QUITO-X: A New Perspective on Context Compression from the Information Bottleneck Theory

Anonymous ACL submission

Abstract

Generative large language models (LLMs) have achieved remarkable success in various industrial applications, owing to their promising In-Context Learning capabilities. However, the issue of long context in complex tasks poses a significant barrier to their wider adoption, manifested in two main aspects: (i) The excessively long context leads to high costs and inference delays. (ii) A substantial amount of task-irrelevant information introduced by long contexts exacerbates the "lost in the middle" problem. Existing methods compress context by removing redundant tokens using metrics such as self-information or perplexity (PPL), which is inconsistent with the objective of retaining the most important tokens when conditioning on a given query. In this study, we introduce information bottleneck theory (IB) to model the problem, offering a novel perspective that thoroughly addresses the essential properties required for context compression. Additionally, we propose a cross-attention-based approach to approximate mutual information in IB, which can be flexibly replaced with suitable alternatives in different scenarios. Extensive experiments on four datasets demonstrate that our method achieves a 25% increase in compression rate compared to the state-of-the-art, while maintaining question answering performance. In particular, the context compressed by our method even outperform the full context in some cases.

1 Introduction

In recent years, LLMs (Achiam et al., 2023) have been widely applied to various tasks in multiple domains, such as text classification (Sun et al., 2023), question answering systems (Wang et al., 2023a), and *etc.*. As one of the most promising capabilities of these models, In-Context Learning (ICL) (Brown, 2020) plays a critical role by enabling the effective use of large language models without requiring additional training. However, in complex tasks, the need to guide the model's adaptation to the task or provide supplementary knowledge often results in excessively long context, leading to high computational costs, increased inference latency, and the "lost in the middle" problem (Tay et al., 2020). Therefore, how to compress context while maintaining model performance has become a widely studied topic.

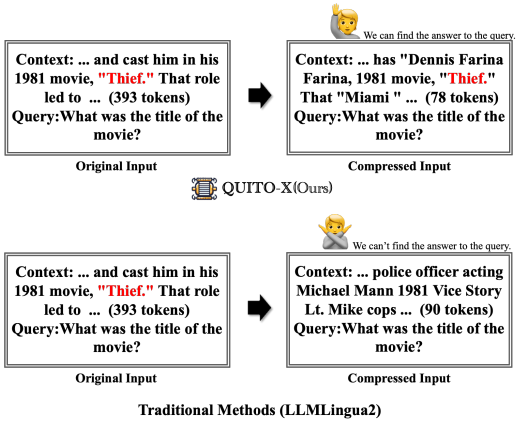

Figure 1: Comparison of our method and baseline approaches for preserving key information in model responses. Our method effectively retains critical context ("Thief"), ensuring accurate interpretation, while baseline methods fail to do so.

In the literature, Liu et al. (2023) utilize language models to compress context in a generative manner, while other methods select the most important lexical units (tokens, words, or sentences) from the original context in an extractive manner. Specifically, the generative-based compression methods typically construct compressors by fine-tuning models to generate summaries of the original text, but they are often constrained by inherent limitations of language models, such as restricted context windows, hallucination phenomena, and the "lost in the middle" problem. The extractive-based compression methods is to design appropriate metrics (e.g., self-information (Shannon, 1951),

perplexity (PPL), self-attention) to assign importance scores to each unit, thereby identifying and removing less salient units. However, the metrics used in previous works are not aligned with the optimization goals of the compressor, which may lead to suboptimal results. For example, these metrics often place excessive emphasis on nouns, while overlooking other crucial elements like prepositional phrases, quantifiers or verbs, which may have lower information entropy. However, neglecting such information can result in highly fragmented compression that is difficult to understand, ultimately leading to incorrect model outputs, as shown in Figure 2.

In this paper, we formulate this problem from an Information Bottleneck (IB) (Tishby et al., 2000; Fischer, 2020) perspective, deriving mutual information as our metric. We also provide a mathematical proof that using mutual information is equivalent to maximizing the likelihood of the compressed output, which is precisely the compressor's optimization objective. In summary, our contributions are twofold:

- **Applying Information Bottleneck Theory to Context Compression**: We introduce a novel perspective by utilizing Information Bottleneck theory to analyze the properties of context compression. This results in the mutual information metric, and we mathematically prove that it is equivalent to maximizing the likelihood of the compressed generation.
- **Experimental Validation**: We conduct extensive experiments that show significant improvements over previous work on long-context question answering. Moreover, our method reduces memory usage to 50% of the most memory-efficient baseline while achieving a 25% improvement in accuracy compared to the best-performing baseline.

2 Related Work

2.1 Extractive Context Compression

Large language models (LLMs) excel at many tasks but struggle with long inputs due to increased token costs and context truncation. ICL (Brown, 2020) alleviates some of these issues by providing task-relevant prompts but also adds to token usage and inference cost.

To address this, extractive context compression methods remove less relevant tokens or phrases while preserving essential content. Selective Context (Li et al., 2023b) ranks tokens by self-information, while LLMLingua (Pan et al., 2024; Jiang et al., 2023) compresses input based on PPL, using a coarse-to-fine strategy. QUITO (Wang et al., 2024) leverages attention from a small LLM to retain query-relevant context.

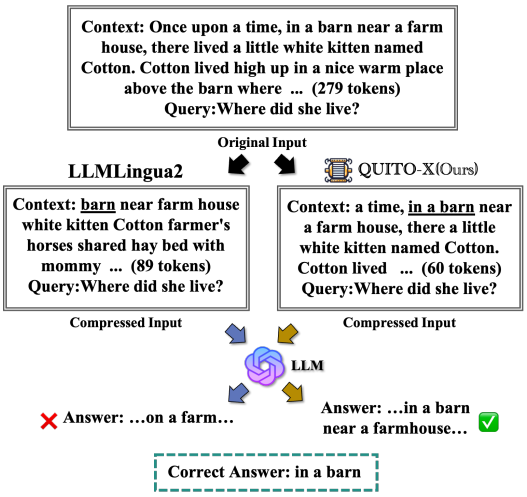

Figure 2: LLMLingua2 overly focuses on high-entropy nouns like 'barn' and 'farmhouse,' while neglecting relational words (e.g., 'near') and verbs, resulting in highly fragmented compression and leading to incorrect answers ('on a farm'). In contrast, QUITO-X retains key relational phrases ('in a barn near a farmhouse'), preserving full meaning and yielding the correct answer.

These approaches use entropy-based metrics (e.g., self-information, PPL) that frequently favor high-entropy tokens such as nouns, while underestimating the importance of function words crucial to relational semantics (Figure 2). Furthermore, these metrics are often not theoretically aligned with the underlying optimization objective, such as minimizing KL divergence, thus leading to suboptimal results.

2.2 Information Bottleneck

The Information Bottleneck (IB) principle (Tishby et al., 2000) aims to compress input X into a representation T that preserves task-relevant information $I(T;Y)$ while discarding irrelevant parts $I(T;X)$:

$$\mathcal{L}_{\text{IB}} = I(T;X) - \beta I(T;Y). \quad (1)$$

In deep learning, IB has been used to interpret representation learning (Shwartz-Ziv and Tishby, 2017) and inform model compression (Alemi et al., 2016). In NLP, recent work (Zhu et al., 2024) applies IB to filter noisy context for LLMs. Inspired

by these works, we build on the information bottleneck principle to derive a token-wise mutual information metric as our optimization objective, using cross-attention scores as a practical proxy. We theoretically prove that this metric is consistent with the maximum likelihood objective, and it achieves state-of-the-art performance across a wide range of long-context evaluation benchmarks.

3 Method

3.1 Theorem

Problem Formulation. Given the original context $X = (x_i)_{i=1}^{L}$ and the query Q, our objective is to filter out unnecessary content from the context $X = (x_i)_{i=1}^{L}$ into a reduced context $\bar{X} = (\bar{x}_i)_{i=1}^{\bar{L}}$, while maximizing the likelihood of the ground truth output Y of the large language model (LLM). This can be formulated as:

$$\max_{\bar{X}} E\left[\log\left(P(Y \mid \bar{X}, Q)\right)\right] \tag{2}$$

where L and $\bar{L}$ represent the sequence lengths of the original context X and the reduced context $\bar{X}$, respectively. The compression ratio τ is defined as $\tau = \frac{\bar{L}}{L}$

IB Perspective. To balance τ and the likelihood of Y, we formulate our task as an optimization problem from an information bottleneck perspective(Tishby et al., 2000):

$$\mathcal{L}_{\text{IB}} = I(\bar{X}; X \mid Q) - \beta I(\bar{X}; Y \mid Q) \tag{3}$$

where minimizing the first term improves efficiency, and maximizing the second term ensures correctness.

In the following discussion, we fix the compression ratio τ as a constant k. Under this condition, the cost savings from compression are fixed, allowing us to ignore the first term and focus solely on maximizing the second term:

$$\max_{\bar{X}} I(\bar{X}; Y \mid Q) \quad \text{s.t. } \tau = k \tag{4}$$

The following Theorem 1 demonstrates the consistency between our modeling and the optimization objective of the task.

Theorem 1. Under our setting, our optimization objective (5) is equivalent to (4):

$$\max_{\bar{X}} I_Q(\bar{X}; Y) \sim \max_{\bar{X}} \mathbb{E}[\log P(Y \mid \bar{X}, Q)] \quad \text{s.t. } \tau = k. \tag{5}$$

The detailed proof is provided in the Appendix B.

Using the chain rule of Mutual Information, we have

$$I(X; Y \mid Q) = I_Q(x_1; Y \mid Q) + ... + I_Q(x_n; Y \mid x_1, x_2 ... x_{n-1}, Q) \tag{6}$$

Thus, We can break the mutual information between X and Y into the mutual information between each token x_i and Y. we utilize

$$s(x_i) = I(x_i; Y \mid x_1, x_2, ... x_{i-1}, Q)$$

as a metric to measure the importance score of token x_i, from which we can identify the tokens to retain and those to remove. However, it is difficult to compute the mutual information $s(x_i)$ directly due to the following reasons: (i) We cannot access the ground truth output Y in practical scenarios. (ii) Even if we use the output of a language model Y_{LM} to approximate Y, the result of $s(x_i)$ cannot be directly inferred from the probability sampled by the language model.

Therefore, we need to establish a computationally feasible metric to approximate mutual information. Inspired by works in the fields of computer vision and multi-modal learning (Dosovitskiy et al., 2021; Esser et al., 2024), which often measure the correlation between two types of information I_1 and I_2 using either cross-attention between them or self-attention after concatenating I_1 and I_2, We conducted several detailed experiments, exploring various strategies for both cross-attention and self-attention, along with other metrics, to determine which method best approximates mutual information. Ultimately, we found that using an encoder-decoder architecture, with X and Q as inputs, and leveraging the cross-attention values between the first token of the output Y and x_i, is the most suitable approach to approximate mutual information in our case. The specific experimental details are provided in the Appendix A.

Merging into Lexical Units. Following Li et al. (2023b), we also merge tokens into words as lexical units to avoid disjoint contexts. We denote w as a word, l_w as the length of the word, and $x_i, x_{i+1}, \ldots, x_{i+l_w-1}$ as the tokens comprising the word w and x_{prev} represents the preceding context. Benefited from the addition of mutual information,

$$I(x_i, ..., x_{i+l_w-1} \mid x_{prev}, Y, Q) = I(x_i \mid x_{prev}, Y, Q) + ... + I(x_{i+l_w-1} \mid x_{prev}, x_i, ..., x_{i+n-2}, Q) \tag{7}$$

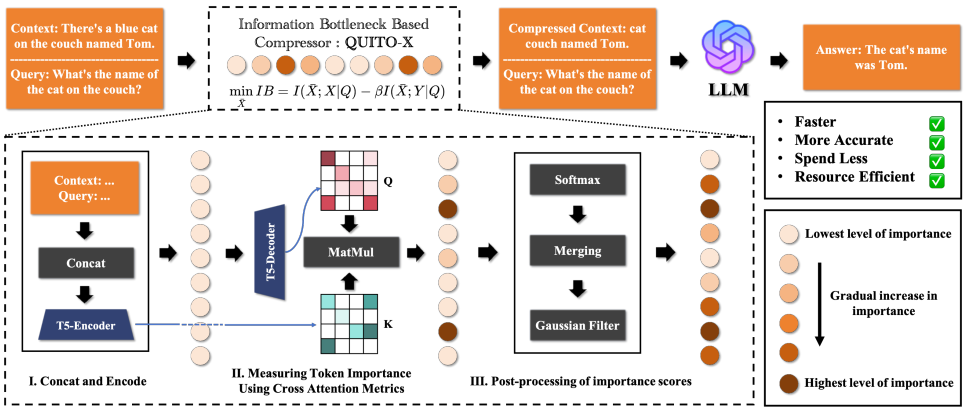

Figure 3: Overview of the proposed method for extracting cross-attention scores using a T5 model. The figure illustrates the process of filtering the context to retain the most relevant information for answering a specific query.

we can directly sum the $s(x_i)$ of all tokens x_i in a word w to represent $s(w)$.

Gaussian Smoothing. We observed that relying solely on independent metrics for each lexical unit often prioritizes nouns, which typically have high information entropy, while overlooking intermediate conjunctions, verbs, and prepositions. This leads to semantic ambiguity and hampers understanding by large models. To mitigate this issue further, we applied a Gaussian filter on word-level scores

$$s(w) = \sum_{k=-K}^{K} s(w+k) \cdot g(k)$$

$$g(k) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{k^2}{2\sigma^2}\right)$$

which helps preserve the information surrounding important units. The detail could be found in section 3.2

3.2 Algorithm

Our method compresses long contexts into concise, informative representations through three key steps, as shown in Figure 3:

Concat and Encode: The X and Q are concatenated into a single input sequence $X + Q$ and fed into the f_{enc}. This produces a sequence of hidden representations that captures the semantic and positional information of the input tokens:

$$\{h_t\} = f_{enc}(X + Q) \tag{8}$$

Here, h_t represents the hidden representation of the t-th token.

Measuring Token Importance: During the decoding process, the cross-attention mechanism f_{attn} is leveraged to compute the importance of each token in the context relative to the query. Specifically, hidden representation of the decoder's first token $h_{<start>}$ attends to all tokens in the encoded sequence via the cross-attention mechanism:

$$\{a_t\} = f_{attn}(\{h_t\}, h_{<start>}) \tag{9}$$

Here, a_t denotes the attention score assigned to the t-th token, reflecting its relative importance with respect to the query.

Post-processing of Importance Score: The attention weights for context tokens are extracted, averaged across all attention heads, and normalized using a softmax function.

$$s(t) = \frac{\exp a_t}{\sum_{token \in f_{tok}(X)} \exp a_{token}}, t \in f_{tok}(X) \tag{10}$$

We use f_{tok} for tokenization, these scores represent the relevance of each token in the tokenized context to the given query.

The normalized token scores are aggregated at the word level:

$$s(w) = \sum_{t \in w} s(t), w \in X \tag{11}$$

To account for the contextual importance of words, a Gaussian filter is applied to the word-level scores. This ensures that words appearing near important terms also receive elevated scores:

$$s(w) = \sum_{k=-K}^{K} s(w+k) \cdot g(k) \tag{12}$$

$$g(k) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{k^2}{2\sigma^2}\right) \tag{13}$$

Based on the smoothed scores, we retain only the most relevant words to form the compressed context. The compression ratio τ can be adjusted to control the level of detail retained. The function f_{top} selects words whose scores are among the top τ proportion:

$$\bar{X} = f_{top}(\{s(w)\}, \tau), w \in X \tag{14}$$

This algorithm effectively reduces context length while retaining essential information, ensuring accurate and efficient performance in downstream tasks.

4 Experiments

4.1 Datasets and Metrics

We conduct experiments on nine datasets that vary in text length and task type, covering both manageable and excessively long contexts:

(i) **CoQA (Reddy et al., 2019) and Quoref (Dasigi et al., 2019):** These datasets feature texts of moderate length, within the processing capability of large models, making them ideal for standard evaluations of model performance.

(ii) **2WikiMultiHopQA (Ho et al., 2020), HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), TriviaQA (Joshi et al., 2017), and Gov_Report (Huang et al., 2021):** These datasets are part of the LongBench benchmark (Bai et al., 2023), which focuses on long-context understanding across diverse NLP tasks such as multi-doc QA, few-shot QA, and summarization. They typically feature excessively long inputs that challenge models' ability to retain and reason over relevant information, often suffering from the "lost in the middle" phenomenon.

To evaluate model accuracy, we adopt the Exact Match (EM) metric for question answering datasets, which measures the percentage of predictions that exactly match the ground truth answers. For the summarization dataset Gov_Report, we report ROUGE-L (Lin, 2004), a widely used metric that assesses the overlap between generated summaries and reference summaries.

4.2 Implementation Details

We employed the FLAN-T5-small model (Chung et al., 2024) for compression. Our approach leverages Huggingface Transformers and PyTorch 2.1.0 with CUDA-12.1. For question-answering tasks, we utilized LongChat-13B-16k (Li et al., 2023a) and LLaMA3-8B-Instruct (AI@Meta, 2024).

In our experiments, we observed that the choice of the parameter σ in (13) does not significantly impact the compression performance as long as $\sigma \neq 0$. Therefore, for consistency, we set $\sigma = 1$ for all subsequent experiments. Detailed parameter search results are provided in the Appendix D.

For CoQA (Reddy et al., 2019) and Quoref (Dasigi et al., 2019), we evaluated model accuracy using the original context and without any context, aiming to assess the models' ability to summarize with full information and rely on prior knowledge. Next, we tested five baseline methods and our proposed approach at compression ratios of 0.75, 0.50, and 0.25, measuring accuracy with the compressed context using both LongChat-13B-16k and LLaMA3-8B-Instruct models.

For datasets with long contexts, including 2WikiMultiHopQA (Ho et al., 2020), HotpotQA (Yang et al., 2018), and MuSiQue (Trivedi et al., 2022), TriviaQA (Joshi et al., 2017), Gov_Report (Huang et al., 2021), sourced from LongBench (Bai et al., 2023), we focused on the LLaMA3-8B-Instruct model. To handle the extreme length of these texts, a chunking strategy was adopted, dividing the context into 512-token chunks. Two strategies were tested:

Strategy 1: Compressing each chunk individually and then merging the compressed representations.
Strategy 2: Calculating attention scores between each chunk and the query, merging these attention scores across all chunks, and then performing a unified compression on the merged context.

4.3 Baseline

We compared against the following context compression baselines in Table 1: (1) **Selective Context** (Li et al., 2023b): Uses GPT-2 (Radford et al., 2019) to retain context segments based on self-information. (2) **LLMLingua** (Pan et al., 2024): Employs Llama-2-7b (Touvron et al., 2023) with dynamic compression driven by context PPL. (3) **LongLLMLingua** (Jiang et al., 2023): Extends LLMLingua for longer contexts, also using Llama-2-7b (Touvron et al., 2023). (4) **LLMLingua2** (Pan et al., 2024): Utilizes XLM-RoBERTa-large (Conneau, 2019), introducing data distillation for compression. (5) **QUITO** (Wang et al., 2024): Applies Qwen2-0.5B-Instruct (Yang et al., 2024) with attention mechanisms to selectively retain query-relevant context.

For datasets with manageable text lengths, such as **CoQA** (Reddy et al., 2019) and **Quoref** (Dasigi

Algorithm	Architecture	Model	Parameters
Selective Context	Transformer Decoder-Only	GPT-2	124M
LLMLingua	Transformer Decoder-Only	Llama-2-7b	7B
LongLLMLingua	Transformer Decoder-Only	Llama-2-7b	7B
LLMLingua2	Transformer Encoder-Only	XLM-RoBERTa-large	355M
QUITO	Transformer Decoder-Only	Qwen2-0.5b-Instruct	500M
QUITO-X	**Transformer Encoder-Decoder**	**FLAN-T5-small**	**80M**

Table 1: Comparison of different compression algorithms in terms of architecture, model, and parameter size. Our method, based on the FLAN-T5-small model, demonstrates the effectiveness of a compact Transformer Encoder-Decoder architecture with only 80M parameters, significantly reducing computational cost while maintaining or exceeding performance compared to larger models like LLMLingua (7B) and QUITO (500M).

dataset	model	ratio	Selective-Context	LLMLingua	LongLLMLingua	LLMLingua2	QUITO	**QUITO-X**
Quoref	LongChat	1.00	70.6	70.6	70.6	70.6	70.6	70.6
		0.75	65.3	46.4	46.5	65.7	65.6	**68.1**
		0.50	55.8	34.5	34.6	55.0	59.4	**65.1**
		0.25	40.9	28.2	28.7	41.5	52.3	**60.8**
		0.00	2.9	2.9	2.9	2.9	2.9	2.9
	Llama-3	1.00	93.1	93.1	93.1	93.1	93.1	93.1
		0.75	90.3	64.9	65.3	90.7	89.8	**92.6**
		0.50	81.3	51.1	51.4	82.6	84.4	**90.2**
		0.25	59.3	43.2	43.3	65.5	75.8	**86.8**
		0.00	6.8	6.8	6.8	6.8	6.8	6.8
CoQA	LongChat	1.00	59.1	59.1	59.1	59.1	59.1	59.1
		0.75	56.6	44.9	45.4	57.5	54.6	**59.6**
		0.50	47.0	36.3	36.4	50.3	50.4	**59.5**
		0.25	32.1	30.4	25.9	41.0	41.4	**55.5**
		0.00	13.8	13.8	13.8	13.8	13.8	13.8
	Llama-3	1.00	79.3	79.3	79.3	79.3	79.3	79.3
		0.75	76.5	62.3	61.8	74.8	73.1	**79.5**
		0.50	64.1	50.9	50.4	69.4	64.6	**78.1**
		0.25	45.3	43.0	37.3	57.7	53.5	**75.5**
		0.00	18.1	18.1	18.1	18.1	18.1	18.1

Table 2: Experimental results of various compression methods applied at different compression ratios on the Quoref and CoQA datasets. The table shows the effectiveness of different methods, including Selective-Context, LLMLingua, LongLLMLingua, LLMLingua2, QUITO, and QUITO-X, across different compression ratios (1.00, 0.75, 0.50, 0.25, and 0.00). Our method consistently achieves the best performance at all ratios.

et al., 2019), we evaluated our method against all listed baselines. These datasets allowed us to test the effectiveness of each approach in compressing contexts without encountering extreme text length challenges.

For datasets with long contexts, including **2WikiMultiHopQA** (Ho et al., 2020), **HotpotQA** (Yang et al., 2018), **MuSiQue** (Trivedi et al., 2022), **TriviaQA** (Joshi et al., 2017), and **Gov_Report** (Huang et al., 2021), we focus our comparison on **LLMLingua2**, as well as two additional baselines: **Selective Context** and **Quito**. These datasets pose different challenges: the QA datasets (multi-doc QA and few-shot QA) often suffer from the "lost in the middle" phenomenon, while the summarization dataset (**Gov_Report**) requires models to preserve critical information across lengthy documents. Together, they provide a comprehensive evaluation of our method's performance in long-context scenarios across diverse task types.

dataset	task	ratio	Selective-Context	QUITO	LLMLingua2	strategy 1	strategy 2
2wikimqa	Multi-Doc QA	1.00	55.0	55.0	55.0	55.0	55.0
		0.75	59.0	56.0	64.0	**64.0**	60.5
		0.50	54.5	58.5	68.0	67.5	**69.0**
		0.25	49.0	51.0	53.5	**61.5**	60.0
hotpotqa	Multi-Doc QA	1.00	15.5	15.5	15.5	15.5	15.5
		0.75	19.0	21.5	25.5	**31.0**	30.0
		0.50	38.5	57.0	57.5	**65.5**	63.0
		0.25	46.5	55.0	52.5	63.0	**69.5**
musique	Multi-Doc QA	1.00	2.5	2.5	2.5	2.5	2.5
		0.75	2.5	2.5	2.5	**4.0**	3.5
		0.50	10.0	37.0	40.5	41.5	**43.5**
		0.25	35.0	36.0	40.0	43.0	**49.0**
Gov_Report	Summ.	1.00	16.50	16.50	16.50	16.50	16.50
		0.75	16.30	17.44	17.39	17.72	**17.95**
		0.50	18.21	**19.12**	18.46	19.12	19.02
		0.25	17.96	19.12	18.04	**19.12**	18.90
TriviaQA	Few-shot QA	1.00	15.0	15.0	15.0	15.0	15.0
		0.75	19.0	20.0	22.0	**28.5**	25.0
		0.50	27.5	32.5	22.0	**42.0**	38.5
		0.25	36.5	**62.5**	37.5	59.0	60.0

Table 3: Performance comparison across datasets under different compression ratios. We evaluate multi-doc QA, summarization, and few-shot QA tasks with Exact Match or ROUGE-L. Bold numbers indicate the best performance for each dataset and ratio combination.

4.4 Experimental Results

The results shown in Table 2 and Table 3 comprehensively demonstrate the effectiveness of our proposed methods across various datasets and compression ratios.

For the Quoref and CoQA datasets (Table 2), our proposed **QUITO-X** consistently outperforms existing baselines, including Selective-Context, LLMLingua, LongLLMLingua, LLMLingua2, and QUITO, under all tested compression ratios (1.00, 0.75, 0.50, 0.25, and 0.00). Remarkably, **QUITO-X** achieves superior performance even at higher compression ratios, where significant portions of context are removed. This robust performance highlights the capability of our method in retaining critical information despite substantial context reductions. In some cases, particularly noted in the underlined sections of Table 2, our method even surpasses the performance of the original, uncompressed context. This suggests that our approach not only removes irrelevant noise but also enables the model to focus better on relevant portions of the context, thereby improving prediction quality.

For long-text datasets (Table 3), including 2WikiMultiHopQA, HotpotQA, MuSiQue, TriviaQA, and Gov_Report, the supplementary experiments further validate the adaptability and robustness of our strategies under varying compression levels.

In the multi-doc QA datasets (2WikiMultiHopQA, HotpotQA, and MuSiQue), both proposed strategies (**Strategy 1** and **Strategy 2**) consistently outperform the baselines. For example, in 2WikiMultiHopQA, **Strategy 1** achieves the best result at a compression ratio of 0.75, while **Strategy 2** excels at 0.50. In HotpotQA, **Strategy 2** demonstrates the highest performance at 0.25 and 0.50 ratios. In MuSiQue, **Strategy 2** shows a clear advantage at lower ratios, particularly under the most aggressive compression (0.25).

On the few-shot QA dataset TriviaQA, our method also achieves consistent improvements over baselines across different compression ratios. This result highlights the effectiveness of our approach even in scenarios with limited supervision and long input contexts.

For the summarization dataset Gov_Report, our

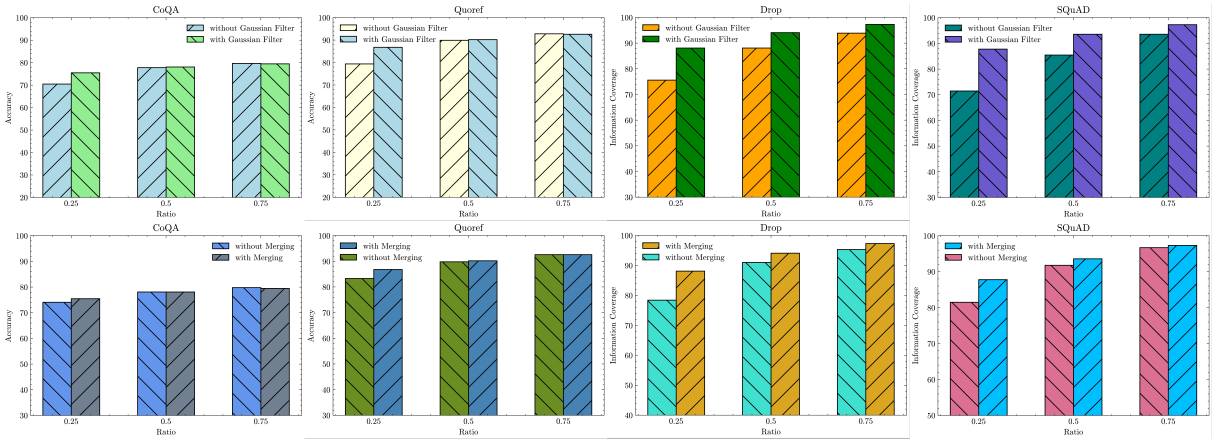

Figure 4: Ablation study results on four datasets (CoQA, Quoref, DROP, SQuAD) under three compression ratios (0.25, 0.5, 0.75). The top row shows the impact of the Gaussian filter on accuracy and information coverage, demonstrating consistent improvements across all datasets and compression ratios. The bottom row illustrates the effect of the merging module, highlighting its importance in recovering meaningful representations, particularly under higher compression ratios.

method yields higher ROUGE-L scores compared to other baselines, particularly under medium and high compression levels. This demonstrates that our strategy not only maintains key information but also preserves summary quality even with significantly reduced context, which is especially important in summarization tasks involving lengthy documents.

These results collectively underscore the robustness, adaptability, and overall effectiveness of our proposed methods for handling compressed contexts across a variety of datasets, task types, and compression scenarios.

4.5 Ablation Study

Gaussian Filter. The top row of Figure 4 shows the effect of the Gaussian filter across different datasets and compression ratios (0.25, 0.5, 0.75). For CoQA and Quoref, we use accuracy as the evaluation metric, while for DROP and SQuAD, we adopt information coverage, which we explain further in the Appendix C. The Gaussian filter consistently improves performance, particularly at lower ratios. For example, in SQuAD, information coverage increases significantly (from 71.5 to 87.8) at the 0.25 ratio. These results demonstrate its effectiveness in retaining critical context information during compression.

Merging. The bottom row of Figure 4 highlights the impact of the merging module. Merging consistently boosts accuracy and information coverage, especially at the 0.25 ratio where context loss is severe. For instance, in DROP, merging improves information coverage by nearly 10 points. This confirms its role in preserving meaningful context under high compression.

4.6 Comparison with Sentence-Level Compression

To further evaluate the effectiveness of our token-level compression approach, we compare it against FILCO (Wang et al., 2023b), a sentence-level method that compresses long contexts by selecting salient sentences. We follow FILCO's experimental protocol and preprocessing pipeline on NQ and TQA, using their released datasets and settings to ensure a fair comparison.

As shown in Appendix G, our method outperforms FILCO under comparable compression ratios (25% and 50%) on both datasets.

5 Conclusion

In this paper, we aim to tackle the challenge of context compression. Leveraging information bottleneck theory, we derive mutual information as the optimization objective, which we prove to be equivalent to maximizing likelihood. Our method significantly outperforms strong baselines in both inference latency and performance. Furthermore, it excels on long texts, occasionally surpassing models that utilize the original context, likely by eliminating inherent redundancy in the context. More effective chunking strategies for long texts are left for future exploration.

Limitations

Despite the strong performance and efficiency gains demonstrated by our method, there are several limitations worth noting:

First, due to the restricted context window of smaller language models, our approach relies on chunking strategies to process long documents. While this proves effective across many datasets, chunking inevitably breaks the global context and may lead to semantic discontinuities between chunks. How to maintain coherence across chunk boundaries—or to quantify the impact of such fragmentation—remains an open research question.

Second, since our method performs compression at the token level, the resulting outputs can suffer from reduced human readability. Compared to sentence-level or summarization-based methods, token-level outputs tend to appear fragmented or syntactically incomplete. Although this does not impair the model's ability to interpret the compressed input and answer questions accurately, it may reduce the interpretability of the compression decisions from a human perspective.

That said, as we demonstrate in Appendix H, our approach retains significantly better semantic continuity and readability compared to other token-level baselines (e.g., LLMLingua2 and QUITO). This highlights the potential of our method to strike a balance between compression granularity and human interpretability. Future work may explore ways to further enhance this trade-off, for example by integrating syntactic structure or discourse markers into the token selection process.

Finally, due to computational constraints, we were unable to conduct broader-scale experiments across more diverse domains. As a result, certain hyperparameters—such as the Gaussian smoothing parameter σ—have not been comprehensively tuned across all datasets. While our experiments suggest the method is relatively stable under reasonable variations of σ, further large-scale validation would strengthen the generalizability claims.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

A Conneau. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Pradeep Dasigi, Nelson F Liu, Ana Marasović, Noah A Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. *arXiv preprint arXiv:1908.05803*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *Preprint*, arXiv:2010.11929.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. 2024. Scaling rectified flow transformers for high-resolution image synthesis. *Preprint*, arXiv:2403.03206.

Ian Fischer. 2020. The conditional entropy bottleneck. *Entropy*, 22(9):999.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Cody C. T. Kwok, Oren Etzioni, and Daniel S. Weld. 2001. Scaling question answering to the web. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, page 150–161, New York, NY, USA. Association for Computing Machinery.

Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. 2023a. How long can context length of open-source llms truly promise? In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. 2023b. Compressing context to enhance inference efficiency of large language models. *arXiv preprint arXiv:2310.06201*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiongnan Liu, Jiajie Jin, Zihan Wang, Jiehan Cheng, Zhicheng Dou, and Ji rong Wen. 2023. Reta-llm: A retrieval-augmented large language model toolkit. *ArXiv*, abs/2306.05212.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *arXiv preprint arXiv:2403.12968*.

Dragomir R. Radev, Hong Qi, Harris Wu, and Weiguo Fan. 2002. Evaluating web-based question answering systems. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Claude E Shannon. 1951. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64.

Ravid Shwartz-Ziv and Naftali Tishby. 2017. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. *arXiv preprint arXiv:2305.08377*.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2020. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*.

Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Jinyuan Wang, Junlong Li, and Hai Zhao. 2023a. Self-prompted chain-of-thought on large language models for open-domain multi-hop reasoning. *arXiv preprint arXiv:2310.13552*.

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*.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023b. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Kun Zhu, Xiaocheng Feng, Xiyuan Du, Yuxuan Gu, Weijiang Yu, Haotian Wang, Qianglong Chen, Zheng Chu, Jingchang Chen, and Bing Qin. 2024. An information bottleneck perspective for effective noise filtering on retrieval-augmented generation. *arXiv preprint arXiv:2406.01549*.

A Experimental Selection of Mutual Information Metric

A.1 Motivating Observation

To identify a metric that best approximates the mutual information $I(X;Y \mid Q)$, we designed the

following experiment: we filtered a subset from the Drop QA dataset, denoted as $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^n = \{X_i, Y_i, Q_i\}_{i=1}^n$. In $\mathcal{D}$, Y_i is a substring of X_i. The substring Y_i within X_i (hereafter referred to as Sub_{Y_i}) captures the majority of the mutual information between X_i and Y_i. Informally, the higher the relative value of a metric on the tokens of these substrings, the better the metric can measure $I(X; Y \mid Q)$.

A.2 Experiment

We tested several commonly used metrics, including self-attention (Wang et al., 2024) and self-information (Li et al., 2023b). Cross-attention is a prevalent metric for measuring the correlation between two pieces of information. We used Flan-T5-small (Chung et al., 2024) to compute cross-attention and implemented the following two strategies for each $\mathcal{D}_i$:

cross attn first. Compute only the cross-attention scores between the first token `<start>` in Y_i and each token in X_i.

cross attn total. Autoregressively generate Y_i and compute the average sum of the cross-attention scores between all tokens in Y_i and all tokens in X_i.

We adopted Mean Reciprocal Rank (MRR) (Kwok et al., 2001; Radev et al., 2002) to evaluate which metric better represents mutual information. Specifically, for each metric, we first calculate the MRR for each data point $\mathcal{D}_i = \{X_i, Y_i, Q_i\}$ individually. For a given $\mathcal{D}_i$, we calculate the value of each token based on the metric, sort them to obtain their rank array, and then compute MRR assuming Sub_{Y_i} has a length of len and appears at positions $k, \ldots, k + len - 1$:

$$\text{MRR}_i = \frac{1}{len} \sum_{j=1}^{len} \frac{1}{\text{rank}_{k+j-1}}$$

Finally, the overall MRR for the dataset $\mathcal{D}$ is obtained by averaging MRR_i across all data points:

$$\text{MRR} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \text{MRR}_i$$

A.3 Result

The experimental results are presented in Figure 5. The results indicate that using the cross-attention value between the first token of output Y and each x_i yields a significantly higher MRR compared to other methods.

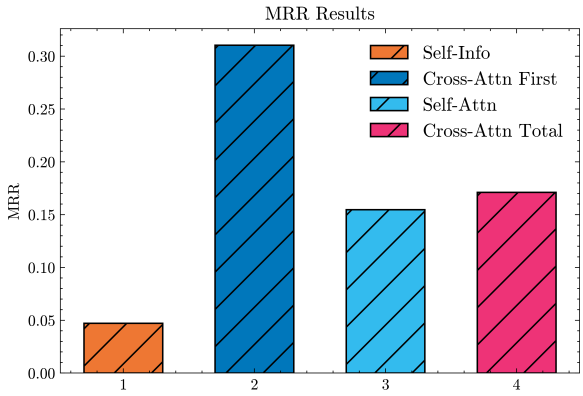

Figure 5: MRR results

B Proof of Theorem 1

Let X be the original context, Q be the query, Y be the output, and $\bar{X}$ be the extractive compressed result. Denote τ as the compression rate, and let k be a constant such that $k \in (0, 1]$.

Theorem

$$\max_{\bar{X}} I_Q(\bar{X}; Y) \sim \max_{\bar{X}} \mathbb{E}[\log P(Y \mid \bar{X}, Q)] \quad \text{s.t. } \tau = k. \tag{15}$$

(To simplify the notation, we use I_Q to represent the condition on Q.)

Proof: We start by expanding the mutual information term $I_Q(\bar{X}; Y)$:

$$\begin{aligned}
&I_Q(\bar{X}; Y) = \\
&\int_{\bar{x},y,q} P(\bar{x}, y \mid q) \log \left(\frac{P(\bar{x}, y \mid q)}{P(\bar{x} \mid q) P(y \mid q)} \right) d\bar{x}\, dy\, dq \\
&= \int_{\bar{x},y,q} P(\bar{x}, y \mid q) \log \left(\frac{P(\bar{x}, y \mid q)}{P(\bar{x} \mid q)} \right) d\bar{x}\, dy\, dq \\
&- \int_{y,q} \log P(y \mid q) \left(\int_{\bar{x}} P(\bar{x}, y \mid q) d\bar{x} \right) dy\, dq \\
&= \int_{\bar{x},y,q} P(\bar{x}, y \mid q) \log \left(\frac{P(\bar{x}, y \mid q)}{P(\bar{x} \mid q)} \right) d\bar{x}\, dy\, dq \\
&- \int_{y,q} \log P(y \mid q) P(y \mid q)\, dy\, dq
\end{aligned}$$

Since $\int_{y,q} \log P(y \mid q) P(y \mid q)\, dy\, dq$ does not affect the optimization, we ignore it:

$$\begin{aligned}
&I_Q(\bar{X}; Y) \\
&\sim \int_{\bar{x},y,q} P(\bar{x}, y \mid q) \log \left(\frac{P(\bar{x}, y \mid q)}{P(\bar{x} \mid q)} \right) d\bar{x}\, dy\, dq \\
&= E_{\bar{X},Y,Q} \left[\log P(y \mid \bar{x}, q) \right].
\end{aligned}$$

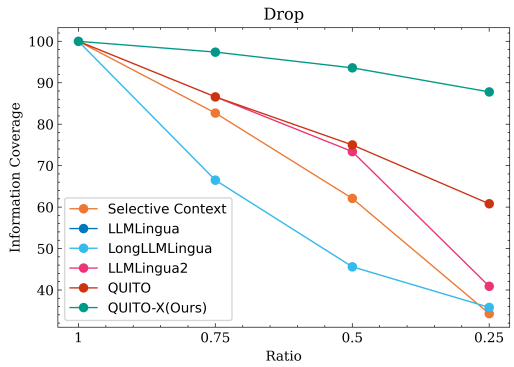

Figure 6: Information coverage on Drop.

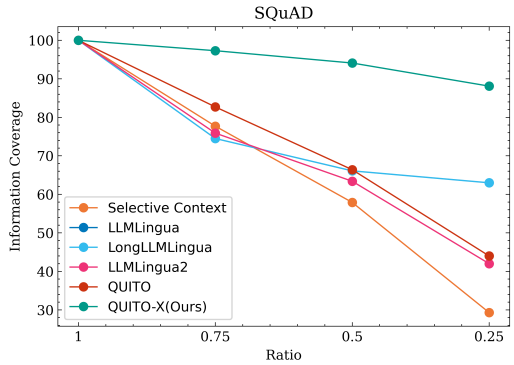

Figure 7: Information coverage on SQuAD.

Here $\bar{x}, y, q$ represent specific data points sampled from the random variables $\bar{X}, Y, Q$, respectively. This completes the proof.

C Information Coverage

In this section, we explain the Information Coverage metric used in our ablation study for DROP and SQuAD datasets. Unlike accuracy, which directly measures the correctness of the model's predictions, Information Coverage focuses on whether key information (i.e., the source of the answer) is preserved after context compression.

Specifically, we adopt EM as the evaluation metric for measuring coverage. Given a compressed context and a target answer, EM evaluates whether the answer's source can still be precisely matched within the compressed context. This ensures that critical information needed to derive the answer is retained post-compression. A higher EM score indicates better preservation of essential information, thus reflecting the compression method's effectiveness in maintaining important content.

Figures 6 and 7 showcase the Information Coverage at different compression ratios (from 1.0 to 0.25) on the DROP and SQuAD datasets. These results are independent of the ablation experiments and are intended to highlight the robustness of our proposed method under varying levels of compression.

From the figures, it is evident that across all compression ratios, our method consistently achieves the highest Information Coverage compared to baseline approaches. This demonstrates the effectiveness of our method in preserving critical answer-related information, even as the context length is reduced. Notably, at lower compression ratios (e.g., 0.25), where information loss is more severe, our approach still outperforms other methods by a clear margin, underscoring its ability to prioritize and retain essential content.

These findings further confirm that our method can effectively mitigate the challenges of information loss during compression while maintaining performance in downstream tasks.

D Parameter Search for σ

In our experiments, we examined the effect of different values of the parameter σ on the performance of the compression technique. Specifically, σ controls the variance of the Gaussian filter used during context compression. To explore its impact, we conducted a parameter search across several values of σ, ranging from 1 to 5, to assess how variations in σ influence model performance at different compression ratios.

Figure 8 shows the results of this search, where we measured the model's accuracy and information coverage at compression ratios of 0.75, 0.50, and 0.25.

From our observations, we found that the value of σ had minimal impact on performance for non-zero values, with only a slight variation in both accuracy and information coverage. Based on these findings, we chose $\sigma = 1$ as the default value for all subsequent experiments, ensuring both consistent and efficient compression without substantial loss in performance.

For a detailed breakdown of the parameter search, see the plot in Figure 8, which illustrates how σ affects model performance across all datasets tested.

E Computational Overhead Analysis

The computational overhead of our approach primarily arises from calculating the cross-attention

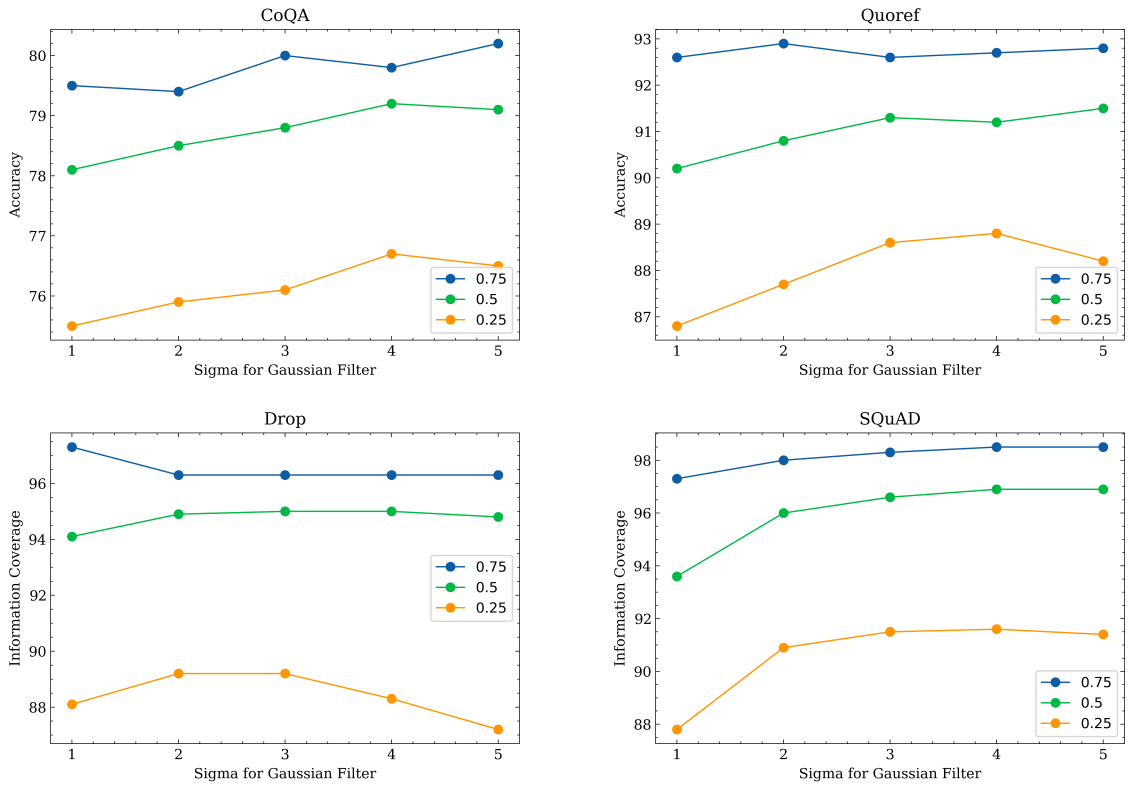

Figure 8: parameter search across several values of σ

during inference with a relatively small proxy model. Similarly, the PPL-based method incurs additional time overhead from computing log-likelihood during inference using the same proxy model. In both methods, the time overhead is approximately equivalent to one round of inference by the proxy model.

E.1 Inference Time per 512 Tokens

The table below details the inference time per 512 tokens for different models:

Model	Time per 512 Tokens
Llama3-8B	2.4251s
Flan-T5-Small	0.3238s

Table 4: Inference time per 512 tokens for different models.

For our method, we use FLAN-T5-Small, a model with only 80M parameters, as the proxy model. This makes the additional time overhead negligible. The efficiency gains from our approach far outweigh this minimal time cost. Furthermore, it is important to note that while our method and the PPL-based method theoretically share the same additional time cost when employing the same proxy model, prior works typically use much larger models as proxies. This makes our method more efficient in practice.

F Comparison with Different FLAN-T5 Model Sizes

To demonstrate the versatility of our approach, we compared models with different sizes of the encoder-decoder architecture. Specifically, we used various models from the Flan-T5 series (Flan-T5-small, Flan-T5-base, Flan-T5-large), as there are no other encoder-decoder models that rival Flan-T5 within the same time frame. Older models like BART (2019) and T5 (2019) show a significant performance gap compared to Flan-T5. For efficiency reasons, we primarily utilized Flan-T5-Small in our experiments. We also benchmarked Flan-T5-Base and Flan-T5-Large, with their results showing similarly promising trends, as shown in the table 5.

G Comparison with Sentence-Level Compression Methods

To compare our token-level compression with sentence-level methods, we replicate FILCO's experimental setup on NQ and TQA, two question answering benchmarks with long input contexts.

Ratio	Dataset	Small	Base	Large
0.75	Squad	97.3	98.3	98.2
0.5		94.1	96.4	95.6
0.25		88.1	92.1	90.4
0.75	Quoref	92.6	92.4	92.2
0.5		90.2	90.1	90.3
0.25		86.8	89.4	89.9
0.75	CoQA	79.5	80.3	80.1
0.5		78.1	78.6	79.9
0.25		75.5	77.8	77.5

Table 5: Evaluation results for different sizes of FLAN-T5 models on various datasets.

We use the same preprocessed datasets and evaluation protocol as described in FILCO's original paper to ensure fair comparison.

Table 6 summarizes the results under 25% and 50% compression ratios.

Method	NQ	TQA
FILCO (44%-64%)	44.24	59.50
Ours (50%)	**60.91**	60.19
Ours (25%)	56.79	**60.95**

Table 6: Comparison with sentence-level compression (FILCO) on NQ and TQA under 25% and 50% compression. Our method consistently outperforms FILCO.

Compared to sentence-level approaches like FILCO, our method achieves superior performance and offers precise compression rate control, making it particularly effective in low-budget scenarios where retaining critical information is crucial.

H Case Studies on Readability and Semantic Continuity

To evaluate the readability and semantic integrity of the compressed outputs, we conducted case studies comparing our method with several strong baselines, including LLMLingua2, QUITO, and their variants. Figure 9 and 10 illustrate representative examples.

These examples support our claim that while token-level compression tends to reduce syntactic completeness, our method produces more coherent and interpretable outputs than other token-level baselines, making it more suitable for applications where transparency matters.

Original Prompt (Census QA Example):
As of the census of 2000, there were 218,590 people, 79,667 households, and 60,387 families residing in the county. The population density was 496 people per square mile (192/km²). There were 83,146 housing units at an average density of 189 per square mile (73/km²). The racial makeup of the county was 86.77% Race (United States Census), 9.27% Race (United States Census), 0.23% Race (United States Census), 1.52% Race (United States Census), 0.06% Race (United States Census), 0.69% from Race (United States Census), and 1.47% from two or more races. 1.91% of the population were Race (United States Census) or Race (United States Census) of any race. 22.5% were of German people, 13.1% Irish people, 9.8% Italian people, 9.2% English, 8.1% "American" and 6.0% Polish ancestry.
Question: Which group from the census is smaller: German or English?

Compressed Prompt (LLMLingua2):
2000, 218,590 79,667 households 60,387 families 496 83,146 units 189 racial makeup 86.77% 1.47% 1.91% 22.5% German 13.1% 9.8% Italian 9.2% 8.1% 6.0% Polish

Compressed Prompt (QUITO):
2000, 79,667 households, and 60,387 families residing There were 86.77% Race (United Race race. 22.5% of German people, 13.1% Irish people, 6.0% Polish ancestry.

Compressed Prompt (Ours):
the people, 79,667 households, and 60,387 families residing 22.5% of German people, 13.1% Irish people, 9.8% Italian people, 9.2% English, 8.1% "American" and 6.0% Polish ancestry.

Answer: English (9.2%) is smaller than German (22.5%)

Figure 9: Case Study 1: Census-based QA under different compression schemes. Our method retains more semantic and numeric fidelity compared to other token-level approaches.

Original Prompt (NFL QA Example):
Hoping to rebound from their tough overtime road loss to the Raiders, the Jets went home for a Week 8 duel with the Kansas City Chiefs. In the first quarter, New York took flight as QB Brett Favre completed an 18-yard TD pass to RB Leon Washington. In the second quarter, the Chiefs tied the game as QB Tyler Thigpen completed a 19-yard TD pass to TE Tony Gonzalez. The Jets would answer with Washington getting a 60-yard TD run. Kansas City closed out the half as Thigpen completed an 11-yard TD pass to WR Mark Bradley. In the third quarter, the Chiefs took the lead as kicker Connor Barth nailed a 30-yard field goal, yet New York replied with RB Thomas Jones getting a 1-yard TD run. In the fourth quarter, Kansas City got the lead again as CB Brandon Flowers returned an interception 91 yards for a touchdown. Fortunately, the Jets pulled out the win with Favre completing the game-winning 15-yard TD pass to WR Laveranues Coles. During halftime, the Jets celebrated the 40th anniversary of their Super Bowl III championship team.
Question: How many yards was the longest TD of the game?

Compressed Prompt (LLMLingua2):
Raiders Jets Week 8 Kansas City Chiefs York Favre 18-yard Washington Chiefs Thigpen 19-yard Gonzalez 60-yard TD Kansas Thigpen 11-yard Bradley third Chiefs Barth 30-yard Jones 1-yard TD fourth Kansas Flowers touchdown Jets Favre 15-yard Coles Jets 40th Super Bowl

Compressed Prompt (QUITO):
Jets the Kansas City Chiefs. as QB Brett Favre completed to RB Leon Washington. In QB Tyler Thigpen TE Tony Gonzalez. run. Kansas WR Mark Bradley. kicker Connor Barth with RB Thomas Jones win with Favre completing to WR Laveranues Coles. During halftime, the

Compressed Prompt (Ours):
completed an 18-yard TD pass RB Tyler completed a 19-yard TD pass getting a 60-yard TD run. completed an 11-yard TD pass nailed a a 1-yard TD the 91 completing the game-winning 15-yard TD pass WR

Answer: 91 yards (Brandon Flowers interception return)

Figure 10: Case Study 2: Sports-related QA. Our method captures the most relevant yardage details, supporting accurate numerical reasoning.