Sentinel: Decoding Context Utilization via Attention Probing for Efficient LLM Context Compression

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) often suffers from long and noisy retrieved contexts. Prior context compression methods rely on predefined importance metrics or supervised compression models, rather than on the model's own inference-time behavior. We propose Sentinel, a lightweight sentence-level compression framework that treats context compression as an understanding decoding problem. Sentinel probes native attention behaviors of a frozen LLM with a lightweight readout to decode which parts of the context are actually utilized when answering a query, rather than using attention as a direct relevance score. We empirically observe that decoded relevance signals exhibit sufficient consistency across model scales to support effective compression with compact proxy models. On LongBench, Sentinel with a 0.5B proxy model achieves up to 5× compression while matching the QA performance of 7B-scale baselines, and despite being trained only on English QA data, generalizes effectively to Chinese and out-of-domain settings.

1 Introduction

Large language models (LLMs) have achieved impressive performance across open-domain question answering, reasoning, and dialogue tasks (Brown et al., 2020; OpenAI, 2024). To scale their capabilities to knowledge-intensive applications, Retrieval-Augmented Generation (RAG) has emerged as a powerful paradigm that augments model inputs with retrieved evidence from external corpora (Lewis et al., 2020; Guu et al., 2020; Shi et al., 2024). However, long retrieved contexts are often noisy, redundant, or exceed model input limits, making context compression essential for both efficiency and effectiveness (Liu et al., 2024; Yoran et al., 2024).

Existing context compression methods can be broadly divided into two categories. Metric-based approaches estimate the utility of context using predefined or model-derived importance metrics, such as perplexity, self-information, mutual information, or query–context similarity (Jiang et al., 2023, 2024a; Li et al., 2023). While lightweight and training-free, these methods estimate relevance via heuristic or proxy importance scores, which are only indirectly related to the model's inference-time behavior. In contrast, data-driven approaches learn compression decisions using external supervision or generator feedback to optimize downstream task performance (Pan et al., 2024; Xu et al., 2024; Hwang et al., 2024). Although effective, these approaches treat context compression as an optimization problem external to the model's inference process, introducing additional training cost and often tying compression behavior to specific training objectives or generator feedback.

Recent mechanistic studies of Transformer-based LLMs have shown that decoder-only models exhibit structured context-utilization behaviors, with specialized attention heads supporting query–context alignment and evidence retrieval (Wu et al., 2024; Jin et al., 2024; Huang et al., 2025). These findings suggest that LLMs actively form query-aware contextual understanding during inference, rather than passively consuming retrieved context. Despite these insights, existing compression methods rarely exploit model-internal understanding signals directly. A key challenge is that such signals are not readily accessible in a stable and lightweight manner in decoder-only LLMs. Moreover, naively using raw attention as a compression signal often proves unreliable, as attention patterns mix informative behaviors with various forms of noise.

Motivated by these observations, we reformulate context compression as an understanding decoding problem, where compression decisions are derived from how an LLM internally utilizes context when answering a query, rather than from external importance heuristics. We propose Sentinel, a

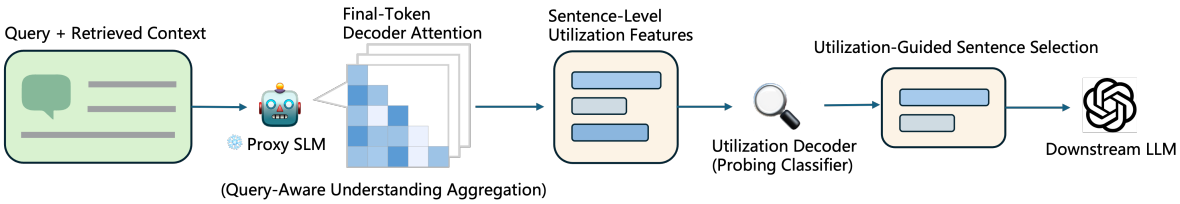

Figure 1: **Sentinel Framework Overview.** Sentinel decodes query-aware context utilization from native attention behaviors of a frozen LLM. By probing sentence-level attention features aggregated at a single decoding step, Sentinel identifies relevant context without training compression models or performing full autoregressive generation.

lightweight context compression framework that decodes query-aware context utilization from native attention behaviors of LLMs. Sentinel requires neither training a dedicated compression model nor running full autoregressive generation. Instead, it probes attention behaviors of a frozen proxy LLM under carefully designed prompts, enabling efficient sentence-level compression in a single forward pass. By grounding compression decisions in the model's own inference-time behavior, Sentinel provides an interpretable and scalable alternative that leverages the LLM's native understanding capabilities, rather than learning task-specific compression policies.

Empirically, Sentinel achieves up to 5× input compression on LongBench while attaining question-answering performance comparable to 7B-scale compression baselines, using only a 0.5B proxy model. The probing classifier relies solely on existing English QA data, yet the resulting compression strategy generalizes effectively to out-of-domain English LongBench tasks and exhibits robust cross-lingual generalization on Chinese benchmarks. Across multiple model families and scales, including Qwen-2.5, Qwen3, and LLaMA-3 variants, Sentinel shows consistent compression behavior under a unified and lightweight probing paradigm.

Our contributions are as follows:

- We introduce a new formulation of context compression that formulates it as an *understanding decoding problem*, grounding compression decisions in how LLMs internally utilize context when answering a query, rather than in external importance metrics or generation-based objectives.
- We propose **Sentinel**, a lightweight context compression framework that decodes query-aware context utilization from the native attention behaviors of frozen LLMs via a probing-based paradigm, eliminating the need for dedicated compression models or full autoregressive generation.
- We empirically observe that the query–context relevance signals decoded by Sentinel remain consistent across proxy model scales and families, enabling compact proxy models to approximate the context compression behavior of much larger LLMs.
- We conduct extensive experiments on long-context benchmarks, demonstrating that Sentinel achieves substantial compression while improving downstream QA performance, with mechanistic analyses supporting consistency with prior findings on retrieval-oriented behaviors in LLMs.

2 Methodology

2.1 Context Compression as Understanding

We propose Sentinel, a lightweight framework that approaches context compression from an understanding-centric perspective. Rather than training a compression model end-to-end or estimating sentence importance using external heuristics, Sentinel treats context compression as a problem of decoding how a language model internally understands and utilizes retrieved context when answering a query.

Formally, given a query q and a retrieved context $C = \{s_1, s_2, \ldots, s_n\}$ composed of sentences, the goal of context compression is to select a subset $C' \subseteq C$ that preserves the information actually used by the model to answer q. Under this formulation, compression decisions are grounded in the model's internal inference behavior, rather than in generation outputs or predefined utility metrics.

2.2 Aggregating Query-Aware Understanding in LLMs

Recent studies suggest that decoder-only LLMs actively form query-aware contextual understanding during inference, which is reflected in their

internal attention behaviors (Wu et al., 2024; Jin et al., 2024; Huang et al., 2025). However, directly exploiting such understanding signals for context compression is non-trivial, as context utilization is typically distributed across multiple decoding steps and raw attention patterns are highly noisy.

An encouraging observation is that, under appropriate prompting, global contextual understanding can be implicitly aggregated at specific stages of the inference process, with the first generation step encoding a compressed representation of the entire input (Jiang et al., 2024c). From an information-theoretic perspective, this aggregation can be viewed as over-squashing, where prior context is progressively compressed into the final token representation (Barbero et al., 2024).

Following this insight, we feed the query and retrieved context into a compact decoder-only proxy model with instruction-following capability, apply a QA-style prompt that encourages semantic compression at the final position, and extract decoder attention from the final decoding token as a compact carrier of query-aware understanding signals.

2.3 Decoding Context Utilization via Probing

We decode context utilization by probing sentence-level attention features extracted from the proxy model, without modifying or fine-tuning the model.

2.3.1 Sentence-Level Attention Features

For each query–context input, we extract the decoder attention tensor from the final decoding token, capturing attention scores across layers, heads, and input tokens. To obtain sentence-level representations, we aggregate the attention weights directed toward the tokens of each sentence, and normalize them by the total attention mass over the context span. This normalization removes the influence of prompt and query tokens and yields comparable relevance signals across sentences.

Averaging the normalized attention weights over tokens within each sentence produces a feature vector for sentence $\mathbf{s}_i$, denoted as $\mathbf{v}_i \in R^{LH}$, where each dimension corresponds to a specific attention head at a particular layer.

2.3.2 Probing Context Utilization

To decode sentence relevance from attention features, we train a lightweight probing classifier on top of the sentence-level representations. Specifically, we adopt logistic regression as a linear probe, which maps each feature vector $\mathbf{v}_i$ to a scalar relevance score via a sigmoid function:

$$\hat{y}_i = \sigma(\mathbf{w}^\top \mathbf{v}_i + b),$$

where $\mathbf{w}$ and b denote the probe parameters.

The resulting probability is interpreted as the degree to which sentence $\mathbf{s}_i$ is utilized by the model when answering the query. We choose a linear probe for two reasons. First, it minimizes the risk of learning new behaviors beyond those already encoded in the model. Second, it enables direct interpretability of individual attention heads, allowing us to analyze which attention head contribute positively or negatively to context utilization.

2.4 Weak Supervision for Probing Context Utilization

To decode the model's internal context utilization behavior, we train the probing classifier using weak supervision derived from question answering data. Importantly, this supervision is not intended to annotate sentence importance in general, but to identify which sentences the model treats as evidence when it genuinely relies on retrieved context to answer a query.

2.4.1 Probing Data Construction

We construct probing examples from existing QA datasets that provide answer span annotations within retrieved contexts, covering both single-hop and multi-hop question answering settings. For each QA instance, sentences containing the gold answer span are labeled as positive, while all other sentences in the retrieved context are labeled as negative. This weak supervision allows us to scale probing without manual relevance annotation and exposes the probe to diverse reasoning patterns, ranging from localized factual evidence to distributed multi-hop evidence.

2.4.2 Selecting Context-Reliant Samples

To purify supervision, we retain only QA examples that require retrieved context for correct answering. Specifically, we keep examples where the model fails without access to the retrieved context but succeeds when the context is provided. This filtering echoes prior work that probes model behavior via intervention-based output changes (Meng et al., 2022). It ensures that positive sentences genuinely contribute essential information for answering, reducing contamination from internal memorization or hallucinated knowledge. By filtering for retrieval

dependency, we focus training on cases where relevance must be decoded from the provided context, aligning with the goal of decoding context utilization rather than internal recall.

2.4.3 Robustness via Sentence Shuffling

To mitigate positional biases (Liu et al., 2024), especially common in multi-document retrieval settings, we apply sentence shuffling during training by randomly permuting sentence order within each passage. This simple perturbation encourages the classifier to rely on semantic relevance rather than fixed positions, improving generalization to real-world RAG inputs with noisy or varied structure.

2.5 Inference-Time Context Compression

At inference time, given a query–context pair (q, C), Sentinel runs a single forward pass of a compact proxy model with a fixed QA-style prompt, extracts final-token decoder attention, and computes sentence-level attention features. A trained probing classifier assigns relevance scores to sentences, based on which a top-ranked subset $C' \subseteq C$ is selected under a length budget and passed to the downstream LLM for answer generation.

3 Experiments

Datasets We evaluate Sentinel on both the English and Chinese subsets of LongBench (Bai et al., 2024). Following our focus on query-conditioned context compression, we report results on question answering tasks and query-conditioned summarization (e.g., QMSum), which involve an explicit query. When using Qwen-2.5-7B as the downstream LLM, we exclude Few-shot, Code, and Synthetic tasks, as they lack an explicit query or rely on strict prompt structure. For fair comparison with prior work, we additionally report results on the full LongBench benchmark when comparing against baseline methods. Detailed dataset descriptions are provided in the appendixA.

Probing Data We train the probing classifier on a small set of English QA examples spanning both single-hop and multi-hop reasoning. In the default setting, we sample 3K QA instances, each yielding one positive sentence containing the gold answer span and one negative sentence from the same context, resulting in 6K sentence-level training examples. We retain only context-reliant examples, where correct answering requires access to the retrieved context. Sentence-level attention features are extracted using a fixed QA-style prompt by collecting decoder attention from the final decoding token. Additional implementation details are provided in Appendix B.

Probing Classifier Training We train a logistic regression probe on attention-derived features using standard cross-validation and regularization. Additional training details are in Appendix B.

Compression Strategy Sentinel performs length-controlled context compression by ranking sentences with the probing classifier and selecting a top-ranked subset under a specified budget. Selected sentences are concatenated in their original order and passed to the downstream LLM.

We consider two budget settings, both measured using the target model's tokenizer: (i) a fixed token budget B (e.g., 2000 tokens), where sentences are selected until the budget is reached; and (ii) a compression ratio $\tau \in [0.1, 0.5]$, where the retained sentences do not exceed a fraction of the original context length.

Proxy Model Setup Unless otherwise specified, Sentinel uses `Qwen-2.5-0.5B-Instruct` as the default proxy model for attention-based feature extraction and sentence relevance probing, with a chunk size of 1024 tokens.

Evaluation Models Following LLMLingua setup, we use `ChatGPT (gpt-3.5-turbo)` as the primary model for evaluation. To assess the generality of our method, we also experiment with `Qwen-2.5-7B-Instruct` in our main results. All evaluations follow the LongBench prompt and decoding setup (Bai et al., 2024), as detailed in Appendix I.

Baselines We compare Sentinel against representative context compression baselines spanning both metric-based and data-driven approaches. Metric-based baselines include LLMLingua-1 (Jiang et al., 2023), LongLLMLingua (Jiang et al., 2024b), and Selective Context (Li et al., 2023). Data-driven baselines include LLMLingua-2 (Pan et al., 2024) and CPC (Liskavets et al., 2024). We include an attention-based heuristic baseline, denoted as **Raw Attention**, which represents a class of methods that directly use aggregated decoder attention weights as relevance scores, such as QUITO (Wang et al., 2024) and AttentionRAG (Fang et al., 2025). We also include non-learning baselines including Ran-

Methods	LongBench-En (GPT-3.5-Turbo, 2000-token constraint)							Compression Stats	
	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	Code	**AVG**	Tokens	$1/\tau$
Selective-Context (LLaMA-2-**7B**-Chat)	16.2	34.8	24.4	15.7	8.4	49.2	24.8	1,925	5x
LLMLingua (LLaMA-2-**7B**-Chat)	22.4	32.1	24.5	61.2	10.4	56.8	34.6	1,950	5x
LLMLingua-2 (XLM-RoBERTa-Large-**0.6B**)	29.8	33.1	25.3	66.4	21.3	58.9	39.1	1,954	5x
LongLLMLingua (LLaMA-2-**7B**-Chat)	39.0	42.2	**27.4**	69.3	**53.8**	56.6	48.0	1,809	6x
CPC (Mistral-**7B**-Instruct-v0.2)	**42.6**	**48.6**	23.7	69.4	52.8	**60.0**	**49.5**	1,844	5x
Sentinel (Qwen-2.5-**0.5B**-Instruct)	40.1	47.4	25.8	**69.9**	46.3	58.0	47.89	1,885	5x
Sentinel (Qwen-2.5-**1.5B**-Instruct)	40.6	48.1	26.0	69.1	49.0	57.6	48.4	1,883	5x
Original Prompt	39.7	38.7	26.5	67.0	37.8	54.2	44.0	10,295	-

Table 1: Performance on English LongBench using GPT-3.5-Turbo as the inference model. Best results are in **bold**, second-best are underlined.

Methods	LongBench-En (2000-token constraint)				LongBench-Zh (2000-token constraint)			Overall AVG
	SingleDoc	MultiDoc	Summ.	En-AVG	SingleDoc	MultiDoc	Zh-AVG	
Empty	10.72	22.26	16.46	16.48	17.71	13.54	15.62	16.05
Random	28.22	30.68	20.33	26.41	43.18	17.22	30.20	28.30
Raw Attention (Qwen-2.5-0.5B-Instruct)	34.92	38.96	21.32	31.74	51.72	17.29	34.50	33.12
Sentinel (Qwen-2.5-0.5B-Instruct)	37.73	**46.16**	**23.03**	**35.64**	**62.24**	**18.57**	**40.41**	**38.02**
Original Prompt	**38.84**	44.74	22.76	35.45	60.06	18.21	39.14	37.30

Table 2: LongBench results under a 2000-token context constraint, evaluated using Qwen-2.5-Instruct-7B as the downstream LLM. Results are reported on both English and Chinese subsets. The **Summ.** column corresponds to query-conditioned summarization tasks (QMSum).

dom Selection and Empty Context. Full descriptions are provided in Appendix C.

Metrics We follow the LongBench evaluation protocol and adopt task-specific metrics for each task category: QA-F1 for Single-Document QA, Multi-Document QA, ROUGE-L for Summarization. All metrics are computed using the official evaluation scripts.

3.1 Results on LongBench

We evaluate Sentinel under two settings: (1) English LongBench tasks using GPT-3.5-Turbo as the inference model, and (2) both English and Chinese LongBench tasks using Qwen-2.5-7B-Instruct. All results are reported under a 2,000-token input constraint.

Strong Performance with Compact Proxies under GPT-3.5-Turbo Table 1 presents results on the English subset of LongBench using GPT-3.5-Turbo as the inference model. Using a compact 0.5B proxy model, Sentinel achieves strong performance across all evaluated task categories.

Despite its small proxy size, Sentinel consistently outperforms task-agnostic compression methods such as LLMLingua and LLMLingua-2, while using only **a 0.5B proxy to perform competitively with 7B-scale compression baselines**, including CPC and LongLLMLingua.

These results demonstrate that effective query-aware sentence selection can be achieved using compact proxy models, without training dedicated compression networks or relying on large-scale generators. Additional results on Chinese tasks under GPT-3.5-Turbo are reported in Appendix D.

Effective Compression and Cross-Lingual Generalization under Qwen-2.5-7B-Instruct Table 2 reports results on both English and Chinese LongBench subsets using Qwen-2.5-7B-Instruct as the downstream LLM. Across all evaluated tasks, Sentinel substantially outperforms Random, Empty Context, and Raw Attention baselines, confirming the effectiveness of decoding query-aware context utilization signals.

On the English subset, Sentinel surpasses the Original Prompt baseline on average and improves performance on most task categories under a strict 2,000-token budget. Notably, despite being trained exclusively on English QA data, Sentinel maintains strong performance on the Chinese subset and achieves clear improvements over Raw Attention, demonstrating **robust cross-lingual generalization** of the decoded relevance signals.

Overall, these results show that Sentinel enables effective query-conditioned context compression by decoding model-internal understanding signals, and that the resulting relevance estimates generalize reliably across languages and inference settings.

3.2 Effect of Proxy Model Family and Size

We evaluate Sentinel using proxy models from three model families, including Qwen-2.5 (Qwen

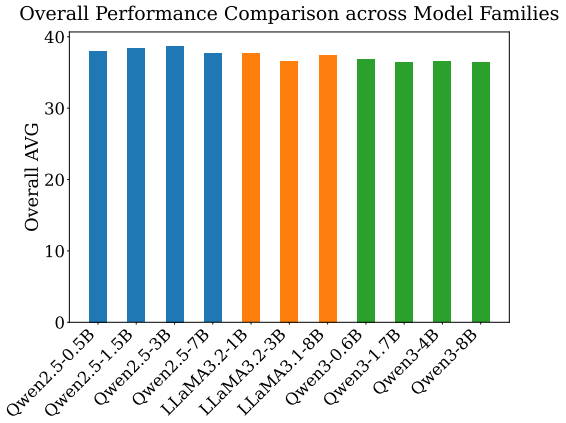

Figure 2: Impact of proxy model family and scale on Sentinel performance under a 2k-token context (LongBench Overall AVG)

et al., 2025), Qwen-3 (Yang et al., 2025), and LLaMA-3 (Grattafiori et al., 2024), spanning parameter scales from 0.5B to 8B. As shown in Figure 2, Sentinel achieves comparable overall performance across proxy models of different sizes and families, indicating that the query-context relevance signals decoded by Sentinel are consistent across model scales. Notably, increasing the proxy model size does not lead to systematic improvements in compression performance. This suggests that the relevance signals exploited by Sentinel are already sufficiently expressed in compact proxy models, rather than emerging only with increased model capacity. As a result, Sentinel can be deployed efficiently using small and computationally inexpensive proxy models without sacrificing compression quality. Detailed per-model and per-task results are provided in Appendix E.

3.3 Ablation

We conduct ablation studies to analyze where Sentinel's performance gains come from and to verify that they arise from decoding model-internal understanding signals rather than from probe capacity, large-scale supervision, or specific inference heuristics. By default, Sentinel is instantiated on `Qwen-2.5-0.5B-Instruct`. Unless otherwise specified, all experiments use `Qwen-2.5-7B-Instruct` as the downstream LLM and are evaluated on LongBench.

3.3.1 Effect of Probing Data Size

A key design principle of Sentinel is to decode relevance signals already encoded in frozen LLMs, rather than to learn a new compression behavior from supervision. To evaluate the dependence of Sentinel on probing data size, we vary the number of QA examples used for probing from 500 to 3000.

Probing Size	Overall AVG
500	38.03
1000	38.24
2000	37.92
3000	38.02

Table 3: Overall performance under different probing data sizes.

Feature	HotpotQA	SQuAD	NewsQA	Overall AUC	Overall AVG
All Layers	0.9228	0.9987	0.9838	0.9700	38.02
Selected	0.9171	0.9943	0.9832	0.9662	36.56
Last Layer	0.8606	0.9538	0.9588	0.9121	37.24

Table 4: AUC comparison of different attention feature extraction strategies.

As shown in Table 3, downstream performance remains nearly invariant across this range, with no clear trend of improvement as more probing data is added. This indicates that Sentinel does not rely on large amounts of supervision: the attention-based relevance signals are already present in the model, and the probing classifier mainly acts as a lightweight readout of these signals. Notably, even a small probing set is sufficient to support effective context compression. Detailed task-level results are provided in Appendix F.

3.3.2 Attention Feature Ablations

We evaluate three attention-based feature construction strategies using `Qwen-2.5-0.5B-Instruct` as the proxy model: (i) aggregating attention from all decoder layers, (ii) using only the final decoder layer, and (iii) selecting a compact subset of attention heads via mRMR (Ding and Peng, 2005), constrained to at most one layer's worth of heads (see Appendix F for details).

As shown in Table 4, aggregating attention across all layers consistently achieves the strongest AUC and downstream performance. Using only the final decoder layer leads to a noticeable performance drop, while the selected-head variant preserves most of the performance with substantially fewer features. These results suggest that query-aware relevance signals are distributed across multiple layers rather than being dominated by the final decoder layer, supporting our choice of aggregating attention information across the full model.

3.3.3 Compression Ratio Variants

We further evaluate robustness under different compression ratios $\tau \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, where smaller τ corresponds to more aggressive pruning. As shown in Figure 3, the Raw Attention baseline degrades sharply with increasing compression,

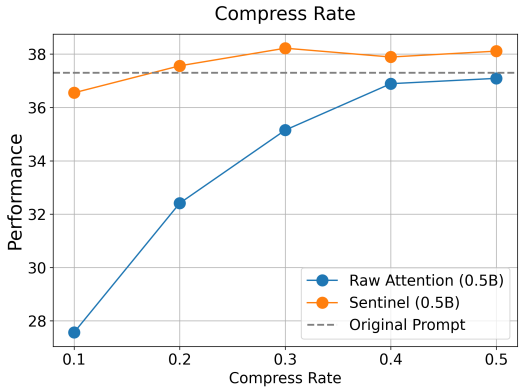

Figure 3: Compression ratio ablation on `Qwen-2.5-7B-Instruct` with a 0.5B proxy.

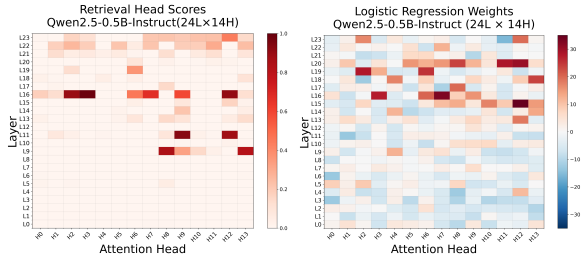

Figure 4: Comparison of attention head importance patterns identified by retrieval-head analysis (left) and Sentinel probing (right).

collapsing when $\tau < 0.4$. In contrast, Sentinel achieves strong performance across all compression levels, consistently outperforming the Raw Attention baseline and surpassing the original prompt performance over a wide range of τ, including the extreme setting of $\tau = 0.2$. This result indicates that decoding query-aware context utilization signals yields more robust context compression than relying on raw attention magnitudes under aggressive pruning, enabling Sentinel to retain the most critical contextual information even under severe compression. Full task-level results are reported in Appendix F.

4 Analysis: Interpreting Context Utilization Decoded from Attention Heads

Alignment with Retrieval-Oriented Attention Heads Prior mechanistic analyses have shown that retrieval-oriented and context-utilization behaviors in decoder-only LLMs are concentrated in a sparse subset of attention heads (Wu et al., 2024; Jin et al., 2024), whose activation depends on the input tokens and contexts.

Using a fundamentally different identification mechanism, Sentinel recovers a meaningful subset of these retrieval-oriented structures. Specifically, when comparing the top-14 positively weighted heads identified by Sentinel with the top-14 retrieval heads obtained by reproducing the retrieval-head identification procedure of prior work (Wu et al., 2024) on the same Qwen-2.5-0.5B-Instruct model, we observe five overlapping heads. As illustrated in Figure 4, these overlapping heads are not randomly distributed, but are predominantly located in middle-to-late layers, with a noticeable concentration around layer 16.

Efficient and Ensemble-Based Decoding of Context Utilization This observed overlap and layer-wise concentration indicate that the proposed probing approach captures core context-utilization behaviors identified by prior mechanistic studies. Whereas prior work identifies retrieval heads individually through autoregressive decoding and token-level copy analysis, Sentinel assigns weights to all attention heads and decodes context utilization through their weighted aggregation. This ensemble-style decoding provides a more efficient and compression-oriented access to model-internal understanding signals.

Beyond Positive Retrieval Heads More importantly, Sentinel goes beyond identifying only positively contributing heads. By assigning signed weights through probing, Sentinel captures both heads that support evidence utilization and heads that systematically interfere with it. We observe that heads assigned negative weights are frequently associated with structurally dominant but semantically uninformative attention patterns, such as attention sinks (Bondarenko et al., 2021; Son et al., 2024). Further analysis of negatively weighted heads is provided in Appendix G.

Robustness under Dynamic and Multi-Functional Head Behavior This distinction is particularly important given that retrieval heads are dynamically activated depending on input tokens and contexts (Wu et al., 2024), reflecting the multi-functional nature of attention heads. A single head may support evidence retrieval in some contexts while exhibiting non-retrieval behaviors in others (Zheng et al., 2024). By aggregating both positive and negative contributions across all heads, Sentinel mitigates instability caused by context-dependent role switching and counterbalances spurious activations, resulting in a more robust decoding of context utilization than approaches that rely exclusively on positively

Methods	LongBench-En (2000-token constraint)				LongBench-Zh (2000-token constraint)			Overall AVG
	SingleDoc	MultiDoc	Summ.	En-AVG	SingleDoc	MultiDoc	Zh-AVG	
Top 14 Retrieval Heads (Qwen-2.5-0.5B-Instruct)	36.53	43.16	22.27	33.99	58.59	18.53	38.56	36.28
Sentinel (Qwen-2.5-0.5B-Instruct)	37.73	46.16	23.03	35.64	62.24	18.57	40.41	38.02

Table 5: Comparison between retrieval-head-based compression and Sentinel on the LongBench benchmark, where retrieval-head compression scores sentences using attention from the top-14 retrieval heads.

identified retrieval heads. Table 5 further supports this analysis: Sentinel consistently outperforms retrieval-head-based compression on LongBench across English, Chinese, and overall averages.

5 Related Work

Metric-Based Context Compression Metric-based approaches estimate context utility using predefined importance scores, such as self-information, mutual information, or query–context similarity, and select tokens or sentences accordingly without training a dedicated compression model. Representative token-level methods include LLMLingua (Jiang et al., 2023) and LongLLMLingua (Jiang et al., 2024a), which prune tokens based on perplexity or query-conditioned probability estimates. QUITO-X (Cao et al., 2024) further introduces mutual information-based scoring to guide context selection. At a coarser granularity, Selective Context (Li et al., 2023) removes low-information content based on token-level self-information scores.

Some recent methods (Wang et al., 2024; Fang et al., 2025) leverage decoder attention as an importance heuristic, using vanilla or heuristically aggregated attention weights as a proxy for relevance. However, attention is used in a post-hoc scoring manner, rather than being decoded to reflect how the model internally utilizes context during inference.

While effective and training-free, these methods rely on predefined or heuristically applied importance scores that are not explicitly tied to decoding-time context utilization.

Data-Driven Context Compression Data-driven approaches learn compression decisions from external supervision, typically by training a ranking or classification model to predict which tokens or sentences should be retained. Token-level methods such as LLMLingua-2 (Pan et al., 2024) leverage distilled labels from large language models to train lightweight compressors. At the sentence level, methods such as RECOMP (Xu et al., 2024) train compressors to produce extractive or abstractive summaries that improve downstream performance, while EXIT (Hwang et al., 2024) learns a sentence-level classifier to select query-relevant sentences. Other works, such as CPC (Liskavets et al., 2024), Refiner (Li et al., 2024), and FineFilter (Zhang et al., 2025), further incorporate query-aware ranking, structure-aware reranking, or multi-hop reasoning objectives. Although these methods often achieve strong performance, they introduce additional training cost and data dependency, which can limit their adaptability across tasks and models.

Our Approach: Utilization-Driven Context Compression In contrast to prior work, we propose Sentinel, a lightweight and model-agnostic framework that adopts a utilization-driven perspective on context compression. Rather than relying on predefined metrics or externally supervised compression models, Sentinel decodes how a frozen LLM internally utilizes context when answering a query. Crucially, Sentinel does not treat attention weights as direct relevance scores. While raw attention answers the question of where the model attends, Sentinel probes attention behaviors with a lightweight readout to decode which parts of the context are actually utilized by the model during inference.

6 Conclusion

We present **Sentinel**, a lightweight context compression framework that decodes how LLMs utilize context when answering a query. By probing native attention behaviors of a frozen proxy model, Sentinel enables effective sentence-level compression without training a dedicated compressor or relying on full autoregressive generation. Empirically, Sentinel achieves up to 5$\times$ compression on LongBench while matching or improving QA performance compared to strong 7B-scale baselines, using only a 0.5B proxy. These results suggest that model-internal utilization signals provide a principled and efficient foundation for query-aware context compression.

Limitations

Query-Conditioned Scope. Sentinel is designed for query-conditioned context compression, where relevance is defined with respect to an explicit question or instruction. Tasks that lack a clear query signal or rely on strict prompt structure, such as free-form summarization or code completion, fall outside the current scope of our framework. An interesting direction for future work is to explore whether such tasks can be reformulated with auxiliary or synthetic queries, enabling utilization-driven compression beyond explicit QA settings.

Evaluation Backbones. Our downstream evaluation focuses on two decoder LLMs, GPT-3.5-Turbo and Qwen-2.5-7B-Instruct, covering both proprietary and open-source settings. While Sentinel demonstrates consistent behavior across multiple proxy model families and scales, further evaluation with additional inference backbones could help characterize its behavior under different instruction styles and decoding configurations.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024. 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)*, pages 3119–3137.

Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, João Madeira Araújo, Oleksandr Vitvitskyi, Razvan Pascanu, and Petar Veličković. 2024. Transformers need glasses! information over-squashing in language tasks. *Advances in Neural Information Processing Systems*, 37:98111–98142.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7947–7969. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zhiwei Cao, Qian Cao, Yu Lu, Ningxin Peng, Luyang Huang, Shanbo Cheng, and Jinsong Su. 2024. Retaining key information under high compression ratios: Query-guided compressor for LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12685–12695.

Chris Ding and Hanchuan Peng. 2005. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205.

Yixiong Fang, Tianran Sun, Yuling Shi, and Xiaodong Gu. 2025. Attentionrag: Attention-guided context pruning in retrieval-augmented generation. *arXiv preprint arXiv:2503.10720*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Yanwen Huang, Yong Zhang, Ning Cheng, Zhitao Li, Shaojun Wang, and Jing Xiao. 2025. Dynamic attention-guided context decoding for mitigating context faithfulness hallucinations in large language models. *arXiv preprint arXiv:2501.01059*.

Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C Park. 2024. Exit: Context-aware extractive compression for enhancing retrieval-augmented generation. *arXiv preprint arXiv:2412.12559*.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024a. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024b. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2024c. Scaling sentence

embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3182–3196.

Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2024. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1193–1215.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353.

Zhonghao Li, Xuming Hu, Aiwei Liu, Kening Zheng, Sirui Huang, and Hui Xiong. 2024. *Refiner*: Restructure retrieved content efficiently to advance question-answering capabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8548–8572.

Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane Luke. 2024. Prompt compression with context-aware sentence encoding for fast and improved llm inference. *arXiv preprint arXiv:2409.01227*.

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

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384.

Seungwoo Son, Wonpyo Park, Woohyun Han, Kyuyeun Kim, and Jaeho Lee. 2024. Prefixing attention sinks can mitigate activation outliers for large language model quantization. *arXiv preprint arXiv:2406.12016*.

Wenshan Wang, Yihang Wang, Yixing Fan, Huaming Liao, and Jiafeng Guo. 2024. Quito: Accelerating long-context reasoning through query-guided context compression. *arXiv preprint arXiv:2408.00274*.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Qianchi Zhang, Hainan Zhang, Liang Pang, Hongwei Zheng, Yongxin Tong, and Zhiming Zheng. 2025. Finefilter: A fine-grained noise filtering mechanism for retrieval-augmented large language models. *arXiv preprint arXiv:2502.11811*.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. Attention heads of large language models: A survey. *arXiv preprint arXiv:2409.03752*.

A Dataset Details

We provide a detailed description of the datasets used in our experiments, based on the English subset of LongBench (Bai et al., 2024). LongBench is

a long-context benchmark covering diverse tasks designed to evaluate the capabilities of language models in understanding and reasoning over extended textual inputs. It consists of six task categories, each comprising multiple representative datasets:

- **Single-Document QA**:
  - NarrativeQA: Answer questions based on a single narrative document, such as a story or movie script.
  - Qasper: Answer questions grounded in a scientific paper.
  - MultiFieldQA-en: Answer factual questions from a long structured encyclopedic entry.
- **Multi-Document QA**:
  - HotpotQA, 2WikiMultihopQA, and MuSiQue: Multi-hop QA tasks requiring reasoning across multiple passages to answer complex factoid questions.
- **Summarization**:
  - GovReport: Summarize long government reports.
  - QMSum: Query-based summarization of meeting transcripts.
  - MultiNews: Summarize multi-source news articles.
- **Few-shot Reasoning**:
  - TREC: Classify question types.
  - TriviaQA: Answer trivia-style factual questions.
  - SAMSum: Summarize short dialogues.
  - LSHT: Classify Chinese news headlines into topic categories.
- **Synthetic Retrieval**:
  - PassageCount: Count the number of unique paragraphs among potentially duplicated inputs.
  - PassageRetrieval-en: Identify the source paragraph corresponding to a given abstract.
- **Code Completion**:
  - LCC: Predict the next line of code given a code block, without an explicit natural language query.
  - RepoBench-P: Predict the next line of a function given multi-file code context and the function signature.

Excluded Task Categories We exclude LongBench task categories that violate the assumptions of query-conditioned context compression, including Code, Synthetic, Few-shot, and Summarization tasks. These tasks either lack explicit queries, depend on strict prompt structure or surface-level consistency, or rely on evaluation metrics insensitive to compression quality. We therefore focus on question answering tasks, where query-conditioned relevance estimation is well defined.

B Additional Probing Data and Training Details

Probing Data Composition The probing classifier is trained on 3,000 QA examples sampled from three widely used QA datasets: NewsQA (50%), SQuAD (20%), and HotpotQA (30%), covering both single-hop and multi-hop reasoning scenarios. Each QA example yields two sentence-level instances: one positive sentence containing the gold answer span and one negative sentence sampled from the same retrieved context, resulting in a total of 6,000 sentence-level training examples.

Context Length Statistics We report the context length distribution for completeness. In NewsQA, 30.1% of examples contain 0–500 tokens and 69.9% contain 500–1,000 tokens when tokenized with the Qwen-2.5 tokenizer. In SQuAD, 99.3% of examples fall within the 0–500 token range. For HotpotQA, all examples are restricted to 0–500 tokens by limiting unrelated content in the retrieved context.

Sentence Segmentation Retrieved contexts are segmented into sentences using spaCy's `sentencizer`. Sentence boundaries are used consistently for both positive and negative sentence extraction as well as for sentence-level attention aggregation.

Prompt Template Sentence-level attention features are extracted using a fixed QA-style prompt applied to each query–context pair. The prompt format is shown below:

```
Given the following information: {context}
Answer the following question based on the
given information with one or few words:
{question}
Answer:
```

Attention Feature Extraction For each prompted input, we collect decoder attention weights from the final decoding token across all layers and attention heads. Attention weights directed to tokens belonging to each sentence are aggregated and normalized to form fixed-length sentence-level feature vectors, which are then used as input to the probing classifier.

Context-Reliant Sample Selection To improve the quality of weak supervision, we retain only *context-reliant* QA examples, where access to the retrieved context is necessary for correct answering. Specifically, we compare model predictions with and without the retrieved context, using exact match (EM) or F1 as appropriate for each dataset.

For NewsQA and SQuAD, we retain examples where the model fails to answer correctly without context (memory-based EM = 0) but succeeds when the context is provided (context-based EM = 1). For HotpotQA, we retain samples with memory-based F1 ≤ 0.2 and context-based F1 ≥ 0.5, reflecting its multi-hop and partial-match evaluation setting.

Probing Classifier Training We train a logistic regression (LR) model on attention-derived features, using 5-fold cross-validation with balanced accuracy as the scoring metric. We perform grid search over regularization strengths $C \in \{0.01, 0.1, 1.0, 10.0, 100.0\}$, and use the liblinear solver with ℓ_2 regularization, class-balanced weighting, and a maximum of 2,000 iterations. The best model is selected based on AUC on the validation set.

C Baseline Descriptions

We compare Sentinel against the following baseline methods, grouped by their design paradigms:

- **LLMLingua-1/2** (Jiang et al., 2023; Pan et al., 2024): Token-level compression methods based on saliency estimation via perplexity and LLM distillation. These methods are task-agnostic and do not condition on the query.
- **Selective-Context** (Li et al., 2023): A sentence-level, task-agnostic method that scores context segments based on general informativeness, independent of the question.
- **LongLLMLingua** (Jiang et al., 2024b): A query-aware, multi-stage compression system using query-conditioned perplexity scoring, document reordering, and adaptive compression ratios.
- **CPC** (Liskavets et al., 2024): A contrastively trained sentence-ranking model that selects sentences based on semantic similarity to the query in embedding space. It is query-aware and trained on synthetic QA data.
- **Raw Attention** (Wang et al., 2024; Fang et al., 2025): A non-learning baseline that selects sentences by averaging attention weights from the final decoder token. This mimics attention-based heuristics used in prior work such as QUITO and AttentionRAG.
- **Random Selection**: Sentences are sampled uniformly at random until the token budget is met. Serves as a lower-bound reference.
- **Empty Context**: The model receives only the question without any retrieved context, serving as a zero-context baseline.

All baselines are evaluated under the same token budget and LLM generation setting for fair comparison.

D Additional Chinese Results with GPT-3.5-Turbo

To assess the cross-lingual robustness of our method, we evaluate Sentinel on LongBench-Zh using GPT-3.5-Turbo as the inference model. We compare against LLMLingua and LLMLingua-2 baselines, which are evaluated under a 3,000-token input constraint. Sentinel uses only 2,000 tokens but consistently outperforms the baselines across all task categories, as shown in Table 7.

E Additional Results on Proxy Model Size and Family

This section provides detailed experimental results of Sentinel using proxy models from different families and parameter sizes, complementing the aggregated analysis presented in the main paper. Table 6 reports the full breakdown across all LongBench tasks.

Methods	LongBench-En (2000-token constraint)				LongBench-Zh (2000-token constraint)			Overall AVG
	SingleDoc	MultiDoc	Summ.	En-AVG	SingleDoc	MultiDoc	Zh-AVG	
Sentinel (Qwen2.5-0.5B-Instruct)	37.73	46.16	23.03	35.64	62.24	18.57	40.41	38.02
Sentinel (Qwen2.5-1.5B-Instruct)	39.48	46.07	23.10	36.22	62.02	18.91	40.47	38.34
Sentinel (Qwen2.5-3B-Instruct)	39.53	47.97	23.06	36.85	62.04	19.23	40.63	38.74
Sentinel (Qwen2.5-7B-Instruct)	38.79	45.56	22.52	35.62	60.88	18.43	39.66	37.64
Sentinel (Llama-3.2-1B-Instruct)	39.43	44.96	21.90	35.43	60.64	19.18	39.91	37.67
Sentinel (Llama-3.2-3B-Instruct)	36.03	44.46	22.00	34.17	59.24	18.89	39.06	36.62
Sentinel (Llama-3.1-8B-Instruct)	36.58	45.15	22.90	34.87	60.84	19.07	39.95	37.41
Sentinel (Qwen3-0.6B)	38.12	42.55	22.77	34.48	60.04	18.51	39.27	36.88
Sentinel (Qwen3-1.7B)	36.52	42.06	22.29	33.62	60.79	17.96	39.38	36.50
Sentinel (Qwen3-4B)	37.15	43.17	22.67	34.33	59.68	17.74	38.71	36.52
Sentinel (Qwen3-8B)	36.31	42.19	22.15	33.55	60.74	17.77	39.26	36.40
Original Prompt	38.84	44.74	22.76	35.45	60.06	18.21	39.14	37.30

Table 6: Detailed Sentinel performance across proxy model families and sizes.

Methods	LongBench-Zh (GPT-3.5-Turbo, 3000-token constraint)						Compression Stats	
	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	**AVG**	Tokens	$1/\tau$
LLMLingua	35.2	20.4	11.8	24.3	51.4	28.6	3,060	5x
LLMLingua-2	46.7	23.0	15.3	32.8	**72.6**	38.1	3,023	5x
Evaluated under 2000-token constraint								
Sentinel (Qwen-2.5-0.5B-Instruct)	**64.8**	25.1	14.3	38.0	89.0	46.2	1,932	5x
Sentinel (Qwen-2.5-1.5B-Instruct)	63.3	24.9	14.8	**40.3**	**95.0**	**47.6**	1,929	5x
Original Prompt	61.2	**28.7**	**16.0**	29.2	77.5	42.5	14,940	-

Table 7: Performance comparison on LongBench-Zh using GPT-3.5-Turbo. LLMLingua baselines are evaluated under a 3,000-token budget. Sentinel uses only 2,000 tokens but consistently outperforms the baselines, demonstrating effective compression across languages.

F Ablation Details

Effect of Probing Data Size. We evaluate how training size affects probing quality. As shown in Table 8, performance remains stable across 500–3000 training examples, with only marginal gains. This suggests that even a small probing set can support effective compression.

Feature Selection Details To construct a compact attention-based feature set, we use the Minimum Redundancy Maximum Relevance (mRMR) algorithm. We first compute mutual information between each feature (i.e., attention head statistics) and the binary relevance label, selecting the most informative one. We then iteratively add features that maximize relevance while minimizing redundancy, measured via Pearson correlation with already selected features. The number of features is capped at the number of heads in a single decoder layer to ensure compactness and interpretability.

Compression Ratio. Table 9 reports results with varying compression ratios ($\tau \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$), under a fixed chunk size of 1024. Sentinel remains robust even at high compression, while Raw attention deteriorates significantly.

G Analysis of Negatively Weighted Attention Heads

To better understand the role of attention heads assigned negative weights by Sentinel (Qwen-2.5-0.5B-Instruct), we analyze their attention distributions on 100 examples from the HotpotQA dataset. This analysis examines which input components these heads predominantly attend to, and whether their negative contributions correspond to known non-informative attention behaviors.

Analysis Setup. We analyze attention patterns on 100 HotpotQA examples by grouping input tokens into four categories: *(i)* sink tokens (e.g., special tokens and structurally dominant positions), *(ii)* supporting evidence sentences, *(iii)* question tokens, and *(iv)* remaining context. For each attention head, we compute the average proportion of attention mass assigned to each category.

Results. As shown in Table 10, attention heads assigned strong negative weights by Sentinel predominantly attend to sink tokens or question tokens, while allocating little to no attention to supporting

Methods	LongBench-En (2000-token constraint)				LongBench-Zh (2000-token constraint)			Overall AVG
	SingleDoc	MultiDoc	Summ.	En-AVG	SingleDoc	MultiDoc	Zh-AVG	
Qwen-2.5-0.5B-Instruct (500)	37.29	46.94	23.25	35.83	62.04	18.42	40.23	38.03
Qwen-2.5-0.5B-Instruct (1000)	38.35	47.43	23.66	36.48	61.43	18.57	40.00	38.24
Qwen-2.5-0.5B-Instruct (2000)	36.70	47.48	22.89	35.69	61.57	18.76	40.16	37.92
Qwen-2.5-0.5B-Instruct (3000)	37.73	46.16	23.03	35.64	62.24	18.57	40.41	38.02

Table 8: Performance of 0.5B models with different probing sizes (500, 1000, 2000, 3000) on LongBench.

Methods	LongBench-En (2000-token constraint)				LongBench-Zh (2000-token constraint)			Overall AVG
	SingleDoc	MultiDoc	Summ.	En-AVG	SingleDoc	MultiDoc	Zh-AVG	
Raw Attention (ratio 0.1)	25.79	36.54	20.39	27.57	35.03	16.33	25.68	26.62
Raw Attention (ratio 0.2)	33.19	41.09	21.63	31.97	48.45	17.23	32.84	32.41
Raw Attention (ratio 0.3)	34.91	43.74	22.39	33.68	55.09	18.14	36.62	35.15
Raw Attention (ratio 0.4)	37.63	45.95	22.88	35.49	58.78	17.82	38.30	36.89
Raw Attention (ratio 0.5)	37.47	44.70	23.25	35.14	60.63	17.42	39.03	37.09
Sentinel (ratio 0.1)	37.72	41.47	22.58	33.93	58.96	19.36	39.16	36.55
Sentinel (ratio 0.2)	39.90	45.97	23.37	36.42	59.50	17.92	38.71	37.56
Sentinel (ratio 0.3)	39.45	46.51	23.86	36.61	60.98	18.68	39.83	38.22
Sentinel (ratio 0.4)	39.93	46.62	23.38	36.65	59.51	18.77	39.14	37.89
Sentinel (ratio 0.5)	38.60	46.77	23.54	36.30	61.41	18.44	39.92	38.11

Table 9: Performance across compression ratios (chunk size = 1024).

Layer	Head	Probe Weight	Sink	Supporting	Question	Others
11	1	-13.16	0.89	0.01	0.05	0.04
3	0	-12.83	0.74	0.01	0.18	0.03
3	10	-10.22	0.08	0.00	0.84	0.02
21	9	-9.95	0.01	0.00	0.98	0.01
14	5	-9.47	0.00	0.03	0.85	0.06
3	5	-9.11	0.74	0.04	0.03	0.18
9	11	-8.15	0.96	0.00	0.03	0.01

Table 10: Examples of attention heads assigned strong negative weights by Sentinel, showing attention mass concentrated on sink or question tokens rather than supporting evidence.

evidence. In contrast, positively weighted heads focus primarily on evidence-bearing context.

Implications. This analysis shows that negatively weighted heads capture structurally dominant but semantically uninformative behaviors, such as attention sinks or self-focused query attention. Explicitly down-weighting these heads allows Sentinel to suppress spurious attention patterns and decode context utilization more robustly than methods that rely on raw attention or positively identified retrieval heads alone.

H Latency and Inference Efficiency

We evaluate end-to-end inference latency across different Sentinel configurations, focusing on the effects of chunk size and attention feature design. Table 11 reports average and median latency per sample on the English LongBench dataset, measured on a single A100 GPU.[1]

With a chunk size of 1024 and All Layers attention features, Sentinel achieves 1.13× speedup over LLMLingua-2 while reaching 38.02 F1.

To further improve runtime, we evaluate SENTINEL (SELECTED), which uses compact mRMR-selected features. At chunk size 1024, this variant reduces latency to 0.60s (1.30×) with only minor performance degradation (37.24 F1), offering an efficient alternative for low-latency scenarios.

I LLM Evaluation Settings

For LLM-based evaluation, we adopt the official prompt templates and decoding settings from LongBench (Bai et al., 2024) to ensure consistency and comparability across methods. Unless otherwise specified, all decoding parameters are fixed for all datasets: the temperature is set to 0.0, the nucleus sampling parameter top_p is 1.0, the random seed is fixed to 42, only a single generation is sampled ($n = 1$), and streaming is disabled.

[1]We monkey-patch the model to extract only the final-token attention used by our method, replacing other activations with None to reduce overhead.

Method	Chunk Size	Proxy Model	Avg. Time (s) ↓	Med. Time (s) ↓	Speedup vs. LLMLingua-2 ↑	Overall-AVG
LLMLingua-2 (trained)	512	XLM-RoBERTa-Large (561M)	0.78	0.70	1.00×	28.35
Raw Attention	512	Qwen-2.5-0.5B-Instruct (494M)	1.01	0.84	0.77×	35.18
Raw Attention	1024	Qwen-2.5-0.5B-Instruct (494M)	0.65	0.54	1.20×	35.03
Sentinel (ours)	512	Qwen-2.5-0.5B-Instruct (494M)	1.02	0.84	0.76×	37.05
Sentinel (ours)	1024	Qwen-2.5-0.5B-Instruct (494M)	0.69	0.57	1.13×	38.02
Sentinel (ours) selected	1024	Qwen-2.5-0.5B-Instruct (494M)	0.60	0.49	1.30×	37.24

Table 11: Inference latency per QA sample on the full LongBench dataset (lower is better). LLMLingua-2 is trained for token-level compression and limited to 512-token chunks. Sentinel uses a smaller, untrained decoder-only proxy and supports larger chunk sizes for improved efficiency. Speedup is relative to LLMLingua-2 (chunk size 512). **Overall-AVG** denotes average accuracy across all chunk settings per method.