twelve tasks match under 0.3% of items in every sample (Table 5). The three exceptions (BoolQ, SciQ, MMLU) peak on FineWeb-Edu, whose educational filtering overlaps encyclopedic passages, and the matches are passage-side surface overlap rather than answer leakage.

Removal at the evaluation end. We flag every scored evaluation item sharing a 13-gram with the corpus and re-score without them, comparing each arm at one matched training seed (three-seed means are in Table 2). On DCLM this removes 161 of 36,934 CSR7 items (HellaSwag 80, SciQ 65, ARC-Easy 8, PIQA 6, ARC-Challenge 2), and the UCE-filter advantage over the random control moves from 3.11 to 3.16 pp at 3.33B and from 4.13 to 4.16 pp at 1.34B. On FineWeb-Edu the removal covers 163 items (SciQ 106, HellaSwag 45, ARC-Easy 6, ARC-Challenge 4, PIQA 2), and the advantage moves from 2.05 to 2.10 pp at 3.33B and from 2.86 to 2.88 pp at 1.34B. The advantage is therefore unchanged in all four cells.

Removal at the training end. On the full DCLM source of 630M rows we remove every row sharing a 13-gram with the index (0.13% of rows), re-run each selection rule over the surviving rows at the same 20% budget (126M rows), and retrain 1.34B from scratch on 70B tokens under the paper's recipe, comparing each arm against its own training seed. Re-scanning one-million-document samples of each new subset finds zero remaining matches. The CSR7 accuracy of all three arms moves within run-to-run noise (Table 31). Each arm's subset UCE shifts by at most 0.005 bits under the same measurement pass, so the selection signal itself is untouched. The UCE-filter keeps $+3.40$ pp over the random control against $+4.13$ pp before, with the ordering unchanged.

Table 31: Decontaminate, re-select, and retrain from scratch (1.34B, DCLM, same-seed comparisons).

CSR7 accuracy (%)	UCE-filter	Random	DSIR-paper
Original	62.39	58.26	59.42
Decontaminated and retrained	62.17	58.77	59.88

A model whose training pool contains no benchmark overlap cannot have memorized the items it is scored on, and re-scoring without the overlapping items removes the evaluation-side channel, so the contamination explanation for the UCE-filter's gain over the random control is ruled out at both ends.

G Per-task UCE-filter accuracy: zero-shot and 5-shot (Section 5.2)

Table 32 complements the cell-level averages of Table 2 with the per-task zero-shot accuracy for each (Source, Scale, Method) cell of Section 5.2, together with BETR (Appendix E.3). The UCE-filter exceeds the random control on each of the seven targeted tasks in every (Source, Scale) cell, so the Targeted averages are not driven by one or two tasks.

G.1 Per-task UCE-filter 5-shot accuracy (Section 5.2)

Table 34 repeats the per-task breakdown of Table 32, without BETR, under 5-shot prompting on the same checkpoints, to verify that the UCE-filter advantage observed in the zero-shot setting is not

Table 32: Per-task zero-shot accuracy (%) for the UCE-filter runs of Section 5.2 and for BETR [45] with the same CSR7 targets, reported as mean $\pm\sigma$ across three independent training seeds (1234, 42, 7777). The left block holds the seven targeted tasks that define the target distribution Q, and the middle block the four out-of-target tasks. **Bold** marks the best mean among {UCE-filter, Random, DSIR-paper, BETR} within each cell. The DSIR-LN row (italics) is the single-seed length-normalized ablation of Appendix E.2 and is excluded from the bold comparison.

		Targeted (UCE-filter targets)							**Out-of-target**				**Average**	
Source	**Method**	ARC-E	ARC-C	HSwag	SciQ	OBQA	PIQA	SWAG	LAMB	CSQA	BoolQ	SIQA	**Targeted**	**Overall**
1.34B parameters														
DCLM	UCE-filter (p=0.20)	68.90 ± 0.26	38.17 ± 0.43	66.84 ± 0.09	90.47 ± 0.60	40.27 ± 1.17	77.86 ± 0.55	**54.96 ± 0.17**	**64.34 ± 0.21**	19.38 ± 1.17	57.84 ± 2.53	**43.31 ± 0.64**	62.49 ± 0.26	**56.58 ± 0.23**
	Random (p=0.20)	62.93 ± 1.39	34.22 ± 0.31	60.58 ± 0.45	89.60 ± 0.85	37.53 ± 0.42	74.90 ± 0.72	52.19 ± 0.14	60.67 ± 0.12	**20.78 ± 0.24**	**62.02 ± 0.97**	42.80 ± 1.10	58.85 ± 0.56	54.38 ± 0.54
	DSIR-paper [43]	65.17 ± 1.88	35.84 ± 1.49	63.00 ± 0.34	88.10 ± 0.78	38.80 ± 0.92	77.02 ± 0.54	50.99 ± 0.06	51.73 ± 0.24	19.49 ± 2.28	61.89 ± 1.97	42.05 ± 0.78	59.85 ± 0.53	54.01 ± 0.28
	BETR [45]	**71.25 ± 1.13**	**43.34 ± 1.19**	**67.10 ± 0.08**	**91.67 ± 1.50**	**40.87 ± 0.83**	**78.75 ± 0.41**	47.75 ± 0.16	43.02 ± 0.73	19.44 ± 0.55	61.28 ± 1.49	41.64 ± 0.50	**62.96 ± 0.43**	55.10 ± 0.25
	DSIR-LN (ablation)	*69.87*	*39.25*	*67.74*	*90.80*	*40.00*	*78.78*	*53.71*	*56.92*	*19.90*	*56.45*	*42.37*	*62.88*	*55.98*
FineWeb-Edu	UCE-filter (p=0.20)	68.38 ± 1.14	39.99 ± 0.86	**64.14 ± 0.35**	**91.20 ± 0.35**	**42.13 ± 0.76**	**76.22 ± 0.54**	**50.86 ± 0.06**	**50.84 ± 0.36**	19.27 ± 0.55	60.78 ± 1.56	**42.70 ± 0.24**	**61.85 ± 0.15**	**55.14 ± 0.02**
	Random (p=0.20)	67.16 ± 0.24	38.11 ± 0.60	57.81 ± 0.29	89.57 ± 0.15	39.27 ± 2.32	73.96 ± 0.44	47.96 ± 0.09	47.26 ± 0.42	**20.28 ± 0.98**	**61.33 ± 3.02**	41.52 ± 0.59	59.12 ± 0.41	53.11 ± 0.64
	DSIR-paper [43]	69.05 ± 0.35	39.70 ± 0.94	58.68 ± 0.14	89.07 ± 0.21	41.80 ± 0.80	74.74 ± 0.57	48.01 ± 0.17	43.42 ± 0.85	19.68 ± 0.17	60.04 ± 4.34	41.42 ± 1.42	60.15 ± 0.08	53.24 ± 0.45
	BETR [45]	**69.88 ± 2.13**	**41.75 ± 1.47**	59.74 ± 0.42	90.17 ± 0.49	41.73 ± 0.42	75.43 ± 0.58	45.71 ± 0.20	36.96 ± 0.52	19.30 ± 0.21	60.89 ± 1.76	40.81 ± 0.87	60.63 ± 0.49	52.94 ± 0.33
	DSIR-LN (ablation)	*67.85*	*39.93*	*63.11*	*89.00*	*41.40*	*75.57*	*49.91*	*47.45*	*20.56*	*59.72*	*41.15*	*60.97*	*54.15*
3.33B parameters														
DCLM	UCE-filter (p=0.20)	73.40 ± 0.54	44.65 ± 0.13	**72.71 ± 0.10**	93.10 ± 0.17	**43.20 ± 0.72**	80.29 ± 0.93	**56.46 ± 0.14**	**68.63 ± 0.49**	19.93 ± 0.47	65.54 ± 1.09	**44.17 ± 0.35**	**66.26 ± 0.20**	**60.19 ± 0.14**
	Random (p=0.20)	70.12 ± 0.40	39.59 ± 0.62	67.36 ± 0.14	92.13 ± 0.31	40.53 ± 0.95	76.71 ± 0.19	54.17 ± 0.11	65.78 ± 0.14	19.71 ± 0.56	**66.02 ± 0.96**	43.47 ± 0.69	62.95 ± 0.03	57.78 ± 0.17
	DSIR-paper [43]	68.73 ± 0.93	39.36 ± 0.10	68.96 ± 0.27	91.07 ± 0.95	40.87 ± 1.21	79.04 ± 0.30	53.05 ± 0.09	56.14 ± 0.30	**21.13 ± 1.42**	63.79 ± 1.38	43.84 ± 0.92	63.01 ± 0.33	56.91 ± 0.35
	BETR [45]	**75.18 ± 1.02**	**46.47 ± 0.69**	71.60 ± 0.22	**93.67 ± 0.15**	41.80 ± 1.22	**80.56 ± 0.39**	49.46 ± 0.02	46.61 ± 0.56	20.15 ± 0.08	64.04 ± 0.12	43.79 ± 0.65	65.53 ± 0.34	57.57 ± 0.15
	DSIR-LN (ablation)	*74.12*	*41.89*	*73.09*	*91.10*	*42.40*	*80.79*	*55.38*	*62.10*	*21.95*	*64.25*	*44.52*	*65.54*	*59.24*
FineWeb-Edu	UCE-filter (p=0.20)	72.66 ± 0.35	**45.02 ± 0.60**	**68.26 ± 0.19**	**93.17 ± 0.71**	43.73 ± 0.46	**77.38 ± 0.25**	**52.24 ± 0.20**	**54.33 ± 0.57**	20.72 ± 0.73	63.21 ± 1.15	**44.18 ± 0.58**	**64.64 ± 0.21**	**57.72 ± 0.14**
	Random (p=0.20)	71.94 ± 0.21	44.11 ± 0.51	63.49 ± 0.19	91.63 ± 0.38	43.13 ± 0.70	75.12 ± 0.46	50.19 ± 0.05	52.31 ± 0.06	**21.57 ± 0.45**	62.67 ± 1.13	42.85 ± 0.52	62.80 ± 0.09	56.27 ± 0.22
	DSIR-paper [43]	71.49 ± 0.42	42.89 ± 0.44	64.16 ± 0.07	91.73 ± 1.19	42.87 ± 0.12	76.21 ± 0.56	50.04 ± 0.22	47.83 ± 0.42	20.67 ± 0.63	62.49 ± 0.73	42.60 ± 0.28	62.77 ± 0.18	55.72 ± 0.24
	BETR [45]	**73.11 ± 0.67**	44.91 ± 0.57	64.13 ± 0.36	92.23 ± 0.21	**43.80 ± 1.22**	76.68 ± 0.17	47.51 ± 0.08	39.25 ± 0.47	19.27 ± 0.95	**63.62 ± 0.38**	42.14 ± 0.11	63.19 ± 0.14	55.15 ± 0.10
	DSIR-LN (ablation)	*72.26*	*42.49*	*67.97*	*91.90*	*45.20*	*78.02*	*51.84*	*52.30*	*21.79*	*59.36*	*41.50*	*64.24*	*56.79*

specific to a particular prompt format. Table 33 summarizes the same runs at the suite level, and the UCE-filter has the highest mean on the Targeted, Out-of-target, and Overall averages in every (Source, Scale) cell. At the task level the UCE-filter has the best mean of the three seeded methods on every Targeted task in the two DCLM cells, while on FineWeb-Edu Random edges it out on SciQ at 1.34B and on ARC-E and OBQA at 3.33B.

Table 33: Suite-level 5-shot accuracy (%), three-seed means. Targeted averages the seven selection targets, Out-of-target the four held-out tasks, Overall all eleven.

Scale	**Source**	**Method**	**Targeted**	**Out-of-target**	**Overall**
1.34B	DCLM	UCE-filter	**64.50**	**46.84**	**58.08**
		Random	61.22	45.80	55.61
		DSIR-paper	61.69	41.83	54.47
1.34B	FineWeb-Edu	UCE-filter	**63.74**	**42.23**	**55.92**
		Random	61.20	40.54	53.69
		DSIR-paper	61.65	39.20	53.49
3.33B	DCLM	UCE-filter	**67.57**	**50.23**	**61.26**
		Random	64.62	48.63	58.80
		DSIR-paper	64.86	44.24	57.36
3.33B	FineWeb-Edu	UCE-filter	**65.68**	**44.50**	**57.98**
		Random	64.46	43.61	56.88
		DSIR-paper	64.00	40.64	55.50

Table 34: Per-task 5-shot accuracy (%) for the UCE-filter runs of Section 5.2, reported as mean $\pm\sigma$ across three independent training seeds (1234, 42, 7777). Conventions follow Table 32.

		Targeted (UCE-filter targets)							Out-of-target				Average	
Source	**Method**	ARC-E	ARC-C	HSwag	SciQ	OBQA	PIQA	SWAG	LAMB	CSQA	BoolQ	SIQA	**Targeted**	**Overall**
1.34B parameters														
DCLM	UCE-filter (p=0.20)	**74.40 ± 0.28**	**42.32 ± 0.81**	**67.88 ± 0.36**	**94.60 ± 0.60**	**41.00 ± 1.22**	**77.95 ± 0.54**	**53.35 ± 0.15**	**59.04 ± 1.39**	19.82 ± 0.75	**61.86 ± 0.87**	**46.66 ± 0.51**	**64.50 ± 0.26**	**58.08 ± 0.30**
	Random (p=0.20)	69.95 ± 0.37	38.40 ± 0.67	61.20 ± 0.18	93.93 ± 0.98	38.73 ± 2.31	75.21 ± 0.60	51.11 ± 0.05	55.62 ± 0.68	**20.17 ± 0.90**	61.55 ± 4.31	45.87 ± 0.37	61.22 ± 0.66	55.61 ± 0.17
	DSIR-paper [43]	70.74 ± 0.84	39.36 ± 0.91	63.32 ± 0.34	92.37 ± 0.21	39.13 ± 2.10	77.02 ± 0.33	49.91 ± 0.13	44.76 ± 1.57	19.85 ± 1.07	58.71 ± 3.94	44.01 ± 0.82	61.69 ± 0.17	54.47 ± 0.47
	DSIR-LN (ablation)	*73.11*	*43.52*	*68.22*	*94.60*	*40.80*	*78.62*	*52.29*	*50.63*	*21.38*	*58.47*	*44.37*	*64.45*	*56.91*
FineWeb-Edu	UCE-filter (p=0.20)	**74.38 ± 0.57**	**44.23 ± 1.13**	**64.61 ± 0.33**	93.90 ± 0.40	**42.33 ± 0.64**	**76.68 ± 0.53**	**50.08 ± 0.32**	**43.63 ± 0.47**	19.33 ± 1.07	**60.62 ± 1.90**	**45.33 ± 0.48**	**63.74 ± 0.29**	**55.92 ± 0.38**
	Random (p=0.20)	73.40 ± 0.15	42.38 ± 0.51	58.09 ± 0.37	**94.27 ± 0.71**	39.13 ± 1.27	73.96 ± 0.82	47.20 ± 0.25	39.40 ± 0.23	19.74 ± 0.59	59.05 ± 5.85	43.98 ± 0.68	61.20 ± 0.35	53.69 ± 0.67
	DSIR-paper [43]	73.91 ± 0.29	43.60 ± 1.09	58.85 ± 0.28	93.03 ± 0.40	39.87 ± 1.14	75.28 ± 0.30	47.04 ± 0.10	33.77 ± 1.21	**19.87 ± 1.11**	59.75 ± 1.57	43.42 ± 0.28	61.65 ± 0.25	53.49 ± 0.11
	DSIR-LN (ablation)	*75.51*	*43.52*	*63.63*	*94.00*	*40.20*	*76.77*	*48.78*	*41.12*	*19.00*	*59.30*	*45.09*	*63.20*	*55.17*
3.33B parameters														
DCLM	UCE-filter (p=0.20)	**77.30 ± 0.30**	**47.16 ± 1.28**	**74.11 ± 0.09**	**95.63 ± 0.49**	**43.27 ± 1.01**	**80.56 ± 0.54**	**54.93 ± 0.13**	**64.93 ± 1.14**	**20.01 ± 0.52**	67.25 ± 1.69	**48.74 ± 0.82**	**67.57 ± 0.28**	**61.26 ± 0.29**
	Random (p=0.20)	74.27 ± 0.30	42.49 ± 0.82	68.41 ± 0.15	95.23 ± 0.49	41.93 ± 0.70	77.15 ± 0.66	52.82 ± 0.09	60.75 ± 0.53	19.06 ± 1.31	**67.35 ± 0.80**	47.36 ± 0.42	64.62 ± 0.15	58.80 ± 0.28
	DSIR-paper [43]	74.20 ± 0.92	43.00 ± 1.20	69.38 ± 0.25	93.90 ± 0.35	42.60 ± 2.31	79.23 ± 0.45	51.70 ± 0.06	49.74 ± 1.04	19.41 ± 0.78	62.00 ± 3.29	45.82 ± 0.50	64.86 ± 0.50	57.36 ± 0.53
	DSIR-LN (ablation)	*76.18*	*48.55*	*74.39*	*95.00*	*41.80*	*80.69*	*53.80*	*56.37*	*19.49*	*65.87*	*47.75*	*67.20*	*59.99*
FineWeb-Edu	UCE-filter (p=0.20)	75.80 ± 1.27	**47.53 ± 0.98**	**69.34 ± 0.41**	**95.73 ± 0.40**	41.87 ± 1.47	**78.24 ± 0.33**	**51.29 ± 0.19**	**48.98 ± 1.16**	19.30 ± 0.91	**63.16 ± 2.43**	**46.57 ± 0.59**	**65.68 ± 0.36**	**57.98 ± 0.10**
	Random (p=0.20)	**76.30 ± 0.46**	47.41 ± 0.94	63.97 ± 0.32	95.30 ± 0.26	**42.93 ± 0.23**	75.99 ± 0.87	49.35 ± 0.00	46.34 ± 0.71	**20.42 ± 0.25**	61.57 ± 7.83	46.11 ± 0.18	64.46 ± 0.25	56.88 ± 0.75
	DSIR-paper [43]	76.15 ± 0.50	46.19 ± 0.99	64.34 ± 0.10	94.43 ± 0.12	41.73 ± 0.42	76.53 ± 0.45	48.60 ± 0.16	38.28 ± 1.81	20.01 ± 0.40	59.57 ± 2.72	44.69 ± 0.59	64.00 ± 0.28	55.50 ± 0.30
	DSIR-LN (ablation)	*75.25*	*45.48*	*68.55*	*95.10*	*40.80*	*77.97*	*50.70*	*46.77*	*18.67*	*64.50*	*44.98*	*64.84*	*57.16*

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract states UCE as a model-free, tokenizer-agnostic measure of corpus–benchmark word-distribution alignment, the descriptive finding that lower UCE tracks higher zero-shot accuracy on every benchmark, and the two prescriptive interventions (document-level data selection and prompt-level drafter routing). The Introduction's Contributions paragraph restates these as three observations (corpus, document, prompt) under one alignment functional, with the scope qualifier that UCE-filter is a controlled probe rather than a stand-alone data-selection method. The claims match the experimental scope and the limits in Section 7.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 discusses (i) what UCE does *not* capture (conditional structure such as syntax, and the grammar, arithmetic, and non-English benchmarks of Appendix D.7), (ii) the whitespace-tokenization caveat for non-segmented or agglutinative scripts, and (iii) the target-adapted nature of the UCE-filter when Q is estimated from the evaluation tasks. Section 5.3 adds the 30B case-study caveats.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Appendix A states the assumptions (stationarity of P_A and P_B, well-defined entropy rates) and gives the unigram identity $H(p_B, p_A) = H(p_B) + D_{\mathrm{KL}}(p_B \| p_A)$. Main-text Section 3 cross-references the appendix at the equation level.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4.1 and Appendix C report the training recipe (optimizer AdamW, cosine LR schedule with warmup, batch size, sequence length, and weight decay) and the per-scale architecture for both the corpus-ranking grid (LLaMA pattern, Tables 6–7) and the UCE-filter grid (DataComp-LM pattern, Tables 8–9), with gradient clipping and BF16 precision also listed for the UCE-filter grid. The UCE-filter algorithm is given in Algorithm 1, and the code at `https://github.com/penpaperkeycode/lexical-imprint` provides reference implementations of UCE, the UCE-filter, the DSIR baseline, and the benchmark preprocessing pipeline. All pre-training corpora and benchmarks are public.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is released at `https://github.com/penpaperkeycode/lexical-imprint`. It covers the word-level UCE metric (`unigram_ce/`), the UCE-filter with multi-worker analytic-ΔH row scoring (`ce_filter/`), the DSIR baseline with hashed-bigram importance resampling and optional Gumbel top-K (`dsir/`), benchmark preprocessing for the evaluation tasks of Sections 4.1 and 5.2 (`benchmarks/`), and a tiny end-to-end demo (`dsir/examples/tiny_demo/`). Each subdirectory has its own quick-start `README.md`. The from-scratch pre-training runs follow the recipes in Appendix C, with the LLaMA pattern for the corpus-ranking grid and the DataComp-LM pattern for the UCE-filter grid. All pre-training corpora (C4, DCLM, FineWeb-Edu, OpenWebText) and benchmarks are publicly available, with evaluation through `lm-evaluation-harness` [16].

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

Justification: Section 4.1 states the shared pre-training protocol and the corpus-ranking setup (corpora, sub-sample sizes, model scales, benchmark pool). Section 5.2 states the UCE-filter setup (sources, scales, top-20% retention, baselines, target suite). Section 6 states the routing setup (verifier, three OpenThoughts-114k [50] specialist drafters under the SAM architecture [51], a generalist-drafter baseline, and AIME/LiveCodeBench/MMLU/MedQA evaluation). The pre-training hyperparameters of the corpus-ranking and UCE-filter grids are in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Within-corpus stability (Figure 3, Section 4.3) reports per-corpus mean $\pm 1\sigma$ across 16 random 8.5B-token subsets at 400M scale, with an OLS fit, 95% CI band, R^2, Spearman ρ, and p-value. Table 2 reports mean $\pm$ std across 3 training seeds (1234, 42, 7777) per (Source, Scale, Selection) cell for UCE-filter, Random, and DSIR-paper, totaling 36 from-scratch runs across the $2\times2\times3$ grid (Appendix E.8). UCE values in Table 1 are selection-deterministic for UCE-Filter and Full, while Random and DSIR vary by less than 0.001 bits across 3 data-selection seeds. A per-task accuracy breakdown is in Table 32.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: All from-scratch runs use NVIDIA B200 GPUs (183GB HBM, 8 per node) on a shared cluster. **Selection.** UCE and the UCE-filter run on a single CPU-only pod (120 cores, no GPU). At the 70B-token output budget the UCE-filter end-to-end wall-time is

roughly 3.5–8.3h per source on a 120-core pod, with aggregate CPU-hours comparable to DSIR's. **Training.** Each cell of the UCE-filter grid (Section 5.2) uses 16 B200 GPUs across two nodes for 70B GPT-2 tokens, with per-cell wall-time of approximately 7h at 1.34B and 17h at 3.33B. The grid covers four selection methods, two sources, and two scales, with UCE-filter, Random, and DSIR-paper at 3 seeds and DSIR-LN at 1 seed (40 runs total). The corpus-ranking grid (Section 4.1) is of the same order. The 30B Stage-2 case study (Section 5.3) runs on in-house infrastructure ($\sim$2T-token baseline vs 200B-token UCE-filtered variant). Inference-time experiments (Section 6) are single-node and negligible.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses only publicly available corpora and benchmarks, conducts no human-subject research, releases no new personal data, and follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: *Positive.* UCE-filter reduces compute spent on uninformative pre-training data, and routing reduces inference cost at fixed model quality. Alignment diagnostics make corpus–benchmark proximity legible, supporting transparent benchmarking. *Negative.* Target-aware curation can "hack" benchmark scores in ways that overstate underlying capability, and over-reliance on benchmark-aligned data may degrade out-of-distribution generalization. Reporting UCE next to accuracy is a partial mitigation, since Section 7.1 notes that zero-shot accuracy partly reflects lexical alignment, and Section 5.2 reports out-of-target accuracy beside the targeted suite.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper releases no model or dataset that poses a high misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Every dataset and code package is cited at first reference. Pre-training corpora (C4 [29], DCLM [28], FineWeb-Edu [27], OpenWebText [30]) and evaluation benchmarks (ARC [33], HellaSwag [34], MMLU [3], MMLU-Pro [4], SciQ [35], OpenBookQA [36], PIQA [37], LAMBADA [38], BoolQ [42], Social IQa [39], CommonsenseQA [41], SWAG [40]) are used under their published licenses. Routing experiments (Section 6) use DeepSeek-R1-Distill-Llama-8B [49] as the verifier and AIME [52], LiveCodeBench [53], and MedQA [54] as benchmarks, with specialist drafters trained on OpenThoughts-114k [50] under the SAM architecture [51]. The verifier is released under the MIT License, and its model card states that it is derived from Llama-3.1-8B, which is distributed under the Llama 3.1 Community License. Evaluation runs through `lm-evaluation-harness` [16]. Tokenization uses GPT-2 [32] byte-level BPE.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release a reference implementation of the UCE-filter (Algorithm 1) including row-scoring and multi-worker batched scoring at `https://github.com/penpaperkeycode/lexical-imprint`. The 1.34B and 3.33B from-scratch base-model checkpoints from Section 5.2 will be uploaded to HuggingFace upon request. Both are accompanied by README/model cards listing training data, hyperparameters (Appendix C), intended research use, and known limitations.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper involves no crowdsourcing or human-subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human-subject research, so IRB review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core methodology (UCE, the UCE-filter, UCE-based drafter routing) does not use LLMs as a research instrument.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.