MEASURING THE ATTENTION SPAN OF LONG CONTEXT TRAINING DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

LLMs are typically pretrained with short context lengths which are later extended through additional long-context training. As long context training is expensive, this stage requires careful data selection to maximize effectiveness. Data selection methods should distinguish a cohesive document with long dependencies across distant sections from a simple concatenation of parts with no real long distance structure. We propose to score documents based on their *AttentionSpan*, which measures the look-back distance covered by the attention mechanism during a forward pass. Naive measurements of attention span are heavily influenced by attention sinks, making them a useless proxy for document structure. We present simple tricks for ignoring attention sinks, resulting in meaningful metrics of long-context structure that are useful for data filtering. When extending OLMo-3-7B from 8K to 32K using 5B tokens sampled from its own long-text mixture, additional AttentionSpan curation improves HELMET-Recall from 62.0 to 75.7. The advantage remains even in a domain-matched experiment, in which context is extended using the same code, arXiv and book proportions for each checkpoint, and AttentionSpan is only used for filtering within domains. These results support sink-corrected attention distance as an interpretable curation signal for long-range retrieval training.

1 INTRODUCTION

Extending the effective context window of an LLM is a central challenge when training transformers. Architectural techniques such as RoPE (Su et al., 2024), positional interpolation, and efficient attention allow models to process increasingly long sequences. Yet supporting a long sequence does not imply that a model can effectively use information throughout it. Models often struggle to retrieve and integrate information over long distances, including the well-known "lost in the middle" behavior (Liu et al., 2024; Levy et al., 2024). Consequently, recent models commonly include a dedicated long-context finetuning stage in which a shorter-context model is further trained after its context window is extended.

This raises a basic question: *what data should a model see during long-context training?* Simply selecting long documents does not guarantee that they contain meaningful long-range dependencies. Recent work instead scores documents using signals such as perplexity or attention (Chen et al., 2024a; 2025; Wu et al., 2025; Fu et al., 2024). These approaches show that data selection can substantially affect long-context performance, but the signal itself is difficult to isolate. Long-range structure is strongly correlated with other factors such as document length, content, inherent complexity etc. (Lv et al., 2024). For example, code is both naturally long and rich in non-local dependencies. Attention-based scoring introduces an additional difficulty in the form of attention sinks that can distort long-range dependencies. Thus, a document can appear rich in long-range dependencies without actually having meaningful long-range signals, or vice versa.

We introduce *AttentionSpan*, a document-level measure of long-range attention usage designed to separate these effects. For each query token, AttentionSpan identifies its most strongly attended valid keys and measures their distance into the past. A key observation of our work is that this metric is only meaningful if we discard attention sinks. By summarizing these distances using an attention-weighted median over the sequence, and by introducing prefill tokens to act as removable attention sinks at early positions, we prevent positional attention sinks from being interpreted as

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.

long-range dependencies. The result is a document score that provided a strong signal for selecting useful long-context data.

We evaluate AttentionSpan through controlled data-selection experiments across model families, context lengths, and training scales. Our experiments range from extending the context length of Llama-2-7B with 4k native context at 16K and 32K using LoRA to full-parameter context extension of OLMo-3-7B from 8K to 32K, as well as Llama-3.1-8B at 128K. Across these settings, training on high-AttentionSpan data consistently improves long-context performance over random selection, including a 13.7-point improvement on HELMET-Recall for OLMo-3-7B and a 7.9-point improvement on HELMET for Llama-3.1-8B. Controlled experiments further show that these gains cannot be explained solely by domain composition. Under matched OLMo-3-7B training, AttentionSpan also substantially outperforms prior attention-based data selectors and remains competitive with perplexity-based selection.

Beyond selecting data with a fixed grader, we study how the choice of grader affects data selection. Different models produce related but non-identical rankings, motivating consensus-based selection using Borda aggregation. We also consider the complementary setting in which a model grades data for its own context extension. Both settings improve over their respective single-grader or random baselines, suggesting that long-context data selection need not depend on a single fixed external scorer.

Our contributions are as follows:

- We propose **AttentionSpan**, a metric for measuring the long-range attention structure of a document. AttentionSpan summarizes the lookback distance to strongly attended keys using an attention-weighted median, providing a robust document-level signal for long-context data selection.
- We introduce a general **attention-sink correction** based on prefill tokens, which absorb content-independent attention sinks and are excluded from scoring. The correction is independent of AttentionSpan and can be applied to other attention-based analyses of long-range dependencies.
- Across multiple model families and context lengths, high-AttentionSpan documents consistently improve long-context performance over random selection, while low-AttentionSpan documents provide a complementary negative control. The signal persists after controlling for domain composition.
- Under matched training conditions, AttentionSpan substantially outperforms prior attention-based selectors, including LongAttn and LADM.
- To reduce scoring cost, we find that three depth-spaced layers preserve full-depth document rankings at $r = 0.975$ while providing an approximately $9\times$ speedup, making AttentionSpan practical at larger data scales (Appendix E).
- Finally, we extend AttentionSpan beyond a single fixed grader through **consensus selection** and **self-selection**. Borda combines rankings across graders to reduce dependence on any individual model, while self-selection allows a model to curate data for its own context extension.

2 Related Work

Long-context extension. A common recipe for long-context language modeling is to start from a shorter-context pretrained model and extend its usable context window through positional interpolation, RoPE (Su et al., 2024) scaling, sparse attention, or continued training on longer sequences. Methods such as position interpolation (Chen et al., 2023), YaRN (Peng et al., 2024), LongLoRA (Chen et al., 2024b), LongRoPE (Ding et al., 2024), and continued pretraining with upsampled long texts (Xiong et al., 2024; Gao et al., 2025) make it possible to train or evaluate models at any token length, irrespective of the base model's training context length. These works primarily address the architectural and optimization challenges of processing longer sequences. But without training on actual long context samples, it is difficult for a model to perform tasks that require true long context capabilities.

General-purpose data selection. A separate line of work studies data selection for improving general language-model training. Distribution-matching and feature-based methods such as DSIR (Xie et al., 2023b) and QuRating (Wettig et al., 2024) select data according to corpus-level similarity, quality, or text features, and data-mixture optimization approaches such as DoReMi (Xie et al., 2023a) and Data Mixing Laws (Ye et al., 2025) re-weight domain proportions to improve downstream performance. Classifier-based pipelines such as DCLM (Li et al., 2024a) target broad web-data quality, while influence- and gradient-based methods such as LESS (Xia et al., 2024) and MATES (Yu et al., 2024b) select examples based on their estimated usefulness for downstream performance (Li et al., 2024b). These methods improve corpus quality and training efficiency, but they do not address the problem of finding good data for long-context training.

Long-context data filtering. Prior works explore several routes for selecting long-context training data. Frequently used and simple baselines are *length-based* selection or long-text upsampling. These ensure that examples fill the target window but do not distinguish documents with meaningful long-range dependencies from those that are merely long. A second family uses *prediction-based signals*. LongFilter (Deng et al., 2025) measures information gain by contrasting long- and short-context predictions. A third family uses *dependency or attention-based signals*. ProLong (Chen et al., 2024a) scores long-range dependencies between document segments, while LADM (Chen et al., 2025) and LongAttn (Wu et al., 2025) use attention-derived signals to estimate dependency strength. These methods are closest to our setting. *AttentionSpan* differs by directly measuring sink-corrected attention distance at the document level; we provide a detailed comparison in Appendix A.

Long-context training Additionally, a complementary line of work to selecting the data is training a long-context method. Gao et al. (2025) show that you need a mixture of long and short context data to retain capabilities of LLMs across all length scales and suggest a dataset called ProLong curated from multiple sources with length constraint in mind. Additionally, they show that certain tasks are only achieved after SFT.

Long-context evaluation. Long-context models are often evaluated with synthetic recall tasks such as needle-in-a-haystack (Kamradt, 2023) or RULER (Hsieh et al., 2024), which test whether a model can retrieve isolated facts from a long input. Additionally, HELMET (Yen et al., 2025) provides a broader evaluation suite covering recall, RAG, long-form QA, and related long-context behaviors. Here, we evaluate on both HELMET and RULER to test whether data selected by *AttentionSpan* improves downstream long-context performance, particularly synthetic retrieval alone.

3 Measuring AttentionSpan

3.1 Prefill tokens for positional attention-sink control

A well-documented artifact of causal LLMs is *attention sinks*: some heads assign a large share of attention mass to early sequence positions, often the first token or a small initial prefix, even when those positions are not semantically important (Xiao et al., 2024; Gu et al., 2025; Barbero et al., 2025). Since these positions are usually near the beginning of the sequence, sink mass can systematically inflate AttentionSpan by making attention appear more long-range than it is. Moreover, sink behavior is not always confined to a single fixed position; it can vary with model, objective, and input format (Gu et al., 2025). Thus, simply excluding the first key is insufficient for a robust document-level attention-distance metric.

We do not remove sink tokens, since doing so would alter the model's own computation, leading to degraded performance (Xiao et al., 2024). Our goal is therefore to absorb sink mass before scoring, not to suppress the sink mechanism. Because the sink mechanism is largely positional and an artifact of training, heads continue to allocate sink mass to the leading positions even after prefill-tokens are prepended (Son et al., 2024). These leading positions are then excluded from the set of valid keys, so sink mass is discarded from the score while content-driven attention over the document remains available to the model. Additional details on design choices surrounding attention sinks and prefill-tokens can be found in Appendix C.

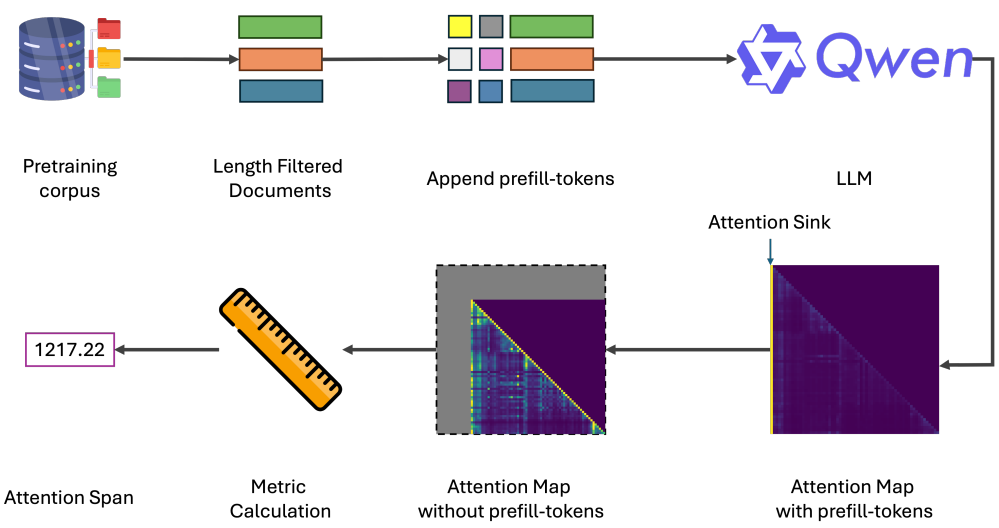


Figure 1: Overview of the AttentionSpan scoring pipeline. A document is tokenized, padded with prefill tokens at the beginning of the sequence, passed through the causal language model, and converted into attention maps. The prefill tokens are then removed from the attention map, addressing most attention sinks. AttentionSpan is then measured as the average over the median top-k key distances for all the valid query tokens.

3.2 Scoring pipeline

Given a tokenized document $x = (x_1, \ldots, x_N)$ from a pretraining corpus, we prepend an array of prefill-tokens of length p, producing an input sequence

$$z = (z_1, \ldots, z_L), \qquad L = p + N.$$

The prefill-tokens occupy positions $1, \ldots, p$, and the document tokens occupy positions $p+1, \ldots, L$. The prefill-tokens absorb positional attention-sink mass at the beginning of the sequence and are excluded from scoring.

We pass z through the LLM and extract the post-softmax attention maps from all layers and heads. $A_{ij}^{(\ell,h)} \in [0,1]$ denotes the attention that head h in layer ℓ assigns to key position j from query position i. By causality, $A_{ij}^{(\ell,h)} = 0$ for $j > i$, and $\sum_{j \le i} A_{ij}^{(\ell,h)} = 1$. We aggregate attention across all M layers and H heads:

$$\bar{A}_{ij} = \frac{1}{MH} \sum_{\ell=1}^{M} \sum_{h=1}^{H} A_{ij}^{(\ell,h)}. \tag{1}$$

The averaged attention map $\bar{A}$ is the input to the metric computation in Figure 2.

3.3 AttentionSpan metric computation

We compute AttentionSpan from the averaged attention map $\bar{A}$ by masking invalid positions, selecting each query's top-k previous valid keys, computing a weighted-median attention distance, and averaging across valid queries.

A position j is a *valid key* if it is not a padding token, a special token including BOS/EOS tokens, a prefill-token, or one of the first k document tokens. We denote the set of valid keys by $\mathcal{V}_K(x)$. Additionally, we define the set of valid queries by

$$\mathcal{V}_Q(x) = \{i \in \mathcal{V}_K(x) : |\{j \in \mathcal{V}_K(x) : j < i\}| \ge k\}. \tag{2}$$

That is, a position i is a *valid query* if $i \in \mathcal{V}_K(x)$ and there are at least k valid keys strictly before it.

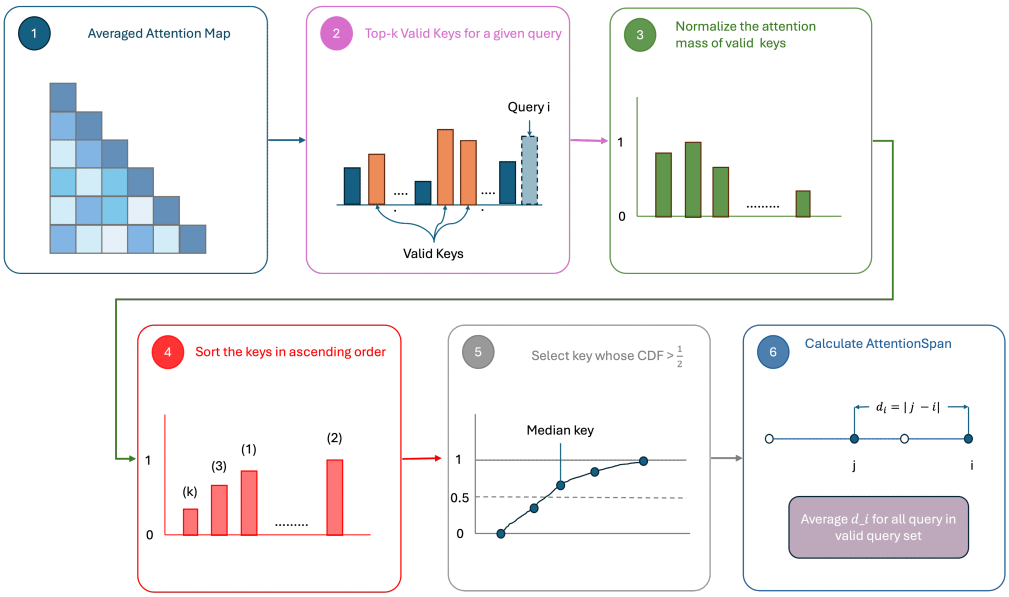


Figure 2: Computing AttentionSpan. Each query is restricted to earlier valid keys; among them, we retain the k keys with the highest attention mass and compute the attention-weighted median of their lookback distances. These values are averaged across the document, with optional distance normalization.

For each valid query $i \in \mathcal{V}_Q(x)$, we restrict $\bar{A}_{i,\cdot}$ to strictly previous valid keys and renormalize the remaining attention mass:

$$\tilde{A}_{ij} = \frac{\bar{A}_{ij}\mathbf{1}[j < i,\ j \in \mathcal{V}_K(x)]}{\sum_{j' \in \mathcal{V}_K(x)} \bar{A}_{ij'}\mathbf{1}[j' < i]}. \tag{3}$$

For positions that are not strictly previous valid keys, $\tilde{A}_{ij} = 0$.

For a query $i \in \mathcal{V}_Q(x)$, denote by $\mathcal{T}_i$ the k eligible keys assigned the largest values by $\tilde{A}_{i,\cdot}$. For each retained key $j \in \mathcal{T}_i$, we define its attention mass and lookback distance as

$$w_{ij} = \tilde{A}_{ij}, \qquad d_{ij} = i - j. \tag{4}$$

We renormalize the attention mass within the top-k set:

$$\hat{w}_{ij} = \frac{w_{ij}}{\sum_{j' \in \mathcal{T}_i} w_{ij'}}. \tag{5}$$

The per-token AttentionSpan is then the attention-weighted median of the top-k lookback distances:

$$s_i = \min\left\{ d : \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} \le d}} \hat{w}_{ij} \ge \frac{1}{2} \right\}. \tag{6}$$

We use the weighted median for robustness to residual sink-like attention; Appendix D formally compares this choice with the weighted mean and maximum. We set $k = 5$ for all of our experiments.

The document-level AttentionSpan averages the per-token scores, optionally normalized by the available distance:

$$\mathrm{AttentionSpan}(x) = \frac{1}{|\mathcal{V}_Q(x)|} \sum_{i \in \mathcal{V}_Q(x)} \frac{s_i}{(N-k)^\gamma}, \qquad \gamma \in \{0, 1\}, \tag{7}$$

where $\gamma = 0$ gives the default unnormalized score in token-distance units, and $\gamma = 1$ gives the normalized score in $[0, 1]$.

Larger AttentionSpan values correspond to attention concentrated farther back in the context, while smaller values correspond to predominantly local attention. Full algorithmic details are given in Appendix B.

3.4 Scoring model choice

For most experiments, we use Qwen2.5-Coder-7B-Instruct (Hui et al., 2024) (32K context length) to grade the data. For the 64K experiments, we extend its context using YaRN. For experiments using Borda consensus, we additionally use Qwen2.5-1.5B-Instruct (Yang et al., 2024) (32K context length) and Llama-3.2-3B-Instruct (Grattafiori et al., 2024) (128K context length).

4 Experiments and Results

We organize our experiments in four stages. First, we evaluate whether AttentionSpan consistently identifies useful long-context training data across model families, context lengths, and training scales (Section 4.1). Second, we study data mixing across AttentionSpan tiers and domain effects (Section 4.2). Third, we examine the choice of grading model and methods for mitigating grader-specific inductive biases (Section 4.3). Finally, we compare AttentionSpan against prior attention based long-context data selection methods (Section 4.4).

4.1 Evaluating AttentionSpan

To test whether AttentionSpan provides a consistent data-selection signa, we extend Llama-2-7B (Touvron et al., 2023) from its native 4K context to 16K and 32K using LoRA (Hu et al., 2022), training each condition on 800 selected documents in a single-pass training. We then scale the experiment to OLMo-3-7B, extending its 8K context to 32K with full-parameter training on 10,000 and 30,000 selected documents, corresponding to approximately 1.7B and 5B training tokens. Across all settings, the candidate pool and training configuration are held fixed within each experiment, while the selected data varies according to its AttentionSpan score. Pool construction and length statistics are given in Appendix I, the shared LoRA configuration in Appendix G, and the compute cost of every phase in Appendix H. Full training details are provided in Appendix N and Appendix O.

Table 1 shows a consistent ordering across these substantially different settings: high-AttentionSpan data outperforms random selection, while low-AttentionSpan data performs worse than random selection. The same pattern therefore holds across model families, context-extension targets, training methods, and training budgets, showing the robustness of the metric. Moreover, Figure 3 shows that the ordering persists to context lengths longer than trained on, 64K for Llama-2-7B trained at 32K, allowing for computational cost savings beyond a given lab's token budget.

Table 1: Evaluation of AttentionSpan across model families, context lengths, and training scales. (a) shows Llama-2-7B trained at 16K and 32K context lengths and evaluated at 8K/16K and 16K/32K, respectively. (b) shows OLMo-3-7B under different training budgets. Subscripts denote 95% CI.

(a) Llama-2-7B: LoRA

	16K training			32K training			
Condition	**Rec 8K**	**Rec 16K**	**Overall**	**Rec 16K**	**Rec 32K**	**RAG**	**Overall**
Long	$\mathbf{57.8}_{[\mathbf{53.9,61.6}]}$	$\mathbf{33.6}_{[\mathbf{30.0,37.2}]}$	$\mathbf{41.29}_{[\mathbf{39.89,42.66}]}$	$\mathbf{57.1}_{[52.7,61.3]}$	$\mathbf{37.1}_{[33.6,40.6]}$	$47.4_{[46.0,48.9]}$	$\mathbf{47.3}_{[45.7,48.9]}$
Random	$44.2_{[40.2,48.1]}$	$22.8_{[19.4,26.2]}$	$37.04_{[35.69,38.37]}$	$44.2_{[39.8,48.4]}$	$25.6_{[22.5,28.7]}$	$47.2_{[45.7,48.8]}$	$41.1_{[39.6,42.6]}$
Short	$22.9_{[19.8,26.1]}$	$12.2_{[10.2,14.5]}$	$30.84_{[29.65,32.05]}$	$36.5_{[32.6,40.6]}$	$22.5_{[19.8,25.2]}$	$46.5_{[45.0,48.0]}$	$38.0_{[36.6,39.4]}$
Baseline	$3.1_{[1.7,4.8]}$	$0.4_{[0.1,0.8]}$	$19.52_{[18.58,20.46]}$	$0.0_{[0.0,0.0]}$	$0.0_{[0.0,0.0]}$	$1.0_{[0.7,1.2]}$	$0.5_{[0.3,0.6]}$

(b) OLMo-3-7B: full-parameter training, 32K context

	10K documents			30K documents		
Condition	**Recall**	**RULER**	**RAG**	**Recall**	**RULER**	**RAG**
Long	$\mathbf{73.7}_{[71.6,75.8]}$	$\mathbf{64.0}_{[63.0,65.0]}$	$\mathbf{50.4}_{[48.6,52.0]}$	$\mathbf{75.7}_{[73.4,77.8]}$	$\mathbf{63.9}_{[62.9,64.9]}$	$\mathbf{49.8}_{[48.2,51.3]}$
Random	$63.1_{[60.8,65.5]}$	$58.0_{[56.9,59.1]}$	$48.0_{[45.8,50.4]}$	$62.0_{[59.1,64.7]}$	$60.3_{[59.2,61.4]}$	$49.4_{[46.8,51.8]}$
Short	$46.7_{[44.2,49.2]}$	$52.3_{[51.3,53.3]}$	$47.1_{[44.1,49.5]}$	$49.0_{[46.5,51.5]}$	$54.6_{[53.6,55.7]}$	$47.2_{[44.3,49.8]}$
Baseline	$19.9_{[18.0,21.8]}$	$41.9_{[40.8,43.1]}$	$47.8_{[45.8,49.8]}$	$19.9_{[18.0,21.8]}$	$41.9_{[40.8,43.1]}$	$47.8_{[45.8,49.8]}$

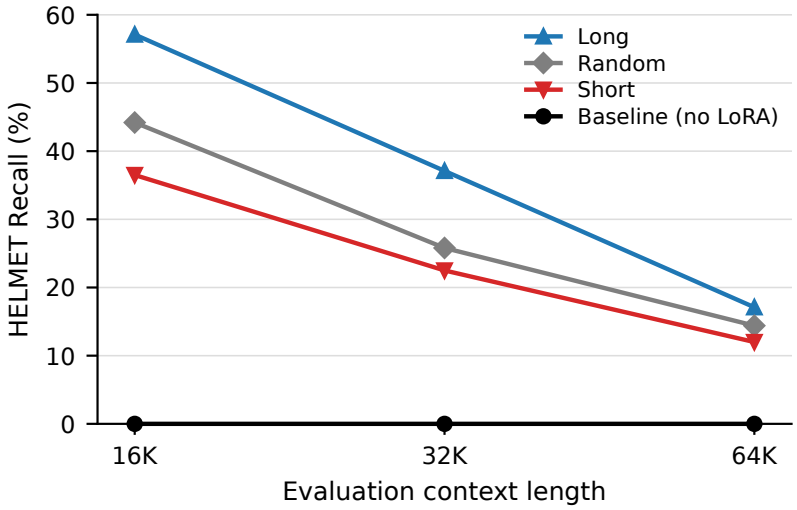


Figure 3: HELMET Recall at 16K, 32K, and 64K evaluation for Llama-2-7B trained at 32K. The AttentionSpan ordering long > random > short is preserved at every evaluation context, including 64K. The untuned baseline collapses to 0% recall, showing that context extension alone is not sufficient for long-context retrieval.

Additional model families. The same tier ordering also appears on Llama-3-8B and SmolLM2-1.7B, providing additional evidence that the signal transfers across model families and scales (Appendix L.2 and Appendix L). At 128K, Llama-3.1-8B-Instruct is already near ceiling on the retrieval tasks before adaptation; in this setting, high-AttentionSpan data largely preserves its existing capability, whereas low-AttentionSpan data degrades it (Appendix L.3).

4.2 Data Mixing and Domain Effects

Global AttentionSpan filtering can change both the long-range attention structure and the domain composition of the selected data (Appendix F). This is particularly visible in the 16K Llama-2 experiments, where the highest-AttentionSpan tier is strongly code-heavy, while the Middle tier contains a more balanced mixture of code and web data. As shown in Table 2, Middle consequently performs close to high-AttentionSpan despite having lower AttentionSpan scores. We therefore test whether mixing AttentionSpan tiers or explicitly balancing domains can improve selection.

At 16K, accounting for domain composition improves over selecting the extreme high-AttentionSpan tier alone: Balanced Top achieves the strongest overall result, while mixtures of high-AttentionSpan and Middle also improve over pure high-AttentionSpan. At 32K, however, mixing high- and low-AttentionSpan tiers provides no additional benefit; the 50/50 mixture falls between its components, while Long remains the strongest condition. Thus, tier mixing itself does not provide a consistent improvement across context lengths, whereas avoiding the low-AttentionSpan tail remains consistent with the results in Section 4.1.

Importantly, the OLMo-3 experiments in Table 1 use domain-balanced selections. The high-AttentionSpan > random > low-AttentionSpan ordering therefore persists when domain composition is controlled, indicating that the AttentionSpan signal is not hampered by the domain balancing nor explainable by trivially selecting only certain domains like code. A matched-domain ablation on Llama-2 at 32K, holding code, arXiv and book proportions fixed and varying only within-domain rank, is reported in Appendix E. In a production scenario, this would lead to benefits gained from introducing diversity to the training data.

4.3 Mitigating Grader-Specific Inductive Biases

A limitation of using a single model for grading is that each model has its own inductive biases, introduced by its training data, architecture, and training procedure. This is particularly important for

Table 2: Data-mixing and domain-balancing ablations for Llama-2-7B at 16K and 32K. Each panel preserves the evaluation metrics reported for that experimental setting. Subscripts are 95% bootstrap intervals over evaluation examples (B=10,000); they capture evaluation variance only, not variation across training runs. Overall in panel (a) is the mean of 22 subtask means; in panel (b) it is over 16 subtasks, the 32K suite having no LongQA category, so the two Overall columns are not comparable. Per-subtask values are in Appendices J and K.

(a) Llama-2-7B, Trained at 16K

Condition	Rec 8K	Rec 16K	HELMET Overall
Balanced Top	$62.8_{[58.6,66.6]}$	$\mathbf{42.4}_{[38.5,46.6]}$	$\mathbf{44.69}_{[43.28,46.13]}$
Diverse Bal. Top	$61.5_{[57.6,65.3]}$	$42.0_{[38.1,46.1]}$	$44.44_{[43.07,45.85]}$
Mix 75/25 (mid/pos)	$61.4_{[57.3,65.3]}$	$37.2_{[33.4,41.2]}$	$43.76_{[42.36,45.16]}$
Diverse Top 50%	$62.4_{[58.1,66.4]}$	$34.9_{[31.2,38.8]}$	$43.57_{[42.18,44.98]}$
Mix 50/50 (mid/pos)	$60.2_{[55.9,64.4]}$	$35.6_{[31.6,39.6]}$	$43.40_{[42.01,44.84]}$
Middle	$61.6_{[57.4,65.6]}$	$36.5_{[32.6,40.4]}$	$43.39_{[41.98,44.78]}$

(b) Llama-2-7B, Trained at 32K

Condition	Recall@16K	Recall@32K	RAG Avg	Overall
Middle	$51.2_{[46.9,55.4]}$	$29.9_{[26.4,33.4]}$	$\mathbf{47.8}_{[46.3,49.3]}$	$44.2_{[42.6,45.7]}$
Diverse Top	$50.1_{[45.8,54.4]}$	$30.0_{[26.9,33.2]}$	$46.6_{[45.2,48.1]}$	$43.3_{[41.8,44.9]}$
Mix 50/50	$40.1_{[36.1,44.2]}$	$26.1_{[23.0,29.3]}$	$47.0_{[45.5,48.4]}$	$40.0_{[38.5,41.5]}$
Diverse Bottom	$38.5_{[34.4,42.7]}$	$22.4_{[19.7,25.2]}$	$47.0_{[45.5,48.5]}$	$38.7_{[37.3,40.2]}$

data selection, where these biases can directly affect which documents are selected. For example, a model specialized for code may naturally favor code-like documents. To reduce the influence of any individual grader, we aggregate rankings across models using Borda consensus.

As shown in Table 3, consensus substantially improves data selection over using any individual grader. Borda achieves 78.4 on Recall and 66.7 on RULER, compared with the best individual-grader scores of 71.1 and 63.8, respectively. Both consensus intervals lie entirely above those of every individual grader. We provide additional analysis of cross-grader agreement, an alternative set of Borda voters, and the resulting data mixtures in Appendix Q.

Table 3: Downstream performance of data selected by individual graders and by Borda consensus. All models are trained on 4,000 documents and 655.35M tokens. Subscripts denote 95% confidence intervals.

Grader Model	Recall	RULER
Llama-3.2-3B	$71.1_{[68.9,73.2]}$	$61.8_{[60.8,62.8]}$
Qwen-2.5-1.5B	$68.0_{[65.8,70.1]}$	$63.4_{[62.4,64.4]}$
Qwen-2.5-Coder-7B-Instruct	$65.1_{[62.8,67.5]}$	$63.8_{[62.8,64.8]}$
Borda (consensus)	$\mathbf{78.4}_{[\mathbf{76.5,80.3}]}$	$\mathbf{66.7}_{[\mathbf{65.8,67.7}]}$

Self-improvement through data selection. Another natural extension of AttentionSpan is self-improvement, where a model scores and selects its own training data. We evaluate this setting using OLMo-3-7B base model and find that self-selected data improves over random selection by 17.8 percentage points on HELMET-Recall and 6.2 percentage points on RULER. Unlike consensus-based selection, self-selection does not mitigate the inductive biases of an individual grader. Instead, it provides a natural mechanism for carrying the model's existing preferences into the data-selection stage of its own training. We explore this setting further in Appendix R.

4.4 Comparison with Prior Long-Context Data Selection Methods

We compare AttentionSpan against prior attention-based methods for selecting long-context training data. For this experiment, we perform this comparison using full-parameter training of OLMo-3-7B. For all methods, we select 4,000 documents while holding the candidate pool, training recipe, token

Table 4: Selection rules compared on OLMo-3-7B at 4,000 documents, with the candidate pool, training recipe and evaluation held fixed. Brackets are 95% CIs over three evaluation seeds.

Selection	HELMET-Recall	RULER	HELMET-RAG	Long-doc QA
AttentionSpan (ours)	$\mathbf{78.4}_{[\mathbf{76.5},\mathbf{80.3}]}$	$\mathbf{66.7}_{[\mathbf{65.8},\mathbf{67.7}]}$	$\mathbf{51.1}_{[\mathbf{48.7},\mathbf{53.2}]}$	$23.5_{[21.0,26.1]}$
LADM	$56.1_{[53.9,58.4]}$	$61.0_{[60.0,62.0]}$	$49.9_{[47.3,52.4]}$	$22.5_{[19.6,25.5]}$
Random	$55.0_{[52.8,57.2]}$	$56.8_{[55.7,57.8]}$	$49.9_{[46.7,52.9]}$	$18.4_{[16.0,20.8]}$
LongAttn	$39.8_{[37.9,41.7]}$	$51.2_{[50.2,52.3]}$	$47.6_{[45.0,50.1]}$	$19.9_{[17.2,22.7]}$
Base + YaRN (no training)	$19.9_{[18.0,21.8]}$	$41.9_{[40.8,43.1]}$	$47.8_{[45.8,49.8]}$	$\mathbf{31.0}_{[\mathbf{26.5},\mathbf{35.4}]}$

budget, and evaluation fixed. Thus, the only difference between conditions is the rule used to rank and select training documents.

AttentionSpan substantially outperforms the other attention-based selection methods, including LongAttn and LADM, showing that the effectiveness of attention-based selection depends critically on how we interpret the signals the attention matrices carry and how we utilize them. The measured layers need to be spread across the model, as the initial layers are biased to local interactions, hiding the long-range dependencies in the model which we into more detail in Appendix E. The sink correction achieves stronger performance by eliminating the confounding signals. We provide additional comparisons and ablations using Llama-2-7B in Appendix P.3.

5 Discussion & Conclusion

AttentionSpan makes long-range dependency structure directly inspectable at the document level. Its token statistic takes the median lookback among the k strongest attended keys, while the prefill prefix removes positional sinks from consideration. Together, these choices prevent a few anomalous attention edges from masquerading as sustained use of distant context.

Attention span metrics expose important information about the nature of publicly available pretraining data. Code, because of its dependencies such as imports, global variables, parentheses (in languages such as C/C++), etc., creates a natural long-context behavior that helps in recall. Importantly, our balanced-domain experiments confirm that *AttentionSpan* is not merely a proxy for code; even when controlling for domain composition, high-scoring data provides a distinct performance lift that directly informs downstream capabilities. However, too much code during training might cause a skewed distribution of what long-range data should look like. Thus, at 16k, it performs better when we have a more balanced domain mix.

The failure of overly code-heavy datasets illustrates an important caveat of AttentionSpan. While *AttentionSpan* provides a robust signal for long-context dependency, high *AttentionSpan* scores alone do not guarantee general document quality or diversity; consequently, training on long-range data that lacks other quality or diversity markers may cause the model to degenerate. *AttentionSpan* should not be used as a standalone data-selection rule; rather, it is most effective when integrated into a broader curation pipeline that includes domain balancing, traditional quality filtering, and deduplication.

Finally, we observe that performance on RAG and QA tasks remains relatively unchanged across all data compositions. While introducing our data provides a clear uplift over the baseline, the lack of variance in performance between compositions likely reflects the nature of the tasks themselves. Advanced RAG and QA often require a niche, specific data format (Yu et al., 2024a; Zhang et al., 2024; Nayak et al., 2024) . This stability is consistent with prior literature, which similarly shows that such specialized tasks are less sensitive to general long-context data filtering than pure retrieval and recall benchmarks.

LLM Usage Disclosure

We used AI tools to polish the writing of this work and to assist with writing and editing code. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including its text and claims edited with the aid of AI tools.

REFERENCES

Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. Why do LLMs attend to the first token? In *Conference on Language Modeling (COLM)*, 2025.

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. Association for Computational Linguistics, 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. Association for Computational Linguistics, 2024a.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. LongLoRA: Efficient fine-tuning of long-context large language models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024b.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 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.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. LongRoPE: Extending LLM context window beyond 2 million tokens. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11091–11104. PMLR, 2024.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979. doi: 10.1214/aos/1176344552.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128K context. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. PMLR, 2024.

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)*. Association for Computational Linguistics, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

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.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Junlong Jia, Ziyang Chen, Xing Wu, Chaochen Gao, Zijia Lin, Songlin Hu, and Binghui Guo. Entropylong: Effective long-context training via predictive uncertainty. In *International Conference on Learning Representations*, volume 2026, pp. 4046–4068, 2026.

Omar Kamali and Omneity Labs. Wikipedia monthly, 2025. URL https://huggingface.co/datasets/omarkamali/wikipedia-monthly.

Greg Kamradt. Needle in a haystack — pressure testing LLMs. https://github.com/gkamradt/LLMTest_Needle_In_A_Haystack, 2023. GitHub repository.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. The stack: 3 tb of permissively licensed source code. *Preprint*, 2022.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. DataComp-LM: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024a.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024b.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638.

Kai Lv, Xiaoran Liu, Qipeng Guo, Hang Yan, Conghui He, Xipeng Qiu, and Dahua Lin. Long-Wanjuan: Towards systematic measurement for long text quality. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5709–5725. Association for Computational Linguistics, 2024.

Nihal Nayak, Yiyang Nan, Avi Trost, and Stephen Bach. Learning to generate instruction tuning datasets for zero-shot task adaptation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12585–12611, 2024.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Seungwoo Son, Wonpyo Park, Woohyun Han, Kyuyeun Kim, and Jaeho Lee. Prefixing attention sinks can mitigate activation outliers for large language model quantization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

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

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. QuRating: Selecting high-quality data for training language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 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. Association for Computational Linguistics, 2025.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

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

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

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. Effective long-context scaling of foundation models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4643–4663. Association for Computational Linguistics, 2024.

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.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Yunhua Zhou, Jun Zhan, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 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. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in Neural Information Processing Systems*, 37:121156–121184, 2024a.

Zichun Yu, Spandan Das, and Chenyan Xiong. MATES: Model-aware data selection for efficient pretraining with data influence models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*, 2024.

Yonghao Zhuang, Lanxiang Hu, Longfei Yun, Souvik Kundu, Zhengzhong Liu, Eric P. Xing, and Hao Zhang. Scaling long context training data by long-distance referrals. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

A Additional Discussion of Long-Context Data Filtering

We group prior long-context data filtering methods by their primary selection signal and expand on the distinctions most relevant to AttentionSpan.

Length-based selection. The simplest heuristic for long-context data curation is length-based thresholding, i.e retraining on documents with length greater than the base model up to a target length, or upsampling sources that naturally feature long documents, such as books and code repositories (Xiong et al., 2024; Grattafiori et al., 2024). Length is an operational prerequisite for long-context training: exposing the model to documents that saturate the target window calibrates positional encodings (such as RoPE base frequencies) far more reliably than relying on interpolation techniques alone. However, raw sequence length merely governs the amount of available context rather than its informational utility, failing to measure whether a document actually demands long-range reasoning. Two sequences of identical token length can diverge drastically in structure: one may contain cross-passage dependencies where distant tokens resolve local uncertainty, whereas another may consist of repetitive logs, boilerplates, or weakly linked sections that are almost entirely predictable from immediate local context. As recent critiques demonstrate, naive length-based filtering predominantly trains the model on short-range n-gram correlations rather than cross-document referrals (Zhuang et al., 2025), failing to provide tokens that genuinely reduce predictive entropy over distant horizons (Jia et al., 2026). Consequently, sequence length is a necessary precondition to cover the context window, but fundamentally insufficient to ensure effective long-context dependency modeling.

Perplexity and prediction-contrast methods. Prediction-based approaches ask whether access to additional context changes the model's ability to predict particular tokens.

LongFilter (Deng et al., 2025) applies this prediction-contrast principle directly to data selection. It measures the information gain provided by extended context by comparing model predictions with long and short contexts, and uses this signal to identify training samples for which long-range information is predictive. Unlike raw perplexity, this explicitly tests whether additional context is useful. However, the resulting signal is still defined through changes in token prediction, whereas AttentionSpan measures the distance structure of the model's attention directly.

Dependency-score methods. ProLong (Chen et al., 2024a) is explicitly designed to identify documents with long-range dependencies. It first divides a document into segments and uses delta perplexity to estimate the *Dependency Strength* between segment pairs. This signal is then combined with *Dependency Distance*, which rewards dependencies spanning larger distances, and *Dependency Specificity*, which discounts trivial dependencies arising from repetitive content. The resulting Long Dependency Score is used to rank and filter training documents. ProLong therefore shares the central motivation of AttentionSpan: document length alone is insufficient, and useful long-context training data should contain dependencies spanning substantial distances. The principal distinction is the measurement signal. ProLong infers dependencies between segments through changes in prediction likelihood, whereas AttentionSpan measures how far the grader model's attention reaches while processing the document.

Attention-based methods. LADM (Chen et al., 2025) and LongAttn (Wu et al., 2025) are most directly related to AttentionSpan because they use attention-derived signals for long-context data selection. LADM operates at the span level and uses a Long Attention Calculator to estimate pairwise dependency scores between spans. LongAttn instead derives token-level statistics from model attention to quantify long-range dependency strength and uses these statistics for document selection. These approaches avoid relying exclusively on document length or prediction loss and instead exploit the model's internal attention patterns.

A central issue for distance-based attention measurements, however, is *attention sinks*. Decoder-only models can assign substantial content-independent attention mass to early sequence positions. Because these positions are also maximally distant from later queries, this positional artifact can be mistaken for genuine long-range dependency and inflate an attention-distance score. AttentionSpan explicitly separates this effect from document content by prepending prefill-tokens to absorb early-position sink mass and excluding these positions from scoring.

AttentionSpan also differs in how the remaining attention is reduced to a document-level score. After restricting attention to valid document positions, it retains the top-k keys for each query and summarizes their lookback distances using an attention-weighted median. This makes the score less sensitive to residual distant attention edges that survive sink correction.

B ATTENTIONSPAN: DETAILED ALGORITHM AND COMPLEXITY

This appendix gives a self-contained specification of the document-level score $\mathrm{AttentionSpan}(x)$ defined in (7), together with its runtime and memory analysis. Algorithm 1 presents the metric in its conceptually simplest form, assuming that the full attention tensor can be materialized. §B.3 then describes the *Tiled Attention Reconstruction* (TAR) implementation used in practice, which preserves the same input-output behavior while reducing peak memory.

B.1 NOTATION

Table 5: Notation used in the AttentionSpan algorithm.

Symbol	Meaning
$x = (x_1, \ldots, x_N)$	original document tokens
$z = (z_1, \ldots, z_L)$	model input after prepending prefill-tokens
N	original document length
L	full sequence length after prefill-tokens insertion, $L = N + f$
p	number of prefill-tokens
B	per-call batch size
M	number of transformer layers
H	number of attention heads after GQA expansion
D	per-head dimension
k	top-k size, fixed to $k = 5$ in our experiments
p	number of initial document tokens ignored after prefill-tokens
$\mathcal{V}_K(x)$	set of valid key positions
$\mathcal{V}_Q(x)$	set of valid query positions
$A^{(\ell,h)}$	post-softmax attention map for layer ℓ, head h
$\bar{A}$	attention map averaged over all layers and heads
$\tilde{A}$	masked and renormalized attention over valid past keys
s_i	per-token AttentionSpan for valid query i

B.2 ALGORITHM

Algorithm 1 $\mathrm{AttentionSpan}(x)$ for one document.

Require: Document tokens $x = (x_1, \ldots, x_N)$, causal language model, p prefill-tokens, top-k size k, normalization parameter $\gamma \in \{0, 1\}$.
Ensure: Document-level $\mathrm{AttentionSpan}(x)$.

Prepend p prefill-tokens to form $z = (z_1, \ldots, z_L)$, where $L = p + N$.
Build valid-key and valid-query sets $\mathcal{V}_K(x)$ and $\mathcal{V}_Q(x)$.
Initialize $\bar{A} \leftarrow \mathbf{0} \in \mathbb{R}^{L \times L}$.
for $\ell = 1$ **to** M **do**
  for $h = 1$ **to** H **do**
    Extract post-softmax causal attention $A^{(\ell,h)} \in \mathbb{R}^{L \times L}$.
    $\bar{A} \leftarrow \bar{A} + \frac{1}{MH} A^{(\ell,h)}$.
  end for
end for
for each $i \in \mathcal{V}_Q(x)$ **do**
  Restrict to previous valid keys and renormalize:

$$\tilde{A}_{ij} = \frac{\bar{A}_{ij} \mathbf{1}[j < i,\ j \in \mathcal{V}_K(x)]}{\sum_{j' \in \mathcal{V}_K(x)} \bar{A}_{ij'} \mathbf{1}[j' < i]}.$$

  Let $\mathcal{T}_i$ be the top-k keys under $\tilde{A}_{i,\cdot}$ and set $d_{ij} = i - j$ for $j \in \mathcal{T}_i$.
  Renormalize attention within $\mathcal{T}_i$:

$$\hat{w}_{ij} = \frac{\tilde{A}_{ij}}{\sum_{j' \in \mathcal{T}_i} \tilde{A}_{ij'}}.$$

  Compute the weighted-median lookback:

$$s_i = \min \left\{ d : \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} \le d}} \hat{w}_{ij} \ge \frac{1}{2} \right\}.$$

end for
return

$$\mathrm{AttentionSpan}(x) = \frac{1}{|\mathcal{V}_Q(x)|} \sum_{i \in \mathcal{V}_Q(x)} \frac{s_i}{(N-k)^\gamma}.$$

$\gamma = 0$ gives the unnormalized score; $\gamma = 1$ gives the normalized score.

B.3 TILED ATTENTION RECONSTRUCTION

Algorithm 1 assumes that the full $H \times L \times L$ attention tensor can be materialized for each layer. At long sequence lengths this is infeasible. For example, when $L = 16{,}384$ and $H = 32$, one layer of per-head attention requires approximately $32 \cdot L^2$ entries, which is roughly 17GB in fp16 or 34GB in fp32. This excludes hidden states, KV tensors, activations, and model weights.

We therefore use *Tiled Attention Reconstruction* (TAR). TAR reconstructs the averaged attention map $\bar{A}$ using query and key tiles together with the same online-softmax identity used in FlashAttention-style (Dao et al., 2022) attention computation. For each layer, TAR processes query blocks of width q_b and key blocks of width k_b. The row-wise softmax statistics are accumulated tile by tile, and the resulting probabilities are folded directly into the averaged attention map $\bar{A}$. Heads and layers are accumulated into the same $L \times L$ accumulator, so the per-head $H \times L \times L$ attention tensor is never stored.

TAR preserves the same $\bar{A}$ as the conceptual full-attention algorithm up to floating-point roundoff. It does not change the asymptotic compute cost, because the same $QK^\top$ products are still formed. Its benefit is memory: the live per-layer attention state scales with the tile size rather than with HL^2.

The only remaining dense quadratic object is the averaged attention map $\bar{A}$, which is used for masked renormalization and top-k scoring.

B.4 TIME COMPLEXITY

Table 6 summarizes the runtime of each component of AttentionSpan for a single sequence of length L. The bounds are stated for the conceptual algorithm, with M layers, H heads, per-head dimension D, and top-k size k. TAR has the same asymptotic time complexity because it reconstructs the same attention probabilities using tiled computation rather than materializing the full per-head tensor.

Table 6: Time complexity of the AttentionSpan algorithm for one document.

Step	Time Complexity
Prepend prefill-tokens and build masks	$O(L)$
Compute averaged attention map $\bar{A}$	$O(MHL^2D)$
Query-key score computation	$O(MHL^2D)$
Causal softmax over keys	$O(MHL^2)$
Average over heads and layers	$O(MHL^2)$
Build valid-key and valid-query sets	$O(L)$
Masked renormalization over valid past keys	$O(L^2)$
Top-k key selection for all valid queries	$O(L^2)$
Per-query top-k selection	$O(L)$
Across all valid queries	$O(L^2)$
Per-token median	$O(Lk \log k)$
Sort top-k distances per query	$O(k \log k)$
Across all valid queries	$O(Lk \log k)$
Document-level averaging	$O(L)$
Total time	$\mathbf{O(MHL^2D + L^2 + Lk \log k)}$
Total time for fixed k	$\mathbf{O(MHL^2D)}$

The dominant cost is reconstructing the attention maps. For each layer and head, the query-key product costs $O(L^2D)$, and across all layers and heads this gives $O(MHL^2D)$. The masked renormalization and top-k selection are also quadratic in L, but they do not include the multiplicative MHD factor. For fixed k, the median step is linear in L up to a small constant, so it is not the bottleneck.

For a batch of B documents with comparable length L, the time bound scales linearly with B:

$$T_{\text{batch}} = O(BMHL^2D). \tag{8}$$

Across N_{doc} documents and G GPUs, the total grading wall-clock cost scales as

$$T_{\text{grade}} = \Theta\left(\frac{N_{\text{doc}}}{G} \cdot MHL^2D\right), \tag{9}$$

up to hardware utilization and batching efficiency. Thus, AttentionSpan scoring is linear in the number of documents and quadratic in context length, matching the cost profile of long-context attention itself.

B.5 SPACE COMPLEXITY

Table 7 summarizes the peak space requirements. We separate the naive full-attention implementation from TAR, which is the implementation used in practice.

Table 7: Space complexity of the AttentionSpan algorithm for one document.

Component	Space Complexity
Token sequence and masks	$O(L)$
Valid-key and valid-query sets	$O(L)$
Per-query top-k indices and weights	$O(Lk)$
Naive per-layer attention tensor	$O(HL^2)$
Averaged attention accumulator $\bar{A}$	$O(L^2)$
Naive peak space	$\mathbf{O(HL^2)}$
TAR live attention tile	$O(Hq_bL)$
Averaged attention accumulator $\bar{A}$	$O(L^2)$
TAR peak space	$\mathbf{O(L^2 + Hq_bL)}$
TAR peak space for fixed q_b	$\mathbf{O(L^2)}$

In the naive implementation, the dominant memory term is the per-layer attention tensor of shape $H \times L \times L$. TAR removes this term by computing attention in query blocks of width q_b. This reduces the live per-layer attention state from $O(HL^2)$ to $O(Hq_bL)$, a factor of approximately L/q_b. The remaining dominant term is the averaged attention accumulator $\bar{A} \in \mathbb{R}^{L\times L}$, so the practical peak memory is $O(L^2)$ for fixed tile size.

For a batch of B documents with comparable length L, the TAR space bound scales as

$$S_{\text{batch}} = O(BL^2 + BHq_bL). \tag{10}$$

The BL^2 term comes from the averaged attention accumulator $\bar{A}$, while the BHq_bL term comes from the live TAR tile state. Since $\bar{A}$ is required for masked renormalization and top-k scoring; the dominant remaining memory cost is quadratic in the sequence length.

C Attention Sinks and prefill-tokens

C.1 Effect of the number of prefill-tokens on sink control

One of the biggest issues that can affect any attention-distance-based metric for computing long-range dependencies in data is attention sinks. However, while the first token of a document is a well-documented attention sink in most LLMs, it is not the only place where attention sinks can appear. This matters because *AttentionSpan* is a distance-based metric: any content-independent attention placed on very early positions will look like long-range attention for most later query tokens, even if the model is not using meaningful information from those positions. A simple first-token exclusion is therefore not the best solution (column 2 in Figure 4).

Figure 4 visualizes this design choice across three 16K-token documents from The Stack (Kocetkov et al., 2022), FineWeb (Penedo et al., 2024), and Wikipedia (Kamali & Labs, 2025). In Figure 4, the no-prefill-token setting shows strong early attention mass, which would artificially increase the measured token-level attention span. Adding and excluding a single prefill-token token reduces only the most concentrated part of the sink, but visible residual sink structure remains in the early document positions. With p prefill-tokens, the sink mass is redirected into the leading prefill-tokens region. Since those prefill-token positions are excluded from valid keys, this mass is removed from the scoring computation without modifying the model's forward pass. After exclusion, the remaining map is less dominated by positional sink behavior and better reflects attention among document tokens. This is why we use multiple prefill-tokens rather than a single-token correction or direct sink removal.

C.2 AttentionSpan is stable to the choice of number of prefill tokens and grading model

Residual sink mass. Let $\mathcal{S}_N = c_1, \ldots, c_N$ denote the first N content tokens of a given document. In this experiment, we fix $N = 64$ for all values of the number of prefill tokens p. We define the

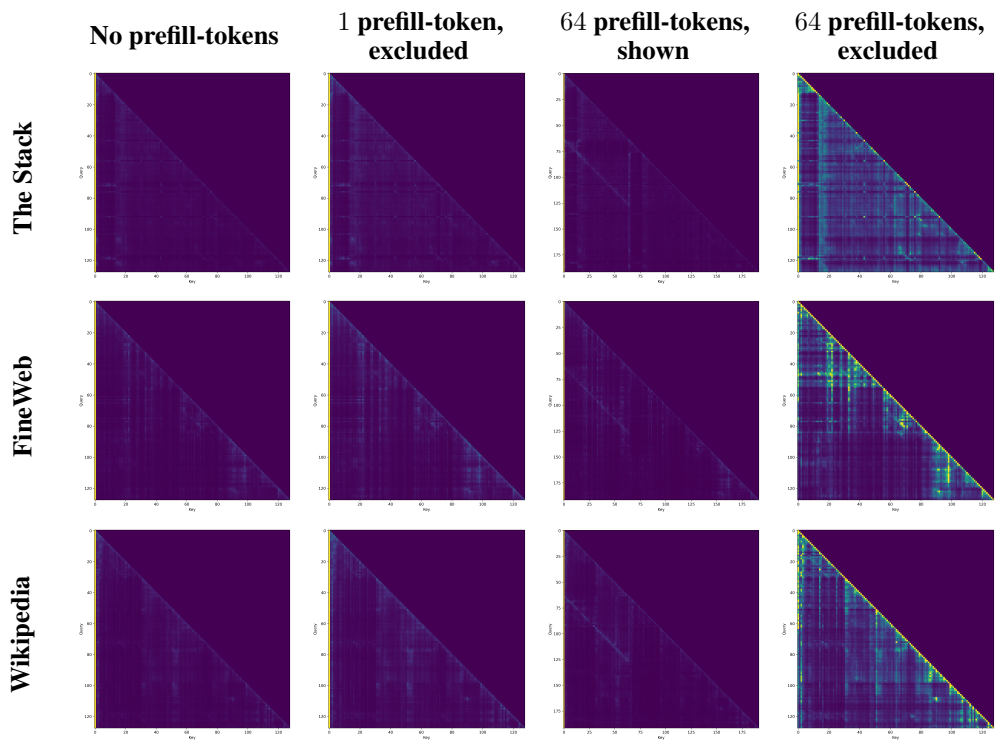


Figure 4: Prefill-tokens ablation on 16K-token documents from The Stack, FineWeb, and `omarkamali/wikipedia-monthly`. Each row corresponds to one dataset, and each column corresponds to one prefill-tokens setting. The *AttentionSpan* score is computed on the full 16K-token sequence, but only the leading attention region is visualized for readability. The no-prefill-tokens, one-prefill-token excluded, and 64 prefill-tokens excluded columns show the first 128 document-token positions, while the 64 prefill-tokens shown column shows the 64-token prefill-tokens plus the first 128 document tokens. Across all three datasets, a single excluded prefill-token only partially removes early sink behavior, while 64 prefill-tokens absorbs the broader sink region. Excluding the 64 prefill-tokens positions then yields an attention map that better reflects attention among document tokens.

residual sink mass of a document as the fraction of attention mass from a given query that still falls on the document's opening content tokens after introducing the filler prefix, averaged over content positions:

$$s^{(d)}(p);=;\frac{1}{|\mathcal{C}|}\sum i \in \mathcal{C};\sum_{j\in\mathcal{S}N,,j\le i}\pi_i(j). \tag{11}$$

Each cell in Figure 5 reports $\mathrm{median}d, s^{(d)}(p)$ over the 200 documents for the corresponding (scorer, domain, length) combination.

Since the filler tokens are included in neither $\mathcal{S}_N$ nor the normalizer of π_i, this quantity measures only the sink mass that *survives* the filler block. Attention absorbed by the filler tokens is excluded from the measurement, whereas attention that spills past the filler block onto the opening real content tokens is retained. Holding N fixed at 64 ensures that the measurement window remains comparable as p varies.

Figure 5 shows that the residual sink mass leaking onto real content tokens is negligible across the tested prefill lengths. Among the evaluated settings, $p = 64$ yields the lowest residual sink mass, suggesting that 64 prefill tokens are already sufficient to absorb nearly all of the attention that would otherwise be assigned to the opening content tokens.

Attention sinks need not be strictly contiguous, but they are typically concentrated near the beginning of the sequence. We therefore expect a sufficiently long filler prefix to absorb most of this sink

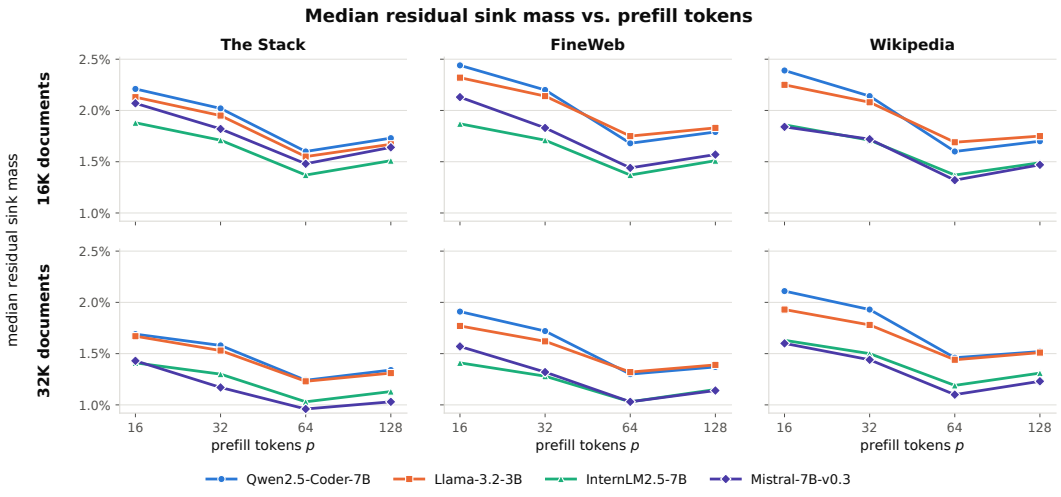


Figure 5: Median residual sink mass versus prefill length p, for four scorer models on three domains, at 16K (top) and 32K (bottom) document lengths. Residual sink mass is the median share of attention mass falling on the first 64 real tokens; 200 documents per cell.

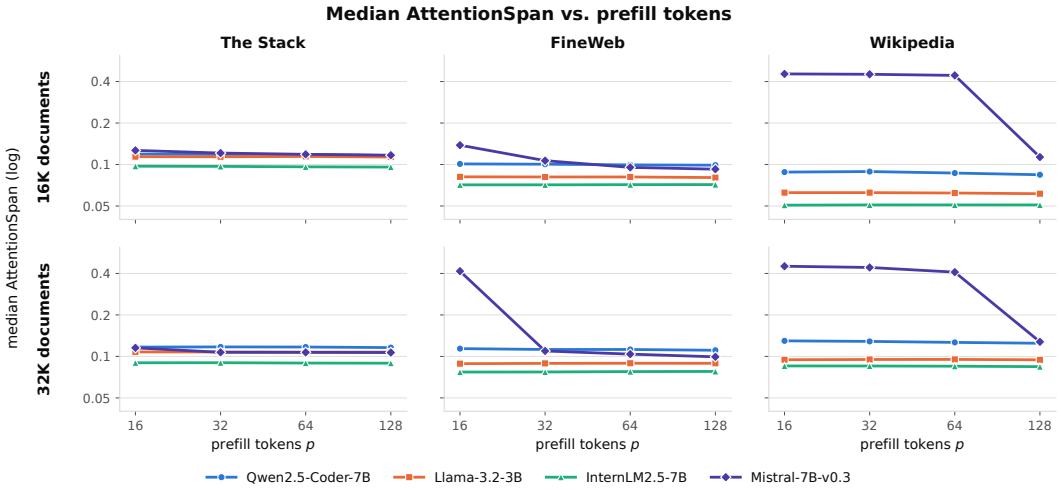


Figure 6: Median AttentionSpan (weighted normalized attention distance) versus prefill length p, for four scorer models on three domains, at 16K (top) and 32K (bottom) document lengths. Log-scaled y-axis; 200 documents per cell.

behavior. Figure 6 examines this from a complementary perspective by measuring how the resulting AttentionSpan score changes as the number of prefill tokens increases. For The Stack, AttentionSpan is stable across the tested prefill lengths for all scorer models. For FineWeb, it generally stabilizes by $p = 32$, while Wikipedia requires up to $p = 128$ for the scores to fully stabilize. Together, these results suggest that prefill lengths in the range of 128–256 are more than sufficient to absorb the sink behavior relevant to our metric, while substantially shorter prefixes can already provide a close approximation.

Using a larger filler prefix, however, increases the sequence length and therefore the computational and memory cost of attention. Since standard attention scales quadratically with sequence length, we seek the smallest value of p that preserves the document ordering obtained with a sufficiently large prefix. Based on Figure 6, we use $p = 128$ as our reference because the AttentionSpan scores have stabilized across the tested domains and scorer models by this point. We then compute the Spearman rank correlation between the per-document AttentionSpan ranking obtained at $p = 128$ and the rankings obtained using smaller prefill lengths.

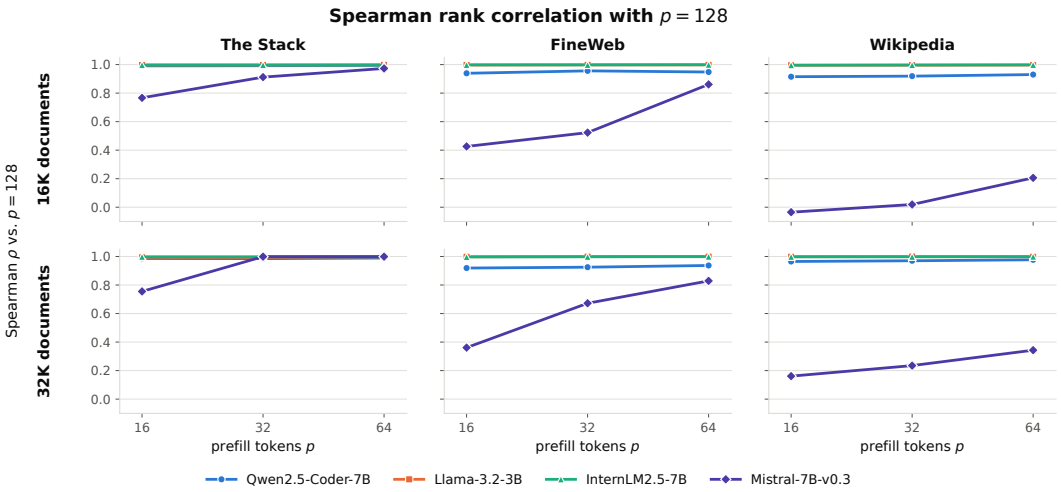


Figure 7: Spearman rank correlation between the per-document AttentionSpan ranking at prefill p and the ranking at $p = 128$, for four scorer models on three domains, at 16K (top) and 32K (bottom) document lengths. $p = 128$ is the reference and is therefore omitted from the x-axis.

As shown in Figure 7, $p = 64$ provides the closest approximation to the $p = 128$ document ranking among the smaller prefill lengths. This is important because our data-selection procedure ultimately depends on the *relative ranking* of documents rather than their exact AttentionSpan values. Combined with the negligible residual sink mass observed in Figure 5, these results motivate our choice of 64 prefill tokens: it provides a conservative balance between absorbing attention-sink behavior, preserving the document ordering obtained with a longer prefix, and limiting the additional computational and memory overhead.

C.3 Do the prefill-tokens preserve document-token attention?

Prior work shows that attention sinks are not merely removable noise: retaining the initial sink tokens in the KV cache is important for stable streaming generation, while evicting them can substantially degrade performance (Xiao et al., 2024). Therefore, our goal is not to remove sink behavior from the model's computation, but to prevent sink mass from contaminating *AttentionSpan*. The prefill-tokens are designed to redirect position-driven sink mass away from document tokens while preserving the relative attention pattern among valid documents keys.

Let e_{ij} be the raw attention logit from query position i to key position j. Let $D_i = \{j \in \mathcal{V}_K(x) : j < i\}$ denote the valid real document keys before query i, and let S_i denote the early sink-support positions. Without prefill-tokens, attention over valid document keys competes with sink positions:

$$A^0_{ij} = \frac{\exp(e^0_{ij})}{\sum_{s\in S_i}\exp(e^0_{is}) + \sum_{r\in D_i}\exp(e^0_{ir})}, \qquad j \in D_i. \tag{12}$$

After prepending the prefill-tokens, let F_i denote the prefill-token keys before query i. Attention to a valid document key becomes

$$A^F_{ij} = \frac{\exp(e^F_{ij})}{\sum_{f\in F_i}\exp(e^F_{if}) + \sum_{r\in D_i}\exp(e^F_{ir})}, \qquad j \in D_i. \tag{13}$$

We do not require $A^F_{ij} = A^0_{ij}$, since the prefill-tokens are meant to change the marginal allocation of attention mass. Instead, we compare the conditional distributions over valid document keys:

$$P_i(j) = \frac{A^0_{ij}}{\sum_{r\in D_i} A^0_{ir}}, \qquad Q_i(j) = \frac{A^F_{ij}}{\sum_{r\in D_i} A^F_{ir}}, \qquad j \in D_i. \tag{14}$$

If the document-key logits are approximately preserved under prefill-tokens insertion, $e^F_{ij} \approx e^0_{ij}$ for $j \in D_i$, then the prefill-tokens and sink terms appear only through the common softmax normalizer

Table 8: **Prefill-tokens fidelity and efficacy across domains.** $\mathrm{FillerMass}$ measures how much attention is absorbed by the prefill-tokens. DocJSD and $\mathrm{Overlap@\,3}$ measure whether the conditional attention distribution over valid document keys is preserved after prefill-token insertion. Negative $\Delta\,\mathrm{AttentionSpan}$ indicates that prefill-token correction reduces distance inflation from positional sinks.

Domain	$\mathrm{FillerMass}\uparrow$	$\mathrm{DocJSD}\downarrow$	$\mathrm{Overlap@\,3}\uparrow$	$\Delta\,\mathrm{AttentionSpan}$
Code / The Stack	0.096 ± 0.010	0.029 ± 0.003	0.780 ± 0.020	-0.370 ± 0.055
FineWeb	0.106 ± 0.009	0.032 ± 0.003	0.768 ± 0.047	-0.416 ± 0.028
Wikipedia	0.111 ± 0.005	0.034 ± 0.002	0.770 ± 0.022	-0.429 ± 0.021
Scientific papers	0.117 ± 0.011	0.037 ± 0.004	0.781 ± 0.049	-0.411 ± 0.018

and cancel under conditioning:

$$Q_i(j) = \frac{\exp(e^F_{ij})/Z^F_i}{\sum_{r\in D_i}\exp(e^F_{ir})/Z^F_i} = \frac{\exp(e^F_{ij})}{\sum_{r\in D_i}\exp(e^F_{ir})} \approx \frac{\exp(e^0_{ij})}{\sum_{r\in D_i}\exp(e^0_{ir})} = P_i(j). \tag{15}$$

Thus, the prefill-tokens are allowed to change how much total attention mass goes to document tokens, but it should leave the conditional distribution over valid document keys approximately unchanged.

Because real transformers use positional encodings and nonlinear layer updates, this logit-preservation condition is only approximate. We therefore validate it numerically across four domains using 500 documents per domain. We use four metrics. First, we measure the document-level Jensen-Shannon divergence between the conditional document-key distributions:

$$\mathrm{DocJSD}(x) = \frac{1}{|\mathcal{V}_Q(x)|}\sum_{i\in\mathcal{V}_Q(x)} \mathrm{JSD}(P_i, Q_i), \tag{16}$$

where

$$\mathrm{JSD}(P_i, Q_i) = \frac{1}{2}\,\mathrm{KL}(P_i\|M_i) + \frac{1}{2}\,\mathrm{KL}(Q_i\|M_i), \qquad M_i = \frac{1}{2}(P_i + Q_i). \tag{17}$$

Lower DocJSD means the conditional attention distribution over valid document keys changes less after prefill-token insertion.

Second, we measure the amount of attention absorbed by the prefill-tokens:

$$\mathrm{FillerMass}(x) = \frac{1}{|\mathcal{V}_Q(x)|}\sum_{i\in\mathcal{V}_Q(x)}\sum_{j\in F_i} A^F_{ij}. \tag{18}$$

Higher $\mathrm{FillerMass}$ means that the filler prefix is actually receiving attention mass.

Third, we measure whether the strongest valid document keys are preserved. Let $\mathcal{T}^0_i$ and $\mathcal{T}^F_i$ be the top-k valid document keys under P_i and Q_i, respectively. We define

$$\mathrm{Overlap@}\,k(x) = \frac{1}{|\mathcal{V}_Q(x)|}\sum_{i\in\mathcal{V}_Q(x)} \frac{|\mathcal{T}^0_i \cap \mathcal{T}^F_i|}{k}. \tag{19}$$

Higher $\mathrm{Overlap@}\,k$ means the same document keys remain among the strongest attended positions. We use $k = 3$.

Finally, we measure the signed change in the final document score:

$$\Delta\,\mathrm{AttentionSpan}(x) = \mathrm{AttentionSpan}_F(x) - \mathrm{AttentionSpan}_0(x). \tag{20}$$

A negative value means that the no-prefill-tokens score was inflated by early-position attention, and that prefill-tokens correction reduces this inflation.

Table 8 shows the desired pattern across all tested domains. The prefill-tokens absorbs non-trivial attention mass, ranging from approximately 9.6% in Code to 11.7% in Scientific papers. Despite this shift in marginal mass, the conditional attention distribution over valid document keys changes only modestly, with $\mathrm{DocJSD} < 0.04$ in every domain. The high $\mathrm{Overlap@\,3}$ values, ranging from 0.768 to 0.781, show that the strongest document keys are mostly preserved after prefill-token insertion. Finally, the consistently negative $\Delta\,\mathrm{AttentionSpan}$ values show that the no-prefill-token score is inflated by early-position attention. Together, these results support the intended behavior: the prefill-tokens redirects sink mass away from document positions while largely preserving the conditional attention pattern over valid document keys.

D FORMAL PROPERTIES OF THE ATTENTIONSPAN

For each valid query i, *AttentionSpan* summarizes the top-k lookback distances by weighting them according to the model's assigned attention mass. Let $\mathcal{T}_i$ denote the set of top-k attended valid keys. Each key $j \in \mathcal{T}_i$ is characterized by its lookback distance $d_{ij} = i - j$ and its normalized top-k attention weight $\hat{w}_{ij}$, such that

$$\sum_{j \in \mathcal{T}_i} \hat{w}_{ij} = 1, \qquad \hat{w}_{ij} \geq 0. \tag{21}$$

The choice of summary statistic can be viewed through the lens of empirical risk minimization. The weighted mean serves as the natural L_2 summary:

$$s_i^{\text{wmean}} = \arg\min_{\theta \in \mathbb{R}} \sum_{j \in \mathcal{T}_i} \hat{w}_{ij}(d_{ij} - \theta)^2 = \sum_{j \in \mathcal{T}_i} \hat{w}_{ij} d_{ij}. \tag{22}$$

By contrast, the weighted median is the corresponding L_1 representative:

$$s_i^{\text{wmed}} \in \arg\min_{\theta \in \mathbb{R}} \sum_{j \in \mathcal{T}_i} \hat{w}_{ij} |d_{ij} - \theta|. \tag{23}$$

The choice between these summaries is a choice between sensitivity to distance and sensitivity to cumulative attention mass. The weighted mean is sensitive to distant outliers because each key contributes proportionally to both its attention weight and its lookback distance. Thus, even a relatively small amount of attention mass assigned to a sufficiently distant token, such as a residual sink token near the beginning of a document, can substantially increase the resulting span.

In contrast, the L_1 objective grows linearly rather than quadratically with distance. Its marginal influence does not increase with the magnitude of an outlier, so the weighted median is governed primarily by where cumulative attention mass reaches one half rather than by the magnitude of individual lookback distances.

While the weighted median is more robust, its protection is not infinite. In the following results, we show that the weighted median agrees closely with the weighted mean when the attended positions are concentrated, while remaining robust to arbitrarily distant outliers as long as proximal positions retain at least half of the top-k attention mass.

Since each key corresponds to a unique sequence position, the lookback distances are distinct. Order them as

$$d_{(1)} < d_{(2)} < \cdots < d_{(k)}, \tag{24}$$

with corresponding weights $\hat{w}_{(1)}, \ldots, \hat{w}_{(k)}$. When the minimizer of (23) is not unique, we use the leftmost weighted median:

$$s_i^{\text{wmed}} = d_{(m)}, \qquad m = \min\left\{ m' : \sum_{r=1}^{m'} \hat{w}_{(r)} \geq \frac{1}{2} \right\}. \tag{25}$$

For comparison, the maximum summary is

$$s_i^{\max} = \max_{j \in \mathcal{T}_i} d_{ij} = d_{(k)}. \tag{26}$$

Theorem 1 (Agreement in coherent regimes)**.** *Suppose all top-k distances are δ-concentrated around a characteristic distance R_i:*

$$|d_{ij} - R_i| \leq \delta \qquad \textit{for all } j \in \mathcal{T}_i. \tag{27}$$

Then

$$|s_i^{\text{wmed}} - s_i^{\text{wmean}}| \leq \delta. \tag{28}$$

Proof. Let

$$a = R_i - \delta, \qquad b = R_i + \delta. \tag{29}$$

By (27), every distance d_{ij} lies in $[a, b]$. Let $m = s_i^{\text{wmed}}$. By the definition of a weighted median, at least one half of the total attention mass lies at distances greater than or equal to m, and at least one half lies at distances less than or equal to m.

If $s_i^{\text{wmean}} \geq m$, then

$$s_i^{\text{wmean}} - m = \sum_{j \in \mathcal{T}_i} \hat{w}_{ij}\, d_{ij} - m \sum_{j \in \mathcal{T}_i} \hat{w}_{ij} \tag{30}$$

$$= \sum_{j \in \mathcal{T}_i} \hat{w}_{ij}(d_{ij} - m) \tag{31}$$

$$= \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} > m}} \hat{w}_{ij}(d_{ij} - m) + \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} \leq m}} \hat{w}_{ij}(d_{ij} - m) \tag{32}$$

$$\leq \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} > m}} \hat{w}_{ij}(d_{ij} - m) \tag{33}$$

$$\leq \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} > m}} \hat{w}_{ij}(b - m) \tag{34}$$

$$= (b - m) \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} > m}} \hat{w}_{ij} \tag{35}$$

$$\leq \frac{1}{2}(b - m) \tag{36}$$

$$\leq \frac{1}{2}(b - a) = \delta. \tag{37}$$

Similarly, if $s_i^{\text{wmean}} \leq m$, then

$$m - s_i^{\text{wmean}} = m \sum_{j \in \mathcal{T}_i} \hat{w}_{ij} - \sum_{j \in \mathcal{T}_i} \hat{w}_{ij}\, d_{ij} \tag{38}$$

$$= \sum_{j \in \mathcal{T}_i} \hat{w}_{ij}(m - d_{ij}) \tag{39}$$

$$= \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} < m}} \hat{w}_{ij}(m - d_{ij}) + \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} \geq m}} \hat{w}_{ij}(m - d_{ij}) \tag{40}$$

$$\leq \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} < m}} \hat{w}_{ij}(m - d_{ij}) \tag{41}$$

$$\leq \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} < m}} \hat{w}_{ij}(m - a) \tag{42}$$

$$= (m - a) \sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} < m}} \hat{w}_{ij} \tag{43}$$

$$\leq \frac{1}{2}(m - a) \tag{44}$$

$$\leq \frac{1}{2}(b - a) = \delta. \tag{45}$$

Therefore,

$$|s_i^{\text{wmed}} - s_i^{\text{wmean}}| \leq \delta. \tag{46}$$

□

Theorem 2 (Robustness to far-back outliers)**.** *Partition the top-k keys into a proximal set G_i and an outlier set O_i. Assume*

$$d_{ij} \leq R_i \quad \textit{for } j \in G_i, \qquad d_{ij} \geq R_{\text{out}} \quad \textit{for } j \in O_i, \tag{47}$$

where $R_{\text{out}} > R_i$. Let

$$W_G = \sum_{j \in G_i} \hat{w}_{ij}. \tag{48}$$

If

$$W_G \geq \frac{1}{2}, \tag{49}$$

then

$$s_i^{\mathrm{wmed}} \leq R_i. \tag{50}$$

Proof. By assumption, every key in G_i has distance at most R_i. Since G_i carries total attention mass at least $1/2$,

$$\sum_{\substack{j \in \mathcal{T}_i \\ d_{ij} \leq R_i}} \hat{w}_{ij} \geq \sum_{j \in G_i} \hat{w}_{ij} = W_G \geq \frac{1}{2}. \tag{51}$$

By the definition of the leftmost weighted median in (25), the first distance at which the cumulative attention mass reaches $1/2$ must be no larger than R_i. Hence,

$$s_i^{\mathrm{wmed}} \leq R_i. \tag{52}$$

□

Corollary 2.1 (Weighted mean and maximum are outlier-sensitive)**.** *Let*

$$W_O = \sum_{j \in O_i} \hat{w}_{ij} = 1 - W_G. \tag{53}$$

If $W_O > 0$, then

$$s_i^{\mathrm{wmean}} \geq W_O R_{\mathrm{out}}. \tag{54}$$

Moreover, if O_i is nonempty, then

$$s_i^{\max} \geq R_{\mathrm{out}}. \tag{55}$$

Proof. Since all lookback distances are nonnegative,

$$\begin{aligned} s_i^{\mathrm{wmean}} &= \sum_{j \in G_i} \hat{w}_{ij} d_{ij} + \sum_{j \in O_i} \hat{w}_{ij} d_{ij} && (56)\\ &\geq \sum_{j \in O_i} \hat{w}_{ij} d_{ij} && (57)\\ &\geq \sum_{j \in O_i} \hat{w}_{ij} R_{\mathrm{out}} && (58)\\ &= W_O R_{\mathrm{out}}. && (59) \end{aligned}$$

Thus, for any fixed $W_O > 0$, the contribution of the outlier set can grow arbitrarily large as R_{out} increases.

For the maximum, if O_i is nonempty, then at least one key has distance at least R_{out}, so

$$s_i^{\max} = \max_{j \in \mathcal{T}_i} d_{ij} \geq R_{\mathrm{out}}. \tag{60}$$

□

Corollary 2.2 (Mass-awareness of the weighted median)**.** *Fix a top-k set $\mathcal{T}_i$ with distinct lookback distances $d_{(1)} < d_{(2)} < \cdots < d_{(k)}$, and let the weights range over the probability simplex*

$$\Delta^{k-1} = \left\{ \hat{w} \in \mathbb{R}^k_{\geq 0} : \sum_{r=1}^{k} \hat{w}_{(r)} = 1 \right\}. \tag{61}$$

Write

$$W_{(m)}(\hat{w}) = \sum_{r=1}^{m} \hat{w}_{(r)}, \qquad W_{(0)}(\hat{w}) = 0, \tag{62}$$

for the cumulative attention mass. Then the leftmost weighted median $s_i^{\mathrm{wmed}}(\hat{w})$ defined by (25) satisfies the following:

1. (Region characterization.) *For each* $m \in \{1, \dots, k\}$,

$$s_i^{\mathrm{wmed}}(\hat{w}) = d_{(m)} \quad \Longleftrightarrow \quad W_{(m-1)}(\hat{w}) < \frac{1}{2} \le W_{(m)}(\hat{w}). \tag{63}$$

The boundaries

$$\left\{ \hat{w} : W_{(m)}(\hat{w}) = \frac{1}{2} \right\}, \qquad m = 1, \dots, k-1, \tag{64}$$

partition the simplex into regions on which $s_i^{\mathrm{wmed}}(\hat{w})$ *takes a fixed value* $d_{(m)}$.

2. (Mass-monotonicity.) *If* $\hat{w}'$ *shifts attention mass toward larger lookback distances relative to* $\hat{w}$*, in the sense that*

$$W_{(m)}(\hat{w}') \le W_{(m)}(\hat{w}) \qquad \textit{for all } m, \tag{65}$$

then

$$s_i^{\mathrm{wmed}}(\hat{w}') \ge s_i^{\mathrm{wmed}}(\hat{w}). \tag{66}$$

3. (Invariance of the unweighted summary.) *Under uniform weights* $\hat{w}_{(r)} = 1/k$*, the leftmost weighted median reduces to the lower sample median*

$$s_i^{\mathrm{umed}} = d_{(\lceil k/2 \rceil)}. \tag{67}$$

This quantity depends only on the locations of the top-k *keys and is therefore unchanged by their observed attention weights.*

Proof. Let

$$F_{\hat{w}}(d) = \sum_{r:\, d_{(r)} \le d} \hat{w}_{(r)} \tag{68}$$

denote the cumulative distribution function of the discrete lookback distribution that places mass $\hat{w}_{(r)}$ at $d_{(r)}$. Since the distances are distinct,

$$F_{\hat{w}}(d_{(m)}) = W_{(m)}(\hat{w}), \tag{69}$$

and the leftmost weighted median in (25) is the generalized inverse

$$s_i^{\mathrm{wmed}}(\hat{w}) = F_{\hat{w}}^{-1}\left(\frac{1}{2}\right) = \inf\left\{ d : F_{\hat{w}}(d) \ge \frac{1}{2} \right\}. \tag{70}$$

(i) By monotonicity of $W_{(m)}(\hat{w})$ in m, the smallest index for which $W_{(m)}(\hat{w}) \ge 1/2$ is exactly the index satisfying

$$W_{(m-1)}(\hat{w}) < \frac{1}{2} \le W_{(m)}(\hat{w}). \tag{71}$$

By (25), the corresponding weighted median is therefore $d_{(m)}$. The boundaries between adjacent regions occur when a cumulative mass $W_{(m)}(\hat{w})$ is exactly $1/2$.

(ii) The condition in (65) implies

$$F_{\hat{w}'}(d) \le F_{\hat{w}}(d) \qquad \text{for all } d, \tag{72}$$

which corresponds to shifting probability mass toward larger lookback distances. Therefore,

$$\left\{ d : F_{\hat{w}'}(d) \ge \frac{1}{2} \right\} \subseteq \left\{ d : F_{\hat{w}}(d) \ge \frac{1}{2} \right\}. \tag{73}$$

Taking infima and applying (70) gives

$$s_i^{\mathrm{wmed}}(\hat{w}') \ge s_i^{\mathrm{wmed}}(\hat{w}). \tag{74}$$

(iii) Under uniform weights $\hat{w}_{(r)} = 1/k$,

$$W_{(m)} = \frac{m}{k}. \tag{75}$$

The first index for which $W_{(m)} \ge 1/2$ is $m = \lceil k/2 \rceil$. Hence,

$$s_i^{\mathrm{wmed}} = d_{(\lceil k/2 \rceil)} = s_i^{\mathrm{umed}}. \tag{76}$$

Unlike the weighted median, this unweighted summary depends only on the ordered top-k distances and not on their observed attention weights. □

E ABLATIONS

Layer subset analysis. We compute per-document AttentionSpan from each candidate layer subset and the full 28-layer analysis on the gold set (4,998 documents from FineWeb and The Stack labeled at 8K context using Qwen2.5-7B-Instruct), then take Pearson correlation across documents. A spread of three layers at $\{7, 14, 21\}$ achieves r=0.975 correlation with the full 28-layer analysis, providing a 9× speedup; early layers alone ($\{0, 1, 2, 3, 4\}$) achieve only r=0.36, motivating layers spanning the full depth.

Table 9: Layer configuration study. Per-document Pearson correlation between the AttentionSpan score from the listed layer subset and from the full 28-layer analysis (4,998 gold documents, FineWeb + The Stack, scored at 8K with Qwen2.5-7B-Instruct, eager attention, bfloat16). OLMo-3 uses the spread_3 subset, whereas the Llama-2 labeling runs use one deep layer. This ablation demonstrates that sparse layer sampling can approximate the full computation; it is not a specification shared by every experiment.

Configuration	Layers	Correlation
spread_3	$\{7, 14, 21\}$	$r = 0.975$
spread_6	$\{4, 9, 13, 18, 22, 27\}$	$r = 0.98$
early_5	$\{0, 1, 2, 3, 4\}$	$r = 0.36$

Cross-context correlation. Attention labels computed at one context length may not predict behavior at a different length. Using a held-out subset of $N \approx 5{,}000$ FineWeb+Stack documents long enough to fill 32K, we relabel the same documents at 4K, 8K, 16K, and 32K (Qwen2.5-7B-Instruct, spread_3) and compute per-document Pearson correlations between short- and long-context scores. Labels at 4K correlate poorly with 32K behavior ($r = 0.41$ for mean distance), while 16K labels achieve $r = 0.84$. We therefore compute training labels at (or near) the target training context length.

Table 10: Cross-context per-document Pearson correlation between attention-pattern scores at a short source context and at a 32K target context. $N \approx 5{,}000$ FineWeb+The Stack documents, Qwen2.5-7B-Instruct, spread_3.

Source to Target	Mean Dist. (r)	Local Ratio (r)	Entropy (r)
4K to 32K	0.41	0.83	0.90
8K to 32K	0.71	0.92	0.92
16K to 32K	0.84	0.95	0.96

Cross-model labeling for Llama-2 at 32K. Llama-2-7B was trained natively at 4K, so applying an 8.0 RoPE factor and asking it to judge dependencies at 32K places its attention far outside the calibrated regime. We instead score the same 10K candidates with four models that natively support long inputs: Qwen2.5-Coder-7B, Qwen2.5-1.5B, Llama-3.2-3B, and Phi-3.5-mini. These labelers span three families, and their document ranks are combined by Borda count. The resulting supervision comes from models other than the Llama-2 target, but it is obtained within the labelers' supported context range; we prefer that tradeoff to self-labels extrapolated eightfold beyond Llama-2's native window.

Domain-controlled ablation (Llama-2-7B, 32K). To remove domain identity as an explanation, we fix the contents of every 800-document condition at 267 code documents, 267 arXiv papers, and 266 books. Selection changes only within each domain: *dc_positive* uses its highest Borda ranks, *dc_negative* its lowest, and *dc_random* uniform draws from the same pool. At step 100, training losses are 1.07, 1.18, and 1.38, respectively. Table 11 shows the analogous 32K evaluation. The first run gives Recall of 47.2 for dc_positive, 36.8 for dc_random, and 29.7 for dc_negative, making the Positive–Random difference 10.4 points. RAG remains essentially unchanged (40.5–40.8). Overall scores for uncontrolled Positive (43.9) and domain-controlled Positive (44.0) are also nearly equal, indicating that within-domain attention rank explains a substantial component of the result. We

repeated the ablation on another cluster with training seeds 42, 43, and 44. Mean Recall was 34.43, 28.98, and 26.18 for dc_positive, dc_random, and dc_negative, respectively. The Positive–Random difference was 5.45 pp and had the same sign for every seed. This confirms the ordering, although the estimated gap is about half that of the initial run. The replication simultaneously changed the cluster, preprocessing, and RoPE scheme (from dynamic to linear), so the source of the smaller gap cannot be isolated. We therefore use the replicated estimate as the more reliable one.

Table 11: Domain-controlled HELMET ablation (Llama-2-7B, 32K). All three conditions use identical domain proportions (1/3 code, 1/3 arXiv, 1/3 books); only within-domain Borda rank varies. The AttentionSpan effect persists with domain held constant and is concentrated in Recall.

Condition	Domain Mix	Recall	RAG	Overall
dc_positive	267/267/266	**47.2**	40.8	**44.0**
dc_random	267/267/266	36.8	40.6	38.7
dc_negative	267/267/266	29.7	40.5	35.1

F Domain Composition and Gzip Proxy

Table 12: Domain composition of AttentionSpan-filtered training sets (Llama-2-7B, 16K). Middle's natural ∼70/30 code/web mix partly explains its strong performance.

Condition	Code %	Web %	Mean AttentionSpan	HELMET
Negative	6	94	0.003	30.84
Random	59	41	0.009	37.04
High-AttentionSpan	97	3	0.026	41.29
Middle	72	28	0.007	43.39
Mix 50/50	83	17	0.016	43.40
Mix 75/25	76	24	0.011	43.76

Table 13: Zero-cost proxies do not survive context extension. The MULTIHOP_NIAH column is the original 4K comparison (Llama-2 at 4K). The Recall columns evaluate Llama-2-7B trained at 32K under the same configuration as Table 2, panel (b). Compression ratio is competitive at 4K, but at 16K and 32K it falls below unfiltered random selection, whereas high-AttentionSpan selection stays above it at every length. Only the two selectors that were retrained at 32K have Recall entries.

Method	Multihop @4K	Recall @16K	Recall @32K	GPU Cost
High AttentionSpan (code)	**36.5%**	**57.1%**	**37.1%**	∼8 GPU-hrs
Compression ratio (gzip)	34.7%	35.6%	16.9%	0
Type-token ratio	16.7%	–	–	0
Cross-half overlap	5.0%	–	–	0
Random	4.0%	44.2%	25.8%	0

G Training Configuration

H Compute Used

Experiments used two HPC systems: one with four AMD MI250X accelerators (eight GCDs) per node, the other with four AMD MI300A accelerators per node. The total cost across both systems was in the tens of thousands of node-hours.

Table 14: Training hyperparameters for all LoRA fine-tuning experiments. The 16K runs use 4-GPU DDP with per-device batch 1 and gradient accumulation 2; the 32K runs train one LoRA condition per GPU (no DDP), with per-device batch 1 and gradient accumulation 8. Both setups give an effective (global) batch size of 8 sequences, i.e. 100 optimizer steps over 800 training documents in 1 epoch.

Hyperparameter	**Value**
Base model	Llama-2-7B-hf (also SmolLM2-1.7B, Qwen2.5-{0.5B,1.5B})
LoRA rank (r)	16
LoRA alpha (α)	32
LoRA target modules	q_proj, k_proj, v_proj, o_proj
LoRA dropout	0.05
Learning rate	2×10^{-4}
LR schedule	Cosine
Warmup ratio	0.1
Optimizer	AdamW
Weight decay	0.01
Gradient clipping (max-norm)	1.0 (HF Trainer default)
Gradient checkpointing	Enabled (`use_reentrant=False`)
Batch size (per GPU)	1
Gradient accumulation (16K, DDP, 4 GPUs)	2
Gradient accumulation (32K, single-GPU per cond.)	8
Effective (global) batch size	8 sequences
Optimizer steps per condition	100 (= 800 docs / batch 8, 1 epoch)
Epochs	1
Tokens seen per condition (16K / 32K)	13.1M / 26.2M
Precision	bfloat16
Max sequence length (16K experiments)	16,384 tokens
Max sequence length (32K experiments)	32,768 tokens
Padding	Right-pad with EOS to max length
Loss masking	Causal LM loss on all non-padding tokens (no packing)
RoPE scaling (8K, Llama-2)	Linear, factor=2.0
RoPE scaling (16K, Llama-2)	Linear, factor=4.0
RoPE scaling (32K, Llama-2)	Linear, factor=8.0
Attention implementation	SDPA (training), Eager (labeling)
DDP backend	NCCL (via ROCm/RCCL)
Hardware	$4\times$ AMD MI300A per node
Seed	42 or 3 seeds

I TRAINING DATA: SOURCES, SCRAPING, AND STATISTICS

This section documents the candidate pools we scraped for each experiment, the tokenizer-aware length filters that produced them, and the per-condition statistics of the final 800-document training sets.

I.1 SOURCES SCRAPED, BY EXPERIMENT

Table 15 lists the upstream datasets streamed for each context bucket and the tokenizer used to compute document length. All counts are post-streaming, post-length-filter document totals; documents shorter or longer than the per-bucket window are discarded at scrape time.

Table 15: Upstream datasets streamed for each context bucket. "Filter window" is the per-document token range for inclusion. "Tokenizer" is the model used to count tokens at filter time.

Bucket	Tokenizer	Filter window	Total docs scraped
16K (Phase 2/3, SmolLM2)	Qwen2.5-Coder-7B-Instruct	12,288–20,480	14,000
16K extended (with arXiv, Wikipedia)	Qwen2.5-Coder-7B-Instruct	12,288–20,480	52,920
32K (Phase 4)	Llama-2-7B-hf	24,576–49,152	17,270

I.2 16K CANDIDATE POOL (PHASES 2–3 AND SMOLLM2)

The 16K candidate pool combines FineWeb (web prose) and The Stack (code, eight programming languages), each filtered by the Qwen-Coder tokenizer to documents of $12{,}288$–$20{,}480$ tokens (a ± 4K-token band around the 16K target). Table 16 reports the scrape yield by source. The aggregate yield is $14{,}000$ documents kept from approximately 5.7M documents streamed; the long tail of FineWeb in particular is overwhelmingly short, with 99.7% of streamed documents below 8K tokens before the length filter kicks in.

Table 16: 16K candidate-pool scrape yield by source. "Scanned" is total documents streamed; "Saved" is documents passing the 12K–20K Qwen-Coder length filter; "Hit %" is yield rate. Code is split across eight Stack languages, each capped at 875 saved documents.

Source	Scanned	Saved	Hit %
FineWeb (web)	5,026,176	7,000	0.14
The Stack – python	62,795	875	1.39
The Stack – javascript	69,916	875	1.25
The Stack – java	133,946	875	0.65
The Stack – c	35,867	875	2.44
The Stack – go	53,833	875	1.63
The Stack – typescript	158,015	875	0.55
The Stack – c-sharp	141,811	875	0.62
The Stack – c++	31,183	875	2.81
Total	**5,713,542**	**14,000**	**0.25**

The extended 16K pool (used for the diverse-domain conditions in Phase 3) adds arXiv ($10{,}000$ docs), Wikipedia ($1{,}656$ docs), and an enlarged FineWeb/Code ($17{,}264 + 24{,}000$) drawn from the same scrape pipeline at the same length window. Wikipedia required additional language filtering after scrape (61.8% clean rate), and the code subset has elevated duplication (776 near-duplicates; 60.9% clean rate) relative to FineWeb (99.8%).

I.3 32K CANDIDATE POOL (PHASE 4)

The 32K candidate pool was scraped with the Llama-2 tokenizer at a $24{,}576$–$49{,}152$ token filter window, drawing from four upstream sources (Table 17). The Pile (uncopyrighted snapshot) provides the largest contribution at $10{,}000$ documents from 2.9M scanned (0.35% hit), followed by pile-of-law

at 5,000 from 978K (0.51%). Total saved: 17,270 documents, of which the 10,000-document subset used for the actual 32K Borda labeling pipeline was drawn proportionally across sources.

Table 17: 32K candidate-pool scrape yield by source (Llama-2 tokenizer, 24,576–49,152 token filter).

Source	Scanned	Saved	Hit %
monology/pile-uncopyrighted	2,887,917	10,000	0.35
pile-of-law/pile-of-law (legal)	978,319	5,000	0.51
allenai/c4 (web)	20,000,000	1,945	0.01
scientific_papers (PubMed)	119,924	325	0.27
Total	**23,986,160**	**17,270**	**0.07**

I.4 Final training-set composition by condition

After scoring with AttentionSpan and selecting per-condition tiers (positive, negative, middle, etc.), the final training sets used in the main experiments are summarized in Table 18. Each row corresponds to one 800-document training set used to train one Llama-2 LoRA at 16K (or, where indicated, 32K).

Table 18: Training data statistics for each condition. All conditions use 800 documents at 16K context unless noted. AttentionSpan (mean). Code/Web percentages may not sum to 100% for diverse conditions with additional domains.

Condition	N	Code %	Web %	Mean AttentionSpan	Selection
Balanced Top	800	50	50	0.0118	50/50 code/web, high AttentionSpan per domain
Diverse Bal. Top	800	46	21	0.0130	7-domain balanced, high AttentionSpan
Mix 75/25	800	76	24	0.0113	75% middle + 25% High-AttentionSpan
Diverse Top 50%	800	59	9	0.0136	7-domain, top 50% AttentionSpan
Mix 50/50	800	83	17	0.0162	50% middle + 50% High-AttentionSpan
Middle	800	72	28	0.0070	Median AttentionSpan (0.006–0.008)
Code Top	800	100	0	0.0264	100% code, high AttentionSpan
Balanced Bottom	800	50	50	0.0050	50/50 code/web, low AttentionSpan per domain
Diverse Bal. Bottom	800	46	21	0.0057	7-domain balanced, low AttentionSpan
Diverse Bottom 50%	800	34	31	0.0052	7-domain, bottom 50% AttentionSpan
Longest Docs	800	100	0	0.0114	Top 800 by token count
High-AttentionSpan	800	97	3	0.0260	Top AttentionSpan (≥ 0.017)
Code Bottom	800	100	0	0.0045	100% code, low AttentionSpan
Random	800	59	41	0.0090	Uniform random (no filter)
Low-AttentionSpan	800	6	94	0.0030	Bottom AttentionSpan (< 0.005)

J Full HELMET Results: Llama-2-7B at 16K

Table 19: Full per-subtask HELMET results for Llama-2-7B at 16K context. Scores are exact match (%) for Recall and RAG, ROUGE-L F1 (%) for LongQA. Each subtask is evaluated at 8K and 16K context with $n = 100$ samples.

	Recall		RAG								LongQA						Rec	RAG	QA
Condition	JSON-KV 8K	JSON-KV 16K	NQ 8K	NQ 16K	TriviaQA 8K	TriviaQA 16K	HotpotQA 8K	HotpotQA 16K	PopQA 8K	PopQA 16K	NarrativeQA 8K	NarrativeQA 16K	InfBench QA 8K	InfBench QA 16K	InfBench Choice 8K	InfBench Choice 16K	Avg	Avg	Avg
Balanced Top	55.0	39.0	49.5	40.2	76.3	79.2	42.3	41.7	40.8	35.2	28.1	28.1	21.2	25.8	26.0	28.0	47.0	50.6	26.2
Diverse Bal. Top	49.0	45.0	50.8	40.7	75.5	78.5	42.0	41.0	42.0	33.8	28.3	30.7	22.1	27.3	24.0	27.0	47.0	50.5	26.6
Mix 75/25	52.0	36.0	50.0	41.3	75.5	79.0	43.3	42.3	41.3	34.7	29.3	28.8	22.4	26.1	26.0	28.0	44.0	50.9	26.8
Diverse Top 50%	50.0	35.0	51.8	38.7	75.8	79.5	43.7	40.0	43.0	34.7	29.5	29.4	21.4	26.4	27.0	28.0	42.5	50.9	27.0
Mix 50/50	49.0	39.0	51.2	39.3	76.3	79.7	42.7	43.0	39.7	34.3	28.7	30.4	21.6	26.6	28.0	30.0	44.0	50.8	27.6
Middle	53.0	41.0	43.8	43.3	75.8	79.8	42.0	41.0	41.0	33.8	29.3	29.1	21.8	24.6	28.0	29.0	47.0	50.1	27.0
Code Top	59.0	42.0	48.8	38.5	76.2	78.2	43.0	42.7	41.3	34.7	27.7	26.1	21.2	26.2	30.0	27.0	50.5	50.4	26.4
Balanced Bottom	48.0	31.0	50.0	40.8	75.8	80.2	41.7	39.7	40.2	34.5	29.3	29.4	21.4	26.0	28.0	28.0	39.5	50.4	27.0
Diverse Bal. Bottom	49.0	32.0	49.5	38.8	75.3	78.8	41.7	40.0	40.5	33.3	29.0	30.9	22.5	25.8	27.0	27.0	40.5	49.8	27.0
Diverse Bottom 50%	45.0	32.0	48.2	39.5	76.0	78.8	40.7	39.0	40.3	34.3	28.6	28.9	22.5	24.9	23.0	28.0	38.5	49.6	26.0
Longest Docs	55.0	33.0	49.5	40.0	76.3	79.5	43.0	40.0	41.0	33.5	29.2	26.8	20.4	26.3	28.0	30.0	44.0	50.4	26.8
High-AttentionSpan	66.0	38.0	40.2	36.2	76.5	77.5	40.7	40.7	40.7	34.8	28.0	23.9	22.5	23.4	28.0	30.0	52.0	48.4	26.0
Code Bottom	51.0	33.0	48.8	40.5	76.5	79.7	41.3	41.3	41.3	33.7	26.6	26.8	19.8	26.5	29.0	33.0	42.0	50.4	26.9
Random	35.0	20.0	39.5	39.0	77.3	78.0	41.3	40.0	39.2	34.8	29.3	24.9	21.5	23.9	26.0	32.0	27.5	48.6	26.3
Low-AttentionSpan	12.0	2.0	37.0	35.7	75.5	78.8	40.3	39.3	36.7	32.7	30.3	27.0	21.2	27.2	28.0	28.0	7.0	47.0	27.0
Baseline (no LoRA)	1.0	0.0	38.0	19.8	66.5	59.2	33.3	22.3	30.0	22.2	26.8	15.4	20.9	15.8	24.0	21.0	0.5	36.4	20.6

J.1 Extended Conditions

Table 20: Extended HELMET results for Llama-2-7B at 16K context: ablations on dataset size, labeling model, and voting. High-AttentionSpan (1600) and Low-AttentionSpan (1600) continue training from their respective 800-document checkpoints. Random, Top 50%, and Bottom 50% (1600) start from the base model, so comparisons among the 1600-document rows are not controlled. 100 Evalution examples per condition and test. Scores as in Table 19.

	Recall		RAG								LongQA						Rec	RAG	QA
Condition	*JSON-KV 8K*	*JSON-KV 16K*	*NQ 8K*	*NQ 16K*	*TriviaQA 8K*	*TriviaQA 16K*	*HotpotQA 8K*	*HotpotQA 16K*	*PopQA 8K*	*PopQA 16K*	*NarrativeQA 8K*	*NarrativeQA 16K*	*InfBench QA 8K*	*InfBench QA 16K*	*InfBench Choice 8K*	*InfBench Choice 16K*	**Avg**	**Avg**	**Avg**
High-AttentionSpan (1600)	57.0	59.0	42.8	40.8	76.8	80.8	42.7	43.0	43.2	35.3	28.7	30.8	21.8	24.9	26.0	30.0	58.0	50.7	27.0
Random (1600)	58.0	40.0	51.5	39.7	76.5	80.5	44.0	40.7	43.0	35.0	27.7	32.0	21.0	25.7	29.0	28.0	49.0	51.4	27.3
Top 50% (1600)	58.0	47.0	51.3	40.5	76.5	79.8	45.0	43.7	43.5	35.8	27.9	29.8	21.3	27.8	27.0	31.0	52.5	52.0	27.5
Bottom 50% (1600)	59.0	42.0	51.8	40.5	77.8	82.7	45.3	39.3	42.5	35.8	28.8	31.5	21.4	26.7	26.0	28.0	50.5	52.0	27.1
Top 50% (800)	53.0	37.0	43.3	41.7	75.8	79.0	40.7	41.7	40.8	35.2	27.8	28.6	20.6	24.8	27.0	29.0	45.0	49.8	26.3
Llama-2 Top 400	34.0	20.0	48.3	38.3	77.2	77.7	44.0	41.0	40.3	32.7	29.0	27.2	22.3	24.7	29.0	30.0	27.0	49.9	27.0
Qwen Top 400	47.0	28.0	46.8	37.3	76.3	78.7	42.3	39.3	41.7	32.3	28.0	27.6	21.6	25.2	31.0	29.0	37.5	49.4	27.1
Qwen Bottom 400	27.0	14.0	43.8	36.2	76.5	76.2	43.0	38.7	38.5	33.3	29.4	28.7	21.3	24.6	26.0	25.0	20.5	48.3	25.8
Vote High-AttentionSpan	53.0	38.0	51.8	40.7	76.5	79.8	40.0	40.3	43.0	33.3	29.6	29.4	20.5	26.7	25.0	27.0	45.5	50.7	26.4
Vote Middle	47.0	33.0	50.2	39.5	75.5	78.5	41.7	40.3	42.8	33.7	28.9	30.0	22.0	26.4	27.0	29.0	40.0	50.3	27.2
Vote Mix 75/25	53.0	35.0	51.0	41.0	76.5	80.0	41.7	41.0	44.2	34.2	29.4	33.1	21.8	27.2	27.0	28.0	44.0	51.2	27.7

K Full HELMET Results: Llama-2-7B at 32K

Table 21: Full per-subtask HELMET results for Llama-2-7B at 32K context. Scores are exact match (%) for Recall and RAG. Each subtask is evaluated at 16K and 32K with $n = 100$ samples.

	Recall		RAG								Rec	RAG
Condition	*JSON-KV 16K*	*JSON-KV 32K*	*NQ 16K*	*NQ 32K*	*TriviaQA 16K*	*TriviaQA 32K*	*HotpotQA 16K*	*HotpotQA 32K*	*PopQA 16K*	*PopQA 32K*	**Avg**	**Avg**
High-AttentionSpan	58.0	27.0	45.2	36.2	79.0	77.7	36.3	35.7	34.0	35.5	42.5	47.4
Low-AttentionSpan	25.0	7.0	41.7	35.2	79.8	74.0	37.0	35.7	33.7	35.3	16.0	46.5
Random	37.0	16.0	41.3	35.7	81.5	76.5	39.0	35.3	34.0	34.7	26.5	47.2
Middle	45.0	18.0	42.0	37.0	80.8	77.5	38.0	38.0	34.2	34.8	31.5	47.8
Mix 50/50	32.0	15.0	42.7	37.8	78.3	75.8	39.0	35.0	33.0	34.0	23.5	47.0
Diverse Top	48.0	16.0	42.0	36.3	79.3	76.5	37.0	35.7	32.0	34.2	32.0	46.6
Diverse Bottom	25.0	5.0	44.7	33.8	78.5	74.7	39.3	35.0	34.8	35.2	15.0	47.0
Baseline (no LoRA)	0.0	0.0	0.0	0.0	3.5	2.8	0.7	0.0	0.7	0.0	0.0	1.0

L Cross-Model Replication: SmolLM2-1.7B and Llama-3-8B

We replicate the AttentionSpan ordering on two additional base models that differ from Llama-2-7B in scale, architecture, and pretraining recipe: SmolLM2-1.7B (a small model trained at 16K) and Llama-3-8B (grouped-query attention, distinct tokenizer, native 8K context).

L.1 SmolLM2-1.7B at 16K

We additionally evaluate AttentionSpan on SmolLM2-1.7B using the same training setup as Llama-2, including the same training hyperparameters and LoRA fine-tuning procedure. Table 22 reports the resulting 16K HELMET scores. SmolLM2 preserves the expected AttentionSpan ordering on the aggregate metrics: High-AttentionSpan $>$ random $>$ Low-AttentionSpan for both Recall and Overall. The separation is particularly clear on Recall, where the high-AttentionSpan, random, and low-AttentionSpan subsets obtain 35.2, 33.4, and 28.2, respectively. However, the margins are smaller than those observed for the larger models in our main experiments, and the individual RAG and LongQA categories show only limited separation.

SmolLM2 and Llama-2 are both adapted with LoRA under the same experimental setup, yet Llama-2 exhibits a substantially stronger separation between the AttentionSpan-selected data mixtures. This

Table 22: SmolLM2-1.7B results on 16K HELMET category averages. The AttentionSpan ordering (High-AttentionSpan > random > Low-AttentionSpan) is preserved on Recall and Overall.

Condition	Recall	RAG	LongQA	Overall
High-AttentionSpan (LoRA)	**35.2**	**42.8**	22.8	**33.6**
Random (LoRA)	33.4	42.4	22.6	32.8
Low-AttentionSpan (LoRA)	28.2	42.0	**23.0**	31.1
Baseline (no LoRA)	19.6	40.9	20.4	27.0

makes the adaptation method alone an unlikely explanation for the weaker SmolLM2 result. A more plausible explanation is the underlying model: SmolLM2 is substantially smaller and differs from the models in our main experiments in its architecture and pretraining regime. These differences can affect both its ability to exploit long-range dependencies and the extent to which additional long-context training can modify that behavior.

Our experiments cannot isolate model scale from pretraining or architecture, so we do not claim which of these factors is responsible. Nevertheless, the broader evidence is consistent: Llama-2, Llama-3, and OLMo-3 all show stronger gains from AttentionSpan-selected data, while SmolLM2 still preserves the expected ordering but with a smaller effect.

Table 23: Full per-subtask HELMET results for SmolLM2-1.7B at 16K context. Scores as in Table 19.

	Recall		RAG								LongQA						Rec	RAG	QA
Condition	*JSON-KV 8K*	*JSON-KV 16K*	*NQ 8K*	*NQ 16K*	*TriviaQA 8K*	*TriviaQA 16K*	*HotpotQA 8K*	*HotpotQA 16K*	*PopQA 8K*	*PopQA 16K*	*NarrativeQA 8K*	*NarrativeQA 16K*	*InfBench QA 8K*	*InfBench QA 16K*	*InfBench Choice 8K*	*InfBench Choice 16K*	Avg	Avg	Avg
High-AttentionSpan	37.0	19.0	36.2	36.3	73.7	65.7	37.3	30.7	32.8	29.8	21.3	18.9	17.1	17.4	32.0	30.0	28.0	42.8	22.8
Low-AttentionSpan	25.0	18.0	36.3	35.3	73.8	63.8	35.3	30.7	32.3	28.0	21.0	19.3	17.6	15.1	33.0	32.0	21.5	42.0	23.0
Random	37.0	18.0	36.5	36.2	74.5	64.8	36.3	30.3	32.5	28.0	20.8	20.3	18.2	17.2	30.0	29.0	27.5	42.4	22.6
Baseline (no LoRA)	18.0	17.0	34.5	34.3	70.8	66.8	33.3	29.7	32.8	25.2	17.2	15.9	18.9	15.6	28.0	27.0	17.5	40.9	20.4
High-AttentionSpan SFT	37.0	16.0	37.8	38.3	73.3	62.2	36.3	27.0	33.0	30.0	21.6	20.2	13.0	12.4	21.0	29.0	26.5	42.2	19.5
Low-AttentionSpan SFT	33.0	18.0	38.0	37.8	73.3	62.5	34.3	27.0	33.3	29.0	21.3	18.7	13.3	12.0	27.0	29.0	25.5	41.9	20.2
Base SFT	33.0	17.0	38.3	37.5	74.7	63.7	35.3	27.3	32.0	29.8	20.0	18.4	13.6	13.0	30.0	31.0	25.0	42.3	21.0

L.2 LLAMA-3-8B AT 64K

The AttentionSpan ordering also holds on Llama-3-8B, a model with grouped-query attention and a different pretraining mixture from Llama-2 and SmolLM2. We train at 32K context on a 200-document subset per condition (one epoch, otherwise identical LoRA hyperparameters), and evaluate at 64K, a 2× extrapolation beyond the training window, using HELMET Recall (RoPE factor 16). Table 24 reports the four Recall subtasks. High-AttentionSpan beats random by $+13.2$ pp on macro Recall and low-AttentionSpan trails random by -6.5 pp; the monotonic High-AttentionSpan > random > Low-AttentionSpan ordering replicates despite the smaller document budget and the cross-model, cross-context setting. RAG and LongQA were not run for this configuration.

Table 24: Llama-3-8B 64K HELMET Recall (%, $n = 100$ samples per subtask). LoRA trained at 32K on 200 documents per condition; eval at 64K with RoPE factor 16. The High-AttentionSpan > random > Low-AttentionSpan ordering replicates the AttentionSpan signal observed on Llama-2-7B and SmolLM2.

Condition	JSON-KV	NIAH MK-2	NIAH MK-3	NIAH MV	Avg
High-AttentionSpan (LoRA)	**65.0**	**61.0**	**66.0**	**71.3**	**65.8**
Random (LoRA)	57.0	54.0	49.0	50.3	52.6
Low-AttentionSpan (LoRA)	46.0	49.0	40.0	49.5	46.1
Baseline (no LoRA)	11.0	0.0	4.0	0.3	3.8

L.3 LLAMA-3.1-8B AT 128K

We also test Llama-3.1-8B-Instruct to determine whether AttentionSpan remains effective at substantially longer context lengths. Unlike our context-extension experiments, this version of Llama-3.1 already natively supports 128K context. Thus, this experiment tests data selection at 128K rather than extension to 128K. All High-AttentionSpan, Random, and Low-AttentionSpan conditions start from the same pretrained checkpoint, isolating the effect of the selected training data.

We score 1,276 documents using Gemma-4-31B at its native 256K length and construct high-AttentionSpan, Random, and low-AttentionSpan pools of 100, 200, and 400 examples. Each model receives one LoRA epoch with the otherwise unchanged configuration and is evaluated at 128K on four HELMET Recall subtasks (NIAH `mk_2`, `mk_3`, `mv`, and JSON-KV; $n = 25$ each). Experiments at this length are substantially more expensive: the quadratic attention cost makes both training and evaluation at 128K computationally and memory intensive. These constraints prevent the larger-scale full-parameter training and broader evaluation used in our shorter-context experiments, so we use LoRA and restrict evaluation to the Recall tasks.

Despite starting from a strong native-128K model, Figure 8 shows that High-AttentionSpan consistently performs best across the tested pool sizes. At 200 documents (Table 25), High-AttentionSpan achieves 97.8 Recall, compared with 79.5 for Random and 86.8 for low-AttentionSpan. The untuned baseline already scores 94.5, leaving limited room for additional gains; nevertheless, High-AttentionSpan preserves and slightly improves this capability, while Random and Low-AttentionSpan substantially degrade it. Thus, although this is not a context-extension experiment, the separation from a common 128K baseline provides evidence that AttentionSpan remains useful for selecting long-context training data even at 128K.

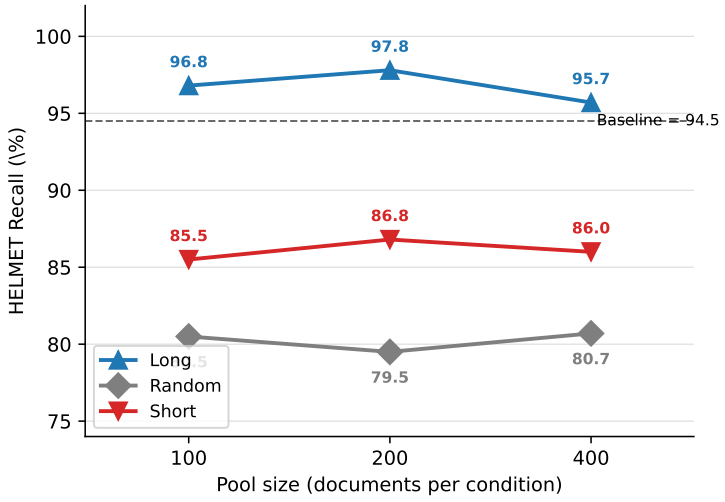


Figure 8: HELMET Recall over four NIAH subtasks for Llama-3.1-8B-Instruct ($n = 25$ per subtask). Model trained on high AttentionSpan data consistently outperforms the random control and low AttentionSpan model and the base model.

Table 25: Per-task HELMET Recall for Llama-3.1-8B at its native 128K length ($n = 25$).

Condition	JSON-KV	NIAH MK-2	NIAH MK-3	NIAH MV	Recall
High-AttentionSpan (LoRA)	**100.0**	**100.0**	**92.0**	**99.0**	**97.8**
Random (LoRA)	76.0	84.0	60.0	98.0	79.5
Low-AttentionSpan (LoRA)	84.0	92.0	72.0	**99.0**	86.8
Baseline (no LoRA)	96.0	96.0	96.0	90.0	94.5

M EARLY CALIBRATION WITH QWEN-2.5-1.5B.

As an early calibration experiment, we trained Qwen-2.5-1.5B at 32K context using documents from the high and low ends of the AttentionSpan ranking. The purpose of this experiment was not to establish scaling behavior, but to test whether the proposed ranking produced a measurable separation in downstream performance before running the larger experiments.

Table 26 shows a clear high–low separation on Recall. Training on positively ranked documents achieves an average Recall score of 76.5, compared with 61.0 for negatively ranked documents. The difference appears across both Recall subtasks, with gains of 12.0 points on JSON-KV at 16K and 19.0 points at 32K. In contrast, RAG performance is essentially unchanged (43.2 versus 43.4).

Table 26: Early calibration results for Qwen-2.5-1.5B at 32K context. Recall and RAG scores are exact-match percentages, with $n = 100$ samples per subtask.

	Recall		**RAG**								**Recall**	**RAG**
Condition	*JSON-KV 16K*	*JSON-KV 32K*	*NQ 16K*	*NQ 32K*	*TriviaQA 16K*	*TriviaQA 32K*	*HotpotQA 16K*	*HotpotQA 32K*	*PopQA 16K*	*PopQA 32K*	**Avg.**	**Avg.**
High-AttentionSpan	84.0	69.0	37.3	32.8	69.0	61.5	41.3	38.3	34.2	30.8	**76.5**	43.2
Low-AttentionSpan	72.0	50.0	37.3	33.5	68.5	61.5	41.3	38.0	35.8	30.8	61.0	**43.4**

These results provided an early indication that AttentionSpan was capturing a useful signal: selecting from opposite ends of the ranking produced a large difference on the long-context Recall tasks. At the same time, the absence of a corresponding improvement on RAG suggested that this signal should not be interpreted as a general measure of document quality. Rather, the result was consistent with AttentionSpan identifying data that is particularly useful for learning long-range retrieval behavior. We therefore treated this experiment as preliminary calibration and used it to motivate the more controlled experiments with larger models and broader evaluations presented in the main study.

N LLAMA-2 AND LLAMA-3 LORA FINE-TUNING

N.1 TRAINING HYPERPARAMETERS

Table 27: Training hyperparameters for the Llama-2 and Llama-3 runs. All conditions within a column share these settings; only the selection rule changes between conditions. Values spanning the table are shared by every run.

Setting	L2 16K	L2 32K	L3-8B	L3.1-8B
Adaptation	LoRA, rank 16, α=32, dropout 0.05			
Target modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`			
Trainable parameters	16.8M			
Optimizer	AdamW, weight decay 0.01			
Learning rate	2×10^{-4}			
LR schedule	Cosine, warmup ratio 0.1			
Maximum gradient norm	1.0			
Precision	bfloat16			
Epochs	1			
Per-device batch size	1 sequence			
Padding	Right-pad with EOS to the maximum length			
Base model	Llama-2-7B	Llama-2-7B	Llama-3-8B	Llama-3.1-8B-Inst.
Native context	4K	4K	8K	128K
Training context	16,384	32,768	32,768	131,072
Position scaling	Linear 4.0	Linear 8.0	Linear 8.0	None
Evaluation context	8K, 16K	16K, 32K	64K	128K
Docs / condition	800	800	200	100/200/400
Gradient accumulation	2	8	8	8
Effective batch	8 sequences			
Optimizer steps	100	100	25	13/25/50
Tokens seen	13.1M	26.2M	6.6M	13–52M
Scoring model	Qwen2.5-Coder	Borda (4 scorers)	Qwen2.5-Coder	Gemma-4-31B

N.2 DATA MIXTURES

Table 28: Candidate pools from which every Llama condition selects. The 16K pool is filtered to at least 16,384 Llama-2 tokens and the 32K pool to at least 32,768; all conditions at a given context length select from the same pool. The eight code languages are Python, JavaScript, Java, C, Go, TypeScript, C# and PHP, 875 documents each.

Pool	Source	Docs	Share
16K	FineWeb (web)	7,000	50.0%
	The Stack (8 languages, 875 each)	7,000	50.0%
	Total	**14,000**	
32K	The Pile (uncopyrighted)	10,000	57.9%
	Pile of Law (legal)	5,000	29.0%
	C4 (web)	1,945	11.3%
	Scientific papers (PubMed)	325	1.9%
	Total	**17,270**	

Per-condition domain shares are in Table 18; scrape yields are in Tables 16 and 17.

O OLMO-3 FULL PARAMETER FINETUNING

O.1 TRAINING HYPERPARAMETERS

Table 29 shows the hyperparameters used in experiments in Section 4.1.

Table 29: Training hyperparameters for the Qwen-graded runs. The two tiers differ only in the number of selected documents (and therefore token budget, steps, and warmup); all other settings are shared.

Hyperparameter	**10k tier**	**30k tier**
Base model	Olmo-3-7B (stage 2, step 47,684)	
Base context length	8k	
Training context length	32,768	
Context extension	YaRN, factor 4, β_{fast}=32, β_{slow}=1	
Number of epochs	5	
Optimizer	SkipStepAdamW (β_1=0.9, β_2=0.95, ϵ=10^{-8})	
Learning rate	2.07×10^{-4}	
Weight decay	0.1 (0 on embeddings)	
LR scheduler	Linear decay to 0 (α_f=0)	
Global batch size	4,194,304 tokens (128 × 32,768)	
Gradient accumulation	None	
Maximum gradient norm	1.0	
z-loss multiplier	10^{-5}	
Precision	bf16 parameters, fp32 gradient reduction	
Packing strategy	Best fit packing	
Number of documents	10,000	30,000
Total tokens seen	1.638B	4.915B
Optimizer steps	391	1,172
Warmup	40 steps	117 steps

O.2 Data mixtures

Here, we report the data mix used for the experiments and composition of the 10k and 30k document mixes.

Table 30: Composition of the candidate pool graded by Qwen-2.5-Coder-7B. All documents are sampled at 32,768 tokens across 14 domains; both tiers select from this same pool.

Domain	**Docs**	**Share**
US GPO	33,103	12.8%
Scientific papers	30,039	11.6%
peS2o	30,037	11.6%
arXiv papers	30,027	11.6%
Caselaw	30,000	11.6%
GitHub archive	29,751	11.5%
Financial (SEC)	27,383	10.6%
Wikipedia	14,589	5.7%
Project Gutenberg	9,286	3.6%
Math	8,849	3.4%
Regulations	7,526	2.9%
UK Hansard	5,338	2.1%
Government reports	1,921	0.7%
LibreTexts	204	0.1%
Total	**258,053**	**100%**

Table 31: Domain composition of the Qwen-2.5-Coder-7B–selected documents (Long: top 25% by AttentionSpan; Short: bottom 25%; Random: uniform), domain-balanced, for the 10k/30k tiers.

	10k			30k		
Domain	L	R	S	L	R	S
Wikipedia	940	754	924	2715	2323	2715
Financial	925	739	893	2715	2323	2715
Scientific papers	924	763	885	2716	2323	2716
peS2o	917	798	931	2716	2323	2716
GitHub	911	782	890	2715	2323	2715
Caselaw	910	767	919	2715	2323	2715
US GPO	891	784	895	2715	2323	2715
arXiv	879	822	925	2715	2323	2715
Gutenberg	759	783	765	2320	2323	2320
Math	739	750	716	2213	2323	2213
Regulations	610	771	649	1879	2323	1879
UK Hansard	428	770	441	1334	2322	1334
Gov. reports	153	657	149	481	1921	481
LibreTexts	14	60	18	51	204	51
Total	**10,000**			**30,000**		

L = Long, R = Random, S = Short.

O.3 Task-wise breakdown of HELMET and RULER scores across 10k and 30k mixes of the OLMo-3 experiments

Table 32: Per-task HELMET scores, 10k tier, seed 42

	Recall					RULER													RAG					LongQA			
Condition	JKV	MK2	MK3	MV	Avg	S1	S2	S3	MK1	MK2	MK3	MQ	MV	CWE	FWE	VT	QA1	QA2	Avg	NQ	TQA	HQA	PQA	Avg	NarrQA	InfQA	InfCh	Avg
Base + YaRN	23.0	17.0	2.0	41.8	20.9	99.0	51.0	57.0	39.0	17.0	2.0	45.8	41.8	3.7	**66.0**	26.0	**53.0**	**44.0**	41.9	**43.3**	77.2	44.7	33.2	49.6	**17.3**	**18.3**	**21.0**	**18.9**
Positive (long)	**91.0**	**98.0**	32.0	**74.2**	**73.8**	**100.0**	97.0	95.0	65.0	**98.0**	32.0	**75.5**	**74.2**	2.7	61.0	39.0	50.0	42.0	**64.0**	37.3	**85.8**	42.3	40.0	**51.4**	14.8	13.6	12.0	13.5
Random	65.0	77.0	**44.0**	66.2	63.1	**100.0**	98.0	95.0	**71.0**	77.0	**44.0**	68.2	66.2	2.8	18.0	31.8	40.0	42.0	58.0	37.2	79.8	44.0	**40.5**	50.4	15.3	9.4	15.0	13.2
Negative (short)	56.0	63.0	19.0	51.2	47.3	**100.0**	**99.0**	**98.0**	47.0	63.0	19.0	52.5	51.2	**8.5**	22.0	**49.8**	28.0	42.0	52.3	34.8	76.8	**45.0**	38.2	48.7	15.7	7.8	7.0	10.2

Table 33: Per-task HELMET scores, 10k tier, seed 123

	Recall					RULER													RAG					LongQA			
Condition	JKV	MK2	MK3	MV	Avg	S1	S2	S3	MK1	MK2	MK3	MQ	MV	CWE	FWE	VT	QA1	QA2	Avg	NQ	TQA	HQA	PQA	Avg	NarrQA	InfQA	InfCh	Avg
Base + YaRN	16.0	17.0	2.0	41.8	19.2	99.0	51.0	57.0	39.0	17.0	2.0	45.8	41.8	3.7	**66.0**	26.0	**53.0**	**44.0**	41.9	**41.5**	79.3	37.0	33.8	47.9	–	–	–	–
High-AttentionSpan	**86.0**	**98.0**	32.0	**74.2**	**72.6**	**100.0**	97.0	95.0	65.0	**98.0**	32.0	**75.5**	**74.2**	2.7	61.0	39.0	50.0	42.0	**64.0**	38.7	**84.3**	39.7	**40.7**	**50.8**	**22.7**	**18.3**	**13.0**	**18.0**
Random	67.0	77.0	**44.0**	66.2	63.6	**100.0**	98.0	95.0	**71.0**	77.0	**44.0**	68.2	66.2	2.8	18.0	31.8	40.0	42.0	58.0	31.5	79.2	40.7	39.2	47.6	19.5	13.0	**13.0**	15.2
Low-AttentionSpan	51.0	63.0	19.0	51.2	46.1	**100.0**	**99.0**	**98.0**	47.0	63.0	19.0	52.5	51.2	**8.5**	22.0	**49.8**	28.0	42.0	52.3	36.3	79.0	**43.7**	35.3	48.6	17.2	9.8	7.0	11.3

Table 34: Per-task HELMET scores, 10k tier, seed 456

	Recall					RULER													RAG					LongQA			
Condition	JKV	MK2	MK3	MV	Avg	S1	S2	S3	MK1	MK2	MK3	MQ	MV	CWE	FWE	VT	QA1	QA2	Avg	NQ	TQA	HQA	PQA	Avg	NarrQA	InfQA	InfCh	Avg
Base + YaRN	17.0	17.0	2.0	41.8	19.4	99.0	51.0	57.0	39.0	17.0	2.0	45.8	41.8	3.7	**66.0**	26.0	**53.0**	**44.0**	41.9	30.5	79.5	**40.0**	33.5	45.9	–	–	–	–
High-AttentionSpan	**95.0**	**98.0**	32.0	**74.2**	**74.8**	**100.0**	97.0	95.0	65.0	**98.0**	32.0	**75.5**	**74.2**	2.7	61.0	39.0	50.0	42.0	**64.0**	**33.0**	**83.2**	34.3	**45.2**	**48.9**	**19.3**	**16.7**	20.0	**18.7**
Random	64.0	77.0	**44.0**	66.2	62.8	**100.0**	98.0	95.0	**71.0**	77.0	**44.0**	68.2	66.2	2.8	18.0	31.8	40.0	42.0	58.0	30.2	77.7	36.3	40.2	46.1	12.4	15.0	**21.0**	16.1
Low-AttentionSpan	54.0	63.0	19.0	51.2	46.8	**100.0**	**99.0**	**98.0**	47.0	63.0	19.0	52.5	51.2	**8.5**	22.0	**49.8**	28.0	42.0	52.3	31.3	73.7	33.7	37.0	43.9	17.6	12.2	5.0	11.6

Table 35: Per-task HELMET scores, 30k tier, seed 42

	Recall					RULER														RAG				
Condition	JKV	MK2	MK3	MV	Avg	S1	S2	S3	MK1	MK2	MK3	MQ	MV	CWE	FWE	VT	QA1	QA2	Avg	NQ	TQA	HQA	PQA	Avg
Base + YaRN	23.0	17.0	2.0	41.8	20.9	99.0	51.0	57.0	39.0	17.0	2.0	45.8	41.8	**3.7**	**66.0**	26.0	**53.0**	44.0	41.9	**43.3**	77.2	**44.7**	33.2	49.6
High-AttentionSpan	**82.0**	**87.0**	**66.0**	71.0	**76.5**	**100.0**	**99.0**	98.0	**83.0**	**87.0**	**66.0**	44.2	71.0	3.5	64.3	32.6	45.0	37.0	**63.9**	41.5	78.2	43.3	**39.7**	**50.7**
Random	58.0	68.0	43.0	**72.5**	60.4	**100.0**	96.0	**99.0**	65.0	68.0	43.0	**70.5**	**72.5**	2.5	54.3	19.6	49.0	**45.0**	60.3	42.8	**81.0**	41.3	36.0	50.3
Low-AttentionSpan	45.0	66.0	28.0	55.8	48.7	**100.0**	93.0	93.0	65.0	66.0	28.0	60.0	55.8	2.9	49.0	**33.8**	26.0	38.0	54.7	35.2	78.0	43.3	34.3	47.7

Table 36: Per-task HELMET scores, 30k tier, seed 123

	Recall					RULER														RAG				
Condition	JKV	MK2	MK3	MV	Avg	S1	S2	S3	MK1	MK2	MK3	MQ	MV	CWE	FWE	VT	QA1	QA2	Avg	NQ	TQA	HQA	PQA	Avg
Base + YaRN	16.0	17.0	2.0	41.8	19.2	99.0	51.0	57.0	39.0	17.0	2.0	45.8	41.8	**3.7**	**66.0**	26.0	**53.0**	44.0	41.9	**41.5**	79.3	37.0	33.8	47.9
High-AttentionSpan	**75.0**	**87.0**	**66.0**	71.0	**74.8**	**100.0**	**99.0**	98.0	**83.0**	**87.0**	**66.0**	44.2	71.0	3.5	64.3	32.6	45.0	37.0	**63.9**	34.8	80.0	**46.3**	39.2	50.1
Random	66.0	68.0	43.0	**72.5**	62.4	**100.0**	96.0	**99.0**	65.0	68.0	43.0	**70.5**	**72.5**	2.5	54.3	19.6	49.0	**45.0**	60.3	36.0	**83.5**	46.0	**40.2**	**51.4**
Low-AttentionSpan	49.0	66.0	28.0	55.8	49.7	**100.0**	93.0	93.0	65.0	66.0	28.0	60.0	55.8	2.9	49.0	**33.8**	26.0	38.0	54.6	31.8	83.3	**46.3**	37.2	49.7

Table 37: Per-task HELMET scores, 30k tier, seed 456

	Recall					RULER														RAG				
Condition	JKV	MK2	MK3	MV	Avg	S1	S2	S3	MK1	MK2	MK3	MQ	MV	CWE	FWE	VT	QA1	QA2	Avg	NQ	TQA	HQA	PQA	Avg
Base + YaRN	17.0	17.0	2.0	41.8	19.4	99.0	51.0	57.0	39.0	17.0	2.0	45.8	41.8	**3.7**	**66.0**	26.0	**53.0**	44.0	41.9	30.5	79.5	**40.0**	33.5	45.9
High-AttentionSpan (long)	**79.0**	**87.0**	**66.0**	71.0	**75.8**	**100.0**	**99.0**	98.0	**83.0**	**87.0**	**66.0**	44.2	71.0	3.5	64.3	32.6	45.0	37.0	**63.9**	**31.0**	81.0	39.3	**43.2**	**48.6**
Random	69.0	68.0	43.0	**72.5**	63.1	**100.0**	96.0	**99.0**	65.0	68.0	43.0	**70.5**	**72.5**	2.5	54.3	19.6	49.0	**45.0**	60.3	28.0	**81.7**	33.7	43.0	46.6
Low-AttentionSpan (short)	45.0	66.0	28.0	55.8	48.7	**100.0**	93.0	93.0	65.0	66.0	28.0	60.0	55.8	2.9	49.0	**33.8**	26.0	38.0	54.6	27.3	75.0	35.0	39.2	44.1

P Additional Comparisons with Prior Data Selection Methods

We provide additional analysis of the data-selection baselines considered in Section 4.4. We first discuss the OLMo-3-7B full-parameter comparison in greater detail and then report additional experiments with Llama-2-7B under the smaller-scale LoRA setting.

P.1 Additional Analysis on OLMo-3-7B

The comparison in Table 4 holds the candidate pool, number of selected documents, training recipe, token budget, and evaluation fixed across selection methods. The differences in downstream performance can therefore be attributed to the composition of the data selected by each scoring method rather than differences in training compute or optimization.

The largest separation occurs on retrieval-oriented tasks. In particular, AttentionSpan substantially outperforms both LongAttn and LADM on HELMET-Recall and RULER. This comparison is informative because all three methods use attention as a signal for long-context data selection, but differ in how that signal is constructed. LongAttn restricts scoring to first-layer attention, whereas AttentionSpan incorporates information from deeper layers while explicitly correcting for positional attention sinks. The resulting differences show that attention alone is not sufficient for identifying useful long-context training data; how the attention signal is extracted and summarized is critical.

Finally, Base + YaRN obtains the strongest long-document QA result despite receiving no additional training. This indicates that long-context training does not uniformly improve every capability measured by the evaluation suite. We therefore interpret the long-document QA results primarily as measuring how well the different training mixtures preserve this capability during context-extension training.

P.2 Task-wise breakdown of HELMET and RULER scores of AttentionSpan and other baselines

Table 38: Per-task HELMET scores for the selection-method comparison, seed 42.

	Recall					RULER													RAG					LongQA			
Selection	JKV	MK2	MK3	MV	Avg	S1	S2	S3	MK1	MK2	MK3	MQ	MV	CWE	FWE	VT	QA1	QA2	Avg	NQ	TQA	HQA	PQA	Avg	NarrQA	InfQA	InfCh	Avg
AttentionSpan (ours)	**80.0**	**97.0**	**70.0**	**65.8**	**78.2**	**100.0**	**99.0**	98.0	**64.0**	**97.0**	**70.0**	**68.2**	**65.8**	3.9	**73.7**	39.8	42.0	**46.0**	**66.7**	40.3	81.2	**48.3**	**41.3**	**52.8**	19.8	17.9	34.5	24.1
LADM	28.0	93.0	43.0	64.2	57.1	**100.0**	97.0	98.0	49.0	93.0	43.0	58.0	64.2	6.8	65.3	**46.8**	32.0	40.0	61.0	39.8	74.2	46.0	35.5	48.9	20.5	17.8	**35.5**	24.6
Random	45.0	90.0	29.0	58.5	55.6	99.0	84.0	**99.0**	57.0	90.0	29.0	59.8	58.5	6.6	48.7	35.2	31.0	40.0	56.7	42.5	**81.8**	47.7	39.3	**52.8**	15.3	14.4	30.3	20.0
LongAttn	15.0	81.0	5.0	55.0	39.0	99.0	66.0	92.0	43.0	81.0	5.0	53.5	55.0	**8.3**	53.7	**46.8**	22.0	41.0	51.3	34.7	73.3	43.0	38.2	47.3	15.1	17.2	22.9	18.4
Base + YaRN (no training)	23.0	17.0	2.0	41.8	20.9	99.0	51.0	57.0	39.0	17.0	2.0	45.8	41.8	3.7	66.0	26.0	**53.0**	44.0	41.9	**43.3**	77.2	44.7	33.2	49.6	**25.7**	**24.1**	26.6	**25.5**

Table 39: Per-task HELMET scores for the selection-method comparison, seed 123.

	Recall					RULER													RAG					LongQA			
Selection	JKV	MK2	MK3	MV	Avg	S1	S2	S3	MK1	MK2	MK3	MQ	MV	CWE	FWE	VT	QA1	QA2	Avg	NQ	TQA	HQA	PQA	Avg	NarrQA	InfQA	InfCh	Avg
AttentionSpan (ours)	**80.0**	**97.0**	**70.0**	**65.8**	**78.2**	**100.0**	**99.0**	98.0	**64.0**	**97.0**	**70.0**	**68.2**	**65.8**	3.9	**73.7**	39.8	42.0	**46.0**	**66.7**	37.2	**85.5**	44.0	**40.5**	**51.8**	**26.4**	**22.7**	28.5	25.9
LADM	20.0	93.0	43.0	64.2	55.1	**100.0**	97.0	98.0	49.0	93.0	43.0	58.0	64.2	6.8	65.3	**46.8**	32.0	40.0	61.0	35.3	80.3	**45.3**	35.2	49.0	22.5	20.2	**35.2**	**26.0**
Random	42.0	90.0	29.0	58.5	54.9	99.0	84.0	**99.0**	57.0	90.0	29.0	59.8	58.5	6.6	48.7	35.2	31.0	40.0	56.7	38.8	80.7	43.7	39.2	50.6	23.7	13.1	31.6	22.8
LongAttn	20.0	81.0	5.0	55.0	40.2	99.0	66.0	92.0	43.0	81.0	5.0	53.5	55.0	**8.3**	53.7	**46.8**	22.0	41.0	51.3	29.7	76.7	42.3	37.8	46.6	20.9	16.4	21.8	19.7
Base + YaRN (no training)	16.0	17.0	2.0	41.8	19.2	99.0	51.0	57.0	39.0	17.0	2.0	45.8	41.8	3.7	66.0	26.0	**53.0**	44.0	41.9	**41.5**	79.3	37.0	33.8	47.9	–	–	–	–

Table 40: Per-task HELMET scores for the selection-method comparison, seed 456.

	Recall					RULER													RAG					LongQA			
Selection	JKV	MK2	MK3	MV	Avg	S1	S2	S3	MK1	MK2	MK3	MQ	MV	CWE	FWE	VT	QA1	QA2	Avg	NQ	TQA	HQA	PQA	Avg	NarrQA	InfQA	InfCh	Avg
AttentionSpan (ours)	**83.0**	**97.0**	**70.0**	**65.8**	**78.9**	**100.0**	**99.0**	98.0	**64.0**	**97.0**	**70.0**	**68.2**	**65.8**	3.9	**73.7**	39.8	42.0	**46.0**	**66.7**	**33.5**	**80.5**	36.7	**43.7**	**48.6**	25.9	**24.4**	31.1	27.1
LADM	25.0	93.0	43.0	64.2	56.3	**100.0**	97.0	98.0	49.0	93.0	43.0	58.0	64.2	6.8	65.3	**46.8**	32.0	40.0	61.0	31.2	79.5	34.3	38.2	45.8	**26.8**	23.7	**41.3**	**30.6**
Random	40.0	90.0	29.0	58.5	54.4	99.0	84.0	**99.0**	57.0	90.0	29.0	59.8	58.5	6.6	48.7	35.2	31.0	40.0	56.7	30.8	79.3	36.3	39.2	46.4	19.7	14.2	29.1	21.0
LongAttn	19.0	81.0	5.0	55.0	40.0	99.0	66.0	92.0	43.0	81.0	5.0	53.5	55.0	**8.3**	53.7	**46.8**	22.0	41.0	51.3	28.3	76.8	32.0	39.2	44.1	19.3	14.3	28.0	20.5
Base + YaRN (no training)	17.0	17.0	2.0	41.8	19.4	99.0	51.0	57.0	39.0	17.0	2.0	45.8	41.8	3.7	66.0	26.0	**53.0**	44.0	41.9	30.5	79.5	**40.0**	33.5	45.9	–	–	–	–

P.3 LLAMA-2-7B COMPARISON WITH LONGATTN

We additionally compare AttentionSpan against LongAttn (Wu et al., 2025) under our original Llama-2-7B experimental setting. This experiment provides an additional comparison under a different model family, a smaller data budget, and parameter-efficient fine-tuning. LongAttn's public scorer ranks the same candidate documents, after which the highest-ranked 800 documents are used to train a 16K Llama-2-7B model with the same LoRA configuration as the other conditions.

Table 41 reports category-level and aggregate *HELMET* results. Paired bootstrapping (Efron, 1979) with 5,000 resamples places the overall advantage of domain-balanced *AttentionSpan* over LongAttn at 2.64 percentage points, with the confidence interval excluding zero.

Table 41: Per-category and overall *HELMET* at 16K using Llama-2-7B with LoRA. Each subtask contains 100 samples. Overall is the mean of 22 subtask means; the 95% CI is obtained using paired bootstrap ($B = 5{,}000$). Category counts are shown in parentheses.

Condition	**Recall** (8)	**RAG** (8)	**LongQA** (6)	**Overall [95% CI]**
Random	33.53	48.65	26.25	$37.04_{[35.69,38.35]}$
AS-High-AttentionSpan	45.69	48.40	25.96	$41.29_{[39.90,42.66]}$
LongAttn	45.88	49.65	**26.79**	$42.05_{[40.64,43.44]}$
AS-balanced	**52.59**	**50.65**	26.20	$\mathbf{44.69}_{[\mathbf{43.26},\mathbf{46.10}]}$

The largest improvement over LongAttn occurs on Recall, where domain-balanced *AttentionSpan* reaches 52.59 compared with 45.88 for LongAttn, while the methods remain much closer on RAG and long-document QA. Pure high-AttentionSpan selection performs similarly to LongAttn, whereas balancing the AttentionSpan-selected documents by domain produces the strongest overall result. This indicates that both the attention-based ranking and the composition of the resulting training mixture contribute to downstream performance.

This smaller-scale experiment also provides an additional robustness check for the OLMo-3-7B results. The advantage of AttentionSpan over LongAttn is observed under both Llama-2-7B LoRA and OLMo-3-7B full-parameter training, despite substantial differences in model family, number of selected documents, and fine-tuning regime.

Q CONSENSUS ANALYSIS AND EXPERIMENTAL DETAILS

We provide additional analysis of the consensus experiments from Section 4.3. We first examine agreement between the graders and test whether the benefit of Borda persists under a different set of voters. We then provide the training configuration and composition of the data selected by each grader.

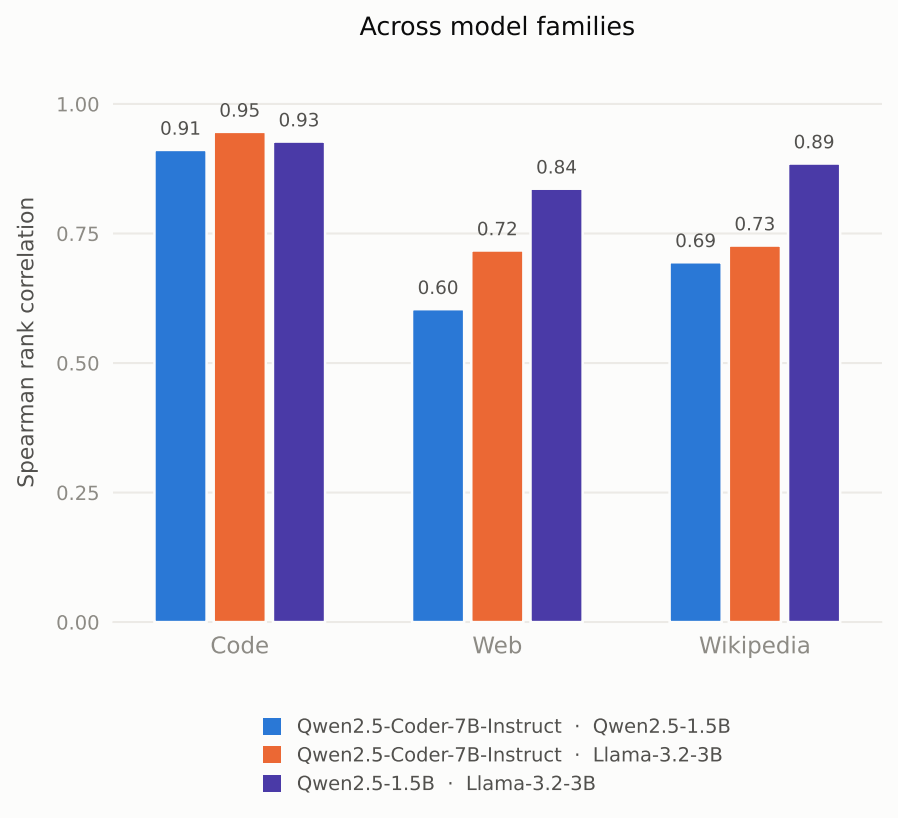


Figure 9: Pairwise Spearman rank correlation between graders on the per-document long-context score, by domain (n=500 per domain).

Q.1 Cross-Grader Agreement

Since Borda operates directly on document rankings, we measure agreement using pairwise Spearman rank correlation. Figure 9 reports correlations over 1,500 held-out documents, with 500 documents each from The Stack, FineWeb, and Wikipedia.

The graders exhibit substantial but non-identical agreement. Qwen-2.5-Coder-7B-Instruct and Llama-3.2-3B have correlations of 0.95, 0.72, and 0.73 across code, web, and Wikipedia, respectively, while Qwen-2.5-Coder-7B-Instruct and Qwen-2.5-1.5B have correlations of 0.91, 0.60, and 0.69. The high correlations indicate that the graders share substantial ranking signal, while the remaining disagreement provides complementary preferences for aggregation. If all graders produced identical rankings, additional voters would not change the Borda ordering. The observed disagreement therefore provides a mechanism for reducing the influence of any single grader's preferences.

Q.2 Robustness to the Choice of Voters

The consensus result in Table 3 uses Qwen-2.5-Coder-7B-Instruct, Qwen-2.5-1.5B, and Llama-3.2-3B. To test whether the improvement depends on this particular set of voters, we replace Qwen-2.5-1.5B with InternLM-2.5-1.8B while keeping the other two graders fixed. This changes one of the model families contributing to the vote while keeping the aggregation rule, candidate pool, selection quota, and training configuration unchanged.

Table 42 shows that the benefit of consensus persists under this substitution. Borda reaches 73.3 on Recall and 66.5 on RULER, compared with the best individual-grader scores of 72.3 and 63.8, respectively. On RULER, the consensus confidence interval lies entirely above those of all individual graders. On Recall, Borda has the highest mean, although its interval overlaps with InternLM-2.5-1.8B.

Table 42: Downstream performance for an alternative Borda voter set in which Qwen-2.5-1.5B is replaced by InternLM-2.5-1.8B. All models are trained on 4,000 documents and 655.35M tokens. Subscripts denote 95% confidence intervals.

Grader Model	Recall	RULER
Llama-3.2-3B	$71.1_{[68.9,\,73.2]}$	$61.8_{[60.8,\,62.8]}$
InternLM-2.5-1.8B	$72.3_{[70.3,\,74.2]}$	$63.2_{[62.2,\,64.3]}$
Qwen-2.5-Coder-7B-Instruct	$65.1_{[62.8,\,67.5]}$	$63.8_{[62.8,\,64.8]}$
Borda (consensus)	$\mathbf{73.3}_{[\mathbf{71.1},\,\mathbf{75.3}]}$	$\mathbf{66.5}_{[\mathbf{65.5},\,\mathbf{67.5}]}$

Table 43: Training hyperparameters shared across the individual-grader and Borda experiments.

Hyperparameter	Value
Base model	OLMo-3-7B (Stage 2, step 47,684)
Base context length	8K
Training context length	32,768
Context extension	YaRN, factor 4
	β_{fast}=32, β_{slow}=1
Number of documents	4,000
Total tokens seen	655.35M
Number of epochs	5
Optimizer	SkipStepAdamW (β_1=0.9, β_2=0.95, ϵ=10^{-8})
Learning rate	2.07×10^{-4}
Weight decay	0.1 (0 on embeddings)
LR scheduler	Linear decay to 0 (α_f=0)
Warmup	16 steps
Global batch size	4,194,304 tokens ($128 \times 32{,}768$)
Gradient accumulation	None
Maximum gradient norm	1.0
z-loss multiplier	10^{-5}
Precision	bf16 parameters, fp32 gradient reduction
Packing strategy	Best fit packing

The magnitude of the gain nevertheless depends on the voters being aggregated. The original voter set improves Recall by 7.3 points over its strongest individual grader, whereas replacing Qwen-2.5-1.5B with InternLM-2.5-1.8B reduces this difference to 1.0 point. Thus, the benefit is not tied to the specific voter set used in the main experiment, but the choice of voters still affects how much is gained from consensus.

Q.3 Consensus Training and Data Details

All individual-grader and Borda experiments use the same candidate pool, training configuration, and token budget. The only difference between runs is the ranking used to select the 4,000 training documents.

Training configuration. Table 43 reports the shared training configuration. We use the OLMo-3-7B Stage-2 checkpoint at step 47,684, whose native 8K context is extended to 32K using YaRN. Each model is trained for five epochs, corresponding to 655.35M tokens.

Data mixture. All selection methods operate on the same candidate pool from `allenai/dolma3_longmino_mix-50B-1025`. The pool contains 30,000 documents spanning scientific PDFs, web data, synthetic long-context data, reasoning traces, code, mathematics, and instruction-tuning data. Table 44 gives the full composition.

Table 45 shows how the different graders change the composition of the selected 4,000 documents. The differences are consistent with the non-identical rankings observed in Figure 9: individual graders

Table 44: Composition of the candidate pool from `allenai/dolma3_longmino_mix-50B-1025`. Braces denote aggregated dataset splits.

Subset	Description	Docs	Share
`olmocr_science_pdfs-high_quality-{topic}-{2e13,2e14,2e15}`	olmOCR scientific PDFs	18,707	62.7%
`common_crawl-high-quality_{19,20}_{topic}`	Filtered Common Crawl	5,383	18.0%
`lc_synth-{yake,cwe}_s2pdf`	Synthetic long-context	1,353	4.5%
`qwq-reasoning-traces`	QwQ traces	822	2.8%
`dolmino-math`	Math	740	2.5%
`r1-reasoning-traces`	R1 traces	726	2.4%
`openthoughts2-reasoning-traces`	OpenThoughts2 traces	525	1.8%
`stack_edu-fim_vigintile_{15,16,17,19}_{lang}`	Code (fill-in-middle)	506	1.7%
`stem-heavy-crawl`	STEM web crawl	493	1.7%
`llama_nemotron-reasoning-traces`	Llama-Nemotron traces	415	1.4%
`tulu-3-sft`	Instruction tuning	153	0.5%
Total		**29,823**	**100%**

Table 45: Composition of the 4,000 documents selected by each grader and Borda consensus from the same 30,000-document candidate pool. Borda (A) uses Qwen-2.5-Coder-7B-Instruct, Qwen-2.5-1.5B, and Llama-3.2-3B; Borda (B) replaces Qwen-2.5-1.5B with InternLM-2.5-1.8B. `unmapped` denotes documents with no entry in the source-mapping index.

Subset	Pool	Coder-7B	Llama-3B	Qwen-1.5B	InternLM-1.8B	Borda (A)	Borda (B)
`olmocr_science_pdfs`	18,707	2,128	2,535	2,204	2,405	2,338	2,399
`common_crawl`	5,383	779	667	1,016	832	804	765
`lc_synth`	1,353	279	301	213	266	280	307
`qwq-reasoning`	822	119	8	0	1	24	16
`dolmino-math`	740	114	147	124	161	134	141
`r1-reasoning`	726	169	23	27	10	58	34
`openthoughts2`	525	108	14	13	5	31	18
`stack_edu-fim`	506	127	185	147	130	177	168
`stem-heavy-crawl`	493	106	71	217	115	109	99
`llama_nemotron`	415	45	1	1	1	1	1
`tulu-3-sft`	153	25	46	36	72	42	50
`unmapped`	177	1	2	2	2	2	2
Total	**30,000**	**4,000**	**4,000**	**4,000**	**4,000**	**4,000**	**4,000**

allocate substantially different portions of their selection budget to several subsets. Borda combines these rankings rather than relying on the preferences of a single grader. We report both the voter set used in the main experiment (Borda A) and the alternative voter set from Section Q.2 (Borda B).

Q.4 Cross-scorer disagreement and the Borda consensus

The 32K AttentionSpan label is a Borda-count consensus over three small attention-scoring models: `coder_7b`, `llama32_3b`, and `qwen_1.5b`. On most documents the three scorers broadly agree. They sometimes disagree dramatically. The example below is the document with the largest cross-scorer rank-percentile spread among the 9,986 graded.

Document: H. G. Wells, *The War That Will End War* (1914) Source: `books` Length: 33,296 tokens

```
Per-scorer evaluation of the same document
(rank percentile within the 9,986-doc 32K pool):

  coder_7b      raw= 101.7      rank   2.2%   (bottom of pool)
  llama32_3b    raw= 175.3      rank  99.9%   (top 0.1% of pool)
  qwen_1.5b     raw= 590.7      rank  25.6%   (bottom quartile)
  ------------------------------------------------------------
  Borda score   = 4,686                       (between the negative
                                                cutoff at borda<2,400
                                                and the positive
    cutoff
                                                at borda>7,700;
                                                NOT selected as
                                                positive or negative
```

```
                                                    for the 32K
    experiment)
```

Two of the three scorers consider this document low long-range; one scorer ranks it as the highest-long-range document in the entire pool. The Borda consensus places it near the median, so it does not enter the positive or negative training subset. This is the design intent of using a multi-scorer consensus rather than a single labeler: idiosyncratic preferences of any one scoring model – here, an unexpectedly strong signal from `llama32_3b` on early-twentieth-century literary prose – are smoothed out, and only documents that all scorers broadly agree on become positive or negative training examples.

R Self-Improvement through Iterative Data Selection

Here, we show a detailed ablation of this self-improvement mechanism.

Round 1: self-selection. We first use the base OLMo-3-7B model to grade the candidate pool and select $4{,}000$ documents for long-context training. We train for five epochs on the selected documents and compare against an otherwise identical run using random selection. As shown in Table 46, self-selected data produces substantially stronger performance on the retrieval-oriented benchmarks. In particular, HELMET-Recall increases from 55.0 under random selection to 72.8 with self-selection, while RULER increases from 56.7 to 62.9. The resulting performance is also comparable to the results obtained when data are selected by external graders, showing that the base model itself can provide a useful signal for selecting data for its own context extension.

Table 46: Round-1 self-selection using OLMo-3-7B as its own grader. We extended the base model (8k context length) to 32k using YaRN with a factor of 4. Under the same training configuration.

Selection	HELMET-Recall	RULER	HELMET-RAG	Long-doc QA
Self-graded	$\mathbf{72.8 \pm 0.5}$	$\mathbf{62.9}$	49.5 ± 1.9	$\mathbf{12.7 \pm 0.7}$
Random	55.0 ± 0.6	56.7	$\mathbf{49.9 \pm 3.3}$	11.2 ± 0.8

Round 2: iterative self-selection. We next test whether this process can be repeated. After Round 1 (R1), we use the resulting checkpoint to grade another set of candidate documents and continue training on the newly selected data in Round 2 (R2). This setting is more challenging because R1 was trained to completion: its learning-rate schedule has already decayed to zero before the second round begins. We therefore experiment with several learning rates for restarting optimization in R2. The results are reported in Table 47.

Table 47: Iterative self-selection in Round 2. R1 denotes the checkpoint obtained after the first round of self-selected training. R2 data are graded using the R1 checkpoint. "Merge" combines the R1 and R2 checkpoints rather than using the R2 checkpoint directly.

Run	Notes	HELMET-Recall	RULER	HELMET-RAG	Long-doc QA
R1	-	72.8 ± 0.5	62.9	49.5 ± 1.9	12.7 ± 0.7
R2 long	2.07×10^{-4}	59.8 ± 0.7	57.7	50.9 ± 1.9	11.4 ± 1.1
R2 random	2.07×10^{-4}	59.8 ± 0.6	62.0	51.9 ± 1.7	11.4 ± 2.2
R2 long	LR $= 2 \times 10^{-5}$	71.6 ± 0.4	64.3	51.6 ± 2.2	14.2 ± 0.4
R2 random	LR $= 2 \times 10^{-5}$	72.3 ± 1.4	65.0	$\mathbf{52.2 \pm 2.1}$	$\mathbf{14.9 \pm 0.5}$
R2 long	LR $= 5 \times 10^{-6}$	73.4 ± 0.7	64.5	52.0 ± 2.4	13.6 ± 0.5
R2 random	LR $= 5 \times 10^{-6}$	71.5 ± 0.8	64.7	51.6 ± 2.3	13.9 ± 0.4
Merged Model	Merge R1 & R2 @ LR $= 2 \times 10^{-5}$	$\mathbf{75.0 \pm 0.6}$	$\mathbf{66.1}$	51.1 ± 2.3	13.8 ± 0.5

The second round reveals an important optimization consideration for iterative self-improvement. Simply restarting R2 with the same learning rate used in R1 substantially degrades the R1 checkpoint. Reducing the R2 learning rate largely avoids this degradation, and the lowest learning rate we test, 5×10^{-6}, improves self-graded Recall from 72.8 after R1 to 73.4 after R2. However, the current

experiments do not establish an optimal second-round training recipe, and additional experiments are needed to understand how the learning-rate schedule should evolve across rounds.

Importantly, the difficulty of directly continuing optimization does not imply that the second round fails to learn useful information. When we merge the R1 checkpoint with the R2 checkpoint trained on self-graded data at 2×10^{-5}, Recall increases from 72.8 to 75.0 and RULER from 62.9 to 66.1. Thus, the R2 update contains useful information even when applying the resulting checkpoint directly does not consistently improve every metric. This suggests that the primary limitation in the current iterative experiment is the optimization strategy across rounds rather than an inability of the updated model to identify useful training data.

Toward interleaved self-grading and training. These results motivate a different strategy for iterative self-improvement. Rather than completing R1 and allowing its learning-rate schedule to decay to zero before beginning another round, data selection could be interleaved with training. An intermediate checkpoint could grade the next portion of the candidate pool while optimization is still in progress, after which training continues on the newly selected data under the same continuous learning-rate schedule. The model would therefore alternate between *training* and *re-grading* rather than treating each round as an independent completed training run.

We do not evaluate this interleaved strategy here, so we leave it as a direction for future work. Nevertheless, the current experiments provide two observations that motivate it: an intermediate model is already capable of producing useful data rankings, and subsequent self-selected training contains useful updates even when restarting optimization after a fully completed first round is suboptimal. If this strategy proves effective, AttentionSpan grading could be performed during long-context training itself, periodically using intermediate checkpoints to refresh the selected training data rather than requiring a separate fixed grader.

S Document examples: high vs. low AttentionSpan

To make the metric concrete, this appendix shows two unedited documents sampled from the pools used to train the 32K Llama-2 conditions (Table 2): one from the high-AttentionSpan ("positive") pool and one from the low-AttentionSpan ("negative") pool. The two pools differ both in surface source mix and, more importantly, in internal structure.

Surface mix. At 32K, the high-AttentionSpan pool of 800 documents is dominated by code and arXiv: only 5 documents (0.6%) come from books, and none come from `govreport` or web. The low-AttentionSpan pool of 800 documents is much more diverse, with one third drawn from books and an additional fifth from `govreport` and web (Table 48).

Table 48: Source mix of the two 32K Llama-2 training pools (800 documents each). The high-AttentionSpan pool is heavily skewed toward code and arXiv; the low-AttentionSpan pool is much more diverse, with books, `govreport`, and web making up a majority of its mass.

Source	High-AttentionSpan pool	Low-AttentionSpan pool
`code`	82.1%	40.2%
`arxiv`	17.2%	7.8%
`books`	0.6%	33.1%
`govreport`	0.0%	8.9%
`web`	0.0%	10.0%

Internal structure. Beyond surface mix, the documents in the two pools have qualitatively different long-range structure. High-AttentionSpan documents tend to introduce entities, equations, or named sections that are recalled tens of thousands of tokens later, so a model that fails to attend across that span cannot follow the document. Low-AttentionSpan documents tend to consist of short, mutually independent items: each item can be read and predicted using only nearby context. The two excerpts below illustrate the contrast.

S.1 High AttentionSpan (positive): an arXiv paper that calls back to itself across ∼30K tokens

The document below is a 33,253-token security paper on the ORide ride-hailing protocol (Borda score 8637, in the top 5% of the 32K pool; "Borda score" is the per-document AttentionSpan label, computed by Borda-count rank aggregation over four independent attention-scoring models – see Section Q.4). The paper introduces an attack section near the start, then, roughly 30K tokens later, the conclusion explicitly recalls that section in order to support its closing claim. The matching identifier `sec:cryptanalysis` appears in both windows; the second window cannot be understood without the first.

Source: `arxiv` `doc_idx=908` AttentionSpan: **8637** (high) Length: 33,253 tokens

```
[char ~1,600 / 124,000, token position ~430]

\section{Analysis of ORide Protocol}
\label{sec:cryptanalysis}                              ← defined here
In this section we briefly recall the ORide protocol of \cite{
    ORidePaper},
followed by a security analysis of the protocol at the rider's end.
We then describe our attack that would allow a rider to predict a
    driver's
location with good accuracy, and present the results of practical
    experiments.

[~30K tokens of construction, threat model, and experiments omitted]

[char ~101,700 / 124,000, token position ~30,100]

Although protocols may seem secure in theory, there may arise several
complications and vulnerabilities when they are deployed practically,
as demonstrated by our attack in Section \ref{sec:cryptanalysis}.
    ← recalled
    here
In the future it will be interesting to experimentally investigate
    the
notion of driver privacy with respect to both the SP and rider in
    more
recent works following ORide (\textit{lpRide}, \textit{pRide}).
```

The two windows are separated by roughly 30,000 tokens of construction, threat-model, and experimental detail. A model whose attention collapses to the last few thousand tokens at 32K context cannot connect the closing sentence to the section it points to, and therefore cannot complete the conclusion in a way consistent with the rest of the paper.

S.2 Low AttentionSpan (negative): a periodical of mutually independent items

The document below is a 29,219-token issue of *Buchanan's Journal of Man* (December 1887, Project Gutenberg; Borda score 252, in the bottom 1% of the 32K pool). The issue begins with a fixed table of contents and is then a sequence of short, topically unrelated items separated by ornamental rule lines. No item refers to any other item in the document, so a model needs only local context to predict each token.

Source: `books` `doc_idx=1069` AttentionSpan: **252** (low) Length: 29,219 tokens

```
[char 0, token position 0]

                          BUCHANAN'S
                       JOURNAL OF MAN.
          VOL. I.      DECEMBER, 1887.        NO. 11.

CONTENTS OF JOURNAL OF MAN.
```

```
  The World's Neglected or Forgotten Leaders and Pioneers
  Social Conditions -- Expenses at Harvard; European Wages; India as
    a
    Wheat Producer; Increase of Insanity; Temperance; Flamboyant
    Animalism
  Transcendental Hash
  Just Criticism
  Progress of discovery and Improvement -- Autotelegraphy; Edison's
    Phonograph; Type-setting Eclipsed; ...
  Life and Death -- Marvellous Examples
  Outlines of Anthropology (continued) Chapter X.

[~7K tokens on Leif Ericson, Norse colonies, and Harvard expenses
    omitted]

[char ~28,500 / 111,200, token position ~7,500]

EUROPEAN WAGES. -- Senator Frye said in a public address in Boston:
"I say from all my observations made there, ... there is only one
country in Europe that comes within half of our wages, and that is
England, and the rest are not one-third ... of our wages."
    ←one-paragraph
    item

INDIA AS A WHEAT PRODUCER. -- "Consul-General Bonham says she is a
dangerous competitor of the United States. The report of Consul-
    General
Bonham at Calcutta, British India, treats at length of the wheat
interests of that country. The area devoted to wheat in 1886 was
about 27,500,000 acres, and the total yield 289,000,000 bushels. ..."
      ←unrelated to
    neighbors

INCREASE OF INSANITY. -- The London Lancet reports that ...
     ←unrelated to
    neighbors
```

The three items in the second window are unrelated to each other and unrelated to the document's earlier articles on Leif Ericson and on Harvard. Within each item, the local paragraph is sufficient to predict every following token, so attending across the 32K window cannot help a model predict this document better.

What the contrast captures. The positive document is a single coherent argument that spans its full length: an attack is named, constructed, evaluated, and then explicitly recalled in the closing discussion. The negative document is, in effect, a sequence of much shorter documents that happen to share a binding. AttentionSpan ranks the first kind of document above the second; the 32K Llama-2 experiment (Table 2) asks whether training on documents of the first kind, rather than on randomly sampled documents (which contain a mix of both), improves a model's ability to use long context.

S.3 Effect of training data on downstream output

The two excerpts above characterize the training data. The example below shows what changes downstream when the same Llama-2-7B base is fine-tuned (LoRA) on each of the three pools and given the same long-context query. Across the 600 `kilt_nq` queries in HELMET at 16K context, the positive-trained model gives the correct answer on 45.2% of queries (substring match), the random-trained model on 41.3%, and the negative-trained model on 41.7%. The query below is one of 30 in which the positive-trained model uniquely produced the correct answer while both other conditions failed.

Task: `kilt_nq` (HELMET, 16K context) Query #23 of 600 Gold answer: **Christina Perri**

```
Use the given documents to write a concise and short answer to the
question. Write your answer in the following format:
Answer: [answer]

Document (Title: Vitamin D): is done in the liver and in the kidneys.
As vitamin D can be synthesized in adequate amounts by most mammals
exposed to sufficient sunlight, it is not an essential dietary factor
    ,
and so not technically a vitamin ...
    ← 1st of 6 retrieved
    docs

[5 more retrieved documents totaling ~16K tokens omitted]

Document (Title: A Thousand Years (Christina Perri song)):
is in the key of B-flat major, with a vocal range of F to C.
The song was written about the love affair between the characters
Edward and Bella from "The Twilight Saga: Breaking Dawn -- Part 1".
Music video. An official lyric video of the song was premiered on
October 17, 2011 via Perri's official Facebook and Twitter pages ...
     ← answer is in this
    document

Question: who sings ive loved you for a thousand years
Answer:
```

Llama-2-7B + LoRA trained on	Generated answer	Correct?
high-AttentionSpan (positive) pool	`Christina Perri`	**yes**
random pool	`Vanessa Carlton`	no (different artist)
low-AttentionSpan (negative) pool	`ive loved you for a thousand years`	no (echoes the question)

The three failure modes are diagnostic. The positive-trained model retrieves the correct entity from the relevant document near the end of the 16K window. The random-trained model returns a plausible-sounding but wrong artist, suggesting it produced an answer from a less directly relevant document earlier in the window. The negative-trained model echoes the question instead of producing a name, the canonical failure of a model that has not learned to attend across the prompt to a supporting document.