BEYOND POSITION BIAS: SHIFTING CONTEXT COMPRESSION FROM POSITION-DRIVEN TO SEMANTIC-DRIVEN

Anonymous authors
Paper under double-blind review

ABSTRACT

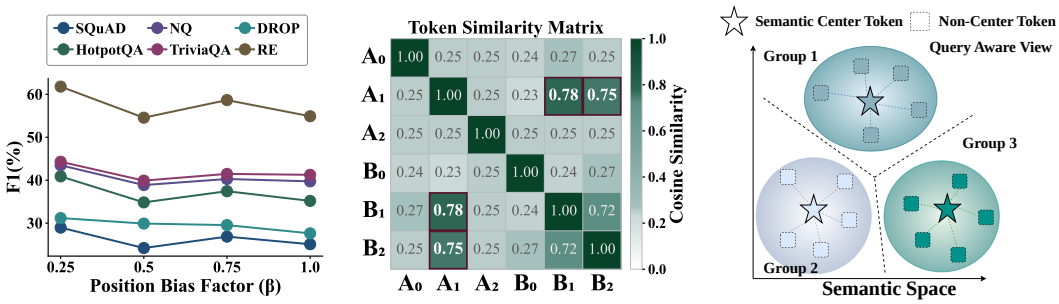
Large Language Models (LLMs) have demonstrated exceptional performance across diverse tasks. However, their deployment in long-context scenarios faces high computational overhead and information redundancy. While soft prompt compression has emerged as a promising way to mitigate these costs by compressing sequences into compact embeddings, existing paradigms remain fundamentally constrained by **position bias**: they primarily rely on learnable tokens insertion at fixed positions or group tokens according to their physical token layout, thereby inducing performance instability and semantic fragmentation. To overcome this bottleneck, we propose **Se**mantic **Co**nsistency Context Compression (**SeCo**), a method that shifts context compression from *position-driven* to *semantic-driven*. Rather than constraint by physical token layout, SeCo dynamically anchors compression *directly* in the semantic space by selecting query-relevant tokens as semantic centers and aggregating remaining tokens via consistency-weighted merging. This design inherently preserves semantic consistency while eliminating position bias. Extensive experiments on 14 benchmarks across two backbone models demonstrate that SeCo consistently shows superiority in downstream tasks, inference latency, and out-of-domain robustness. The code is available at `https://anonymous.4open.science/r/SeCo-52YH`.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, catalyzing a paradigm shift in Natural Language Processing (NLP) (Yang et al., 2024; DeepSeek-AI et al., 2025; Team et al., 2025). Real-world applications increasingly demand longer prompt inputs, such as Retrieval-Augmented Generation (RAG) (Lewis et al., 2020), In-Context Learning (ICL) (Dong et al., 2024), and the deployment of LLMs as agents for long-horizon task planning (Sun et al., 2025). Despite the enhanced contextual coverage afforded by long sequences, deploying LLMs directly on such inputs faces (1) significant information redundancy (Liu et al., 2024b; Jiang et al., 2024; Ge et al., 2024; Tang et al., 2025a;b) and (2) high computational load caused by Transformer's quadratic time complexity (Vaswani et al., 2017; Tang et al., 2026a;b) on its self attention mechanism, limiting their real-world utility.

Prompt compression (Li et al., 2025a) has emerged as a promising way to shorten sequence lengths and mitigate redundancy. Existing methods can be mainly categorized into hard prompt compression (Jiang et al., 2024; Pan et al., 2024; Tang et al., 2025a; Chirkova et al., 2025; Luo et al., 2025; Hwang et al., 2025; Liu et al., 2025) and soft prompt compression (Ge et al., 2024; Zhang et al., 2025a; Li et al., 2025b; Chen et al., 2025; Dai et al., 2025; Rau et al., 2025; Tang et al., 2025b; 2026a;b). Hard compression methods typically discard non-selected tokens entirely, failing to leverage secondary contextual information and compromising semantic coherence, which limits information utility (Zhang et al., 2025b). In contrast, soft prompt compression exploits inherent redundancy in the semantic space (Ethayarajh, 2019; Aghajanyan et al., 2021) by learning compact soft embeddings, typically preserving richer semantic information, especially under high compression rates (Tang et al., 2025b). However, we identify a critical bottleneck: existing soft compression methods, which can be categorized into learnable token insertion (Ge et al., 2024; Zhang et al., 2025a; Li et al., 2025b; Chen

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



(a) Positional Encoding Bias (b) Group-based Position Bias (c) SeCo

Figure 1: Comparison between existing methods and SeCo. (a) **Performance Instability.** Learnable token insertion method, such as ICAE (Ge et al., 2024), exhibit pronounced sensitivity to the position bias factor β, which controls the positional encoding of learnable tokens (as detailed in Figure 5). Consequently, F1 scores fluctuate substantially across various benchmarks under different β configurations. (b) **Semantic Fragmentation.** Token A_1 in Group A is semantically highly correlated with tokens B_1 and B_2 in Group B. However, position-based grouping rigidly partitions these semantically related tokens (highlighted in red boxes) into separate compression groups. (c) **Semantic Consistency Compression (SeCo).** SeCo selects the top-k tokens most relevant to the query as semantic centers. Subsequently, remaining tokens are dynamically assigned and merged *directly* in the semantic space based on query-centric semantic consistency.

et al., 2025; Dai et al., 2025; Rau et al., 2025) and position-based grouping (Cao et al., 2024; Tang et al., 2025b; 2026a;b), rely heavily on positional priors, introducing **position bias**.

Specifically, current soft prompt compression methods suffer from two systematic forms of **position bias**: (1) **Positional encoding bias.** As illustrated in Figure 1a, methods employing learnable token insertion (e.g., ICAE (Ge et al., 2024)) exhibit significant instability in downstream tasks depending on the Positional Encodings (PEs) assigned to these tokens. This sensitivity to perturbations in PEs (Vaswani et al., 2017; Su et al., 2024) aligns with recent observations in Zhao et al. (2025). (2) **Group-based position bias.** The principle of semantic consistency posits that semantically coherent vector structures facilitate more effective compression (Gray & Neuhoff, 1998; Cover & Thomas, 2006; Donoho, 2006). However, position-based grouping methods *rely on the physical token layout*, potentially partitioning semantically related tokens into distinct compression groups (see Figure 1b). This induces semantic fragmentation, thereby violating semantic consistency. Furthermore, due to the inherent sparsity of task-relevant information (Li et al., 2023; Jiang et al., 2024; Tang et al., 2025a), the correlation between such derived groups and the query is often highly imbalanced (i.e., certain groups are entirely irrelevant to the query). These findings collectively highlight a core limitation: *over-reliance on positional priors constrains compression effectiveness.* This motivates our research: *How can we mitigate the negative effects of position bias in soft prompt compression?*

To address this, we draw inspiration from cognitive science: human cognition is fundamentally organized as a semantic network rather than a linear sequence, where concepts are linked by meaning rather than physical proximity (Collins & Loftus, 1975; Quillian, 1967). Motivated by this, we shift the compression paradigm from *position-driven* to *semantic-driven*, proposing the **Se**mantic Consistency **Co**ntext Compression (**SeCo**) framework (Figure 1c). SeCo eliminates the need for injecting learnable tokens at fixed positions, instead *modeling context structure directly within the semantic space to avoid positional encoding bias*. To resolve semantic fragmentation caused by group-based position bias, SeCo reconstructs the grouping logic guided by query semantics: we first compute the relevance between context tokens and the query to identify the top-k most relevant tokens as semantic centroids. Subsequently, remaining tokens are dynamically assigned to these centroids based on semantic consistency, and compressed representations are generated via consistency-weighted aggregation within each center. *This process is driven by semantic relevance rather than physical token layout; each token autonomously identifies its suitable compression group directly in the semantic space, effectively mitigating group-based position bias.*

Our contributions are summarized as follows: (1) We identify and analyze position bias as a bottleneck that fundamentally limits existing soft prompt compression methods. (2) We propose SeCo, a

semantic-driven compression framework designed to mitigate position bias and enhance semantic consistency. (3) We conduct extensive experiments across two backbone models (i.e., LLaMA and Qwen) and 14 benchmarks (e.g., long context question-answering and summarization), including both in-domain and out-of-domain evaluations. Experiment results demonstrate that SeCo consistently delivers superior performance in terms of downstream tasks, computational efficiency, and robustness.

2 Related Work

Hard Prompt Compression. Hard prompt compression aims to reduce input length by explicitly shortening the original prompt, either through removing less informative tokens or generating condensed textual summaries. The resulting compressed prompt remains in the form of discrete natural language. Existing approaches can be broadly grouped into four categories (Tang et al., 2025b). (1) Perplexity-based methods estimate token importance according to perplexity and typically adopt a coarse-to-fine compression strategy to progressively discard less critical content, as exemplified by LLMLingua (Jiang et al., 2023), LongLLMLingua (Jiang et al., 2024), and Perception Compressor (Tang et al., 2025a). (2) Bidirectional semantic-based methods address the limitations of the unidirectional perplexity signal by incorporating bidirectional semantic information, as in LLMLingua-2 (Pan et al., 2024), MOOSComp (Zhou et al., 2025), Provence (Chirkova et al., 2025), and SAC (Liu et al., 2025). (3) Attention-based methods exploit the intrinsic attention patterns of LLMs to identify and retain salient information during compression, such as AttnComp (Luo et al., 2025). (4) Summary-generation methods compress long contexts by producing concise linguistic summaries that preserve task-relevant information, as demonstrated by CompACT (Yoon et al., 2024), RECOMP (Xu et al., 2024), and EXIT (Hwang et al., 2025). *Despite their effectiveness in improving inference efficiency, these hard compression methods inevitably operate in a lossy manner and may impair the semantic coherence of the original prompt.*

Soft Prompt Compression. Compared with hard prompt compression methods that rely on discrete selection strategies, soft prompt compression compress context by distilling contextual information into a compact continuous embedding space. This line of research has attracted considerable attention in recent years. Some methods, such as GIST (Mu et al., 2023), AutoCompressor (Chevalier et al., 2023), ICAE (Ge et al., 2024), LLoCO (Tan et al., 2024), Beacon (Zhang et al., 2025a), 500xCompressor (Li et al., 2025b), typically obtain compact compressed representations by introducing appending learnable tokens. Other methods, including QGC (Cao et al., 2024), GMSA (Tang et al., 2025b), RAM (Tang et al., 2026a), and COMI (Tang et al., 2026b), instead group tokens according to their physical token layout, thereby learning compressed representations. While these methods achieve relative rich semantic density, they are constrained by *position priors*. Whether through the injection of learnable tokens at fixed positions or grouping based on physical token layout, such reliance introduces **position bias** that leads to performance instability or semantic fragmentation. *In contrast, SeCo avoids position constraints by selecting semantic centroids within the semantic space and allocating remaining tokens based on semantic consistency for compression, thereby effectively mitigating position bias.*

3 Method

In this section, we present SeCo, a *semantic-driven* framework that compresses long contexts into compact soft tokens while preserving task-relevant information. SeCo comprises an encoder for feature extraction and a decoder for downstream generation. The core compression mechanism proceeds in three stages: *Semantic Center Selection*, *Semantic Assignment*, and *Consistency Merging*. By decoupling compression process from physical token layout, SeCo mitigates position bias and ensures semantic consistency within each compressed tokens.

3.1 Semantic Feature Extraction

Given an input sequence X composed of a context X^{context} and a query X^{query}, we employ a pre-trained language model as the encoder to extract contextualized hidden states:

$$H = \texttt{Encoder}(X), \tag{1}$$

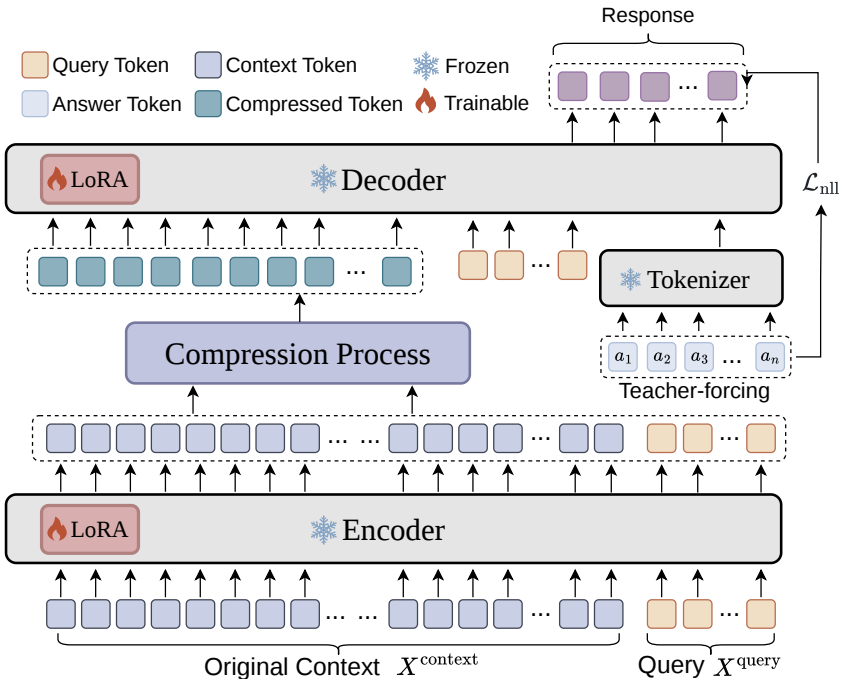


Figure 2: The Training Paradigm of SeCo. SeCo is based on an encoder-decoder architecture. The original context X^{context} and query X^{query} are first encoded into hidden states, which are then compressed via a compression process (see Figure 3). During training, the encoder and decoder are fine-tuned with Low-Rank Adaptation (LoRA) (Hu et al., 2022).

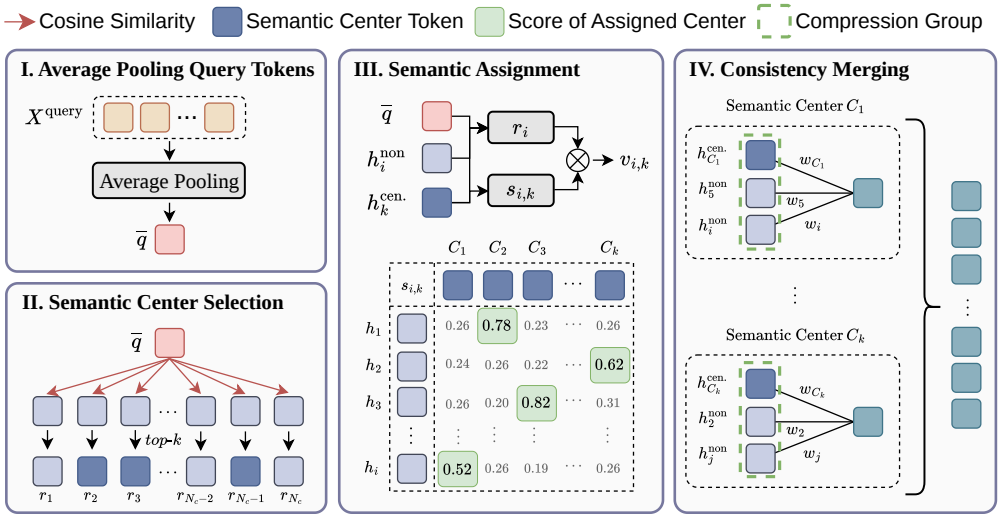


Figure 3: The Compression Process of SeCo. Specifically, it sequentially performs four steps: **I. Average Pooling Query Tokens.** Obtain a single query representation $\bar{q}$ via average pooling; **II. Semantic Center Selection.** The *top-k* tokens are selected as semantic center tokens based on the cosine similarity between each context token and $\bar{q}$; **III. Semantic Assignment.** $h^{\text{cen.}}$ represents the selected semantic centers while h^{non} denotes the remaining tokens. Each non-center token is assigned to the most relevant semantic center based on the cosine similarity score s, which is defined in Eq. 6. **IV. Consistency Merging.** Each compression group is merged into a compressed representation based on the weights (defined in Eq. 8) of the tokens within the group. Specifically, the weight of the semantic center is its cosine similarity with $\bar{q}$, and the weight of the non-center token is the product of their cosine similarity with $\bar{q}$ and the center of their respective group.

where $H \in \mathbb{R}^{N \times d}$ denotes the last-layer hidden states, with N being the total sequence length and d the hidden dimension. We partition H into context and query components, $H^{\text{context}} \in \mathbb{R}^{N_c \times d}$ and $H^{\text{query}} \in \mathbb{R}^{N_q \times d}$, where N_c and N_q represent their respective sequence lengths.

3.2 Semantic Center Selection

To achieve a target compression rate τ, we determine the number of compressed tokens K as:

$$K = \max\left(2, \left\lceil \frac{N_c}{\tau} \right\rceil\right). \tag{2}$$

Guided by the principle of task relevance, we first aggregate the query information into a single representation $\bar{q}$ by averaging its token embeddings:

$$\bar{q} = \frac{1}{N_q} \sum_{i=1}^{N_q} h_i^{\text{query}}, \tag{3}$$

where $h_i^{\text{query}} \in \mathbb{R}^d$ is the i-th token embedding in H^{query}.

Next, we quantify the semantic relevance of each context token to the query via cosine similarity:

$$r_i = \frac{h_i^{\text{context}} \cdot \bar{q}}{\|h_i^{\text{context}}\| \|\bar{q}\|}. \tag{4}$$

We select the K context tokens with the highest similarity scores as semantic centers:

$$\mathcal{C} = \left\{ i \in \{1, \ldots, N_c\} \;\middle|\; r_i \in \textit{top-}k_{j=1}^{N_c}\{r_j\} \right\}, \quad |\mathcal{C}| = K. \tag{5}$$

These centers serve as semantic anchors, ensuring that the compressed representation retains the most task-critical information.

3.3 Semantic Assignment and Consistency Merging

Semantic Assignment. Remaining non-center tokens, denoted as $H^{\text{non}} = H^{\text{context}} \setminus \{h_j\}_{j \in \mathcal{C}}$, we first compute the cosine similarity $s_{i,k}$ between h_i^{non} and every center h_k^{center}:

$$s_{i,k} = \frac{h_i^{\text{non}} \cdot h_k^{\text{center}}}{\|h_i^{\text{non}}\| \|h_k^{\text{center}}\|}. \tag{6}$$

Token i is assigned to the center k^* that maximizes this score: $k^* = \arg\max_k \; s_{i,k}$. This mechanism ensures that tokens are grouped by their semantic coherence.

Consistency Merging. For each non-center token, we compute a joint representation score $v(i,k)$:

$$v(i,k) = r_i \cdot s_{i,k}, \tag{7}$$

which jointly captures the semantic coherence of the tokens and their relevance to the query.

Within each compression group $\mathcal{G}_k$ (comprising the center and its assigned members), we first define a consistency weight w_i for each token:

$$w_i = \begin{cases} r_i & \text{if } i \in \mathcal{C}, \\ v(i,k^*) & \text{otherwise.} \end{cases} \tag{8}$$

Center tokens are weighted by their direct task relevance (r_i), while non-center tokens are weighted by their joint representation score ($v(i,k^*)$), balancing local semantic consistency with global task alignment.

We then generate a compressed representation $\tilde{h}_k$ via a consistency-weighted aggregation:

$$\alpha_i = \frac{\exp(w_i)}{\sum_{j \in \mathcal{G}_k} \exp(w_j)}, \quad \tilde{h}_k = \sum_{i \in \mathcal{G}_k} \alpha_i \cdot h_i. \tag{9}$$

The final compressed context is formed by concatenating these aggregated representations:

$$\widetilde{H} = \left[\tilde{h}_1, \ldots, \tilde{h}_K\right] \in \mathbb{R}^{K \times d}. \tag{10}$$

$\widetilde{H}$ serves as the compressed embeddings for the subsequent training/decoding stage.

3.4 Training Objective

The training loss is:

$$\mathcal{L}_{\text{nll}} = -\sum_{i=1}^{N_a} \log p_\phi\Big(a_i \mid \widetilde{H}, X^{\text{query}}, a_{<i}\Big) \,, \tag{11}$$

where $p_\phi(\cdot)$ is the decoder's output distribution; a_i denotes the i-th answer token; N_a represents the token length of answer. Both the encoder and decoder are fine-tuned using LoRA.

4 Experiment

In this section, we aim to answer the following Research Questions (RQs): (1) How does SeCo perform on various downstream tasks (RQ1)? (2) Compared with other methods, does SeCo have an advantage in inference latency (RQ2)? (3) How effective are the individual components of SeCo (RQ3)?

4.1 Settings

Training. The training paradigm of SeCo is shown in Figure 2. Our method uses the MRQA (Fisch et al., 2019) datasets, partitioned into in-domain and out-of-domain sets. The in-domain data, used for both training and internal testing, comprises six subsets: SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), TriviaQA (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (Yang et al., 2018) and NaturalQuestions (Kwiatkowski et al., 2019). To evaluate generalization, we employed an out-of-domain test set consisting of DROP (Dua et al., 2019), BioASQ (Tsatsaronis et al., 2015), DuoRC.ParaphraseRC (Saha et al., 2018), TextbookQA (Kembhavi et al., 2017), RelationExtraction (Levy et al., 2017) and RACE (Lai et al., 2017). During training, we set two compression rates for each training sample (i.e., 16× and 32×). SeCo is implemented based on `LLaMA-3.2-1B-Instruct` and `Qwen3-4B-Instruct`. Furthermore, to assess SeCo's performance in summarization and long-context QA, we incorporate DialogSum (Chen et al., 2021), a multi-turn dialogue summarization benchmark, and NarrativeQA (Kociský et al., 2018), which demands deep comprehension and reasoning over long-form stories and scripts. Further dataset descriptions and training details are provided in Appendix D and Appendix E, respectively.

Evaluation Metrics. We evaluate QA performance using Exact Match (EM) (Liu et al., 2025) and F1 score (Lin, 2004), and assess summarization quality via BERT-F1 (Zhang et al., 2020) computed with `roberta-large`[1].

Baselines. We conduct a comparison with encoder-decoder based context compression methods, including task-agnostic methods (i.e., ICAE (Ge et al., 2024), EPL (Zhao et al., 2025), SAC (Liu et al., 2025), 500xCompressor (Li et al., 2025b)) and task-aware methods (i.e., RAM (Tang et al., 2026a), COMI (Tang et al., 2026b)). For methods employing specific position layout strategies, we append the suffixes (EPL) or (DPL) to denote the adopted scheme. Specifically, Enhanced Position Layout (EPL) reassigns position IDs to learnable or compressed tokens to optimize structural alignment. In contrast, Default Position Layout (DPL) retains the standard sequential positional indices for these tokens without additional reorganization. **All baselines use the same training datasets and hyperparameters.**

4.2 Main Result

For RQ1, we analyze the performance of SeCo via its performance on diverse downstream tasks. Tables 1 and 2 report the results on in-domain and out-of-domain test sets. Under compression constraints of 16× and 32×, SeCo demonstrates superior performance across various backbone models (i.e., `LLaMA3.2-1B-Instruct` and `Qwen3-4B-Instruct`). SeCo consistently outperforms both *task-agnostic* and *task-aware* baselines, across most test datasets in terms of average EM and F1. This highlights SeCo's ability to aggregate semantic space knowledge based on semantic cues provided by queries while eliminating the need of *positional priors* used in the baselines.

[1] `https://huggingface.co/FacebookAI/roberta-large`

Figure 4 presents the results of the long-context QA and summarization. SeCo exhibits a consistent performance advantage over task-aware baselines (i.e., RAM and COMI). Furthermore, we conduct comprehensive stress tests on SeCo across varying compression rates, which demonstrate that SeCo consistently maintains superior performance in both in-domain and out-of-domain evaluations, underscoring its robustness (see Appendix C).

Table 1: Main results of benchmark tests on In-Domain (ID) datasets. We **bold** the optimal results.

Methods	SQuAD		NewsQA		TriviaQA		SearchQA		HotpotQA		NQ		Average	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
16x compression constraint														
LLaMA-3.2-1B-Instruct														
RAM	34.12	51.49	19.90	35.18	41.07	50.23	51.47	61.69	27.94	47.25	27.49	43.97	36.99	51.03
COMI	27.26	42.42	14.62	27.13	50.85	57.79	62.88	70.65	33.67	51.28	34.21	50.88	42.07	54.37
500x (DPL)	14.92	26.96	10.02	23.06	39.99	49.73	47.34	61.00	26.28	41.97	25.64	41.88	30.89	44.46
500x (EPL)	17.98	32.42	11.35	26.51	42.36	52.50	48.49	62.25	29.84	47.19	26.39	44.16	32.72	47.46
SAC (EPL)	19.00	33.96	11.89	27.93	41.01	51.57	47.66	61.47	30.35	46.98	26.38	43.76	32.56	47.38
ICAE (EPL)	17.94	32.09	10.59	24.37	36.06	45.71	44.17	57.67	26.98	42.56	24.70	42.82	29.89	44.24
SeCo	**51.84**	**70.89**	**28.89**	**49.94**	**67.30**	**75.06**	**65.89**	**74.82**	**45.55**	**67.17**	**39.05**	**59.35**	**52.89**	**68.16**
Qwen3-4B-Instruct														
RAM	17.28	29.25	7.48	16.87	40.35	45.97	49.83	56.26	18.86	31.75	15.70	29.40	28.96	38.75
COMI	15.88	27.85	6.77	16.31	39.87	45.67	49.27	55.96	17.69	31.08	16.63	29.95	28.52	38.39
500x (DPL)	23.93	39.96	14.58	31.06	50.96	60.73	55.91	69.28	35.74	52.68	35.74	54.02	39.99	55.03
500x (EPL)	32.50	52.67	20.16	39.70	55.00	65.15	59.45	71.93	44.62	63.13	42.09	61.55	45.82	62.04
SAC (EPL)	33.77	53.63	20.25	41.08	54.28	64.78	56.50	69.66	44.06	62.16	42.05	60.64	45.03	61.30
ICAE (EPL)	32.21	50.40	15.08	32.20	49.36	59.30	53.56	66.95	37.10	53.80	39.51	58.35	41.59	57.20
SeCo	**57.38**	**75.56**	**33.43**	**55.45**	**69.34**	**76.65**	**73.77**	**81.27**	**48.23**	**69.48**	**42.78**	**62.84**	**57.88**	**72.50**
32x compression constraint														
LLaMA-3.2-1B-Instruct														
RAM	28.81	43.75	17.57	30.33	39.38	47.96	46.80	56.83	23.91	40.98	25.05	41.23	33.33	46.32
COMI	18.99	32.70	12.89	24.73	44.38	51.64	59.05	66.69	26.76	42.76	29.10	45.99	36.64	48.52
500x (DPL)	12.51	23.54	9.43	21.82	36.69	46.16	46.63	59.93	22.44	36.39	23.09	38.67	28.81	41.69
500x (EPL)	13.98	26.15	9.76	22.21	37.93	48.15	44.75	58.16	23.49	38.21	23.28	39.65	28.86	42.34
SAC (EPL)	14.75	27.13	9.26	22.74	38.79	48.52	46.73	60.56	23.22	37.93	23.96	40.77	29.78	43.52
ICAE (EPL)	15.44	28.30	8.86	20.65	34.80	44.93	44.13	58.21	24.17	38.30	23.86	40.34	28.66	42.36
SeCo	**51.63**	**70.94**	**28.82**	**50.03**	**64.55**	**72.94**	**65.18**	**74.37**	**44.92**	**66.61**	**39.33**	**60.01**	**52.27**	**67.85**
Qwen3-4B-Instruct														
RAM	13.45	24.10	5.25	13.01	37.19	42.56	46.13	52.36	14.01	25.13	13.38	25.51	25.60	34.42
COMI	13.11	24.11	4.77	12.51	37.55	42.77	46.50	52.61	13.83	25.12	14.16	26.46	25.82	34.70
500x (DPL)	23.34	38.30	14.34	30.50	50.89	60.38	56.65	71.69	34.43	51.05	34.22	50.97	39.61	54.52
500x (EPL)	23.69	40.29	14.84	32.31	50.98	61.01	58.28	70.56	37.54	54.89	**36.55**	37.54	41.03	52.18
SAC (EPL)	25.02	41.64	15.86	33.88	53.50	63.32	56.41	69.45	36.84	53.93	35.46	54.03	40.82	56.07
ICAE (EPL)	25.96	43.78	12.51	27.81	46.68	56.73	46.40	60.19	32.20	48.20	33.42	52.06	36.00	51.41
SeCo	**50.92**	**69.63**	**30.41**	**50.59**	**65.11**	**73.57**	**72.24**	**80.23**	**42.70**	**64.07**	36.29	**56.71**	**53.49**	**68.46**

4.3 Efficiency Analysis

For RQ2, we conduct the following inference latency experiment. SeCo replaces the original context with soft prompt to accelerate the inference process, thereby reducing the inference cost of processing the original context in the generation stage by a factor of τ. The overall computational overhead includes the computational costs of both the compression and generation stages.

The compression process of SeCo comprises three distinct phases: (1) a single full forward pass of the encoder over the input sequence; (2) compute the cosine similarity between each of the context

Table 2: Main results of benchmark tests on Out-Of-Domain (OOD) datasets using `LLaMA-3.2-1B-Instruct` as backbone.

Methods	DROP		BioASQ		DuoRC		TQA		RE		RACE		Average	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
16x compression constraint														
RAM	11.11	18.39	22.67	34.72	19.79	30.47	22.09	29.26	32.94	46.93	13.50	25.33	22.83	33.74
COMI	16.10	24.67	27.93	40.08	9.86	16.42	15.17	22.55	42.57	58.81	9.35	19.37	24.46	35.54
500x (DPL)	21.42	29.94	24.34	35.96	6.80	12.89	17.43	30.78	21.64	34.15	2.52	14.16	17.72	28.54
500x (EPL)	23.02	32.85	20.28	34.05	15.12	23.66	24.75	41.10	30.66	43.08	3.41	18.90	22.60	35.05
SAC (EPL)	24.55	33.89	20.41	31.60	15.19	24.34	23.35	39.34	33.96	47.04	3.71	18.78	23.68	35.86
ICAE (EPL)	21.82	31.55	26.53	38.18	11.53	18.38	18.50	30.76	38.74	54.18	2.08	14.54	24.23	36.14
SeCo	**26.35**	**40.99**	**31.45**	**48.76**	**37.24**	**50.84**	**42.51**	**53.71**	**56.21**	**72.93**	**21.96**	**38.64**	**40.20**	**55.33**
32x compression constraint														
RAM	16.50	23.67	25.53	32.91	16.99	25.95	20.63	28.37	24.86	37.15	12.91	23.60	20.94	30.32
COMI	13.91	21.38	24.60	36.59	8.59	13.72	16.50	24.92	36.43	51.46	7.27	16.56	21.58	31.98
500x (DPL)	19.96	28.35	22.47	32.47	6.26	11.54	16.43	28.16	28.26	39.96	1.48	11.96	18.91	28.75
500x (EPL)	20.96	29.43	20.01	30.79	10.73	18.73	21.29	35.80	31.07	42.59	2.97	15.49	21.10	32.02
SAC (EPL)	20.83	30.42	19.95	31.32	10.13	17.27	21.62	37.18	25.37	37.15	3.12	15.75	19.30	30.60
ICAE (EPL)	21.22	30.45	23.01	34.70	3.80	7.31	19.49	32.57	36.23	49.57	2.97	14.81	21.83	32.60
SeCo	**25.62**	**40.40**	**31.72**	**47.90**	**36.11**	**50.17**	**41.78**	**53.05**	**55.60**	**72.76**	**23.00**	**39.97**	**39.72**	**54.94**

Table 3: Latency evaluation (in milliseconds) under a 16× compression constraint across different datasets. Compression latency and generation latency denote the latency for the compression and generation phases, respectively. Total latency is the sum of both phases. The best results are highlighted in **bold**, and the second-best results are underlined.

Methods	SQuAD			NewsQA			NQ		
	Comp. Lat.	Gen. Lat.	Total Lat.	Comp. Lat.	Gen. Lat.	Total Lat.	Comp. Lat.	Gen. Lat.	Total Lat.
COMI	14.10	**44.93**	59.03	28.11	<u>46.80</u>	74.91	17.81	<u>49.24</u>	<u>67.05</u>
500x (DPL)	11.40	47.54	58.94	21.29	50.46	71.75	14.45	91.86	106.31
500x (EPL)	<u>11.22</u>	48.30	59.52	21.22	48.74	69.96	<u>14.25</u>	57.52	71.77
SAC (EPL)	12.31	51.32	63.63	22.59	56.58	79.17	15.10	66.09	81.19
ICAE (EPL)	11.56	45.90	<u>57.46</u>	<u>21.16</u>	48.47	<u>69.63</u>	14.56	61.24	75.80
SeCo	**10.23**	<u>45.67</u>	**55.90**	**14.30**	**46.43**	**60.73**	**11.90**	**46.77**	**58.67**

tokens and the query semantics $\bar{q}$; (3) compute the pairwise similarity matrix between the non-center tokens and the semantic centers, followed by consistency-weighted aggregation.

The compression process can be represented as:

$$\text{FLOPs}^{\text{comp.}} = F^{\text{Encoder}}(L) + F^{\text{Selection}}(L_c, d) + F^{\text{Assignment}}(L_c, K, d) + F^{\text{Merging}}(L_c, K, d), \tag{12}$$

where L_c and d represent the context lengths and hidden dimension, respectively. K is the number of semantic centers defined in Eq. 2. Specifically, $F^{\text{Encoder}}(L)$ is the cost of prefilling process; $F^{\text{Selection}}(L_c, d)$ denotes the *Semantic Center Selection* cost (complexity $O(L_c d)$, with $L_c \ll L^2$); $F^{\text{Assignment}}(L_c, K, d)$ refers the *Semantic Assignment* cost (complexity $O(L_c K d)$, with $L_c K \ll L^2$); and $F^{\text{Merging}}(L_c, K, d)$ represents the *Consistency Merging* cost (complexity $O(L_c d)$, with $L_c \ll L^2$). Therefore, aforementioned operations introduce only lightweight overhead.

For the generation stage, assuming the answer length is L_a, the decoder performs L_a forward passes autoregressively over the compressed prefix of length $K + L_q$. The FLOPs for the i-th forward pass are:

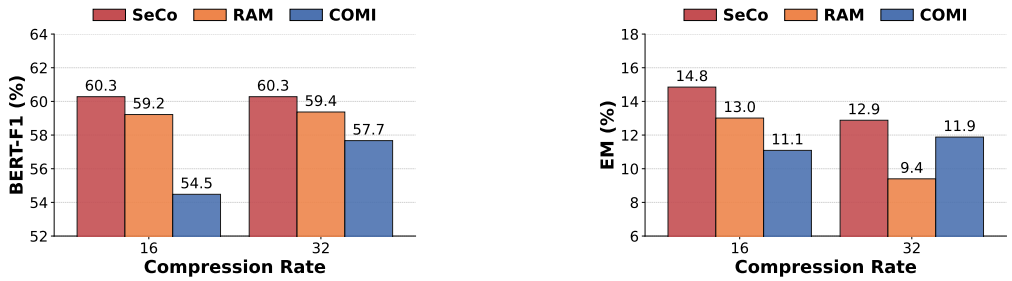
$$\text{FLOPs}^{\text{generation}} = \sum_{i=1}^{L_a} F^{\text{Decoder}}(K + L_q,\ i), \tag{13}$$

where $F^{\text{Decoder}}(n, i)$ denotes the FLOPs of a single decoder forward step with prefix length n at generation step i. SeCo achieves the lowest total latency among all baselines, delivering superior speed under a 16× compression rate across multiple test datasets (see Table 3).

4.4 Ablation Study

To investigate RQ3 and understand the specific contributions of each component within SeCo, we conduct the following three ablation experiments. (see Table 4): (1) Our *w/o* Query indicates that during the assignment and merging process, we do not use the cosine similarity between the non-center tokens and $\bar{q}$ as a weight reference, but instead rely solely on the cosine similarity between the non-center tokens and the semantic centers; (2) Our *w/o* Consist. Merging refers to directly select the *top-k* most relevant tokens based on their similarity scores with $\bar{q}$ to achieve the target compression rate, bypassing the entire assignment and merging process; (3) Our *w/* Uniform Merging treats the center and all assigned members within each compression group equally, and constructs the compressed representation by taking the average of their hidden representations.

Omitting any component leads to a substantial performance drop, validating the necessity of each module. Specifically, removing query guidance (*w/o* Query) eliminates relevance weighting between non-center tokens and the query, resulting in a lack of *task-specific* focus during semantic assignment and consistency merging. Disabling the assignment mechanism (*w/o* Consist. Merging) restricts the model to hard filtering, thereby discarding valuable contextual information from unselected tokens. Finally, replacing Consistency Merging with uniform averaging (*w/* Uniform Merging) removes the semantic-aware weighting among the center and its assigned members, potentially overemphasizing less relevant information and weakening the informativeness of the compressed representations.



(a) DialogSum (b) NarrativeQA

Figure 4: Additional results on long context and summarization benchmarks compared with task-aware methods (i.e., COMI and RAM) using `Qwen3-4B-Instruct` as the backbone.

Table 4: Ablation study on diverse benchmarks under 32× compression constraint. We use `Qwen3-4B-Instruct` as the backbone. ID and OOD refer to in-domain and out-of-domain datasets, respectively.

Methods	ID						OOD						Overall	
	SQuQA		SearchQA		HotpotQA		DROP		TQA		RE		Average	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Default	**50.92**	**69.63**	**72.24**	**80.23**	**42.70**	**64.07**	**37.46**	**52.86**	**58.48**	**69.52**	**56.41**	**71.87**	**59.07**	**72.90**
w/o Query	47.83	66.77	69.66	78.03	40.48	61.63	35.13	50.11	53.43	64.97	54.78	70.40	56.40	70.43
w/o Consist. Merging	44.02	63.02	65.13	74.06	36.91	58.26	32.67	47.57	50.17	61.32	52.99	68.38	52.54	66.82
w/ Uniform Merging	48.03	66.84	69.98	78.25	41.26	62.31	35.06	50.21	54.03	65.58	55.26	70.75	56.76	70.69

5 Conclusion

We identify position bias as a bottleneck in existing soft prompt compression: compression effectiveness is often constraint by position prior, which can lead positional encoding bias and group-based bias. To overcome this limitation, we propose SeCo, a framework that selects task-relevant centers and aggregates remaining context tokens via consistency merging while eliminating the need of positional priors. Across extensive benchmarks, backbones, and compression rates, SeCo consistently delivers superior downstream performance and better efficiency than prior prompt compression baselines.

AI USE STATEMENT

Throughout the development of this paper, we integrated a large language model into our drafting process to assist in synthesizing our technical notes into a coherent narrative. The model served as a supportive tool for structuring arguments and ensuring consistent terminology, rather than generating the underlying research content. We exercised complete editorial oversight, conducting rigorous manual reviews of all AI-generated suggestions to ensure scientific precision.

ETHICS STATEMENT

This work presents SeCo, a framework based on the encoder-decoder architecture that enables task-aware context compression by performing clustering in the semantic space. Regarding the broader societal impact, SeCo enhances the efficiency and efficacy of context compression within machine learning pipelines. We have evaluated the potential implications of this framework and concluded that it does not introduce novel ethical risks or societal harms beyond those inherent in existing context compression technologies. We remain committed to ethical AI development and will continue to monitor the implications of our work as it is integrated into broader applications.

REPRODUCIBILITY STATEMENT

The main results are fully reproducible using the code provided in our anonymous repository: `https://anonymous.4open.science/r/SeCo-52YH`. Detailed descriptions of the training and test datasets are provided in Appendix D, while all hyperparameters and optimization settings are documented in Subsection 4.1 and Appendix E. The repository contains the complete implementation, training and evaluation scripts. We also provide instructions for reproducing the main experimental results and specify the required dependencies and execution settings.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *ACL/IJCNLP (1)*, pp. 7319–7328. Association for Computational Linguistics, 2021.

Zhiwei Cao, Qian Cao, Yu Lu, Ningxin Peng, Luyang Huang, Shanbo Cheng, and Jinsong Su. Retaining key information under high compression ratios: Query-guided compressor for llms. In *ACL (1)*, pp. 12685–12695. Association for Computational Linguistics, 2024.

Shaoshen Chen, Yangning Li, Zishan Xu, Yongqin Zeng, Shunlong Wu, Xinshuo Hu, Zifei Shan, Xin Su, Jiwei Tang, Yinghui Li, and Hai-Tao Zheng. DAST: context-aware compression in llms via dynamic allocation of soft tokens. In *ACL (Findings)*, pp. 20544–20552. Association for Computational Linguistics, 2025.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. Dialogsum: A real-life scenario dialogue summarization dataset. In *ACL/IJCNLP (Findings)*, Findings of ACL, pp. 5062–5074. Association for Computational Linguistics, 2021.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *EMNLP*, pp. 3829–3846. Association for Computational Linguistics, 2023.

Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. Provence: efficient and robust context pruning for retrieval-augmented generation. In *ICLR*. OpenReview.net, 2025.

Allan Collins and Elizabeth Loftus. A spreading activation theory of semantic processing. *Psychological Review*, 82:407–428, 11 1975. doi: 10.1037//0033-295X.82.6.407.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

Yuhong Dai, Jianxun Lian, Yitian Huang, Wei Zhang, Mingyang Zhou, Mingqi Wu, Xing Xie, and Hao Liao. Pretraining context compressor for large language models with embedding-based memory. In *ACL (1)*, pp. 28715–28732. Association for Computational Linguistics, 2025.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In *EMNLP*, pp. 1107–1128. Association for Computational Linguistics, 2024.

David L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL-HLT (1)*, pp. 2368–2378. Association for Computational Linguistics, 2019.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *CoRR*, abs/1704.05179, 2017.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings. In *EMNLP/IJCNLP (1)*, pp. 55–65. Association for Computational Linguistics, 2019.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *MRQA@EMNLP*, pp. 1–13. Association for Computational Linguistics, 2019.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *ICLR*. OpenReview.net, 2024.

Robert M. Gray and David L. Neuhoff. Quantization. *IEEE Trans. Inf. Theory*, 44(6):2325–2383, 1998.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net, 2022.

Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C. Park. EXIT: context-aware extractive compression for enhancing retrieval-augmented generation. In *ACL (Findings)*, pp. 4895–4924. Association for Computational Linguistics, 2025.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *EMNLP*, pp. 13358–13376. Association for Computational Linguistics, 2023.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *ACL (1)*, pp. 1658–1677. Association for Computational Linguistics, 2024.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL (1)*, pp. 1601–1611. Association for Computational Linguistics, 2017.

Aniruddha Kembhavi, Min Joon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *CVPR*, pp. 5376–5384. IEEE Computer Society, 2017.

Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Trans. Assoc. Comput. Linguistics*, 6:317–328, 2018.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: large-scale reading comprehension dataset from examinations. In *EMNLP*, pp. 785–794. Association for Computational Linguistics, 2017.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *CoNLL*, pp. 333–342. Association for Computational Linguistics, 2017.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*, 2020.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *EMNLP*, pp. 6342–6353. Association for Computational Linguistics, 2023.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey. In *NAACL (Long Papers)*, pp. 7182–7195. Association for Computational Linguistics, 2025a.

Zongqian Li, Yixuan Su, and Nigel Collier. 500xcompressor: Generalized prompt compression for large language models. In *ACL (1)*, pp. 25081–25091. Association for Computational Linguistics, 2025b.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

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

Xin Liu, Runsong Zhao, Pengcheng Huang, Xinyu Liu, Junyi Xiao, Chunyang Xiao, Tong Xiao, Shengxiang Gao, Zhengtao Yu, and Jingbo Zhu. Autoencoding-free context compression for llms via contextual semantic anchors. *CoRR*, abs/2510.08907, 2025.

Xinyu Liu, Runsong Zhao, Pengcheng Huang, Chunyang Xiao, Bei Li, Jingang Wang, Tong Xiao, and Jingbo Zhu. Forgetting curve: A reliable method for evaluating memorization capability for long-context models, 2024b. URL https://arxiv.org/abs/2410.04727.

Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–136, 1982.

Lvzhou Luo, Yixuan Cao, and Ping Luo. Attncomp: Attention-guided adaptive context compression for retrieval-augmented generation. In *EMNLP (Findings)*, pp. 8456–8472. Association for Computational Linguistics, 2025.

Jesse Mu, Xiang Li, and Noah D. Goodman. Learning to compress prompts with gist tokens. In *NeurIPS*, 2023.

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. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *ACL (Findings)*, pp. 963–981. Association for Computational Linguistics, 2024.

M. Ross Quillian. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, 12(5):410–430, 1967. doi: https://doi.org/10.1002/bs.3830120511. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/bs.3830120511.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, pp. 2383–2392. The Association for Computational Linguistics, 2016.

David Rau, Shuai Wang, Hervé Déjean, Stéphane Clinchant, and Jaap Kamps. Context embeddings for efficient answer generation in retrieval-augmented generation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, WSDM '25, pp. 493–502, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713293. doi: 10.1145/3701551.3703527. URL https://doi.org/10.1145/3701551.3703527.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. Duorc: Towards complex language understanding with paraphrased reading comprehension. In *ACL (1)*, pp. 1683–1693. Association for Computational Linguistics, 2018.

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

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling long-horizon LLM agent via context-folding. *CoRR*, abs/2510.11967, 2025.

Sijun Tan, Xiuyu Li, Shishir G. Patil, Ziyang Wu, Tianjun Zhang, Kurt Keutzer, Joseph Gonzalez, and Raluca A. Popa. Lloco: Learning long contexts offline. In *EMNLP*, pp. 17605–17621. Association for Computational Linguistics, 2024.

Jiwei Tang, Jin Xu, Tingwei Lu, Zhicheng Zhang, Yiming Zhao, Lin Hai, and Hai-Tao Zheng. Perception compressor: A training-free prompt compression framework in long context scenarios. In *NAACL (Findings)*, pp. 4093–4108. Association for Computational Linguistics, 2025a.

Jiwei Tang, Zhicheng Zhang, Shunlong Wu, Jingheng Ye, Lichen Bai, Zitai Wang, Tingwei Lu, Jiaqi Chen, Lin Hai, Hai-Tao Zheng, and Hong-Gee Kim. GMSA: enhancing context compression via group merging and layer semantic alignment. *CoRR*, abs/2505.12215, 2025b.

Jiwei Tang, Shilei Liu, Zhicheng Zhang, Qingsong Lv, Runsong Zhao, Tingwei Lu, Langming Liu, Haibin Chen, Yujin Yuan, Hai-Tao Zheng, Wenbo Su, and Bo Zheng. Read as human: Compressing context via parallelizable close reading and skimming. *CoRR*, abs/2602.01840, 2026a.

Jiwei Tang, Shilei Liu, Zhicheng Zhang, Yujin Yuan, Libin Zheng, Wenbo Su, and Bo Zheng. COMI: coarse-to-fine context compression via marginal information gain. *CoRR*, abs/2602.01719, 2026b.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence, 2025. URL https://arxiv.org/abs/2507.20534.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *Rep4NLP@ACL*, pp. 191–200. Association for Computational Linguistics, 2017.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artières, Axel-Cyrille Ngonga Ngomo, Norman Heino, Éric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinform.*, 16:138:1–138:28, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pp. 5998–6008, 2017.

Christopher K. I. Williams and Matthias W. Seeger. Using the nyström method to speed up kernel machines. In *NIPS*, pp. 682–688. MIT Press, 2000.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *AAAI*, pp. 14138–14148. AAAI Press, 2021.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: improving retrieval-augmented lms with context compression and selective augmentation. In *ICLR*. OpenReview.net, 2024.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, et al. Qwen2 technical report. *CoRR*, abs/2407.10671, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pp. 2369–2380. Association for Computational Linguistics, 2018.

Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. Compact: Compressing retrieved documents actively for question answering. In *EMNLP*, pp. 21424–21439. Association for Computational Linguistics, 2024.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. In *ICLR*. OpenReview.net, 2025a.

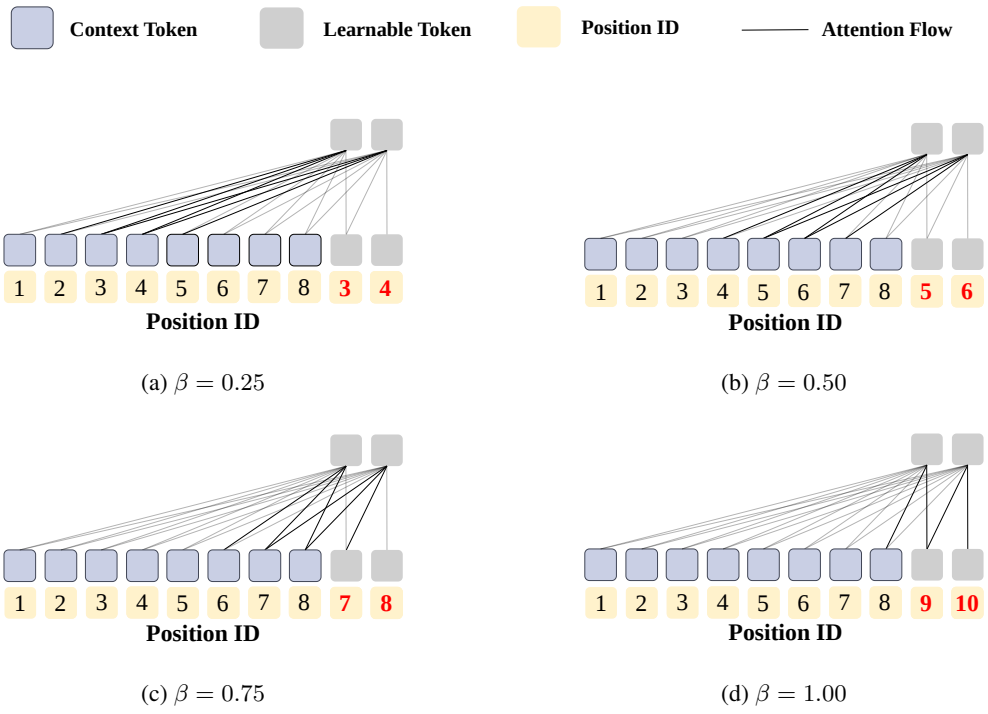
Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *ICLR*. OpenReview.net, 2020.

Tinghui Zhang, Yifan Wang, and Daisy Zhe Wang. SCOPE: A generative approach for LLM prompt compression. *CoRR*, abs/2508.15813, 2025b.

Runsong Zhao, Xin Liu, Xinyu Liu, Pengcheng Huang, Chunyang Xiao, Tong Xiao, and JingBo Zhu. Position ids matter: An enhanced position layout for efficient context compression in large language models. In *EMNLP (Findings)*, pp. 17715–17734. Association for Computational Linguistics, 2025.

Fengwei Zhou, Jiafei Song, Wenjin Jason Li, Gengjian Xue, Zhikang Zhao, Yichao Lu, and Bailin Na. Mooscomp: Improving lightweight long-context compressor via mitigating over-smoothing and incorporating outlier scores. *CoRR*, abs/2504.16786, 2025.

A Positional Encoding Bias Statement



(a) $\beta = 0.25$ (b) $\beta = 0.50$

(c) $\beta = 0.75$ (d) $\beta = 1.00$

Figure 5: Without modifying the attention mask, the bias factor β is used to control the Positional Encodings (PEs) of learnable tokens. Subfigures (a) to (d) correspond to the changes in the PEs of learnable tokens when β from $\{0.25, 0.50, 0.75, 1.00\}$.

To analyze the impact of *position bias* on methods based on learnable tokens, we define the *position bias factor* $\beta \in [0, 1]$, which is used to determine the starting Positional Encoding (PE) pos_{insert} of these learnable tokens:

$$pos_{\text{insert}} = \lfloor L \cdot \beta \rfloor + 1\,, \tag{14}$$

where L denotes length of the input sequence.

Consequently, the K learnable tokens are assigned as a contiguous segment with position IDs $\{pos_{\text{insert}}, \ldots, pos_{\text{insert}} + K - 1\}$, as depicted in Figure 5.

B Theoretical Analysis of Position Bias Mitigation

Overview

We establish three properties of the semantic aggregation operator $\mathcal{A}$: **(i)** permutation invariance (Theorem. 1), which structurally eliminates ordering-dependent position bias (Liu et al., 2024a); **(ii)** positional noise attenuation at rate $\mathcal{O}(|\mathcal{G}_k|^{-1})$ (Theorem. 2), which bounds residual variance after compression; and **(iii)** equivalence to a query-adaptive Nyström approximation (Proposition. 4), which connects the compression to kernel methods (Williams & Seeger, 2000; Xiong et al., 2021). Section B.5 instantiates these results on sinusoidal and RoPE encodings.

B.1 Preliminaries

Let $H = \{h_i\}_{i=1}^{n} \subset \mathbb{R}^d$ be the encoder hidden states, decomposed as

$$h_i = s_i + p_i, \tag{15}$$

where s_i represents the position-invariant semantic component, while p_i denotes the positional component. The semantic aggregation operator $\mathcal{A}$ with compression rate τ produces $K = max(2, \lceil n/\tau \rceil)$

aggregated tokens:

$$\alpha_i = \frac{\exp(w_i)}{\sum_{j\in\mathcal{G}_k}\exp(w_j)}, \quad \tilde{h}_k = \sum_{i\in\mathcal{G}_k}\alpha_i \cdot h_i, \tag{16}$$

where $\mathcal{G}_k = \{i : k = \arg\max_j s_{i,j}\}$ is the Voronoi partition (Lloyd, 1982) induced by centroids $\{c_k\}$, with $s_{i,j}$ as defined in Eq. 6, and w_i as defined in Eq. 8.

B.2 Permutation Invariance

Theorem 1 (Permutation Invariance). *For any permutation $\sigma \in S_n$, where S_n denotes the symmetric group of all $n!$ bijections on $\{1,\dots,n\}$,*

$$\mathcal{A}((h_1,\dots,h_n)) = \mathcal{A}((h_{\sigma(1)},\dots,h_{\sigma(n)})), \tag{17}$$

where $\mathcal{A} : \mathbb{R}^{n\times d} \to \mathbb{R}^{K\times d}$ is the semantic aggregation operator defined in Eq. 16, and $(h_{\sigma(1)},\dots,h_{\sigma(n)})$ denotes the input sequence reindexed under σ.

Proof. The centroid c_k is computed as a symmetric function of the input multiset $\{\{h_1,\dots,h_n\}\}$, so it is unaffected by any permutation $\sigma \in S_n$. Given fixed centroids, compression group membership $i \in \mathcal{G}_k$ is determined solely by the value of h_i, not by the index i. A permutation σ therefore preserves the content multiset $\mathcal{H}_k = \{\{h_i : i \in \mathcal{G}_k\}\}$ of every compression group. Since $\tilde{h}_k$ is a symmetric (softmax-weighted) function of $\mathcal{H}_k$, it is invariant to the ordering of its elements, and the conclusion follows. □

B.3 Positional Noise Attenuation

Let $\tilde{s}_k = \sum_{i\in\mathcal{G}_k}\alpha_i s_i$ be the target semantic aggregation representation, where $\{\alpha_i\}_{i\in\mathcal{G}_k}$ are non-negative weights satisfying $\sum_{i\in\mathcal{G}_k}\alpha_i = 1$. Define the positional residual $\epsilon_k = \tilde{h}_k - \tilde{s}_k = \sum_{i\in\mathcal{G}_k}\alpha_i p_i$.

Theorem 2 (Noise Attenuation). *Assume $\{p_i\}_{i\in\mathcal{G}_k}$ are independent with $\mathbb{E}[p_i] = 0$ and $\mathrm{Cov}(p_i) = \Sigma_p$ for all $i \in \mathcal{G}_k$. Then*

$$\mathbb{E}\|\tilde{h}_k - \tilde{s}_k\|^2 = \Big(\sum_{i\in\mathcal{G}_k}\alpha_i^2\Big)\,\mathrm{tr}(\Sigma_p). \tag{18}$$

In particular, under uniform weights $\alpha_i = 1/|\mathcal{G}_k|$, justified by semantic redundancy within a high-density cluster, this reduces to

$$\mathbb{E}\|\tilde{h}_k - \tilde{s}_k\|^2 = \frac{\mathrm{tr}(\Sigma_p)}{|\mathcal{G}_k|}. \tag{19}$$

Proof. Since $\mathbb{E}[p_i] = 0$ for all i, linearity of expectation gives $\mathbb{E}[\epsilon_k] = \sum_{i\in\mathcal{G}_k}\alpha_i\,\mathbb{E}[p_i] = 0$, so the bias term vanishes and

$$\mathbb{E}\|\epsilon_k\|^2 = \mathbb{E}\|\epsilon_k - \mathbb{E}[\epsilon_k]\|^2 = \mathrm{tr}\big(\mathrm{Cov}(\epsilon_k)\big).$$

By independence of $\{p_i\}$ and the identical covariance structure,

$$\mathrm{Cov}(\epsilon_k) = \sum_{i\in\mathcal{G}_k}\alpha_i^2\,\mathrm{Cov}(p_i) = \Big(\sum_{i\in\mathcal{G}_k}\alpha_i^2\Big)\Sigma_p,$$

and thus $\mathrm{tr}(\mathrm{Cov}(\epsilon_k)) = \big(\sum_{i\in\mathcal{G}_k}\alpha_i^2\big)\,\mathrm{tr}(\Sigma_p)$. Under uniform weights $\alpha_i = 1/|\mathcal{G}_k|$, we have $\sum_i \alpha_i^2 = 1/|\mathcal{G}_k|$, yielding the stated bound. □

Remark 3 (Correlated noise). If positional components are stationary with integer indices $i, j \in \mathcal{G}_k$ and $\mathrm{Cov}(p_i, p_j) = \Sigma_p\,\rho(|i-j|)$ for some scalar function $\rho : \mathbb{Z}_{\geq 0} \to [-1, 1]$ with $\rho(0) = 1$ (Toeplitz structure), then under uniform weights

$$\mathrm{Cov}(\epsilon_k) = \frac{\Sigma_p}{|\mathcal{G}_k|^2}\sum_{i,j\in\mathcal{G}_k}\rho(|i-j|).$$

A sufficient condition for preserving the $\mathcal{O}(|\mathcal{G}_k|^{-1})$ rate of Theorem 2 is absolute summability of the correlation function, i.e., $\sum_{\ell=0}^{\infty}|\rho(\ell)| < \infty$, which holds in particular when $\rho(\ell)$ decays exponentially. Under this condition, $\frac{1}{|\mathcal{G}_k|}\sum_{i,j}\rho(|i-j|) = \mathcal{O}(1)$, so $\mathrm{tr}(\mathrm{Cov}(\epsilon_k)) = \mathcal{O}(|\mathcal{G}_k|^{-1})\,\mathrm{tr}(\Sigma_p)$, and the off-diagonal contributions remain lower-order.

B.4 Nyström Approximation Perspective

Proposition 4 (Structural Correspondence to Nyström)**.** *Define the kernel matrices $\mathbf{K}_H \in \mathbb{R}^{n\times n}$, $\mathbf{K}_{HZ} \in \mathbb{R}^{n\times K}$, and $\mathbf{K}_{ZZ} \in \mathbb{R}^{K\times K}$ with entries $(\mathbf{K}_H)_{ij} = s_{i,j}$, $(\mathbf{K}_{HZ})_{ik} = s_{i,k}$, and $(\mathbf{K}_{ZZ})_{kl} = s_{k,l}$, respectively, where the similarity $s_{\cdot,\cdot}$ is given by Eq. 6. Then the aggregated representations $\{\tilde{h}_k\}$ satisfy*

$$\tilde{H} = A \cdot H, \qquad A_{ki} = \alpha_i^{(k)} \cdot \mathbf{1}[i \in \mathcal{G}_k], \tag{20}$$

where the assignment matrix $A \in \mathbb{R}^{K\times n}$ shares the similarity-weighted structure of the Nyström projection $\mathbf{K}_{ZZ}^{-1}\mathbf{K}_{ZH}$ in the sense that both operators assign weight to token i under landmark k proportionally to $s_{k,i}$: the former via softmax normalization within $\mathcal{G}_k$, the latter via global kernel inversion.

Proof. By definition, $\tilde{h}_k = \sum_{i\in\mathcal{G}_k} \alpha_i h_i$, so the full output $\tilde{H} \in \mathbb{R}^{K\times d}$ is a linear map $A \cdot H$ where $A_{ki} = \alpha_i^{(k)} \cdot \mathbf{1}[i \in \mathcal{G}_k]$. The Nyström approximation reconstructs $\mathbf{K}_H \approx \mathbf{K}_{HZ}\mathbf{K}_{ZZ}^{-1}\mathbf{K}_{ZH}$, where the per-landmark reconstruction weight for token i under landmark k is given by $(\mathbf{K}_{ZZ}^{-1}\mathbf{K}_{ZH})_{ki}$, which is proportional to $s_{k,i}$ when $\mathbf{K}_{ZZ}$ is well-conditioned. Since $\alpha_i^{(k)}$ is derived from the same similarity $s_{k,i}$ via softmax normalization over $\mathcal{G}_k$ (Eq. 16), A shares the same similarity-weighted structure restricted to each partition cell, establishing the structural correspondence. □

Remark 5 (Query-Adaptive Landmarks and Approximation Quality). The Nyström approximation error for a rank-K reconstruction is known to depend on how well the landmark subspace $\mathrm{span}\{c_k\}$ captures the dominant eigendirections of $\mathbf{K}_H$ (Williams & Seeger, 2000). Fixed-position landmarks (e.g., uniform or stride-based sampling) align with positional eigendirections that may be orthogonal to the query-relevant subspace. In contrast, SeCo selects landmarks by maximizing similarity between c_k and $\bar{q}$, concentrating $\mathrm{span}\{c_k\}$ on query-relevant eigendirections of $\mathbf{K}_H$. Under the standard Nyström error decomposition (Xiong et al., 2021), this implies lower reconstruction error on the query-relevant subspace.

B.5 Instantiation on Sinusoidal and RoPE Encodings

We instantiate Theorems 1–2 under the query-aware semantic aggregation of SeCo, focusing on the dominant behavior of positional residuals.

B.5.1 Sinusoidal Positional Encoding

In sinusoidal encoding, the positional component is additive and independent of semantics.

Permutation Invariance. Since both semantic assignment and consistency merging depend only on semantic similarity and query relevance, aggregation is permutation invariant.

Noise Attenuation.

Proposition 6 (Sinusoidal Case)**.** *Under bounded and normalized weights, the positional residual in each compression group satisfies*

$$\mathbb{E}\|\epsilon_k\|^2 = \mathcal{O}(|\mathcal{G}_k|^{-1}). \tag{21}$$

Sketch. The residual takes the form

$$\epsilon_k = \sum_{i\in\mathcal{G}_k} \alpha_i \sin(i\omega), \tag{22}$$

which is a weighted average of bounded oscillatory terms. Since $\sin(i\omega)$ is approximately zero-mean over phases, and SeCo assigns tokens based on semantic similarity rather than position, the indices within $\mathcal{G}_k$ can be viewed as approximately unordered samples.

Expanding the second moment,

$$\mathbb{E}\|\epsilon_k\|^2 = \sum_i \alpha_i^2 \mathbb{E}[\sin^2(i\omega)] + \sum_{i\neq j} \alpha_i\alpha_j \mathbb{E}[\sin(i\omega)\sin(j\omega)]. \tag{23}$$

The cross terms are suppressed due to oscillation and weak correlation between phases, leaving the dominant contribution proportional to $\sum_i \alpha_i^2$. Under bounded weights, $\sum_i \alpha_i^2 = \mathcal{O}(|\mathcal{G}_k|^{-1})$, which yields the stated result. □

B.5.2 Rotary Position Embedding (RoPE)

RoPE introduces position-dependent rotations, leading to a residual that depends on both content and position.

Permutation Invariance. Permutation invariance does not hold strictly, but is partially mitigated by semantic aggregation.

Noise Attenuation and Bias.

Proposition 7 (RoPE Case)**.** *Assume bounded hidden states and smooth variation of rotations. Then the residual satisfies*

$$\|\epsilon_k^{(\mathrm{R})}\| = \mathcal{O}(\Delta\theta_k) + \mathcal{O}(|\mathcal{G}_k|^{-1/2}), \tag{24}$$

where $\Delta\theta_k$ is the angular spread within the compression group.

Sketch. The residual can be written as

$$\epsilon_k^{(\mathrm{R})} = \sum_{i\in\mathcal{G}_k} \alpha_i(\mathbf{R}_{\theta,i} - \mathbf{I})h_i. \tag{25}$$

We decompose it into a mean and fluctuation term:

$$\epsilon_k^{(\mathrm{R})} = (\bar{\mathbf{R}}_k - \mathbf{I})\bar{h}_k + \sum_{i\in\mathcal{G}_k} \alpha_i(\mathbf{R}_{\theta,i} - \bar{\mathbf{R}}_k)h_i, \tag{26}$$

where $\bar{\mathbf{R}}_k = \sum_i \alpha_i \mathbf{R}_{\theta,i}$ and $\bar{h}_k = \sum_i \alpha_i h_i$.

The first term represents a bias induced by the average rotation, whose magnitude scales with the variation of rotation angles within the compression group, i.e., $\mathcal{O}(\Delta\theta_k)$. The second term aggregates zero-mean fluctuations around the mean rotation and behaves like a weighted average of weakly correlated terms, yielding a decay rate of $\mathcal{O}(|\mathcal{G}_k|^{-1/2})$. Combining both terms gives the stated bound. □

C Stress Test

To further evaluate the robustness of methods, we conduct a stress test experiment. In this experiment, SeCo, COMI, and RAM all adopted a random compression constraint strategy during the training phase: in each training iteration, we randomly selected a compression rate τ from the set $\{2, 4, 8, 16, 32\}$ (`LLaMA-3.2-1B-Instruct` as backbone). As shown in Figure 6, while increasing the compression rate from 2 to 32 leads to varying degrees of performance degradation across methods, *SeCo consistently maintains superior performance compared to COMI and RAM on both in-domain and out-of-domain settings.* This performance gap indicates that SeCo possesses stronger robustness under various compression constraints.

D Dataset Details

SQuAD. SQuAD (Rajpurkar et al., 2016) is a reading comprehension benchmark derived from Wikipedia, where models must locate answer spans directly within the provided passage to respond to given questions.

NewsQA. NewsQA (Trischler et al., 2017) is a reading comprehension dataset built on CNN news articles, posing questions that frequently require inference beyond surface-level text matching.

TriviaQA. TriviaQA (Joshi et al., 2017) is a large-scale QA dataset pairing independently authored trivia questions with evidence documents sourced from Wikipedia and the web.

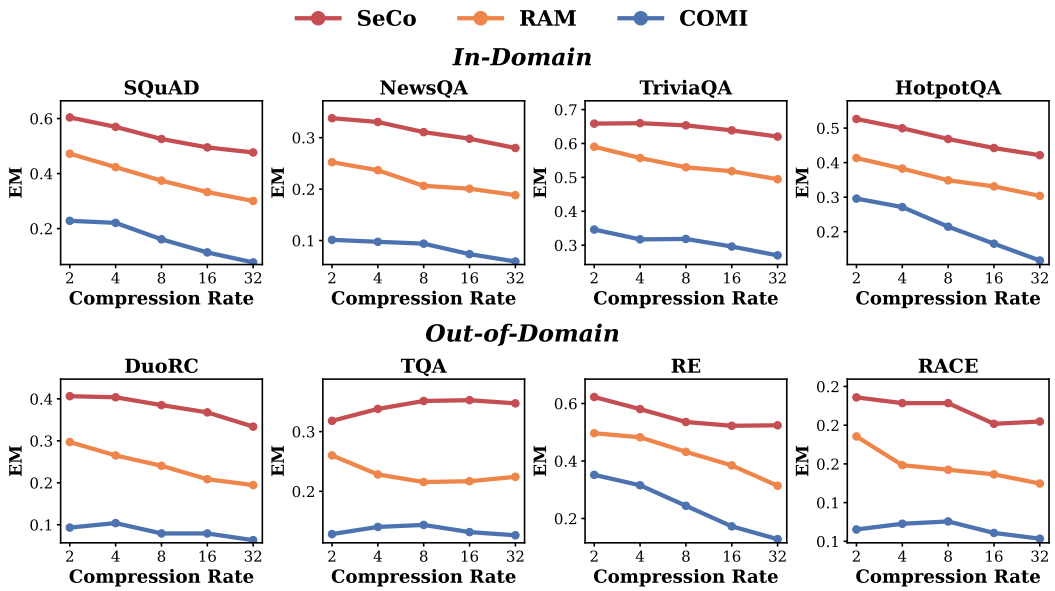


Figure 6: Exact Match (EM) across both in-domain and out-of-domain QA datasets at different compression rates $\{2, 4, 8, 16, 32\}$.

SearchQA. SearchQA (Dunn et al., 2017) is a QA dataset that pairs Jeopardy! clues with web-retrieved snippets as context, challenging models to synthesize information across multiple documents.

HotpotQA. HotpotQA (Yang et al., 2018) is a multi-hop QA dataset requiring models to jointly reason over several supporting documents, with annotated supporting facts provided to facilitate explainability.

NaturalQuestions. NaturalQuestions (Kwiatkowski et al., 2019) is an open-domain QA benchmark composed of real user queries issued to the Google search engine. Models must extract both short and long-form answers from Wikipedia articles, making it a realistic testbed for evaluating reading comprehension under natural information-seeking conditions.

DROP. DROP (Dua et al., 2019) is a reading comprehension benchmark emphasizing discrete reasoning over text, where answering questions may involve numerical operations such as addition, subtraction, counting, or sorting.

BioASQ. BioASQ (Tsatsaronis et al., 2015) is a biomedical QA dataset featuring expert-curated questions paired with PubMed abstracts, demanding specialized domain knowledge to generate exact or abstractive answers.

DuoRC.ParaphraseRC. DuoRC.ParaphraseRC (Saha et al., 2018) is a reading comprehension dataset grounded in movie plot summaries, where the ParaphraseRC variant supplies a paraphrased context document, substantially reducing lexical overlap with the original question.

TextbookQA. TextbookQA (Kembhavi et al., 2017) is a QA dataset drawn from middle school science textbooks, requiring comprehension of both textual descriptions and accompanying diagrams to answer curriculum-based questions.

RelationExtraction. RelationExtraction (Levy et al., 2017) casts relation extraction as a reading comprehension problem, mapping each relation type to a natural language question and requiring the model to extract the corresponding answer span from the input sentence.

RACE. RACE (Lai et al., 2017) is a multiple-choice reading comprehension dataset sourced from English exams administered in China, encompassing diverse topics and demanding skills such as reasoning, summarization, and commonsense understanding.

DialogSum. DialogSum (Chen et al., 2021) is a dialogue summarization dataset pairing everyday spoken conversations with human-written summaries, with the goal of distilling key information and communicative intent from conversational text.

NarrativeQA. NarrativeQA (Kociský et al., 2018) is a reading comprehension dataset grounded in full-length books and movie scripts, where questions are crafted by annotators who only see human-written summaries, necessitating deep narrative understanding to answer correctly. We truncate each input to at most 32K tokens.

E Training Details

We fine-tune SeCo with bf16 precision using data sampled from the MRQA benchmark (Fisch et al., 2019). For the QA task, we construct our training set by randomly selecting 4,000 samples per subset. LLaMA-3.2-1B-Instruct[1] is trained on 4 NVIDIA RTX 4090 GPUs, while Qwen3-4B-Instruct[2] is trained on 2 NVIDIA A800 GPUs. For the summarization task and long-context QA, we fine-tune Qwen3-4B-Instruct[2] on DialogSum and NarrativeQA using 2 NVIDIA A800 GPUs and 32 NVIDIA H200 GPUs, respectively. **To ensure a fair comparison, all baselines are trained using the same datasets and hyperparameters.** Detailed training configurations are summarized in Table 5.

Table 5: Hyperparameters for fine-tuning.

Hyperparameter	**Value**
Per Device Batch Size	1
Optimizer	AdamW
Betas	$(0.9, 0.999)$
Learning Rate	1×10^{-5}
LoRA Rank	128
LoRA Alpha	32
LoRA Dropout	0.05
Target Modules	all-linear
Warmup Ratio	0.1
Weight Decay	0.01
Scheduler	cosine
Precision	bf16

F Limitation

While SeCo effectively mitigates position bias and enhances compression quality through semantic space clustering, it still possesses certain limitations. Specifically, SeCo introduces a query-aware clustering process during the compression phase. Although this process is lightweight compared to backbone model inference and the overhead of other compression methods (see Tab. 3), it still introduces additional computational costs during the end-to-end generation stage. Notably, however, the process of assigning non-centroid tokens to cluster centers is inherently parallelizable, which alleviates the aforementioned efficiency concerns.

G Comparison with Prompting and Full Fine-Tuning

We further compare SeCo with three standard baselines: zero-shot prompting, the original prompt without additional training, and full-parameter fine-tuning (FT). As shown in Tables 5 and 6, SeCo achieves substantial improvements over both zero-shot and original prompting under the strict $16\times$ compression constraint. On the ID benchmarks, SeCo obtains an average F1 score of 68.16 and 72.50 with LLaMA-3.2-1B and Qwen3-4B, respectively, substantially outperforming the original prompts (50.05 and 57.03) and zero-shot prompting (16.40 and 22.78). Notably, SeCo is competitive with full fine-tuning, slightly surpassing it with Qwen3-4B (72.50 vs. 71.24) and remaining close with LLaMA-3.2-1B (68.16 vs. 70.05). Similar trends are observed on OOD datasets, where SeCo consistently outperforms zero-shot prompting and provides a clear gain over the original prompt for Qwen3-4B. SeCo also achieves the best performance on several individual benchmarks, including

[1]https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct

[2]https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507

SearchQA and TQA. These results demonstrate that SeCo preserves strong task performance under aggressive context compression, substantially narrowing the gap to full fine-tuning while avoiding the cost of updating all model parameters.

Table 5: Comparison with zero-shot prompting, original prompting, and full-parameter fine-tuning on in-domain (ID) benchmarks.

Methods	SQuAD		NewsQA		TriviaQA		SearchQA		HotpotQA		NQ		Average	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
LLaMA-3.2-1B-Instruct														
SeCo	51.84	70.89	28.89	49.94	67.30	75.06	**65.89**	**74.82**	45.55	67.17	39.05	59.35	52.89	68.16
LLaMA-ZeroShot	3.45	11.38	1.09	3.22	24.46	31.13	14.26	19.74	3.20	8.54	4.99	15.10	9.55	16.40
LLaMA-Original	50.35	68.15	25.33	47.18	53.96	63.87	21.82	29.91	41.16	59.05	33.15	50.30	35.98	50.05
LLaMA-FT	**75.00**	**83.42**	**39.03**	**55.07**	**70.62**	**76.30**	59.72	66.56	**55.09**	**70.94**	**51.18**	**64.43**	**60.08**	**70.05**
Qwen3-4B-Instruct														
SeCo	57.38	75.56	33.43	55.45	69.34	76.65	**73.77**	**81.27**	48.23	69.48	42.78	62.84	57.88	**72.50**
Qwen-ZeroShot	5.06	15.23	0.69	3.76	28.66	37.00	27.99	35.60	4.83	11.72	4.62	14.68	14.47	22.78
Qwen-Original	34.28	60.81	8.74	38.18	44.86	59.31	57.96	67.51	43.53	63.34	14.35	41.95	37.30	57.03
Qwen-FT	**79.50**	**88.87**	**43.45**	**61.98**	**76.13**	**81.93**	51.25	56.81	**61.63**	**77.43**	**54.32**	**69.59**	**60.84**	71.24

Table 6: Comparison with zero-shot prompting, original prompting, and full-parameter fine-tuning on out-of-domain (OOD) benchmarks.

Methods	DROP		BioASQ		DuoRC		TQA		RE		RACE		Average	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
LLaMA-3.2-1B-Instruct														
SeCo	26.35	40.99	31.45	48.76	37.24	50.84	**42.51**	**53.71**	56.21	72.93	21.96	38.64	40.20	55.33
LLaMA-ZeroShot	4.13	13.53	14.83	27.11	1.00	2.74	18.96	26.47	4.65	9.94	1.04	4.38	7.57	14.25
LLaMA-Original	21.16	36.61	35.04	59.14	32.78	47.23	41.78	52.08	65.57	77.50	24.93	40.32	42.21	56.97
LLaMA-FT	**36.39**	**44.29**	**59.84**	**74.79**	**51.50**	**61.11**	41.58	49.35	**74.83**	**85.61**	**33.98**	**46.59**	**54.81**	**65.27**
Qwen3-4B-Instruct														
SeCo	38.26	53.18	48.34	64.12	43.04	56.15	**55.62**	**66.23**	57.84	73.49	29.82	48.89	48.69	63.30
Qwen-ZeroShot	1.86	13.34	24.14	45.05	0.73	4.05	40.39	52.46	5.90	15.24	1.34	5.56	12.37	22.99
Qwen-Original	5.52	30.85	30.32	61.50	15.46	37.65	50.77	65.35	56.51	72.66	15.73	38.05	34.32	55.37
Qwen-FT	**58.82**	**68.09**	**62.77**	**78.92**	**55.16**	**65.29**	54.96	62.93	**76.53**	**87.91**	**42.88**	**56.27**	**62.57**	**73.78**