BEYOND LENGTH: QUANTIFYING LONG-RANGE INFORMATION FOR LONG-CONTEXT LLM PRETRAINING DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-context language models unlock advanced capabilities in reasoning, code generation, and document summarization by leveraging dependencies across extended spans of text. However, much readily available long-text data does not genuinely require extended context, as most spans can be predicted with only short-range context while only a small fraction truly depends on long-distance dependencies, making it important to identify and select training data with stronger long-context dependencies. Therefore, we introduce LongFilter, a framework for curating training data tailored to long-context pretraining. LongFilter measures the information gain provided by extended context by contrasting model predictions under long-context versus short-context settings, thereby identifying samples where long-range dependencies are essential. Experiments with LLaMA-3-8B, extending its context length from 8K to 64K, show that LongFilter efficiently selects high-quality data and yields substantial improvements on benchmarks such as HELMET, LongBench, and RULER. Moreover, our analyses further confirm that different types of text segments vary in their reliance on extended context, highlighting which data truly benefits from long-context modeling.

1 INTRODUCTION

Modern large language models have shown remarkable capabilities when processing short spans of text, but many real-world tasks, such as reasoning across documents, generating long codebases, or summarizing entire chapters—require understanding and integrating information over much longer contexts. To enable these capabilities, models are typically first trained on standard short-context corpora and then further pre-trained on long-context data, which activates their long-context reasoning abilities. Recent techniques, including modifications to RoPE (Su et al., 2024) and attention interpolation (Peng et al., 2023; Ding et al., 2024), can accelerate this process and reduce overall training costs.

While existing methods improve long-context pretraining efficiency, the quality of long-context training data remains a critical factor for unlocking a model's long-context abilities. Current data engineering approaches primarily focus on sequence length, for example by increasing the proportion of long sequences in the training set (Fu et al., 2024; Abdin et al., 2024; Yang et al., 2025) or adjusting the ratio between long and short sequences (Gao et al., 2024). However, relying solely on sequence length cannot distinguish truly long-context-dependent data from long sequences that largely consist of repetitions, independent contexts, or tokens predictable from short preceding spans. Consequently, a substantial portion of long sequences in widely used corpora does not require extended context, and even high-quality data may be better suited for short-text training rather than long-text pretraining.

For instance, consider books and poetry collections. Individual poems, even by the same author, often lack inter-poem dependencies, and their relatively short length makes them suitable for short-context models. In contrast, textbooks are more appropriate for long-context pretraining, as chapters are tightly interconnected and understanding one chapter often requires access to preceding chapters. These examples illustrate that not all long sequences provide meaningful long-context information, and including sequences that do not require extended context can dilute the training signal.

Existing long-text pretraining strategies can be viewed as a “0-to-1” step: by increasing the proportion of long sequences, the model begins to learn long-context dependencies. However, because the training loss is averaged over all tokens, sequences that do not truly depend on long-range context contribute equally to the learning signal, which is suboptimal. In this work, we take a “1-to-2” step by further increasing the proportion of sequences that genuinely require long-context understanding. This approach assigns more learning signal to tokens that depend on extended context, improving the efficiency of long-context pretraining and enabling the model to better leverage long-range dependencies.

To distinguish truly useful long-context data from merely long-length sequences, we propose LongFilter, a data selection framework for continued pre-training. Our method is founded on a simple yet powerful principle: data is valuable for long-context training only if the long context actually helps the model make better predictions. We operationalize this insight by developing a scoring function to quantify this “information gain.” The score’s formulation is derived from the Kullback-Leibler (KL) divergence between a model’s next-token prediction distributions conditioned on a long versus a short context. A high score signifies that the extended context provides crucial information, making the sequence a high-quality candidate for training.

Contributions

1. This paper suggests that long-context continued pretraining should be conducted on data whose extended context provides additional information for next-token prediction.
2. We propose LongFilter, a data curation method that quantifies the information gain provided by an extended context. Using a transformer-based causal language model, LongFilter efficiently scores and selects high-quality long-context pre-training data.
3. Extensive experiments show that, without modifying the model or training setup, simply selecting training data with richer long-range information can substantially improve a language model’s long-text processing ability during continued pre-training. Models trained on LongFilter-selected data achieved average gains of over 2 points on benchmarks including HELMET, LongBench, and RULER.

2 Related Work

2.1 Long-Context Language Model Pretraining

Long-context language models have garnered significant attention within the community in recent years due to their high practical value in applications such as code generation and reasoning. A current mainstream approach involves extending the context of an existing language model with short-term context. On top of this, certain techniques have been developed to reduce the amount of training required. For example, some works employ position interpolation (Chen et al., 2023; Peng et al., 2023; Bertsch et al., 2023; Ding et al., 2024; Liu et al., 2024b; Zhang et al., 2024; Zhu et al., 2024) on RoPE (Su et al., 2024) to enable the model to better adapt to the positional encoding of extended context, or manipulating attention module (Xiong et al., 2025; Jin et al., 2024; Bertsch et al., 2023). Some of these methods have been applied in certain enterprise-level models (Liu et al., 2024a; Yang et al., 2025).

2.2 Data Curation and Filtering for Language Model Pretraining

The quality of data exerts a direct influence on the performance of language models. This has become a standard process for enterprise-level langauge models (Gunasekar et al., 2023; Abdin et al., 2024; Abouelenin et al., 2025). Typically, this complex process involves multiple steps, including heuristic approaches (Gao et al., 2020; Laurençon et al., 2022; Rae et al., 2021), data quality classification (Longpre et al., 2024; Wettig et al., 2024; Xie et al., 2023), domain-specific selection (Feng et al., 2022), deduplication (Borgeaud et al., 2022; Abbas et al., 2023), multilingual filtering (Wenzek et al., 2019), removing toxic content (Penedo et al., 2023; Jansen et al., 2022). These methods have achieved tremendous success in short-context model pretraining, yet few of them are specifically designed for long-context data.

2.3 DATA ENGINEERING FOR LONG-CONTEXT PRETRAINING

Existing data engineering approaches for long-context pretraining primarily focus on the length of training data, specifically by adjusting data proportions to increase the proportion of longer-length training examples within the text corpus (Abdin et al., 2024; Yang et al., 2025). Fu et al. (2024) recommends increasing the proportion of data with longer length while maintaining domain balance. Gao et al. (2024) investigated the impact of the ratio of long-to-short data mixing and the data source on the performance of long-text pretraining. A similar idea to this paper is LongWanjuan (Liu et al., 2024c), which proposes several metrics to measure the quality of long text data. However, most of its metrics are also applicable to short texts, and the context length of the model-based filtering method used in its paper is too short (the longest window in its paper is as short as the short windows in this paper). Another related approach, LongAttn (Wu et al., 2025), uses attention scores to model long-range dependencies, but studies have shown that these attention scores do not reliably capture token importance.

3 METHODOLOGY

Our method, LongFilter, is designed to identify and select training data where long-range dependencies are semantically meaningful and essential for accurate token prediction. The core insight is to quantify the "information gain" provided by an extended context. We formalize this gain as the Kullback-Leibler (KL) divergence between the predictive distributions of a language model given a long context versus a short one. Based on this principle, our framework follows a three-step pipeline: (1) score each data instance for its long-context informational value using our proposed metric, (2) rank the instances by this score, and (3) select a high-scoring subset for continued pre-training.

3.1 EVALUATING THE INFORMATION CONTRIBUTION OF EXTENDED CONTEXT

We operate within the standard causal language modeling framework, where the objective is to predict the next token x_t given a preceding context $x_{<t}$.

Let a sequence of tokens be denoted by $X = (x_1, x_2, \dots, x_N)$. For any given token x_t in the sequence, we define two distinct context windows:

- **Short Context** (S): The sequence of ℓ_{Short} tokens immediately preceding x_t. Formally, $S(t) = (x_{t-\ell_{\text{Short}}}, \dots, x_{t-1})$.
- **Long Context** (L): The sequence of ℓ_{Long} tokens immediately preceding x_t, where $\ell_{\text{Long}} > \ell_{\text{Short}}$. Formally, $L(t) = (x_{t-\ell_{\text{Long}}}, \dots, x_{t-1})$.

The **extended context**, denoted E, is the portion of the long context that precedes the short context, i.e., $E(t) = (x_{t-l_l}, \dots, x_{t-l_s})$. The long context is therefore the concatenation of the extended and short contexts, $L = E \circ S$. See Figure 1.

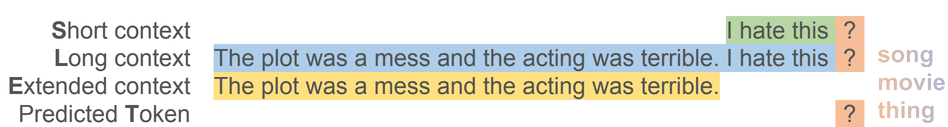


Figure 1: An illustration of the token-level long-context information gain. Given only the Short Context (S) "I hate this", the predictive distribution for the next token has high entropy, as many words ('song', 'thing', 'movie') are plausible. The Extended Context (E), "The plot was a mess...", provides critical information that reduces this entropy, concentrating the probability on "movie".

Given a pre-trained language model M, we can obtain two conditional probability distributions for the next token:

$$P_{\text{short}}(\cdot) = P_M(\cdot \mid S(t)) \quad \text{and} \quad P_{\text{long}}(\cdot) = P_M(\cdot \mid L(t))$$

The central question LongFilter addresses is:

How can we quantify the additional information that the extended context E provides for predicting x_t beyond what is already available in the short context S?

3.2 Information-Theoretic Formulation of Contextual Gain

The ideal theoretical tool to answer our question is *Conditional Mutual Information (CMI)*. The CMI $I(T; E \mid S)$ measures the reduction in uncertainty about a target variable T (the next token) after observing an extended context E, given that a short context S is already known (Cover & Thomas, 2006).

The CMI can be expressed in two well-known, equivalent forms. The first defines CMI as the reduction in conditional entropy:

$$I(T; E \mid S) = H(T \mid S) - H(T \mid S, E) \tag{1}$$

where $H(\cdot \mid \cdot)$ is the conditional entropy. A second, equivalent formulation expresses the CMI as the expected Kullback-Leibler (KL) divergence between the predictive distributions with and without the extended context:

$$I(T; E \mid S) = \mathbb{E}_{p(s,e)} \left[D_{KL}\big(p(T \mid S = s, E = e) \parallel p(T \mid S = s)\big) \right] \tag{2}$$

This second form is particularly insightful, as it frames the information gain as the expected "distance" between the posterior belief $p(T \mid S, E)$ and the prior belief $p(T \mid S)$. For completeness, we derive the equivalence of these two definitions in Appendix C.

For a given context instance (e^*, s^*), to evaluate the effect of the extended context e^* on the next token T prediction, we consider the one sample estimate of the above CMI:

$$\hat{I}(T; E = e^* \mid S = s^*) = D_{KL}\big(p(T \mid S = s^*, E = e^*) \parallel p(T \mid S = s^*)\big) \tag{3}$$

3.3 A Practical Scoring Function for Contextual Gain

Expanding the KL divergence by its definition in equation 3, we have

$$D_{KL}\big(p(T \mid S = s^*, E = e^*) \parallel p(T \mid S = s^*)\big) = \sum_{t \in \mathcal{V}} p(t \mid s^*, e^*) \log \frac{p(t \mid s^*, e^*)}{p(t \mid s^*)}. \tag{4}$$

This formula has two drawbacks: does not leverage the ground-truth information of $T = t^*$, i.e., the value of $D_{KL}\big(p(T \mid S = s^*, E = e^*) \parallel p(T \mid S = s^*)\big)$ does not depend on t^* and it requires a costly summation over the entire vocabulary $\mathcal{V}$. To create a practical score for a single ground-truth instance (t^*, s^*, e^*), we focus on the term corresponding to t^*, which yields a surrogate for KL divergence:

$$\text{score}(t^*, s^*, e^*) = p(T = t^* \mid E = e^*, S = s^*) \log \frac{p(T = t^* \mid E = e^*, S = s^*)}{p(T = t^* \mid S = s^*)} \tag{5}$$

This score can be interpreted as the gain for predicting the specific target t^* that is contributed by the extended context e^*, given that s^* was already observed. A positive value indicates that e^* made the correct target s^* more likely, while a negative value indicates it was made less likely.

To score an entire document $X^* = (x_1^*, \ldots, x_N^*)$, we average the per-token scores defined in equation 5. The final *LongFilter score* is

$$\begin{aligned} \text{Score}(X^*) &= \frac{1}{N} \sum_{i=1}^{N} \text{score}(x^*_{i-\ell_{\text{Long}}:i-\ell_{\text{Short}}-1}, x^*_{i-\ell_{\text{Short}}:i-1}, x_i^*) \\ &= \frac{1}{N} \sum_{i=1}^{N} p(x_i^* \mid x^*_{i-\ell_{\text{Long}}:i-1}) \log \frac{p(x_i^* \mid x^*_{i-\ell_{\text{Long}}:i-1})}{p(x_i^* \mid x^*_{i-\ell_{\text{Short}}:i-1})} \end{aligned} \tag{6}$$

For a more practical perspective, we can reformulate the LongFilter score in terms of the standard per-token cross-entropy loss, which is equivalent to the negative log-likelihood. Let $\mathcal{L}^{\text{long}}$ and $\mathcal{L}^{\text{short}}$ be the losses for predicting the ground-truth token x_i^* given the long and short contexts, respectively:

$$\begin{aligned} \mathcal{L}_i^{\text{long}} &= H_c\left(\mathbf{1}\{x_i = x_i^*\}, p(\cdot \mid x^*_{i-\ell_{\text{Long}}:i-1})\right) = -\log p(x_i^* \mid x^*_{i-\ell_{\text{Long}}:i-1}), \\ \mathcal{L}_i^{\text{short}} &= H_c\left(\mathbf{1}\{x_i = x_i^*\}, p(\cdot \mid x^*_{i-\ell_{\text{Short}}:i-1})\right) = -\log p(x_i^* \mid x^*_{i-\ell_{\text{Short}}:i-1}), \end{aligned}$$

where $H_c(p, q) = -\mathbb{E}_p \log q$ denotes the cross-entropy of the distribution q relative to p.

Then we have

$$\text{Score}(X^*) = \frac{1}{N}\sum_{i=1}^{N} \exp(-\mathcal{L}_i^{\text{long}})(\mathcal{L}_i^{\text{short}} - \mathcal{L}_i^{\text{long}}). \tag{7}$$

This loss-based view offers a clear interpretation: the score gives preference to examples where the reduction in prediction loss from using a longer context (the term $\mathcal{L}^{\text{short}} - \mathcal{L}^{\text{long}}$) is large. This loss reduction is then weighted by the model's confidence on the token given the full context $(\exp(-\mathcal{L}^{\text{long}}) = p(x_i^* \mid x_{i-\ell_{\text{Long}}:i-1}^*))$, ensuring that the gains are on tokens the model considers plausible.

3.4 LongFilter

The framework of LongFilter is shown in Figure 2. LongFilter utilizes a pre-trained causal language model to estimate the distribution of the next token across varying context lengths. LongFilter consists of three steps: Long-context Modeling, Short-Context Modeling, and LongFilter Scoring.

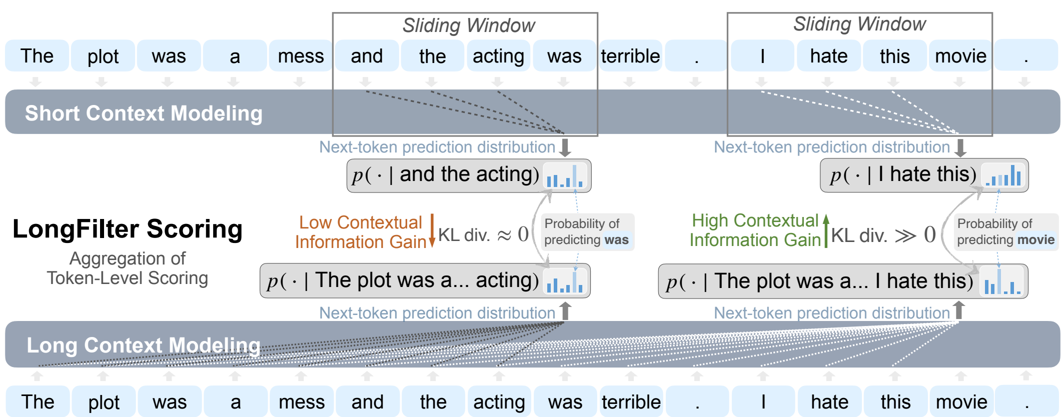


Figure 2: The **Upper part** computes the next-token probability distribution using a short-context sliding window (shown as 4 tokens for illustration, though our experiments use 4K), while the **Lower part** computes it using the full long context. LongFilter then scores the information gain (**Middle part**) by calculating a token-level surrogate KL divergence between these two distributions. This gain is low for locally predictable tokens (such as 'was'), but high for tokens that require extended context (such as 'movie'). Finally, these token-level scores are aggregated to produce a single score for the entire data instance.

Long-Context Modeling For an input sequence, we compute the probability distribution obtained by predicting the next token for each position based on its prefix context, in a manner analogous to the training stage of causal language model. The process of getting the next token distribution in the long context utilizes the prefix context from all positions.

Short-Context Modeling For predicting the distribution of the next token in a short context, the LongFilter first segments the entire text into shorter chunks. Each short chunk is then fed into a pre-trained causal language model, thereby constraining the context of the predicted output to the boundaries of the short chunk. To avoid predicting tokens with insufficient context at the beginning of each short chunk, we chose to introduce overlap between different chunks during segmentation.

LongFilter Scoring After obtaining the probabilities for predicting the next token from both short and long contexts, the final score is calculated using Equation 6. All scores are sorted, and a portion of the higher-scoring entries are selected as the chosen data.

4 EXPERIMENTS

4.1 SETUP

We conduct experiments by continually pre-training LLaMA-3-8B (Dubey et al., 2024), which has an initial effective context length of 8K, on different training datasets to extend its context length to 64K. For each dataset, we pack the tokenized text into sequences of 64K tokens and score each sample with LongFilter. The model is then continually pre-trained on these filtered samples, and its performance is evaluated on long-context benchmarks to assess the effectiveness of the data selection process. As this study focuses on the pre-training stage, our evaluation targets fundamental long-context capabilities; we exclude reasoning and open-ended generation tasks, which typically necessitate post-training.

Datasets We use SlimPajama-627B (Soboleva et al., 2023) as the primary source for long-context pretraining. LongFilter is applied to extract high-quality long-text samples from this dataset. SlimPajama has been widely adopted in recent long-context data engineering works (e.g., Gao et al. (2024), Fu et al. (2024)). Specifically, we select three corpora from SlimPajama, that is ArXiv, Books, and CommonCrawl, for our experiments. We similarly categorized each corpus by length, selecting thresholds of 16K, 64K, and 32K for long and short texts in ArXiv, Book, and CommonCrawl, respectively. After applying these thresholds, the volume of data classified as long texts was approximately 19 billion tokens. We constructed the model training dataset with 80% long texts and 20% short texts. Data selection was applied exclusively to the long text portion.

Table 1: Number of Long Context Tokens

Name	**#Tokens**
SlimPajama-Book	19,535,822,848
SlimPajama-Arxiv	19,489,295,000
SlimPajama-CommonCrawl	19,284,099,072

Model and Training Configuration For training, we adopt the same configurations as ProLong (Gao et al., 2024) when scaling LLaMA-3-8B from an 8K to a 64K context, including optimizer, learning rate, and RoPE base frequency. The only difference lies in our choice of training data, which is guided by LongFilter-based selection. Apart from increasing the RoPE base frequency from 5×10^5 to 8×10^6, we made no further modifications to the model in our experiments. Drawing on configurations from previous studies on long text training (Fu et al., 2024), we set the batch size to 4M tokens and trained for 1,000 steps, processing a total of 4B tokens.

Baselines We compare our method against two baselines. We first compared our model with ProLong (Gao et al., 2024), but unified the training data to three corpora from the SlimPajama dataset and adopted the same short-to-long ratio. For a fair comparison, we did not use ProLong's ShortMix dataset. ProLong's training data was sampled from all training data, while LongFilter's training data was sampled from the selected data. We did not exclude the selected data from ProLong's training set, meaning ProLong and LongFilter share a portion of high-quality long-context training data. We also compared our approach with LongWanjuan (Liu et al., 2024c), conducting comparative experiments using their best-performing aggregated and holistic data ratio of 1:1 as specified in their paper.

Setting of LongFilter We set the short context window to 4K and the long context window to 64K, using the Llama-3.1-8B model (which supports 128K contexts) for scoring. We sorted the scores and selected the top 20% of data as the final training dataset of LongFilter. We run LongFilter on 32 NVIDIA H100 GPUs, enabling each corpus to complete all scoring within a single day.

4.2 EVALUATION ON RECALL (NEEDLE-IN-A-HAYSTACK) TASKS

We first report the performance of different data strategies on a series of Recall tasks. This series of tasks has also been referred to as Needle-in-a-Haystack (NIAH) (Kamradt, 2023). This type of tasks directly tests a model's ability to utilize information from any position, often serving as one of the most important metrics for evaluating a model's performance on long text.

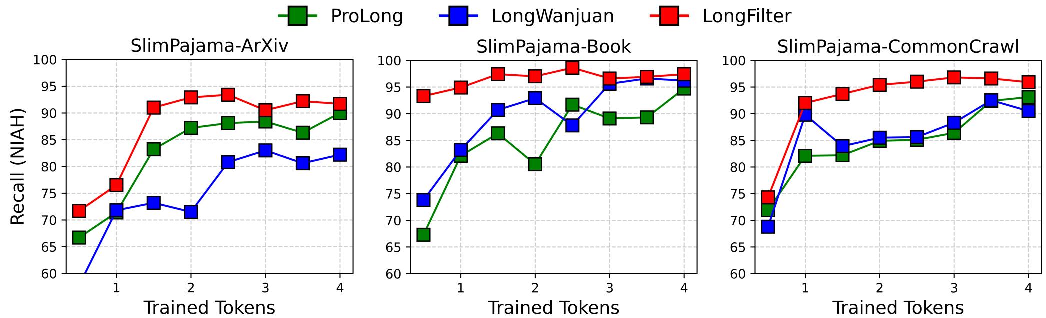


Figure 3: Performance on Recall tasks (Needle-in-a-Haystack) w.r.t trained tokens

Specifically, we reported on the SubEM of Recall task within the HELMET benchmark (Yen et al., 2025), which encompasses four distinct NIAH tasks: JsonKV, Needle retrieval with multiple keys, UUID retrieval with multiple keys, and value retrieval with multiple keys. Experimental results are shown in Figure 3.

Across the three experimental settings with different type of training data, we consistently observe that LongFilter achieves the best overall performance and exhibits the most stable improvement as the training scale increases. For example, in all three groups, LongFilter rapidly surpasses both ProLong and LongWanjuan at small scales (0.5B–1B) and maintains a clear advantage when scaling up to 4B, reaching performance above 90 in every case. This indicates that filtering with LongFilter can effectively enhance data quality and maximize the benefits of scaling.

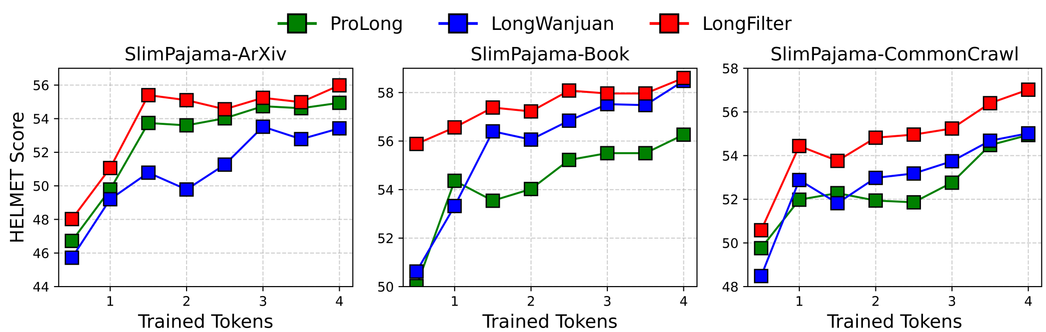


Figure 4: Performance on HELMET w.r.t trained tokens

4.3 Evaluation on Long-Context Benchmarks

To validate whether the data selection strategy of LongFilter is beneficial for training long-text language models, we evaluate the continually pre-trained models on 3 widely used long-context benchmarks: HELMET (Yen et al., 2025), LongBench (Bai et al., 2024), and RULER (Hsieh et al., 2024). Since RULER and LongBench require language models to comprehend instructions, we SFT models with 1B data using UltraChat dataset (with settings consistent with Gao et al. (2024)).

The final reported score is the average of all non-model-based evaluation metrics across all tasks in HELMET, encompassing five tasks: Recall, RAG, Re-rank, ICL, and QA. We report the overall performance on HELMET benchmark with respect to the number of trained tokens in Figure 4.

According to the experimental results, the quality of long-text training data undergoes a clear, significant, and sustained improvement after LongFilter's data selection. LongFilter significantly improves training efficiency. Compared to unfiltered data, length extension training with 1.5B filtered tokens already achieves performance comparable to training on 3-4B tokens, indicating that only about half the data is required to reach the same level of effectiveness.

Table 2: Experimental Result on LongBench

Dataset & Model	SingleQA	MultiQA	Summ	ICL	Synthetic	Code	Overall
Arxiv							
ProLong	25.26	16.24	24.83	63.04	51.76	68.69	38.58
LongWanjuan	28.43	18.25	25.09	62.85	49.92	69.22	39.36
LongFilter	28.32	17.99	24.48	63.15	51.41	70.00	**39.52**
Book							
ProLong	21.25	15.06	23.64	62.60	52.41	69.76	37.47
LongWanjuan	22.33	14.84	22.75	63.46	52.62	70.05	37.69
LongFilter	26.58	20.14	24.36	62.69	53.58	70.08	**39.81**
CC							
ProLong	21.95	16.83	25.03	63.87	52.18	69.33	38.37
LongWanjuan	33.64	18.58	25.18	61.95	47.87	69.79	40.02
LongFilter	30.54	17.21	25.58	62.84	57.02	69.11	**40.66**

On LongBench, the three methods demonstrate different trade-offs across datasets. LongFilter consistently achieves the highest overall scores, showing its robustness across diverse domains. In particular, it yields notable gains in Synthetic tasks (e.g., 57.02 on CC) and maintains competitive performance in Code, where both categories heavily rely on the model's ability to leverage information from arbitrary positions within the context. This suggests that LongFilter effectively improves the long-range information of training data, thereby benefiting tasks needs long-range dependency.

Table 3: Experimental Result on RULER

Dataset & Model	NIAH Single	NIAH MultiKey	NIAH MultiValue	NIAH MultiQuery	Other	Overall
Arxiv						
ProLong	77.90	85.40	89.35	85.73	47.12	69.28
LongWanjuan	83.8	80.57	83.16	85.04	49.15	69.69
LongFilter	78.68	86.20	89.76	86.92	48.07	**70.13**
Book						
ProLong	93.83	83.80	92.76	95.08	46.56	73.35
LongWanjuan	90.08	91.03	91.38	82.75	48.23	73.74
LongFilter	95.33	97.87	93.15	80.10	54.71	**78.95**
CommonCrawl						
ProLong	91.31	86.58	94.32	77.87	47.54	72.59
LongWanjuan	90.85	84.65	89.76	80.49	41.25	74.08
LongFilter	92.58	94.50	94.80	77.80	31.71	**75.37**

On the RULER benchmark, LongFilter consistently achieves the highest overall scores across all three datasets. Its advantage is particularly pronounced in structured data tasks, such as MultiKey, MultiValue, and MultiQuery, where careful filtering likely enhances the model's ability to capture long-range information.

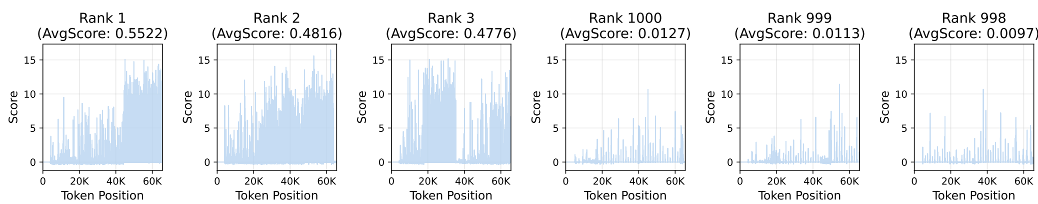
Overall, these results reinforce the pattern observed in LongBench: data quality and filtering (LongFilter) provide more consistent and robust improvements than original training data and LongWanjuan.

4.4 Case Study: Token-Level Analysis

In this case study, we analyze a subset of the processed SlimPajama-Arxiv dataset containing 1000 samples, each a sequence of 65536 tokens. The analysis is presented in Figure 5. The token-level score is visualized by color intensity in Figure 5a (prose) and Figure 5b (code), where darker text indicates a higher score. These results support the intuition that repetitive content like TikZ code, which lacks long-range semantic structure, receives low scores. Figure 5c shows the token-level scores for the top-three and bottom-three ranked documents. The abrupt score jumps seen in plots of Rank 1 and 3 are artifacts created by concatenating multiple `.tex` files from a single arXiv submission during data preprocessing.

(a) High-score text segment: Thesis

(b) Low-score text segment: TikZ codes



(c) Context Score at each token position for documents of varying ranks.

Figure 5: Token-level context score analysis on a subset of the processed SlimPajama-Arxiv dataset. In the top examples, the color of each token is determined by its score: the darker the color, the higher the score. (a) A high-scoring segment of well-formed academic prose from a PhD thesis. (b) A low-scoring segment containing non-prose LaTeX TikZ drawing commands. (c) Context Scores across the full token sequence for documents of the top three ranks and the bottom three ranks.

5 Conclusion

This paper proposes a data filtering framework tailored for pretraining long-context language models. Unlike short-context language models, long-context models require leveraging semantic information from longer range of positions. Based on intuition, we recommend that long-context language models should be trained on data where this additional length provides information for the next word prediction.

We formalize this process as identifying training data where additional context yields higher conditional mutual information for predicting the next token. Based on this formulation, we develop a scoring function that estimates the informational gain of context using a trained language model. To apply this method to practical data filtering, we design a model called LongFilter to score the informational value of additional context in long training data, recommending training on data with higher scores.

Sufficient experimental resutls demonstrates the effectiveness of LongFilter. We achieve sustained and significant improvements in long-text capabilities for long-text models solely through data filtering. After expanding the Llama-3-8B model from 8K to 64K context, experiments on benchmarks like HELMET, LongBench, and RULER demonstrate that this simple yet effective method yields up to a 10% accuracy gain on recall tasks when training on 1B tokens.

REFERENCES

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, 2024.

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew Gormley. Unlimiformer: Long-range transformers with unlimited length input. *Advances in Neural Information Processing Systems*, 36:35522–35543, 2023.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

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.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Hoboken, NJ, 2nd edition, 2006.

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 *Forty-first International Conference on Machine Learning*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Yukun Feng, Patrick Xia, Benjamin Van Durme, and João Sedoc. Automatic document selection for efficient encoder pretraining. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9522–9530, 2022.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128k context. *arXiv preprint arXiv:2402.10171*, 2024.

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

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*, 2024.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024.

Tim Jansen, Yangling Tong, Victoria Zevallos, and Pedro Ortiz Suarez. Perplexed by quality: A perplexity-based method for adult and harmful content detection in multilingual heterogeneous web data. *arXiv preprint arXiv:2212.10440*, 2022.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. Llm maybe longlm: Selfextend llm context window without tuning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 22099–22114, 2024.

Greg Kamradt. Needle in a haystack – pressure testing llms. `https://github.com/gkamradt/LLMTest`, 2023. Accessed: 2025-09-25.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826, 2022.

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

Jiaheng Liu, ZhiqiBai ZhiqiBai, Yuanxing Zhang, Chenchen Zhang, YuangZh YuangZh, Ge Zhang, JiakaiWang JiakaiWang, Haoran Que, Yukang Chen, Wenbo Su, et al. E2-llm: Efficient and extreme length extension of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4243–4253, 2024b.

Xiaoran Liu, Kai Lv, Qipeng Guo, Hang Yan, Conghui He, Xipeng Qiu, and Dahua Lin. Longwanjuan: Towards systematic measurement for long text quality. In *EMNLP (Findings)*, 2024c.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, et al. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3245–3276, 2024.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. *Advances in Neural Information Processing Systems*, 36:79155–79172, 2023.

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*, 2023.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Daria Soboleva, Faisal Al-Khateeb, Joel Hestness, Nolan Dey, Robert Myers, and Jacob R. Steeves. Slimpajama: A 627b token, cleaned and deduplicated version of redpajama. `https://www.cerebras.ai/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama`, June 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*, 2019.

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

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. *arXiv preprint arXiv:2502.16860*, 2025.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36: 34201–34227, 2023.

Jing Xiong, Jianghan Shen, Chuanyang Zheng, Zhongwei Wan, Chenyang Zhao, Chiwun Yang, Fanghua Ye, Hongxia Yang, Lingpeng Kong, and Ngai Wong. Parallelcomp: Parallel long-context compressor for length extrapolation. In *Forty-second International Conference on Machine Learning*, 2025.

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

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. Helmet: How to evaluate long-context models effectively and thoroughly. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yikai Zhang, Junlong Li, and Pengfei Liu. Extending llms' context window with 100 samples. *arXiv preprint arXiv:2401.07004*, 2024.

Dawei Zhu, Nan Yang, Liang Wang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. Pose: Efficient context window extension of llms via positional skip-wise training. In *The Twelfth International Conference on Learning Representations*, 2024.

A Reproducibility

All code and scripts used in this work are publicly available at: https://anonymous.4open.science/r/LongFilter.

This repository contains all necessary files for data preprocessing, model training, evaluation, and visualization.

B Use of LLM

During the preparation of this paper, a large language model (LLM) was used solely for the following purposes:

- Sentence-level polishing of grammar and wording.
- Translation.

The LLM was not used to generate original content, draft sections of the paper, or make any scientific claims. The authors take full responsibility for all content in the submission.

C Equivalence of two Definitions of Conditional Mutual Information

The equivalence of Eq. equation 1 and Eq. equation 2 can be shown by expanding the definition of entropy:

$$
\begin{aligned}
I(T;E \mid S) &= H(T \mid S) - H(T \mid S, E) \\
&= \left(-\sum_{s,t} p(s,t)\log p(t \mid s)\right) - \left(-\sum_{s,e,t} p(s,e,t)\log p(t \mid s,e)\right) \\
&= -\sum_{s,e,t} p(s,e,t)\log p(t \mid s) + \sum_{s,e,t} p(s,e,t)\log p(t \mid s,e) \\
&= \sum_{s,e,t} p(s,e,t)\left(\log p(t \mid s,e) - \log p(t \mid s)\right) \\
&= \sum_{s,e,t} p(s,e,t)\log\frac{p(t \mid s,e)}{p(t \mid s)} \\
&= \sum_{s,e} p(s,e)\sum_{t} p(t \mid s,e)\log\frac{p(t \mid s,e)}{p(t \mid s)} \\
&= \sum_{s,e} p(s,e) D_{KL}\big(p(T \mid S=s, E=e) \parallel p(T \mid S=s)\big) \\
&= \mathbb{E}_{p(s,e)}\left[D_{KL}\big(p(T \mid S,E) \parallel p(T \mid S)\big)\right].
\end{aligned}
\tag{8}
$$

D Additional Experiments

D.1 Using Smaller Models to Score Data in Order to Train Larger Models.

We conduct experiments to investigate how using smaller scoring models affects the performance of LongFilter. Specifically, we perform experiments with Qwen3-0.6B(Yang et al., 2025), a model containing only 0.6B parameters. All other experimental settings remain the same as those described in Section 4. The experimental results are shown in Figure 6.

The experimental results show that LongFilter remains effective even when using a smaller model as the scoring model, yielding a clear improvement in data quality compared to using unfiltered data. However, compared with the 8B model, the 0.6B model performs worse in data selection.

We believe this is because LongFilter relies on the language model's ability to capture long-range dependencies, and in this regard the 0.6B model cannot match the capabilities of the 8B model. Overall, the results demonstrate that smaller language models can still be used as scoring models to improve data efficiency for continued long-context pretraining of larger models.

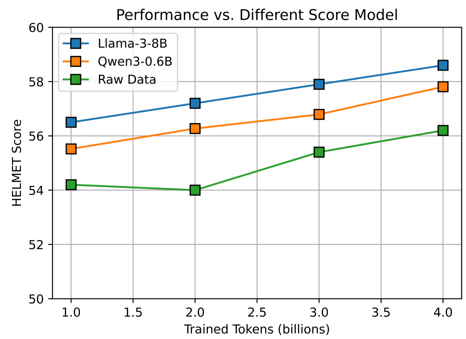


Figure 6: LongFilter Performance on HELMET Benchmark When Scored by a 0.6B-Parameter Model.

D.2 Experiments on the Proportion of Data Selected

We conducted experiments to examine how selecting different proportions of data affects the final results. In the experiments described in Section 4, we used the top 20% of data selected by LongFilter as the dataset for continued long-context pretraining. In this experiment, we additionally selected the top 30% and 40% of the data to observe how these selection ratios influence the final continued-pretraining performance. The experimental results are shown in Figure 7.

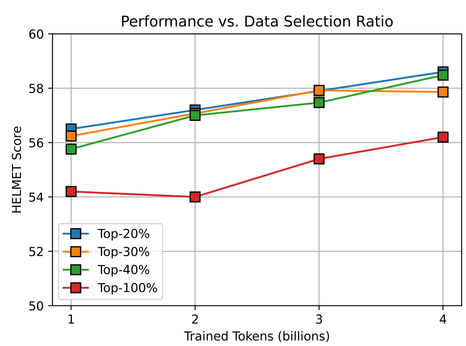


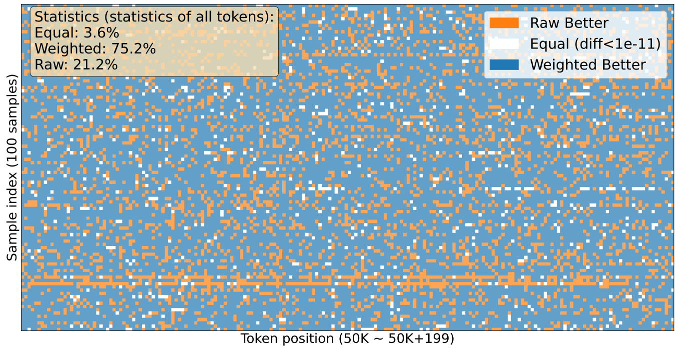
Figure 7: Performance comparison under different data-selection ratios. We vary the proportion of top-ranked data selected by LongFilter (20%, 30%, 40%, and 100%) to evaluate its robustness to selection ratio.

The experimental results show that LongFilter remains effective across a fairly wide range of data-selection ratios. We recommend that users choose the selection ratio based on how many training tokens they ultimately need for their application.

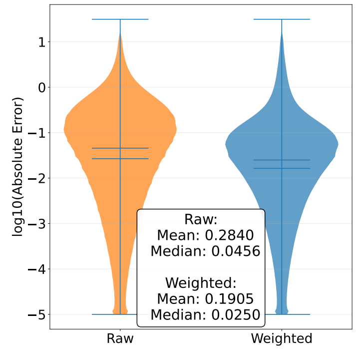
E Discussion on an Alternative Scoring Formulation

In this section, we discuss an alternative score formulation which removes the penalty weighting term in equation 7 and equation 5:

$$\text{score}(t^*, s^*, e^*) = \log \frac{p(T = t^* \mid E = e^*, S = s^*)}{p(T = t^* \mid S = s^*)} = \mathcal{L}^{\text{short}} - \mathcal{L}^{\text{long}} \tag{9}$$



(a) Pairwise Comparison



(b) Error Distribution

Figure 8: **Approximation Fidelity of Scoring Formulations relative to KL Divergence.** We evaluate the absolute error of the *raw* loss difference ($\mathcal{L}^{\text{short}} - \mathcal{L}^{\text{long}}$) versus the *weighted* loss difference ($\exp(-\mathcal{L}^{\text{long}}) \cdot (\mathcal{L}^{\text{short}} - \mathcal{L}^{\text{long}})$) relative to the ground truth KL divergence. **(a) Pairwise Comparison.** A heatmap visualizing 100 randomly selected sequences (y-axis) across 200 consecutive token positions (x-axis, indices 50k–50.2k). Pixels are colored by the method with the lower absolute error: Blue indicates the weighted score is closer to the true KL, Orange indicates the raw score is closer, and White indicates equality (difference $< 10^{-11}$). The weighted score outperforms the raw score in 75.2% of instances. **(b) Global Error Distribution.** Violin plots illustrating the density of $\log_{10}$ absolute errors computed over the full validation set. The weighted formulation (right) exhibits a significantly lower error profile (lower mean and median) compared to the raw formulation (left), validating the effectiveness of the exponential weighting term.

We empirically demonstrate that the weighted loss difference defined in equation 5 serves as a superior surrogate for the KL divergence compared to the unweighted variant. Following the setup in Section 4, we analyze a subset of 1000 samples from the processed SlimPajama-Arxiv dataset.

To establish a ground truth, we compute the full probability distribution over the vocabulary (using the same model as Section 4) at each position to derive the exact KL divergence between P_{short} and P_{long}. We compare this against two candidate scores derived only from the probability of the ground-truth token in the data sequence: the raw loss difference ($\mathcal{L}^{\text{short}} - \mathcal{L}^{\text{long}}$) and the weighted loss difference ($\exp(-\mathcal{L}^{\text{long}}) \cdot (\mathcal{L}^{\text{short}} - \mathcal{L}^{\text{long}})$). The results, illustrated in Figure 8, indicate that the weighted formulation approximates the KL divergence with consistently higher fidelity. We attribute this to the theoretical properties of the metrics: while the raw score captures the pointwise log-likelihood ratio (which can be noisy for low-probability tokens), the weighted score scales this ratio by the token probability. This effectively approximates the term-wise contribution to the expected KL divergence, thereby suppressing outliers in the distribution's tail.