SELECTING NATURAL LONG DOCUMENTS FOR CONTEXT EXTENSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Natural long documents are widely used to extend language models to longer contexts, yet length alone does not reveal whether they provide useful long-range supervision. We present a controlled study of how such documents should be selected, and scored. Using Qwen2.5-3B, 128K-token continued training, and matched 6B-token budgets, we compare repacking shorter documents from the preceding training stage into long sequences, random sampling of natural long documents, and generic-quality, LLM-judge, and likelihood-based selection; HELMET tasks are used for model selection and RULER for held-out evaluation. Random sampling is only slightly better than repacking at shorter evaluation lengths, but leads by approximately 2.5 HELMET points and 3.3 RULER points at 128K. Within a fixed pool of natural long documents, the evaluated generic-quality and LLM-judge pipelines do not exceed random sampling, whereas the likelihood-based long-dependency pipelines achieve higher observed scores. Selecting documents by their density of LongPPL key tokens—tokens whose prediction improves markedly when the long context is available—achieves the highest observed scores among the evaluated pipelines. Its most consistent cross-benchmark gains are retrieval-oriented and widen toward 64K–128K, whereas QA effects are benchmark-dependent and aggregation and variable tracking show no consistent gain. Because scoring every document over the full 128K span is expensive, we introduce an efficient variant, truncLongPPL, that restricts scoring to the first 64K target positions and yields downstream scores close to those of the full-128K pipeline while providing a measured $2.28\times$ end-to-end scoring speedup. Finally, in one evaluated 66/34 short/long mixture, replacing random long documents with selected ones improves macro-averaged HELMET by 4.6 points and RULER by 3.3 points without a material additional observed reduction on seven short-context tasks, and this advantage is further corroborated after supervised fine-tuning on LongBench v2 and MRCR. Overall, we hope the controlled and comprehensive comparisons will provide more practical references for long-context training.

1 INTRODUCTION

Extending language models to longer contexts generally requires continued training at longer sequence lengths (Gao et al., 2025c; Yang et al., 2025; Liu et al., 2024). Natural long documents can supply cross-section structure, delayed references, and distributed evidence (Gao et al., 2025c; Olmo et al., 2025), but length does not guarantee useful long-range supervision: a long document may merely collect locally coherent, mutually independent passages (Deng et al., 2025; Chen et al., 2024; Fang et al., 2025). The data-selection problem is therefore to identify documents in which distant context changes what a model can predict.

Existing pipelines offer different answers. Scoring-free constructions either repack the preceding training mixture or randomly sample natural long documents (Team et al., 2026b; Gao et al., 2025c). Generic selectors use perplexity or gzip compressibility, while dependency-aware selectors use model judgments or likelihood changes induced by additional context (Olmo et al., 2025; Deng et al., 2025; Chen et al., 2024; Fang et al., 2025). Their conclusions are difficult to reconcile because candidate pools, scorers, training budgets, context lengths, and evaluations vary jointly. OLMo 3, for example, reports no clear advantage of its LongPPL-derived selectors over gzip (Olmo et al., 2025). It remains

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

unclear which pipeline works best under a shared protocol and whether effective dependency scoring can be made economical at corpus scale.

We study this as a controlled comparison of complete training pipelines, asking four questions: (1) Does random sampling of natural long documents improve over preceding-mixture repacking? (2) Which selection signals work best within a fixed pool, and which capabilities benefit? (3) Can a shorter scoring span retain downstream quality at lower cost? (4) Does the selection benefit persist in a fixed short/long mixture and after instruction tuning?

Our main experiments extend Qwen2.5-3B (Yang et al., 2024) to 128K with matched 6B-token budgets. HELMET (Yen et al., 2024) determines all selection settings; RULER (Hsieh et al., 2024) remains held out, and seven short-context tasks measure retention. Because selectors differ in both signal and implementation, our comparison concerns their downstream pipelines rather than directly comparable proxy scores. The contributions are:

- **A controlled comparison.** We compare scoring-free, generic-quality, LLM-judge, and likelihood-based pipelines. LongPPL key-token density achieves the highest observed aggregate scores; its most consistent gains are retrieval-oriented and concentrate at 64K–128K.
- **An efficient selector.** truncLongPPL restricts LongPPL scoring to the first 64K target positions. It achieves downstream quality close to full-128K scoring with a measured 2.28× speedup, establishing an empirical cost–quality trade-off rather than equivalent document scores or selected sets.
- **Evidence within a mixed-data recipe.** At a fixed 66/34 short/long token ratio, selected long documents improve long-context performance with essentially unchanged short-context performance relative to random-long. The advantage carries over to LongBench v2 and MRCR after identical supervised fine-tuning.

Overall takeaways: summary of the main empirical findings

- **RQ1—Data construction (§4.1):** Random sampling of natural long documents and preceding-mixture repacking are similar at shorter evaluation lengths, but the gap widens toward the target length. At 128K, random sampling is higher by approximately 2.5 HELMET points and 3.3 RULER points.
- **RQ2—Selection signals and scope (§4.2 and §6):** Within the fixed pool of natural long documents, the evaluated perplexity-, gzip-, and LLM-judge-based pipelines do not exceed random sampling, while the likelihood-based long-dependency pipelines are higher. The front-64K LongPPL key-token-density pipeline (truncLongPPL) achieves the highest observed macro-averages, exceeding random sampling by 4.57 HELMET points and 3.18 RULER points. The largest differences occur on retrieval-oriented tasks at 64K–128K.
- **RQ3—Reduced-span scoring (§5):** We evaluate the first 64K target positions as a lower-cost LongPPL key-token-density pipeline. Under the evaluated absolute threshold, this front-64K pipeline yields downstream scores close to those of the full-128K pipeline while providing a measured 2.28× end-to-end scoring speedup. This is a pipeline-level cost–quality result, not a claim that the two document scores or selected sets are equivalent.
- **RQ4—Selection in a fixed mixture (§7):** In one OLMo-3-inspired 66/34 short/long mixture (Olmo et al., 2025), replacing randomly sampled long documents with selected documents from the same source improves macro-averaged HELMET by 4.6 points and RULER by 3.3 points, without a material additional observed reduction relative to the random-long control on the seven-task short-context average; the same ranking holds after supervised fine-tuning on LongBench v2 and on MRCR (§8).

2 Related Work

Long-context training recipes. Long-context extension commonly uses late continued training. ProLong emphasizes downstream evaluation (Gao et al., 2025c); MAI-Thinking-1 repacks its preceding mixture at 256K (Team et al., 2026b); and OLMo 3 combines gzip-filtered documents, synthetic

augmentation, and a 66/34 short/long mixture (Olmo et al., 2025). Qwen3 trains at 32K (Yang et al., 2025), DeepSeek-V3 extends to 128K (Liu et al., 2024), and GLM-5 reaches 200K using PPL-filtered documents alongside synthetic agent and repository data (Zeng et al., 2026). Their architectures, curricula, mixtures, and budgets vary jointly, precluding direct selector comparisons; we compare pipelines under a shared protocol.

Long-document selection. Generic document-level proxies include PPL in GLM-5 (Zeng et al., 2026) and gzip in OLMo 3, which reports no clear advantage for its evaluated LongPPL-derived alternatives (Olmo et al., 2025). Long-dependency selectors include LongFilter's token-loss reductions (Deng et al., 2025), Prospector's block-pair dependency strength, distance, and specificity (Chen et al., 2024), and LongPPL's tokens benefiting from added context (Fang et al., 2025). LADM (Chen et al., 2025) and LongAttn (Wu et al., 2025) instead use explicit 32K attention statistics, which do not directly scale to our 128K scoring stack. We compare generic-, judge-model-, and likelihood-based pipelines with repacking and random-sampling references, and measure the cost of reduced-span LongPPL-density scoring.

Long-context data construction. In-Context Pretraining orders related documents (Shi et al., 2024); QUEST groups documents by predicted queries and keywords (Gao et al., 2024a); NExtLong interleaves meta-chunks with retrieved hard negatives (Gao et al., 2025b); EntropyLong retains contexts reducing uncertainty at difficult targets (Jia et al., 2025); and LiteLong organizes retrieved documents into coherent sequences (Jia et al., 2026). These approaches alter document order or insert retrieved content. We select natural long documents while preserving their internal structure; combining these approaches remains future work.

3 Experimental Setup and Evaluation Protocol

3.1 Task and Candidate Pool

Let $\mathcal{C}$ be a pool of natural long documents. A selector assigns a score $s(x)$ to each document x and produces a subset $\mathcal{D} \subseteq \mathcal{C}$ for continued training. We compare downstream models trained on each resulting dataset under the same token budget. We sample 25% of the *documents* in our internal pretraining corpus, then retain those containing at least 128K tokens. The resulting pool contains 3M documents from books, scanned documents, manuals, academic papers, and web sources, covering English, Chinese, and mixed/other languages. Repository-level code is excluded.

3.2 Data Construction Methods

We distinguish scoring-free constructions from generic document-level, judge-model long-dependency, and likelihood-based long-dependency selectors. REPACK preserves the preceding-stage mixture weights and packs its mostly shorter documents into 128K sequences; random uniformly samples natural long documents. Other pipelines rank documents from the same pool while preserving document boundaries. Perplexity selects the top or middle 40%; gzip retains the middle 60%; and an LLM judge selects the top 5% by the fraction of sampled distant fragment pairs judged dependent. LongFilter aggregates probability-weighted loss reductions from short to long context, while Prospector combines dependency strength, distance, and specificity over block pairs (Deng et al., 2025; Chen et al., 2024).

Table 1: Shared protocol for the main data-selection comparisons.

Component	Setting
Backbone	Qwen2.5-3B
Target context length	128K tokens
Continued-training budget	6B tokens per configuration
Validation benchmark	HELMET at 8K–128K
Held-out evaluation benchmark	RULER at 8K–128K
Short-context evaluation	Mean over seven `lm-eval` tasks

We adapt LongPPL (Fang et al., 2025) to select by *key-token density*, $\rho_W(x) = |W|^{-1} \sum_{i \in W} k_i$. A token is key when its long-context log-likelihood exceeds -2.0 and improves over its 4K short-context log-likelihood by more than 2.0. The selector retains documents with $\rho_W(x) > \tau$. We fix the scorer, tokenizer, and token-level thresholds, varying the target-position window W and document threshold τ. Unless stated otherwise, truncLongPPL denotes the HELMET-selected front-64K configuration with $\tau = 0.05$.

The two token-level conditions serve different purposes: the long-context likelihood condition excludes tokens that remain difficult to predict, while the improvement condition identifies tokens for which additional context is useful. Averaging these indicators measures how frequently such tokens occur within the scored window. This density is a model-dependent selection proxy; its value is assessed through downstream training outcomes rather than treated as a direct measure of a document's reasoning content.

PPL, LongFilter, and LongPPL use Gemma-4-26B-A4B (base); the judge uses its instruction-tuned counterpart (Team et al., 2026a). Our 128K Prospector adaptation uses SmolLM2-135M (Allal et al., 2025) for its 5,000 block-pair comparisons per document, rather than the original OPT-350M. Thus scorer choice is part of each evaluated pipeline. Appendix A gives all selection rules; Appendices C–E provide exact scores, prompts, and implementation details. Explicit-attention selectors are omitted because their statistics do not scale readily to our 128K stack.

3.3 Training and Evaluation

All main comparisons start from Qwen2.5-3B and train at 128K for 6B tokens, increasing the RoPE base to 2×10^8. Training uses bf16 AdamW, a constant learning rate of 10^{-6}, and approximately 8M tokens per update. Appendix B supplies the remaining parameters and task definitions. The fixed-mixture experiment uses a separate, matched 10B-token budget (Section 7).

HELMET alone selects the density threshold, scoring-window length and position, and default configuration. RULER is evaluated after these choices are fixed; reported non-default RULER configurations are post-selection diagnostics and do not inform tuning. Both benchmarks cover 8K, 16K, 32K, 64K, and 128K. HELMET averages recall, in-context learning, re-ranking, long-document QA, and retrieval-augmented generation; RULER covers retrieval, aggregation, variable tracking, and QA. We report length-specific results and macro-averages over lengths. Short-context retention is the mean zero-shot raw accuracy over seven `lm-eval` tasks, detailed in Appendix B.

4 Comparison of Data Construction and Selection Pipelines

4.1 Natural Long Documents versus Preceding-Mixture Repacking

We first compare Repack, which preserves the preceding-stage mixture and repacks its documents into 128K sequences, with random sampling of natural long documents under the same 6B-token budget. Figure 1 shows a small difference at 8K but an increasing gap with evaluation length. At 128K, random sampling is higher by approximately 2.5 validation HELMET points and 3.3 held-out RULER points, motivating selection within the natural-long-document pool. This comparison concerns the complete construction pipelines: repacking retains the preceding mixture, whereas random sampling draws from natural long documents. The subsequent selector comparison shares the same natural-document candidate pool and preserves document boundaries.

4.2 Selecting Natural Long Documents

We compare the complete pipelines in Section 3.2. Figure 2 uses the HELMET-selected truncLongPPL configuration (front-64K, $\tau = 0.05$), whose threshold and window are analyzed below. None of the evaluated perplexity-, gzip-, or LLM-judge-based pipelines exceeds random sampling; top-40% perplexity performs worst on both aggregates. This finding concerns the evaluated configurations, rather than these signals in general. Under OLMo 3's symmetric middle-60% selection rule, LongPPL and gzip differ only slightly (Appendix F), consistent with its reported comparison.

Prior long-dependency pipelines modestly exceed random sampling: LongFilter scores 53.72 HELMET / 79.54 RULER and our 128K Prospector adaptation scores 52.26 / 78.36, versus 51.66 / 78.05

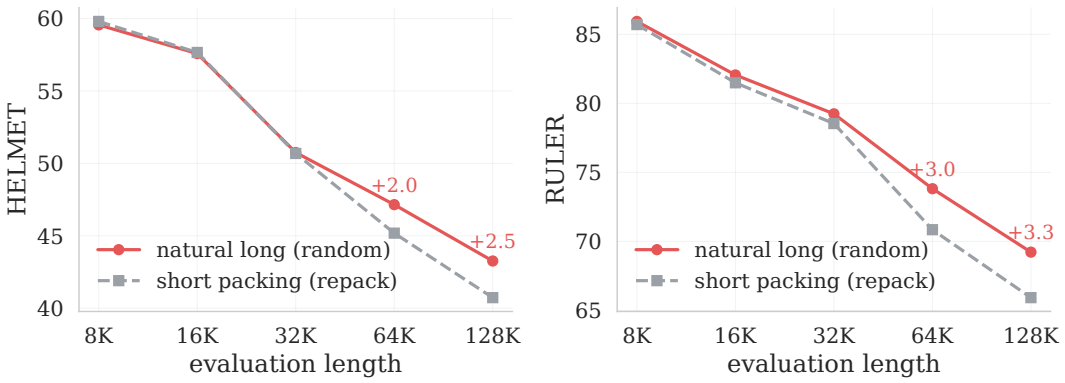

Figure 1: Random natural long documents versus preceding-mixture repacking at 6B tokens: validation HELMET (left) and held-out RULER (right), by evaluation length.

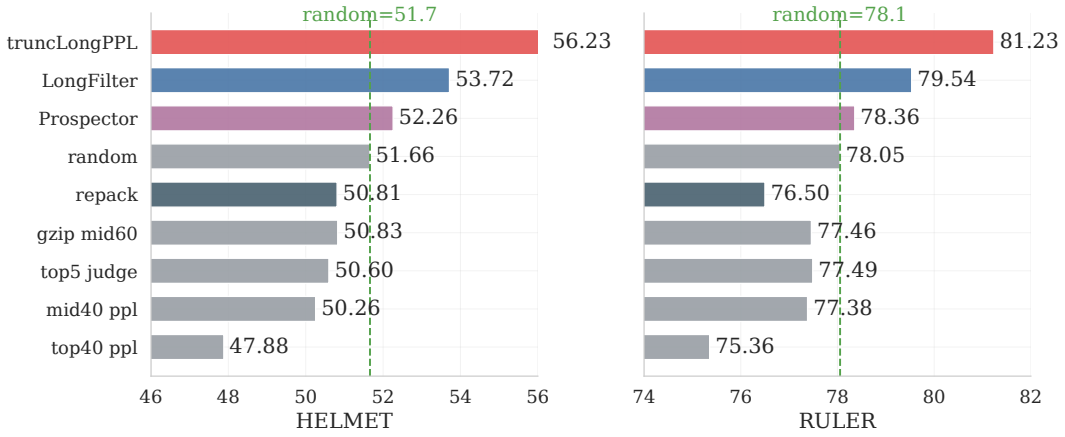

Figure 2: Selection pipelines at 6B tokens: macro-averaged validation HELMET (left) and held-out RULER (right). Dashed lines indicate random sampling of natural long documents.

for random sampling. truncLongPPL achieves the highest observed scores, 56.23 / 81.23, improving by 4.57 / 3.18 points. Restoring the candidate pool's book, web, and paper proportions within selected data changes HELMET by at most 0.3 point and RULER by at most 0.2 point across thresholds (Appendix I). Thus, coarse source reweighting alone does not explain the observed advantage.

5 LongPPL-Density Selection: Sensitivity and Efficiency

We examine the density threshold, scored-span length, and window placement to identify an efficient operating point. All configuration choices use HELMET validation; RULER remains held out.

5.1 Key-Token-Density Threshold

With full-128K scoring, HELMET improves substantially between $\tau = 0.01$ and 0.03, rises from 55.30 to 56.23 between 0.03 and 0.05, and then plateaus through 0.08 (Figure 3). Validation distinguishes permissive selection from the high-density regime but gives little evidence for ordering configurations within $[0.05, 0.08]$. We choose $\tau = 0.05$, the least restrictive plateau setting, retaining the largest unique-document pool. Stricter thresholds continue to shrink the pool without a corresponding validation improvement, so the evidence favors a high-density regime rather than a uniquely optimal cutoff. Held-out RULER follows a similar trend: approximately 78.9 at 0.01, 80.34 at 0.03, 81.34 at 0.05, and 81.6 at 0.08. The cutoff is specific to our scorer, corpus, and scoring configuration.

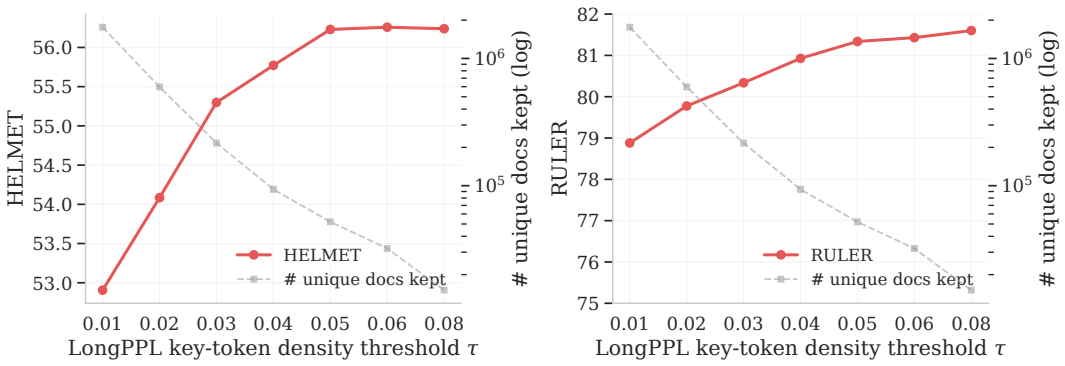

Figure 3: Full-128K density-threshold sweep at 6B tokens: HELMET (left), RULER (right), and retained unique documents (gray, secondary axes). HELMET plateaus from $\tau = 0.05$ through 0.08.

5.2 Scoring-Window Length and Placement

Full-128K scoring costs approximately 10^4–10^5 H20 GPU-hours for the candidate pool (Appendix E). We therefore fix $\tau = 0.05$ and compare front windows of 16K, 32K, and 64K with the full 128K span. A prefix bounds the processed span, whereas scoring tail targets still requires preceding context. Only scoring changes: selected documents retain their original training spans.

Differences grow with evaluation length (Figure 4). At 128K, HELMET scores are 46.1, 47.2, 49.9, and 50.0 for front-16K, front-32K, front-64K, and full-128K, respectively. Front-64K and full-128K also share the same macro-average to two decimal places (56.23). We select front-64K as the shortest evaluated span with comparable validation performance. After this choice is fixed, RULER differs by only 0.2 point at 128K; front-16K instead scores 70.3.

Alternative placements do not improve this choice consistently. Front-64K and tail-64K differ only slightly on HELMET, with no observed tail advantage, and their intersection does not outperform front-64K at the default threshold (Appendix H.3). Across $\tau \in \{0.03, 0.04, 0.05\}$, eight scattered 8K windows and contiguous front-64K differ by approximately one validation point or less, without a consistent direction (Appendix H.4). These variants alter scored target positions, not the key-token definition or training objective.

5.3 Scoring Efficiency and Agreement

Front-64K halves the target positions included in the density and bounds the processed span. Measured end-to-end scoring time falls from 16.55 to 7.25 seconds per document, a 2.28× speedup (Appendix G); the full-context stage's superlinear attention component permits savings beyond

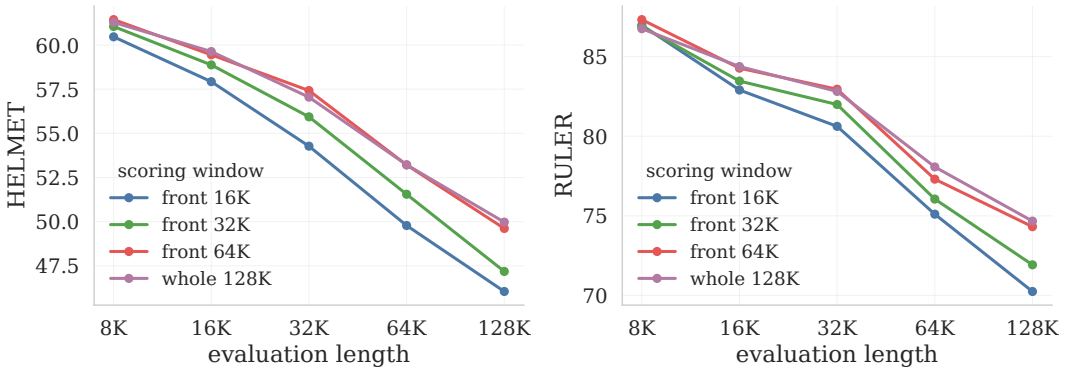

Figure 4: Window-length comparison at $\tau = 0.05$ and 6B training tokens: validation HELMET (left) and held-out RULER (right). Front-64K and full-128K remain close at long evaluation lengths.

the twofold position reduction. Together with comparable validation performance, this supports front-64K as the selected cost–quality operating point.

The selection-conditioned audit in Appendix G nevertheless finds imperfect density agreement: one-way front-to-full selected-set containment ranges from 76% to 92%. This audit is conditioned on selection and does not measure agreement over an unselected sample of the entire candidate pool. The result therefore concerns downstream pipeline quality, not equivalent scores or selected corpora. Unless stated otherwise, truncLongPPL denotes front-64K LongPPL key-token-density selection at $\tau = 0.05$. Appendix H includes training-budget sensitivity and a window-length comparison at $\tau = 0.03$.

6 Capability-Level Effects of Selection

The selector favors tokens whose predictions improve with distant context, suggesting a retrieval-oriented benefit. We test this interpretation by capability and evaluation length, reporting held-out RULER first and then comparing validation HELMET.

Held-out RULER. Retrieval has the most length-consistent gain, increasing from a small gap at 8K to approximately eight points at 128K (Figure 5). QA improves at every length, but non-monotonically: approximately eight points at 16K and five at 128K, with smaller gaps at 32K and 64K. Aggregation and variable tracking show no consistent benefit, and selection is slightly below random sampling at 128K. The gains therefore support improved access to remote evidence more clearly than improved aggregation or manipulation of that evidence.

Validation HELMET. Recall corroborates this pattern: the selected–random gap grows from 1.8 points at 8K (98.9 versus 97.1) to approximately 21 at 128K (83.5 versus 62.8; Figure 6). At 128K, re-ranking and RAG improve by approximately four points and in-context learning by two. Long-document QA instead differs by at most 1.1 points across lengths. Retrieval gains thus generalize across both suites, whereas QA effects depend on the benchmark. Selection does not uniformly improve long-context reasoning or resolve low absolute aggregation and QA performance at 128K. Whether complementary synthetic training data can address these capabilities remains an open question rather than a conclusion of the present comparisons.

7 Selection within a Fixed Short/Long Mixture

The 6B-token pipelines lower the seven-task short-context average from the uncontinued base model's 60.1 to 59.0–59.5; truncLongPPL scores 59.2 (Appendix J). We therefore test selection within one OLMo-3-inspired 66/34 short/long mixture, retaining a substantial short-data component. This separate 10B-token experiment is compared only within its group, not directly with the 6B-token ablations.

We train three 128K configurations: short-only packing; short + random-long, using 6.6B short tokens and 3.4B randomly sampled natural long tokens; and short + truncLongPPL-long, substituting selected documents from the same pool. The mixed configurations share the budget, short component,

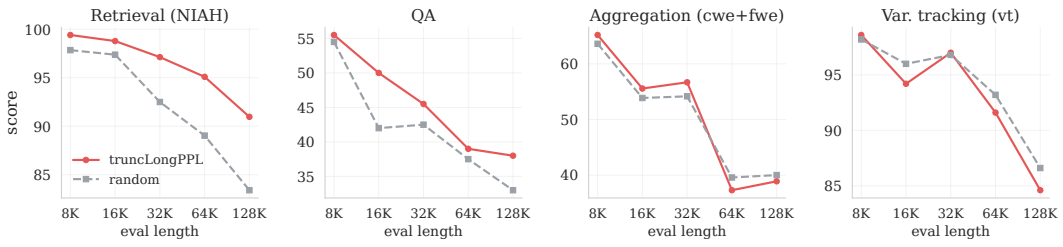

Figure 5: Held-out RULER capabilities by evaluation length for random sampling and truncLongPPL at 6B tokens. Retrieval gains increase with length; aggregation and variable tracking do not consistently improve.

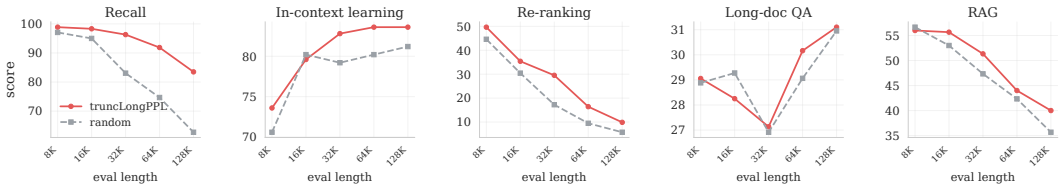

Figure 6: Validation HELMET capabilities for the same 6B-token comparison. Recall gains reach approximately 21 points at 128K; long-document QA differs by at most 1.1 points across lengths.

long-document source, and mixing ratio, differing only in long-data selection. We keep the already-fixed front-64K, $\tau = 0.05$ selector and do not optimize the mixture ratio.

Replacing random long documents with selected ones raises HELMET from 50.7 to 55.3 (+4.6) and RULER from 78.6 to 81.9 (+3.3), with essentially unchanged short-context averages (59.8 versus 59.8; Figure 7). The short-only reference scores 49.4 HELMET, 77.2 RULER, and 60.0 on short-context tasks. Relative to this reference, random long data adds 1.3 HELMET and 1.4 RULER points, whereas selected long data adds 5.9 and 4.7 points. The selected mixture's short-context average is 0.2 point below short-only and remains close to the uncontinued base model's 60.1. Selection therefore retains its benefit when long data provides 34% of tokens, without a material additional observed short-context reduction relative to random-long. Across three independent training seeds, standard deviations are at most 0.27 point on these aggregates, and the ranking is stable (Appendix K). This supports the evaluated mixture, without establishing an optimal ratio or guaranteeing short-context preservation more broadly.

8 Downstream Effects after Supervised Fine-Tuning

We fine-tune the three fixed-mixture models with identical recipes and hyperparameters on Long-Magpie (Gao et al., 2025a), then evaluate on held-out LongBench v2 (Bai et al., 2024) and MRCR (Vodrahalli et al., 2024),[1] probing long-document reasoning and multi-round retrieval. This comparison holds SFT fixed to test whether the pretraining-data selection benefit persists after instruction tuning. Tables 2 and 3 report means over three runs; Appendix L retains per-cell standard deviations and LongBench v2 difficulty splits.

truncLongPPL leads on LongBench v2 overall (32.5 versus 30.9 for random-long and 29.6 for short-only) and on every context-length split. Its gains over random-long are 2.1, 1.0, and 2.1 points on Short, Medium, and Long, respectively; these are positive but not monotonic with length. On MRCR, it improves overall from 17.62 to 18.91, with gains of approximately 1.9, 0.8, and 1.2 points on the 2-, 4-, and 8-needle subsets. The selected mixture thus retains its ranking after SFT on both suites, extending the observed benefit beyond direct base-model evaluation.

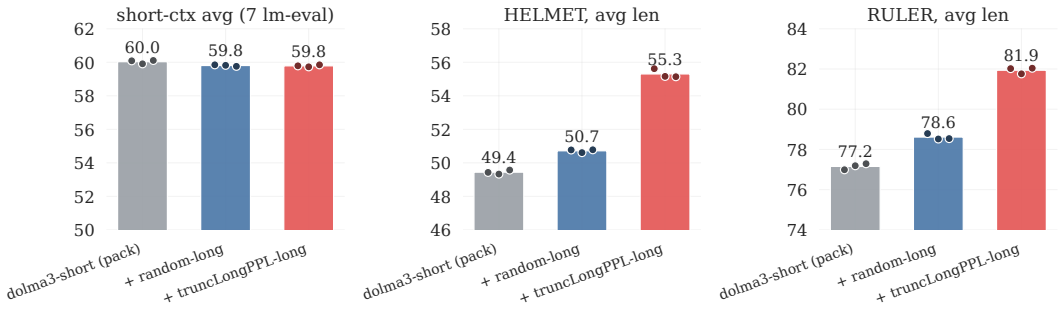

Figure 7: Fixed-mixture comparison at 10B tokens: short-context average, HELMET, and RULER. Bars show three-seed means; points show individual seeds (Appendix K).

[1] We use the OpenAI MRCR set: https://huggingface.co/datasets/openai/mrcr.

Table 2: LongBench v2 accuracy (%) after SFT, averaged over three runs. Best values are bold.

Configuration	Overall	Short	Medium	Long
dolma3-short	29.6	37.8	25.7	23.7
+ random-long	30.9	38.1	26.3	27.8
+ truncLongPPL-long	**32.5**	**40.2**	**27.3**	**29.9**

Table 3: MRCR accuracy (%) up to 128K, averaged over three seeds ($N = 1476$). Best values are bold.

Configuration	Overall	2-needle	4-needle	8-needle
dolma3-short	16.16	22.27	15.13	11.08
+ random-long	17.62	24.13	16.74	11.98
+ truncLongPPL-long	**18.91**	**26.00**	**17.53**	**13.16**

9 LIMITATIONS

Results cover one backbone and corpus, a fixed LongPPL scorer, and a 128K protocol at tested budgets; pipeline comparisons do not isolate individual signals. With LongPPL's token thresholds fixed, document thresholds and scoring windows may not transfer across scorers, tokenizers, corpora, or context lengths. Growing prefixes can affect positional density, and shared thresholds need not yield equal selection rates across windows. Front-64K is therefore an empirical cost–quality tradeoff, without implying score or selected-set equivalence to full-128K. The fixed 66/34 mixture does not establish an optimal ratio. Repository-level code and broader multilingual evaluation remain untested. Configuration selection uses HELMET exclusively; RULER is held out for final evaluation.

10 CONCLUSION

We study natural long-document selection for 128K context extension, separating HELMET configuration selection from held-out RULER evaluation. LongPPL key-token density achieves the highest observed aggregate scores among evaluated pipelines, especially on retrieval-oriented tasks. truncLongPPL approaches full-128K downstream scores at evaluated thresholds with a measured $2.28\times$ end-to-end scoring speedup. In the fixed 66/34 mixture, selected documents outperform random sampling without a material additional observed short-context reduction relative to random-long. This ranking persists after identical instruction tuning, with evaluation on LongBench v2 and MRCR, supporting a cost-aware selection recipe within the evaluated setting.

AI USE STATEMENT

Generative AI tools assisted with manuscript preparation, including LaTeX formatting, reference-style conversion, organization and editing for clarity and length, and checking the consistency of result interpretations with the reported numerical results. These editing steps did not generate new experimental measurements. Separately, pretrained language models are components of the evaluated data-selection methods: the LLM-judge pipeline assesses dependencies between document fragments, while the likelihood-based pipelines compute document scores as described in Section 3.2 and Appendices D–E. Beyond these methodological uses and manuscript-preparation assistance, no other generative-AI assistance was used in this work. The authors take responsibility for the final text, claims, and artifacts.

ETHICS STATEMENT

This work studies document selection from an internal pretraining corpus and reports aggregate evaluation results. Long-context likelihood and dependency scores assess training utility; they do not establish that documents are free of private information, copyright restrictions, harmful content, or social biases. Applying these selectors therefore requires separate assessment of data permissions, privacy, and content safety. The coarse source-reweighting control does not constitute a fairness audit. Improved retrieval of distant information may also increase the ability to recover sensitive content from a context; privacy and misuse risks are outside the present benchmark evaluation. Reduced scoring cost does not remove the computational cost of long-context training.

REPRODUCIBILITY STATEMENT

Section 3 and Appendices A–E document the candidate-pool construction, selection rules, scorer configurations, scoring formulas, judge prompt, training settings, and evaluation protocol. Section 5 and Appendices G–I provide threshold and window analyses, timing and agreement diagnostics, and source-reweighting controls. Section 7 and Appendix K describe the fixed-mixture comparison and report variation across three seeds; Section 8 and Appendix L report the post-SFT results and full per-cell statistics. The candidate pool comes from an internal corpus, so exact replication depends on access to that corpus; the disclosed procedures support reimplementation on other corpora.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020. Issue 05.

Jianghao Chen, Junhong Wu, Yangyifan Xu, and Jiajun Zhang. Ladm: Long-context training data selection with attention-based dependency measurement for llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3076–3090, 2025.

Longze Chen, Ziqiang Liu, Wanwei He, Yinhe Zheng, Hao Sun, Yunshui Li, Run Luo, and Min Yang. Long context is not long at all: A prospector of long-dependency data for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8222–8234, 2024.

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

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

Haoran Deng, Yingyu Lin, Zhenghao Lin, Xiao Liu, Yizhou Sun, Yi-An Ma, and Yeyun Gong. Beyond length: Quantifying long-range information for long-context llm pretraining data. *arXiv preprint arXiv:2510.25804*, 2025.

Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. What is wrong with perplexity for long-context language modeling? In *International Conference on Learning Representations*, volume 2025, pp. 94541–94563, 2025.

Chaochen Gao, Xing Wu, Qi Fu, and Songlin Hu. QUEST: Query-centric data synthesis approach for long-context scaling of large language model. *arXiv preprint arXiv:2405.19846*, 2024a.

Chaochen Gao, Xing Wu, Zijia Lin, Debing Zhang, and Songlin Hu. Longmagpie: A self-synthesis method for generating large-scale long-context instructions. *arXiv preprint arXiv:2505.17134*, 2025a.

Chaochen Gao, Xing Wu, Zijia Lin, Debing Zhang, and Songlin Hu. NExtLong: Toward effective long-context training without long documents. *arXiv preprint arXiv:2501.12766*, 2025b.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 2024b. URL https://zenodo.org/records/12608602.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. How to train long-context language models (effectively). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7376–7399, 2025c.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Junlong Jia, Ziyang Chen, Xing Wu, Chaochen Gao, Zijia Lin, Debing Zhang, Songlin Hu, and Binghui Guo. Entropylong: Effective long-context training via predictive uncertainty. *arXiv preprint arXiv:2510.02330*, 2025.

Junlong Jia, Xing Wu, Chaochen Gao, Ziyang Chen, Zijia Lin, Zhongzhi Li, Weinong Wang, Haotian Xu, Donghui Jin, Debing Zhang, et al. Litelong: Resource-efficient long-context data synthesis for llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31274–31282, 2026. Issue 37.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

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

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan 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 (volume 1: Long papers)*, pp. 1525–1534, 2016.

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

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Wen tau Yih, and Mike Lewis. In-context pretraining: Language modeling beyond document boundaries. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LXVswInHOo.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026a.

Microsoft AI Team et al. Mai-thinking-1: Building a hill-climbing machine, 2026b.

Kiran Vodrahalli, Santiago Ontanon, Nilesh Tripuraneni, Kelvin Xu, Sanil Jain, Rakesh Shivanna, Jeffrey Hui, Nishanth Dikkala, Mehran Kazemi, Bahare Fatemi, et al. Michelangelo: Long context evaluations beyond haystacks via latent structure queries. *arXiv preprint arXiv:2409.12640*, 2024.

Longyun Wu, Dawei Zhu, Guangxiang Zhao, Zhuocheng Yu, Junfeng Ran, Xiangyu Wong, Lin Sun, and Sujian Li. Longattn: Selecting long-context training data via token-level attention. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19367–19380, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. Helmet: How to evaluate long-context language models effectively and thoroughly. *arXiv preprint arXiv:2410.02694*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

A Data-Construction Pipelines

A.1 Scoring-Free Constructions: Repacking and Random Sampling

Repack preserves the preceding-stage mixture weights and packs that mixture to our 128K target length, adapting MAI-Thinking-1's 256K recipe (Team et al., 2026b). random uniformly samples documents from $\mathcal{C}$ without applying a quality or dependency score.

A.2 Generic Document-Level Selectors

Following the perplexity definition used by LongPPL (Fang et al., 2025), the perplexity selector scores each document by its token-level perplexity under the frozen Gemma-4-26B-A4B (base) scoring model (Team et al., 2026a). The top-40% rule selects documents above the global 60th percentile, whereas the middle-40% rule selects documents between the 30th and 70th percentiles.

The gzip selector instead scores each document by the ratio of its gzip-compressed UTF-8 byte length to its original byte length. Lower ratios indicate greater compressibility and can expose highly repetitive documents, whereas unusually high ratios can indicate poorly compressible or noisy content. Following OLMo 3 (Olmo et al., 2025), we remove both extremes and select the middle 60% of gzip scores. Appendix C gives the exact score definitions.

A.3 Judge-Model Long-Dependency Selector

The LLM-judge selector uses Gemma-4-26B-A4B-instruct (Team et al., 2026a) to classify whether a later fragment depends on an earlier fragment across M sampled fragment pairs at varying token distances. We rank documents by the fraction of pairs classified as dependent and select the top 5%; Appendix D gives the exact score definition and the system- and user-prompt structure. For the PPL, gzip, and LLM-judge selectors, we draw a fixed number of qualifying documents by reservoir sampling with seed 42.

A.4 Likelihood-Based Long-Dependency Selectors

All three selectors derive their signals from likelihood changes induced by additional context, but differ in granularity and aggregation. Appendix E gives their complete scoring definitions and shared implementation details.

LongFilter. LongFilter (Deng et al., 2025) uses Gemma-4-26B-A4B (base) (Team et al., 2026a) to aggregate continuous token-level loss reductions from short to long context, weighting each reduction by the model's long-context probability of the observed token. We select the highest-scoring documents up to the 6B-token training budget.

Prospector. The original Prospector implementation uses OPT-350M as its scoring model (Chen et al., 2024). Our 128K Prospector adaptation instead uses `SmolLM2-135M` (Allal et al., 2025), a newer and more capable small language model despite its lower parameter count, to score 5,000 sampled pairs of 512-token blocks within each document.[2] Its structured block-pair score combines dependency strength, dependency distance, and dependency specificity; we select the highest-scoring 50% of documents.

LongPPL. We adapt the LongPPL key-token detector (Fang et al., 2025) to document selection. A token is marked as key when a long prefix improves its log-likelihood over a 4K short prefix by more than α and its long-context log-likelihood exceeds β. For a scoring window W, we select documents whose key-token density $\rho_W(x) = |W|^{-1} \sum_{i \in W} k_i$ exceeds τ. We fix the scorer, tokenizer, short-context construction, and (α, β), varying only W and τ. Section 5 evaluates thresholds and

[2] Because Prospector evaluates 5,000 block-pair comparisons per document, its scoring cost in our 128K setting remains substantially higher than that of LongFilter or LongPPL. We therefore use `SmolLM2-135M` rather than Gemma-4-26B-A4B for this pipeline. This scorer substitution should be viewed as an implementation adaptation rather than an exact reproduction of the original Prospector pipeline.

Table 4: Definitions of the data-construction pipelines compared in Section 4, grouped by the signal taxonomy used in Section 2. The first column maps each method to the shorthand used in the result figures. Except for REPACK, every method operates on the candidate pool $\mathcal{C}$ of natural long documents and preserves document boundaries. All percentile thresholds are computed globally over $\mathcal{C}$.

Method	Document signal	Selection rule or construction
Scoring-free constructions		
REPACK	None	Preserve the preceding-stage mixture weights and pack its documents into 128K sequences; documents may be concatenated.
random	None	Uniformly sample documents from $\mathcal{C}$ without document scoring.
Generic document-level selectors		
PPL (top40 / mid40)	Token perplexity under Gemma-4-26B-A4B (base)	Select scores above the 60th percentile or between the 30th and 70th percentiles, respectively.
gzip (mid60)	gzip compression ratio	Select scores between the 20th and 80th percentiles.
Judge-model long-dependency selector		
LLM judge (top5)	Fraction of sampled long-distance fragment pairs judged dependent	Select the top 5% of documents.
Likelihood-based long-dependency selectors		
LongFilter	Probability-weighted token-loss reduction from short to long context	Select the highest-scoring documents up to the 6B-token budget.
Prospector	Dependency strength, distance, and specificity over sampled block pairs	Select the highest-scoring 50% of documents.
LongPPL / truncLongPPL	Density $\rho_W(x)$ of LongPPL key tokens in scoring window W	Select documents satisfying $\rho_W(x) > \tau$; Section 4.2 reports truncLongPPL (front-64K, $\tau = 0.05$), with threshold and span analyses in Section 5 and position analysis in Appendix H.3.

scored spans of 16K–128K. Equal-length window positions and more complex layouts are evaluated separately in Appendix H.3 and Appendix H.4.

B TRAINING AND EVALUATION DETAILS

Positional encoding and optimization. We extend Qwen2.5-3B from its base context directly to 128K by increasing the RoPE base frequency to $\theta = 2 \times 10^8$, with no other positional-encoding change. Training uses bf16 and AdamW (β_1=0.9, β_2=0.95, weight decay 0.1, gradient clip 1.0), with a constant learning rate of 1×10^{-6} (no warmup or decay), a global batch of approximately 8M tokens per optimizer update, and 6B tokens in total.

Short-context tasks. The seven-task short-context average uses `lm-evaluation-harness` (Gao et al., 2024b) in a zero-shot setting on `hellaswag` (Zellers et al., 2019), `lambada_openai` (Paperno et al., 2016), `arc_challenge` and `arc_easy` (Clark et al., 2018), `piqa` (Bisk et al., 2020), `winogrande` (Sakaguchi et al., 2021), and `logiqa` (Liu et al., 2020), evaluated in bfloat16 over each task's full test set. We report the unweighted mean of the raw `acc` metric across the seven tasks.

C Implementation Details for Generic Document-Level Selectors

For a document x, the perplexity selector uses the frozen scoring model p_{θ_0} and computes

$$s_{\mathrm{PPL}}(x) = \exp\left(-\frac{1}{|x|}\sum_{i=1}^{|x|}\log p_{\theta_0}(x_i \mid x_{<i})\right). \tag{1}$$

The gzip selector computes the ratio of the byte length after level-6 gzip compression to the original UTF-8 byte length:

$$s_{\mathrm{gzip}}(x) = \frac{|\,\mathrm{gzip}_6(\mathrm{UTF8}(x))|}{|\,\mathrm{UTF8}(x)|}. \tag{2}$$

D LLM-Judge Selector Details and Prompt

The LLM-judge pipeline in Section 3.2 evaluates whether a later fragment depends on an earlier fragment. Given M sampled fragment pairs and judge predictions y_m, its document score is the fraction classified as dependent:

$$s_{\mathrm{judge}}(x) = \frac{1}{M}\sum_{m=1}^{M}\mathbb{I}[y_m = \text{dependent}]. \tag{3}$$

The judge is queried with a single user message and no separate system prompt; the complete message is reproduced verbatim in the box below. The placeholders {`head_text`} and {`tail_text`} are replaced by the earlier and later fragment, respectively (each truncated to its leading 512 and trailing 256 tokens, with the middle elided as "…" when longer). The model is instructed to return a single JSON object with a one-sentence rationale, a binary `verdict`, and the matched criterion.

Prompt used by the LLM-judge selector

```
You are a rigorous expert in judging inter-passage DEPENDENCY within a long
document.

Below are TWO NON-ADJACENT fragments from the SAME long document: Fragment 1
comes earlier and Fragment 2 comes later, and they may be far apart in the
original text. To save space, an overly long fragment keeps only its beginning
and end, with the middle elided as "...".

[Fragment 1 (earlier)]
{head_text}

[Fragment 2 (later)]
{tail_text}

# Task
Imagine a reader who has read ONLY Fragment 2 and has never seen Fragment 1:
would they fail to correctly understand Fragment 2 because some specific piece
of information supplied by Fragment 1 is missing?
In other words - does Fragment 2 contain a "dangling" reference / premise /
constraint / response whose gap can ONLY be filled by Fragment 1?

# Criteria for "DEPENDENT (1)" (any one suffices; the dependency MUST point to
specific content in Fragment 1)
  C1 Entity reference
    Meaning: Fragment 1 first defines or introduces a specific entity, person,
    concept, term, abbreviation, symbol, variable, or label; Fragment 2 then
    uses it directly without restating its identity or meaning.
    Typical signs: Fragment 2 says "this method / the above system / he / it /
    this parameter / as mentioned earlier", or directly uses a proper noun,
    abbreviation, or label that only Fragment 1 explains.
    Positive: F1 defines "we denote our model as M"; F2 says "M reaches 90% on
    the test set" - a reader who skips F1 has no idea what M is.
    Boundary: if the entity is common domain knowledge (e.g. "CPU", "DNA") or
    Fragment 2 explains it itself, it is NOT a dependency.

  C2 Instruction-response
    Meaning: Fragment 1 poses a specific question, task, requirement, prompt,
    or to-do; Fragment 2 is precisely its answer, solution, execution, or
    fulfillment.
    Typical signs: Fragment 1 is a question / problem statement / requirement,
```

```
    and Fragment 2 is the answer / solution / implementation / conclusion; or
    Fragment 2 continues with "to answer the above / to accomplish that task".
    Positive: F1 asks "how to reduce GPU memory usage?"; F2 answers "using a
    paged KV cache cuts it by 40%".
    Boundary: if Fragment 2 merely happens to discuss a related topic but does
    not address that specific question in Fragment 1, it does NOT count.

  C3 Numeric / constraint carry-over
    Meaning: Fragment 1 sets a specific value, parameter, unit, range, initial
    condition, assumption, or constraint; Fragment 2 builds on it to continue
    computing, deriving, or arguing.
    Typical signs: Fragment 2 uses "that value / the above parameter / under
    this setting", or directly reuses a constant, threshold, formula symbol, or
    experimental config from Fragment 1 without re-declaring it.
    Positive: F1 sets "learning rate = 1e-4, batch = 256"; F2 says "under this
    config it converges in 3 epochs".
    Boundary: if the two fragments each provide their own independent values and
    do not reuse each other's, it does NOT count.

  C4 Causal / temporal continuation
    Meaning: Fragment 2 is the direct result, subsequent step, next stage, or
    temporal continuation of an event, action, reasoning, or procedure
    described in Fragment 1, forming a before->after / cause->effect chain.
    Typical signs: Fragment 2 continues Fragment 1's narrative or reasoning with
    "therefore / thus / then / next / after that / finally"; or it is clearly
    the next step of the same procedure or storyline.
    Positive: F1 "mix the reactants and heat to 80C"; F2 "once the solution
    turns clear, filter it" - the next step of the same procedure.
    Boundary: if they are merely two unrelated facts under the same topic with
    no causal or sequential link, it does NOT count.

  C5 Citation / call-back
    Meaning: Fragment 2 explicitly cites, responds to, rebuts, supplements, or
    extends a specific claim, stance, conclusion, evidence, or viewpoint
    established in Fragment 1.
    Typical signs: Fragment 2 contains call-backs like "as noted above / as
    argued earlier / on the contrary / building on this / this confirms the
    earlier conclusion", and the referent is specific content in Fragment 1.
    Positive: F1 argues "a long context does not necessarily lower perplexity";
    F2 "based on the above conclusion, we redesigned the evaluation".
    Boundary: if Fragment 2 only vaguely mentions a similar viewpoint rather
    than calling back the specific claim/evidence built in Fragment 1, it does
    NOT count.

# When to judge "NOT DEPENDENT (0)" (avoid false positives)
  - Fragment 2 is self-contained and can be fully understood without Fragment 1.
  - The two fragments only share "same topic / same domain / similar wording /
    both mention some common-knowledge entity", but Fragment 2 does not actually
    need any specific information from Fragment 1 - such surface similarity does
    NOT count as dependency.
  - The two fragments are independent, parallel items/sections (e.g.
    encyclopedic listings, mutually-unreferenced problem lists), even if the
    same terminology recurs.

# Output
First give a one-sentence rationale that pinpoints WHICH specific element of
Fragment 2 does or does not depend on Fragment 1 (be concrete, not generic),
then give the verdict. Output ONLY the JSON, nothing else:
{"reasoning":"...","verdict":0 or 1,"criterion":"C1/C2/C3/C4/C5 or none"}
```

E Implementation Details for Likelihood-Based Selectors

This appendix gives the full score definitions and scoring configuration summarized in Section 3.2.

LongFilter. LongFilter (Deng et al., 2025) contrasts token losses under long and short contexts. Let ℓ_i^{long} and ℓ_i^{short} denote the corresponding negative log-likelihoods. Its document score is

$$s_{\text{LongFilter}}(x) = \frac{1}{|x|} \sum_{i=1}^{|x|} \exp(-\ell_i^{\text{long}}) \big(\ell_i^{\text{short}} - \ell_i^{\text{long}}\big), \tag{4}$$

which weights each loss reduction by the model's long-context probability of the observed token.

Prospector. The original Prospector pipeline uses OPT-350M and divides 32K documents into $N = 256$ blocks of 128 tokens (Chen et al., 2024). Our 128K adaptation uses the more capable

`SmolLM2-135M` scorer and keeps $N = 256$ by increasing the block size to 512 tokens. For an earlier block c_j and a later block c_i, it computes dependency strength

$$\mathrm{DST}(i, j) = 1 - \frac{\mathrm{PPL}(c_i \mid c_j)}{\mathrm{PPL}(c_i)}, \tag{5}$$

and dependency distance $\mathrm{DDI}(i, j) = (i - j)/(N - 1)$. With $\Delta_{ij} = \mathrm{PPL}(c_i) - \mathrm{PPL}(c_i \mid c_j)$ and $q_{ij} = \exp(\Delta_{ij}) / \sum_{k<i} \exp(\Delta_{ik})$, dependency specificity for target blocks with at least two possible predecessors ($i \geq 3$) is the normalized inverse entropy

$$\mathrm{DSP}(i) = 1 - \frac{-\sum_{j<i} q_{ij} \log q_{ij}}{\log(i - 1)}. \tag{6}$$

The document score is the sum of $(a\,\mathrm{DST}(i, j) + b\,\mathrm{DDI}(i, j))\,\mathrm{DSP}(i)$ over 5,000 sampled block pairs with $i \geq 3$ satisfying $\mathrm{DST}(i, j) > 0.1$. We follow Chen et al. (2024) with $(a, b) = (1, 1)$.

LongPPL key-token density. Let $x_{<i}^{\mathrm{long}}$ be the available long prefix of target token x_i, and $x_{<i}^{\mathrm{short}}$ its most recent K prefix tokens. Following LongPPL (Fang et al., 2025), we define the long–short likelihood difference as

$$\mathrm{LSD}(i) = \log p_{\theta_0}(x_i \mid x_{<i}^{\mathrm{long}}) - \log p_{\theta_0}(x_i \mid x_{<i}^{\mathrm{short}}) \tag{7}$$

and let $\mathrm{LCL}(i) = \log p_{\theta_0}(x_i \mid x_{<i}^{\mathrm{long}})$. A token is classified as a key token when

$$k_i = \mathbb{I}[\mathrm{LSD}(i) > \alpha \ \wedge \ \mathrm{LCL}(i) > \beta]. \tag{8}$$

The LCL condition removes tokens that remain intrinsically unpredictable even with long context. For a scoring window W, the document-level key-token density is

$$\rho_W(x) = \frac{1}{|W|} \sum_{i \in W} k_i, \tag{9}$$

and the corresponding selected set is

$$\mathcal{D}_{W,\tau} = \{x \in \mathcal{C} : \rho_W(x) > \tau\}. \tag{10}$$

Thus, the document thresholds from 0.01 to 0.10 in our experiments apply to ρ_W, not to the LongPPL token-level parameters α and β.

Scoring configuration and windows. The frozen scoring model p_{θ_0} is Gemma-4-26B-A4B (base) (Team et al., 2026a), served in bf16 through vLLM (Kwon et al., 2023) with a FlashAttention backend (Dao et al., 2022) on H20 GPUs. We use its native tokenizer, set the short-context length to $K = 4096$, and follow LongPPL (Fang et al., 2025) by fixing the token-level thresholds to $(\alpha, \beta) = (2.0, -2.0)$. All LongPPL scoring-window lengths and the 128K truncation length are measured after tokenization with this Gemma tokenizer. We compute LSD and LCL for every target token from full-context log probabilities (`echo`, one log probability per position), without target-token subsampling; short-context losses use a sliding window with stride 1024. We score documents up to 128K Gemma tokens. Full-context scoring of the complete pool requires on the order of 10^4–10^5 H20 GPU-hours, making the scored-span length an important computational variable.

We score the first 16K, 32K, or 64K tokens; the last 32K or 64K tokens; or the complete 128K span. We also test the intersection of documents independently selected by the front and tail 64K windows. These configurations compare window position and length. Across the document-selection sweeps, we hold the scoring model, tokenizer, short-context construction, and token-level thresholds (α, β) fixed; only the document-level threshold τ and scoring window W vary.

F OLMo-Style Middle-60% Selection Control

OLMo 3 selects scientific PDFs by removing the most and least compressible 20% and reports no clear observed advantage for its LongPPL-derived alternatives based on key-token fraction or positional spread over gzip (Olmo et al., 2025). Its evaluated LongPPL rules include one-sided bottom-20% removal and symmetric top-and-bottom-20% removal. Our main key-token-density

configurations instead target a substantially narrower high-density region, although they also differ from OLMo 3 in scorer, tokenizer, corpus, base model, and training recipe.

To examine this selection convention, we reproduce the symmetric middle-60% LongPPL-versus-gzip comparison in our setting. Table 5 shows the same qualitative result as OLMo 3: the two pipelines differ by at most 1.17 points across all evaluated budgets and by at most 0.51 points from 4B tokens onward. At the shared 6B-token budget, both remain below the random baseline of 51.66 validation HELMET / 78.05 held-out RULER. LongPPL consistently yields the higher held-out RULER score, whereas gzip yields the higher validation HELMET score from 6B tokens onward; under this design, these small differences do not establish a reliable ordering. This aligned control covers only symmetric middle-range selection rather than the complete OLMo 3 recipe.

This aligned experiment therefore reproduces the qualitative OLMo 3 conclusion: when LongPPL uses a broad middle-60% selection rule, its observed scores differ only slightly from gzip, and at the shared 6B-token budget neither selector is higher than random sampling on both aggregates. Our main configurations instead apply the absolute key-token-density criterion $\rho_W(x) > \tau$ with $\tau \in [0.03, 0.05]$. With full-128K scoring, these thresholds select 216,448 (7.2%) and 51,959 (1.7%) documents, respectively, from the 3M-document candidate pool. Taken together with the threshold sweep in Section 5.1, this contrast suggests that the OLMo-style middle-60% rule may be too permissive to isolate the high-key-token-density tail targeted by our main configurations. The absolute density values remain specific to our scorer, tokenizer, corpus, and token-level thresholds and should therefore be treated as a training-recipe reference rather than a transferable cutoff.

G Front-64K versus Full-128K: Agreement Diagnostics and Scoring Cost

As a partial, selection-conditioned audit, we compare front-64K and full-128K scoring among documents that exceed a low density threshold in at least one window. Their per-document key-token densities are strongly correlated (Pearson $r = 0.83$; Spearman $\rho = 0.77$). Because this subset is conditioned on the observed scores, these coefficients describe association within the audited subset rather than unconditional estimator accuracy over the candidate pool. Let $S_{64}(\tau)$ and $S_{128}(\tau)$ denote the sets selected by front-64K and full-128K scoring at threshold τ. Across $\tau \in \{0.02, 0.03, 0.05\}$, the one-way front-to-full containment rate

$$\frac{|S_{64}(\tau) \cap S_{128}(\tau)|}{|S_{64}(\tau)|} \tag{11}$$

ranges from 76% to 92%, with full-128K scoring generally selecting a larger set. This quantity is the fraction of front-64K-selected documents also selected by full-128K scoring, not a symmetric agreement measure.

For this model, scoring the first 64K rather than the full 128K reduces the estimated prefill FLOPs to approximately $0.37\times$ (a $2.7\times$ reduction), based on its 25 sliding-window and 5 full-attention layers. We also directly measure per-document wall-clock time for the complete scoring pipeline. The full-context loss (stage 1) decreases from 7.5 to 2.55 s ($2.94\times$), and the sliding-window short-context loss (stage 2) decreases from 9.05 to 4.70 s ($1.93\times$). The combined time therefore decreases from 16.55 to 7.25 s per document, yielding an end-to-end $2.28\times$ speedup.

The stages scale differently by construction. Stage 1 comprises a single long-sequence prefill whose attention incurs a superlinear $O(n^2)$ term, so halving the sequence length reduces its cost by more than $2\times$. Stage 2 uses a fixed $\sim$5K-token sliding window, and its number of windows scales linearly with document length; halving the scored span therefore approximately halves the number of windows. Because stage 2 accounts for most of the total cost (55% at 128K and 65% at 64K), the end-to-end $2.28\times$ speedup is closer to the stage-2 ratio of approximately $2\times$ than to the stage-1 or FLOPs ratio.

H Additional LongPPL-Density Analyses

H.1 Position-Wise Key-Token Density

Figure 8 shows that the raw mean density increases monotonically across the 8K bins. This trend should not be attributed solely to intrinsic positional differences among document regions: under

Table 5: OLMo-style middle-60% selection comparison in our training setup across budgets of 2B–10B tokens. Columns are grouped into validation HELMET and held-out RULER; within each group, the two subcolumns compare LongPPL and gzip selection. The higher value in each (budget, metric) pair is bold.

	HELMET (validation)		RULER (held-out)	
Training tokens	LongPPL	gzip	LongPPL	gzip
2B	**50.41**	49.24	**76.67**	75.86
4B	**50.56**	50.36	**77.84**	77.55
6B	50.40	**50.71**	**77.91**	77.45
8B	50.33	**50.68**	**77.93**	77.72
10B	50.54	**51.05**	**78.15**	78.01

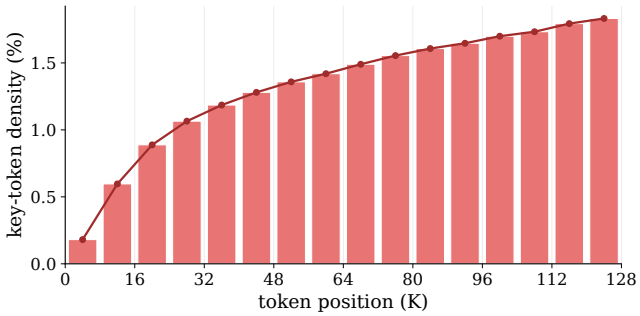

Figure 8: Raw mean LongPPL key-token density across 8K positional bins for 30,000 natural long documents truncated to 128K tokens. Density increases with position, but the available prefix also grows; the curve is therefore descriptive rather than evidence of a position-only effect.

the LongPPL definition, later target tokens can use a longer available prefix, whereas early targets have less context beyond the 4K short-context reference, so the observed curve combines document structure with this construction effect. The front-64K choice is instead evaluated as an empirical cost–quality operating point through the downstream comparison in Section 5.2.

H.2 Sensitivity to the Training-Token Budget

The main comparisons fix the training budget at 6B tokens. We vary the budget from 2B to 10B tokens while holding the front-64K scoring window fixed to test whether the threshold ordering persists across continued-training scales.

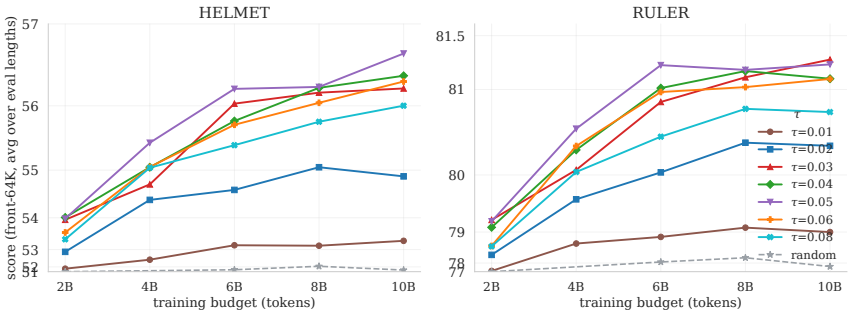

Figure 9: Validation HELMET (left) and held-out RULER (right) by continued-training budget for front-64K LongPPL key-token-density selection. Low thresholds remain lower, while $\tau \in \{0.03, 0.04, 0.05, 0.06\}$ forms a higher cluster from approximately 6B tokens onward.

As shown by HELMET validation in Figure 9, the permissive $\tau \in \{0.01, 0.02\}$ configurations remain lower, indicating that additional training tokens do not compensate for insufficient selectivity. Thresholds from 0.03 to 0.06 form a higher-performing cluster from approximately 6B tokens onward, with generally small subsequent gains. Within this cluster, $\tau = 0.06$ has no consistent validation advantage over $\tau = 0.05$, while $\tau = 0.08$ remains lower. We therefore retain $\tau = 0.05$ as the default high-density configuration.

H.3 Window Position at Threshold 0.05

Having fixed the window length, we ask *where* in the document the density should be measured. We compare front and tail windows of equal length, their intersection, the full span, and eight scattered 8K windows.

Figure 10 shows small observed differences among the tested positions. On HELMET validation, front-64K achieves 56.23 compared with 55.70 for tail-64K, and neither the front–tail intersection nor the scattered layout provides a consistent validation advantage over the front window. We therefore select the front window as the simplest contiguous default. After this choice is frozen, its held-out RULER score is 81.23, compared with 81.00 for tail-64K.

H.4 Scattered versus Contiguous Windows

To test whether broader positional coverage helps, Table 6 compares eight randomly located 8K windows (64K target-token positions in total) with one contiguous front-64K window at the same threshold. Across $\tau \in \{0.03, 0.04, 0.05\}$, validation HELMET differs by approximately one point or less, with no consistent direction, so broader positional coverage does not change the selected default. Held-out RULER, reported after this decision, also differs by less than one point with no consistent direction.

H.5 Window-Length Sensitivity at Threshold 0.03

The main window-length analysis uses the default $\tau = 0.05$, the least restrictive point on the 0.05–0.08 validation plateau in Section 5. We repeat that comparison at the more permissive threshold $\tau = 0.03$.

The qualitative downstream pattern matches the $\tau = 0.05$ analysis: the front-64K and full-128K pipelines differ only slightly in the observed scores, whereas front-16K is lower at long evaluation lengths.

I Coarse Domain Reweighting Control

Selection may change the proportions of books, web pages, and papers. To assess whether this shift explains the observed improvement, we resample the front-64K selected documents so that their

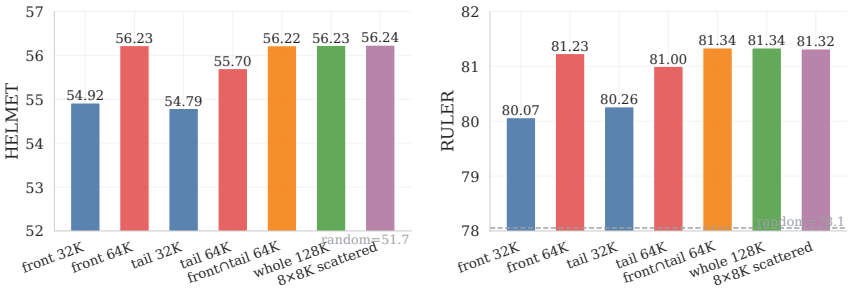

Figure 10: Macro-averaged validation HELMET (left) and held-out RULER (right) for scoring-window layouts at $\tau = 0.05$ under the 6B-token budget. Dashed lines mark random sampling of natural long documents. All layouts exceed this baseline.

Table 6: Scattered (8×8K, randomly located) versus contiguous (front-64K) scoring at shared absolute thresholds, 6B tokens. Cells report validation HELMET / held-out RULER, each averaged over evaluation lengths.

Threshold τ	Scattered 8×8K	Contiguous front-64K
0.03	55.01 / 80.55	56.03 / 80.87
0.04	56.23 / 80.97	55.79 / 81.01
0.05	56.24 / 81.32	56.23 / 81.23

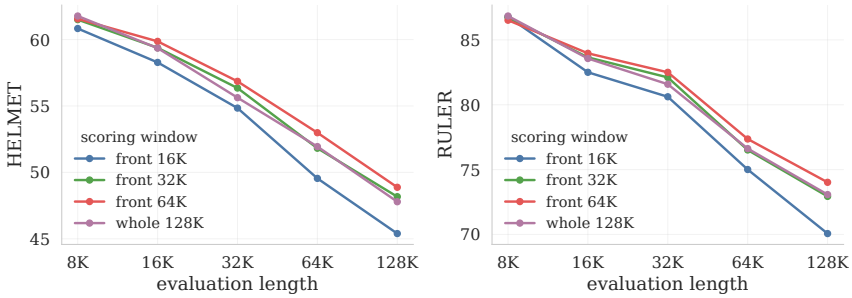

Figure 11: Scores by evaluation length for front-16K, front-32K, front-64K, and full-128K scoring at $\tau = 0.03$ under the 6B-token budget. At 128K, front-16K is lower, while front-64K and full-128K remain close (cf. Figure 4).

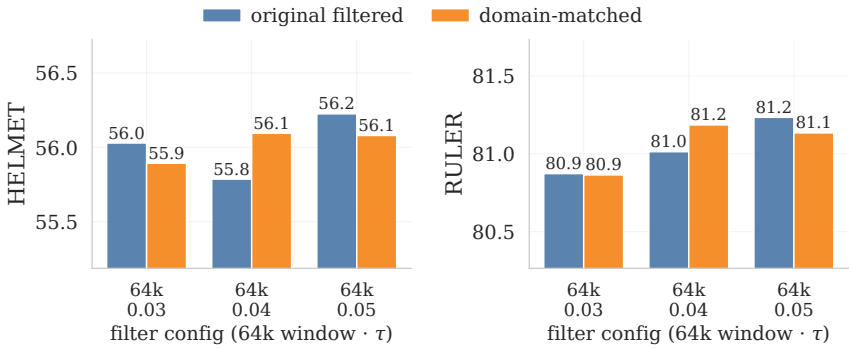

Figure 12: Domain-mixture control for front-64K selection under the 6B-token budget. Resampling restores the candidate pool's coarse domain proportions and changes validation HELMET (left) by at most 0.3 point and held-out RULER (right) by at most 0.2 point.

domain proportions match those of the original candidate pool. If implicit domain reweighting were the primary explanation, the improvement would be expected to diminish after this correction.

Across all three thresholds, domain-matched resampling changes HELMET by at most 0.3 point and RULER by at most 0.2 point, with no systematic direction. Coarse domain proportions alone are therefore insufficient to account for the observed difference, although this control does not rule out finer within-domain shifts in topic, length, duplication, or quality.

J Short-Context Performance under Long-Context Continued Training

Figure 13 compares the uncontinued base model with all 6B-token continued-training pipelines from Section 4.2. Every continued-training configuration has a lower seven-task short-context average than the base model, but the selected-long configuration does not show an additional reduction relative to the other long-context extensions. Table 7 reports the per-task short-context scores for the three fixed-mixture configurations of Section 7 (Figure 7): the short-only packing reference (dolma3-short),

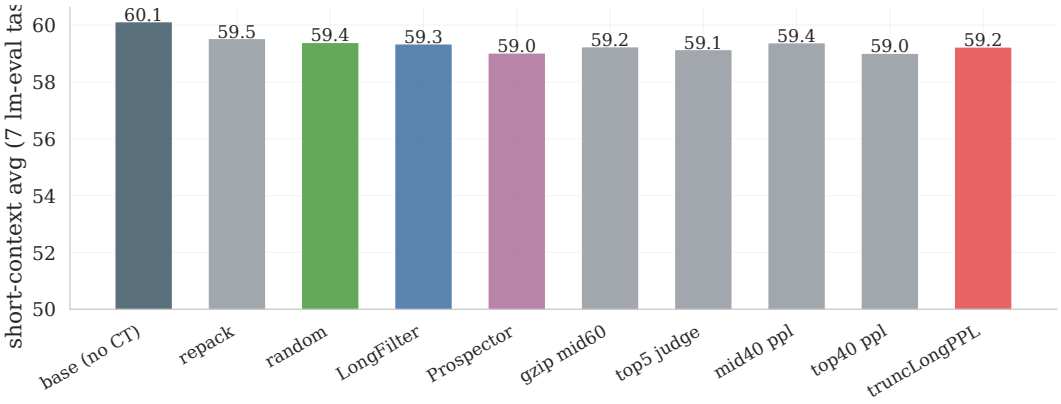

Figure 13: Seven-task short-context average for the uncontinued base model and the 6B-token continued-training pipelines. The base model scores 60.1, the continued models span 59.0–59.5, and truncLongPPL scores 59.2.

Table 7: Per-task short-context accuracy (%) for the uncontinued base model (BASE, no long-context extension) and the three fixed-mixture configurations in Section 7 under the shared 10B-token budget: short-only packing, and the 66/34 mixtures adding random versus truncLongPPL-selected long documents. Values are for a single representative seed; the corresponding three-seed averages of the final column appear in Table 8. Columns are the seven LM-Evaluation-Harness tasks and their unweighted average. The highest value in each column is shown in bold.

Configuration	ARC-C	ARC-E	HellaSwag	LAMBADA	LogiQA	PIQA	WinoG.	Avg.
BASE (no CT)	45.05	77.36	**55.00**	**66.89**	**29.95**	**78.24**	68.35	**60.12**
dolma3-short (pack)	**48.21**	**79.17**	54.27	66.31	26.27	78.18	68.19	60.09
+ random-long	47.01	78.58	54.17	65.63	27.34	78.02	68.19	59.85
+ truncLongPPL-long	46.84	78.45	54.10	65.26	27.34	78.13	**68.43**	59.79

Table 8: Seed stability of the three fixed-mixture configurations of Section 7 at the shared 10B-token budget, reported as mean ± standard deviation over three independent training seeds. Short is the seven-task short-context average; HELMET is the macro-averaged validation score; RULER is the macro-averaged held-out score. The highest value in each column is in bold.

Configuration	Short	HELMET	RULER
dolma3-short (pack)	$\mathbf{60.04 \pm 0.11}$	49.44 ± 0.12	77.16 ± 0.15
+ random-long	59.81 ± 0.04	50.72 ± 0.10	78.61 ± 0.15
+ truncLongPPL-long	59.79 ± 0.06	$\mathbf{55.31 \pm 0.27}$	$\mathbf{81.94 \pm 0.16}$

and the two 66/34 mixtures that add random versus truncLongPPL-selected natural long documents from the same pool. We also include the uncontinued base model (BASE, no long-context extension) as a reference. The seven tasks are ARC-Challenge, ARC-Easy, HellaSwag, LAMBADA (OpenAI), LogiQA, PIQA, and WinoGrande, each evaluated with the LM Evaluation Harness (Gao et al., 2024b); the final column is their unweighted average. All four configurations stay within 0.3 point on the average: adding selected long documents matches the random-long mixture to within 0.1 point and the short-only reference to within 0.3 point, and remains comparable to the uncontinued base model.

K SEED STABILITY OF THE FIXED-MIXTURE COMPARISON

To verify that the fixed-mixture ranking in Section 7 is not an artifact of a single random seed, we retrain each of the three configurations under three independent seeds and report the resulting

seven-task short-context average, macro-averaged validation HELMET, and held-out RULER. Table 8 gives the mean $\pm$ standard deviation over the three runs. The between-seed standard deviations are small on all three metrics (≤ 0.27 HELMET, ≤ 0.16 RULER, ≤ 0.11 short-context), so the gaps between configurations are much larger than the seed-level variation: on HELMET and RULER, truncLongPPL-long exceeds random-long by 4.59 and 3.33 points, respectively, both an order of magnitude above the corresponding standard deviations. Short-context averages remain within 0.25 point across all configurations.

L Supervised Fine-Tuning Evaluation Details

Tables 9 and 10 report the full mean $\pm$ standard deviation results for the SFT comparison in Section 8. The main text reports means for readability; the LongBench v2 difficulty splits are retained here as additional breakdowns.

Table 9: LongBench v2 accuracy (%) after supervised fine-tuning for the three fixed-mixture configurations of Section 7, reported as mean $\pm$ standard deviation over three fine-tuning runs. Columns follow the LongBench v2 overall score, the difficulty splits (Easy/Hard), and the context-length splits (Short/Medium/Long). The highest value in each column is in bold.

Configuration	Overall	Easy	Hard	Short	Medium	Long
dolma3-short (pack)	29.6 ± 0.2	32.3 ± 1.4	28.0 ± 1.1	37.8 ± 0.9	25.7 ± 1.1	23.7 ± 1.1
+ random-long	30.9 ± 0.1	33.7 ± 0.6	29.2 ± 0.5	38.1 ± 0.3	26.3 ± 0.3	27.8 ± 0.9
+ truncLongPPL-long	$\mathbf{32.5 \pm 0.4}$	$\mathbf{34.4 \pm 0.5}$	$\mathbf{31.3 \pm 0.5}$	$\mathbf{40.2 \pm 2.3}$	$\mathbf{27.3 \pm 2.3}$	$\mathbf{29.9 \pm 1.4}$

Table 10: MRCR accuracy (%) at context lengths up to 128K for the three fixed-mixture configurations of Section 7, reported as mean $\pm$ standard deviation over three independent training seeds ($N = 1476$). Columns are the overall score and the 2-, 4-, and 8-needle subsets. The highest value in each column is in bold.

Configuration	Overall	2-needle	4-needle	8-needle
dolma3-short (pack)	16.16 ± 0.02	22.27 ± 0.23	15.13 ± 0.14	11.08 ± 0.30
+ random-long	17.62 ± 0.04	24.13 ± 0.19	16.74 ± 0.02	11.98 ± 0.15
+ truncLongPPL-long	$\mathbf{18.91 \pm 0.22}$	$\mathbf{26.00 \pm 0.14}$	$\mathbf{17.53 \pm 0.36}$	$\mathbf{13.16 \pm 0.18}$