AUTOENCODING-FREE CONTEXT COMPRESSION FOR LLMS VIA CONTEXTUAL SEMANTIC ANCHORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Context compression presents a promising approach for accelerating large language model (LLM) inference by compressing long contexts into compact representations. Current context compression methods predominantly rely on autoencoding tasks to train context-agnostic compression tokens to compress contextual semantics. While autoencoding tasks enable compression tokens to acquire compression capabilities, compression via autoencoding tasks creates a fundamental mismatch: the models are optimized for reconstruction that diverge from actual downstream tasks, thereby weakening the features more beneficial for real-world usage. We propose Semantic-Anchor Compression (SAC), a novel method that shifts from autoencoding task based compression to an architecture that is equipped with this compression capability *a priori*. Instead of training models to compress contexts through autoencoding tasks, SAC directly selects so-called anchor tokens from the original context and aggregates contextual information into their key-value (KV) representations. By deriving representations directly from the contextual tokens, SAC eliminates the need for autoencoding training. To ensure compression performance while directly leveraging anchor tokens, SAC incorporates two key designs: (1) anchor embeddings that enable the compressor to identify critical tokens, and (2) bidirectional attention modification that allows anchor tokens to capture information from the entire context. Experimental results demonstrate that SAC consistently outperforms existing context compression methods across various compression ratios. On out-of-distribution evaluation using MRQA, SAC achieves 1 EM improvement at 5x compression over strong baselines, with increasing advantages at higher compression ratios.

1 INTRODUCTION

The expanding scope of large language models (LLMs) to tasks like processing long documents (Liu et al., 2024b; Li et al., 2024; Duan et al., 2025), maintaining multi-turn dialogue coherence (Zhang et al., 2025; Yi et al., 2025; Guan et al., 2025), and generating responses grounded in extensive external knowledge (Lewis et al., 2020; Karpukhin et al., 2020; Huang et al., 2025) necessitates the incorporation of vast contexts into the model input. However, directly processing such extremely long contexts is fraught with challenges, including prohibitive computational costs, significant inference latency, and performance degradation, largely caused by the "lost-in-the-middle" phenomenon (Liu et al., 2024a).

To address these challenges, recent studies have proposed context compression (Chang et al., 2024; Li et al., 2025a), a technique that typically appends special tokens (i.e. compression tokens) to the end of the context and leverages the LLM's causal attention mechanism to compress contextual information into a compact representation within these tokens. Once this compact representation is obtained, the LLM can generate responses conditioned on it, rather than being conditioned on the entire original context. This significant reduction in context length leads to substantial decreases in both inference time and GPU memory consumption. While effective, these approaches face a key limitation: the compression tokens are randomly initialized and lack inherent semantic information. To compensate, they typically rely on extensive pretraining on both autoencoding (AE) and language modeling (LM) tasks (illustrated in Figure 1) to endow the compression tokens with the ability to carry contextual information. While AE task has shown to be necessary in ICAE (Ge et al., 2024) since compression tokens lack context-relevant semantics, the AE task requires the compressed

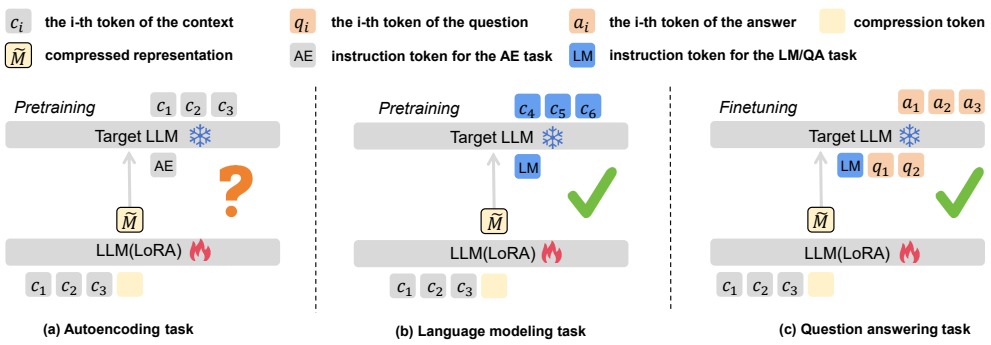

Figure 1: Three tasks for training the context compressor introduced by ICAE and followed by numerous works. The training uses (a) Autoencoding task and (b) Language modeling task to pretrain the encoder, then finetunes on (c) Question answering task.

representation to reconstruct all tokens in the context, even low-information tokens. This reliance on a suboptimal and costly pretraining stage raises a critical research question: Is it possible to design a compression architecture that inherently understands context without a demanding AE phase?

To answer this question, this work introduces Semantic-Anchor Compression (SAC) (Figure 2), a novel architecture for the context compression task. Instead of appending new special tokens and requiring extensive autoencoding pretraining to learn their representations, SAC directly selects representative tokens from the original context to act as 'anchor tokens' for compression. By leveraging these semantically meaningful anchors from the input itself, SAC incorporates natural semantic priors that obviate the need for autoencoding pretraining. To signify their special role, these selected tokens are then augmented with dedicated 'anchor embeddings', enabling the LLM to distinguish them from regular tokens. Furthermore, to enhance their compression capabilities, we modify the standard causal attention to a bidirectional attention mechanism. This allows anchor tokens to access information from the entire context, rather than being restricted to only preceding tokens. These modifications collectively foster a more effective context compression by providing anchor tokens with both distinct representations and comprehensive contextual awareness. Empirically, we test SAC on the MRQA (Fisch et al., 2019a) dataset and confirm that it outperforms existing strong context compression baselines. For example, compared to 500xCompressor (Li et al., 2025b) at 5x compression, the average exact match (EM) improves from 25.4 to 32.3. Results show that 1) our proposed method improves more in absolute accuracy over strong baselines on more challenging high compress ratio scenarios 2) our proposed architecture achieves its best performance in a simpler training setting without autoencoding training arguably because the anchor tokens already contain enough information about the original context. Our analysis reveals that SAC's compressed representations more closely resemble original context token KVs in feature space, so that LLMs performing inferenece can arguably better understand them.

2 Related Works

2.1 Compression Method

Many methods focus on reducing prompt lengths. CC (Wingate et al., 2022) utilizes contrastive learning to compress specific natural language prompts into shorter and unique soft prompt tokens. However, it cannot generalize to unseen prompts and requires retraining for new prompts. GIST (Mu et al., 2023) compresses original prompts into KV values through finetuning and can handle arbitrary unseen contexts. AutoCompressor (Chevalier et al., 2023) recursively combines compressed vectors with sub-prompts and aggregates all compressed vectors to construct the final representation, enabling compression of longer contexts. However, both GIST and AutoCompressor require finetuning the LLMs performing inference (referred to later as target LLM), which may affect LLMs' original capabilities.

ICAE (Ge et al., 2024) formulates context compression as training a general encoder that compresses contexts into compact representations understandable by target LLMs without finetuning. To achieve

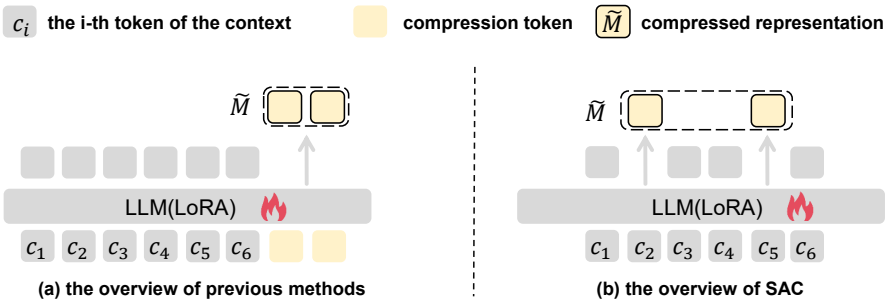

Figure 2: The difference between SAC and previous methods. While previous methods (a) compress contextual information into context-agnostic special tokens (referred to as compression tokens), SAC (b) compresses the context directly into the original contextual tokens themselves. Here, $\tilde{M}$ can represent either the output from the final layer of the LLM or the Key-Value pairs, being later on used as compressed representations for LLM inference.

this, ICAE introduces autoencoding tasks and performs joint pretraining with language modeling tasks, followed by finetuning on downstream tasks. 500xCompressor (Li et al., 2025b) improves upon ICAE by replacing the compression carrier from the last layer output of compression tokens with KV values at each layer, achieving higher compression ratios. EPL (Zhao et al., 2025) identifies that ICAE and 500xCompressor neglect the impact of positional encoding and proposes distributing compression token position IDs uniformly across the entire context rather than placing them at the end. However, these methods still rely on autoencoding tasks to endow the compression tokens with the ability to carry contextual information.

Another category of prompt compression methods is based on token selection, which selects representative tokens from contexts based on token importance. SelectiveContext (Li et al., 2023), LLMLingua (Jiang et al., 2023), and LongLLMLingua (Jiang et al., 2024) employ causal small language models to evaluate token importance based on information entropy. LLMLingua-2 (Pan et al., 2024a) distills a token classifier to compute the probability of each token to be preserved. These works demonstrate that LLMs can understand original contexts using a small number of representative tokens. However, they do not perform compressed tokens training which limits the usability of the selected tokens by target LLMs. Our proposed SAC can be seen as a combination of token selection methods and compressed token training methods: it derives and train compressed representations that are based on tokens selected directly from the context and indeed is compatible with the token selection methods above.

2.2 Bidirectional Attention

Recent studies have shown that, removing the decoder's unidirectional causal constraint and introducing bidirectional attention can effectively enhance the model's representational capacity (Wang et al., 2020). For instance, NV-Embed (Lee et al., 2025) replaces causal attention with bidirectional attention during contrastive training, achieving strong performance on general text embedding and dense vector retrieval tasks. LLM2Vec (BehnamGhader et al., 2024), by enabling bidirectional attention alongside masked next-token prediction, significantly improves the model's ability to capture global semantics in text embedding tasks. These works indicate that bidirectional attention is advantageous for acquiring global semantic information and robust contextual representations. However, its effectiveness in text compression tasks remains underexplored. Motivated by these findings, we incorporate bidirectional attention into the compressor to enhance contextual modeling during the compression phase.

3 METHOD

3.1 TASK FORMULATION

Context compression is formally defined as follows: an encoder $\mathcal{E}$ compresses a context $C = (c_1, c_2, \ldots, c_{|C|})$ into a compact representation $\tilde{M}$ with $\tilde{M} = \mathcal{E}(C)$. Subsequently, a target LLM leverages the compressed representation $\tilde{M}$ in place of the original context C to perform various tasks, such as question answering.

To train the encoder $\mathcal{E}$ to effectively extract contextual information, ICAE introduces three objective functions. The autoencoding loss $\mathcal{L}_{\mathbf{AE}}$ ensures that the compressed representation $\tilde{M}$ generated by $\mathcal{E}$ preserves all tokens in the context C, regardless of their relative importance, as shown in Figure 1a; mathematically, $\mathcal{L}_{\mathbf{AE}} = -\log P(C|\tilde{M})$. The language modeling loss $\mathcal{L}_{\mathbf{LM}}$ encourages $\tilde{M}$ to maintain predictive capability for future context $C' = (c_{|C|+1}, c_{|C|+2}, \ldots, c_{|C|+|C'|})$, enabling proactive information planning, as shown in Figure 1b; mathematically, $\mathcal{L}_{\mathbf{LM}} = -\log P(C'|\tilde{M})$. During pretraining, $\mathcal{L}_{\mathbf{AE}}$ and $\mathcal{L}_{\mathbf{LM}}$ are jointly optimized to obtain an initially effective encoder $\mathcal{E}$.

Additionally, during finetuning, the question answering loss $\mathcal{L}_{\mathbf{QA}}$ enhances the ability of $\tilde{M}$ to extract information that is potentially relevant for downstream tasks (e.g. QA). Since the encoder operates without knowledge of what questions might be asked later, it learns to identify and preserve information that is likely to be queried, enabling accurate answer generation $A = (a_1, a_2, \ldots, a_{|A|})$ when presented with subsequent questions $Q = (q_1, q_2, \ldots, q_{|Q|})$, as shown in Figure 1c; mathematically, $\mathcal{L}_{\mathbf{QA}} = -\log P(A|\tilde{M}, Q)$.

3.2 SEMANTIC-ANCHOR COMPRESSOR

A key distinction between our approach SAC and previous methods is that we derive compressed representations directly from selected context tokens, as shown in Figure 2. This involves selecting a subset of tokens from context C as anchor tokens $S \subseteq C$. We believe that a good selection strategy benefits SAC. Following EPL, our default strategy divides the entire context C into $|S|$ chunks and selects the middle token from each chunk. This setting helps maximize coverage of context C. As illustrated in Figure 3a, selected tokens $c_i \in S$ are enhanced with anchor embeddings e_A, yielding an embedding sequence $E = (e_1, e_2, \ldots, e_{|C|})$:

$$e_i = \mathbf{Emb}(c_i) + \mathbf{1}_{c_i \in S} \cdot e_A \tag{1}$$

where $\mathbf{1}_{c_i \in S}$ is an indicator function that equals 1 when $c_i \in S$ and 0 otherwise. Following previous works, we employ a LLM with LoRA parameters θ_{LoRA} as the compressor: $\tilde{M} = \mathcal{E}(C) = \mathbf{LLM}(E|\theta_{LoRA})$.

While using original tokens from the context avoids learning compressed tokens from scratch and potentially improves the learning efficiency. We notice that because the encoder uses causal attentions, the anchor tokens S do not have visibility to the full sentence, limiting its representation power. Hence we modify the LLM from using causal attention to use bidirectional attention (see Figure 3b), enhancing the LLM's encoding capability. $\tilde{M}$ can be either the output of anchor tokens from the LLM's final layer or the Key-Value pairs from each layer. Following 500xCompressor, we use Key-Value pairs as the compressed representation $\tilde{M}$. During pretraining, we only use $\mathcal{L}_{\mathbf{LM}}$ and do not use $\mathcal{L}_{\mathbf{AE}}$ to train the compressor. Following previous work, we use $\mathcal{L}_{\mathbf{QA}}$ for finetuning.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTING

Dataset. For continued pretraining, we utilize the large-scale corpus SlimPajama-6B (Soboleva et al., 2023). During finetuning and evaluation, we employ the standard MRQA (Fisch et al., 2019b) question-answering dataset, which consolidates multiple QA tasks and standardizes them into a unified format. We evaluate SAC on both test sets, namely in-domain (ID) and out-of-domain (OOD), to comprehensively assess its in-distribution fitting ability and cross-domain generalization performance.

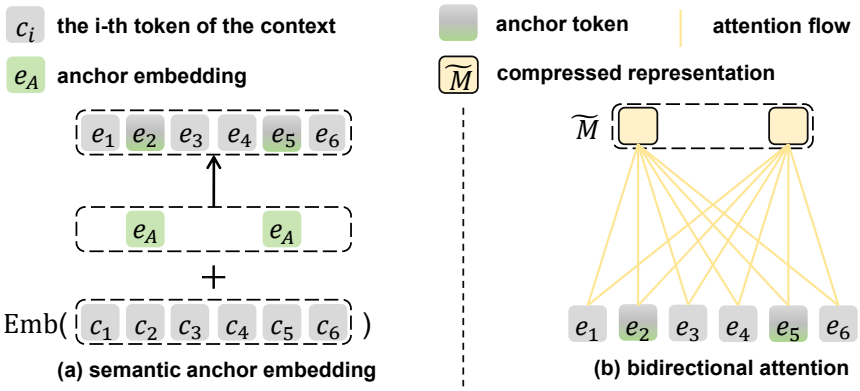

Figure 3: Key differentiators within SAC model architecture. (a) Representative tokens are transformed into anchor tokens through anchor embeddings. (b) The encoder in SAC adopts bidirectional attention, while the decoder operates with causal attention.

Implementation Details. SAC utilizes Llama-3.2-1B (Grattafiori & Dubey, 2024) as both the encoder and target LLM. The encoder is equipped with trainable LoRA (Hu et al., 2022) adapters (rank = 128, $\alpha = 256$), while the target LLM parameters remain frozen. For each context, we partition it into sub-contexts of 510 tokens each. The compressor compresses each sub-context into a sub-compressed representation, and subsequently concatenates these sub-compressed representations to form the complete compressed representation. The number of anchor tokens $|S| = \lfloor L/r \rfloor$ is determined by the compression ratio r and the length of the sub-context L. We train all models in two stages: pretraining for 20,000 optimization steps followed by finetuning for an additional 20,000 steps, both conducted with a batch size of 16. Complete hyperparameter configurations are provided in Appendix A.

Baselines. We use the Llama-3.2-1B model trained on the MRQA (Fisch et al., 2019b) dataset as an uncompressed baseline(denoted as "Full-FT"). We compare our method against several context compression techniques. For hard compression, we choose LLMLingua-2 (Pan et al., 2024b) and evaluate its performance on the Full-FT model. For soft compression, we select ICAE (Ge et al., 2024), 500xCompressor (Li et al., 2025b), and EPL (Zhao et al., 2025). To ensure a fair comparison, all these soft compression baselines are trained on the same dataset as our SAC method.

4.2 Finetuning Results

Tables 1 and 2 report the evaluation results of SAC on in-domain and out-of-domain MRQA datasets, which we analyze from three perspectives: overall performance, effect of compression ratio, and domain generalization.

Overall Performance. SAC consistently outperforms all baselines across a variety of conditions, including compression ratios, and both in-domain and out-of-domain tests, as shown in Tables 1 and 2. Averaging the results of the context compression methods across different compression ratios, SAC shows a maximum improvement of 24.6% F1 / 28.6% EM and a minimum improvement of 4.6% F1 / 5.7% EM in in-domain evaluations. For out-of-domain tests, the maximum improvement is 32.5% F1 / 36.2% EM, with a minimum improvement of 4.6% F1 / 6.9% EM.

Impact of Compression Ratio. We conducted a detailed evaluation of model performance under different compression ratios (5x, 15x, and 51x), as shown in Tables 1 and 2. As expected, F1 and EM scores of all methods decrease with increasing compression ratio, from 5x to 51x, since higher compression ratios result in more information being discarded. At the highest compression rate of 51×, the performance of different compression methods is not consistent. While one method may perform well on certain datasets, it may underperform on others. Nonetheless, SAC consistently achieves the best average performance.

Cross-Domain Generalization. We evaluated the generalization capability of SAC on out-of-domain datasets, as shown in Table 2. Under all compression ratio constraints, SAC consistently achieves the highest average F1/EM scores among all methods. Specifically, at a 5x compression ratio, SAC

Table 1: For the finetuning results, we report in-domain performance using ROUGE-1 F1 (Lin, 2004) and exact match (EM) (Maalouly, 2022) scores on the following datasets: 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 (NQ) (Kwiatkowski et al., 2019).

Methods	SQuAD		NewsQA		TriviaQA		SearchQA		HotpotQA		NQ		Avg	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Full-FT	77.69	59.71	63.5	46.04	68.80	60.54	73.25	62.07	74.78	59.26	71.01	53.47	71.51	56.85
Lingua-2	32.93	19.57	26.78	13.20	9.67	8.12	45.4	31.80	36.1	22.05	40.08	22.01	31.83	19.46
5x compression constraint														
ICAE	36.20	22.12	28.06	13.77	54.63	45.59	65.12	53.06	48.79	33.40	52.36	34.99	47.53	33.82
500x	51.62	33.63	39.70	22.63	57.62	48.76	66.43	54.38	59.10	42.20	57.11	39.26	55.26	40.14
EPL	64.72	44.28	48.74	**27.45**	63.75	54.54	69.69	57.73	67.16	49.79	63.32	44.16	62.90	46.33
SAC	**65.37**	**44.83**	**49.39**	27.14	**65.06**	**55.93**	**69.99**	**58.06**	**67.41**	**50.28**	**64.56**	**45.44**	**63.63**	**46.95**
15x compression constraint														
ICAE	31.90	18.91	25.25	11.97	51.78	42.94	64.81	52.89	45.22	30.32	48.01	30.67	44.50	31.28
500x	40.68	24.97	32.01	16.76	53.84	44.86	65.65	53.70	53.01	36.30	50.93	33.26	49.35	34.98
EPL	44.58	27.91	33.34	16.69	56.16	47.09	66.36	54.13	54.88	38.38	53.80	35.71	51.52	36.65
SAC	**47.43**	**30.25**	**36.55**	**18.07**	**61.13**	**52.19**	**68.97**	**56.76**	**58.83**	**41.86**	**56.79**	**38.88**	**54.95**	**39.67**
51x compression constraint														
ICAE	26.17	14.58	22.48	9.69	47.62	39.23	64.31	52.80	38.91	24.78	42.87	26.86	40.39	27.99
500x	30.09	17.11	25.06	12.20	50.84	42.13	64.92	53.29	42.15	27.32	46.07	29.53	43.19	30.26
EPL	30.09	17.49	24.49	11.54	51.15	42.38	65.12	53.16	42.19	27.23	46.29	29.77	43.22	30.26
SAC	**31.81**	**18.78**	**27.36**	**13.56**	**56.73**	**47.85**	**65.82**	**53.76**	**48.28**	**32.84**	**48.22**	**31.70**	**46.37**	**33.08**

attains average F1 and EM scores of 47.72 and 32.30, outperforming the second-best EPL method by 0.77 and 1.0 points, respectively. At a more challenging 15x compression ratio, SAC achieves average F1 and EM scores of 39.26 and 26.02, surpassing EPL by 2.52 and 2.19 points, with an EM improvement approaching 10%. Even at an extreme 51x compression ratio, SAC maintains average F1 and EM scores of 32.24 and 21.44, still leading EPL by 2.02 and 1.96 points, respectively. These results indicate that the compressed representations learned by SAC exhibit strong cross-domain robustness.

Table 2: For the finetuning results, we report out-of-domain performance using ROUGE-1 F1 and exact match (EM) scores on the following datasets: BioASQ (Tsatsaronis et al., 2015), DROP (Dua et al., 2019), DuoRC (Saha et al., 2018), RACE (Lai et al., 2017), Relation Extraction (RE) (Levy et al., 2017), and TextbookQA (TQA) (Kembhavi et al., 2017).

Methods	BioASQ		DROP		DouRC		RACE		RE		TQA		Avg	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Full-FT	49.37	36.77	44.67	34.46	48.82	35.51	35.57	9.64	83.34	72.46	53.32	32.4	52.51	36.87
Lingua-2	27.76	19.48	27.28	18.83	27.07	18.32	17.54	4.15	39.30	20.59	28.42	15.83	27.90	16.20
5x compression constraint														
ICAE	36.08	26.06	28.95	21.09	16.67	10.79	15.65	3.12	54.73	41.01	35.24	20.96	31.22	20.51
500x	40.30	28.99	35.40	25.55	29.43	19.32	21.57	4.90	65.43	50.88	38.62	22.75	38.46	25.40
EPL	**46.05**	**32.58**	39.94	28.94	39.10	**27.12**	**30.99**	6.08	76.07	62.31	49.54	30.74	46.95	31.30
SAC	44.66	31.45	**41.55**	**30.87**	**39.48**	26.92	30.53	**6.23**	**77.87**	**65.40**	**52.24**	**32.93**	**47.72**	**32.30**
15x compression constraint														
ICAE	35.51	24.47	30.39	21.96	13.78	9.06	15.21	3.71	55.24	40.33	34.75	21.56	30.81	20.18
500x	36.30	25.93	33.46	23.55	20.53	12.72	18.49	3.41	54.37	41.11	41.09	25.82	34.04	22.09
EPL	40.52	28.52	32.16	22.29	25.70	16.39	20.97	4.01	59.75	46.34	41.31	25.42	36.74	23.83
SAC	**41.31**	**28.66**	**36.72**	**27.48**	**28.94**	**18.99**	**23.35**	**4.90**	**61.04**	**47.90**	**44.21**	**28.21**	**39.26**	**26.02**
51x compression constraint														
ICAE	33.82	23.67	27.94	19.29	11.14	6.86	14.89	3.41	47.02	34.02	33.08	19.83	27.98	17.85
500x	32.17	23.07	**30.11**	**21.76**	13.42	8.53	15.18	2.67	**54.62**	**41.86**	37.10	22.62	30.43	20.09
EPL	32.52	22.21	29.64	20.89	13.16	8.13	**17.15**	3.12	53.72	40.37	35.15	22.16	30.22	19.48
SAC	**36.95**	**26.86**	29.52	20.89	**21.85**	**14.26**	15.87	**4.00**	48.19	36.43	**41.05**	**26.21**	**32.24**	**21.44**

4.3 Ablation Study

To verify the effectiveness of each key component and strategy in the SAC architecture, we conduct three groups of ablation studies, all performed under a $5\times$ compression ratio.

Component Ablation. As shown in Table 3, our ablation study clearly demonstrates the critical roles of the bidirectional attention and anchor embedding. Removing either component leads to substantial performance degradation in both in-domain (ID) and out-of-domain (OOD) settings. The bidirectional attention mechanism enables anchor tokens to more effectively integrate information from the entire context, producing compressed representations that are more beneficial for downstream tasks. Meanwhile, the anchor embedding provides explicit structural signals that guide the model to accurately identify and process these key tokens, thereby ensuring the effectiveness of information compression.

Table 3: Component ablation results. We report the average F1/EM performance of the model on in-domain (ID) and out-of-domain (OOD) tasks after removing the bidirectional attention (w/o mask) and the anchor embedding (w/o anchor). Full results on all tasks are provided in the Appendix B.2.

Methods	ID						OOD					
	TriviaQA		HotpotQA		Avg		BioASQ		TextbookQA		Avg	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
SAC	**65.06**	**55.93**	**67.41**	**50.28**	**66.24**	**53.11**	**44.66**	31.45	**52.24**	**32.93**	**48.45**	**32.19**
SAC(w/o mask)	62.60	53.27	64.63	47.43	63.62	50.35	41.93	30.65	48.29	29.67	45.11	30.16
SAC(w/o anchor)	63.90	54.81	65.25	48.31	64.58	51.56	43.70	**31.78**	51.59	32.20	47.65	31.99

Token Selection. As shown in Table 4, our ablation study investigates the effect of different token selection strategies on the performance of SAC. The results indicate that random selection (*Random*) significantly degrades performance, not only because the selected tokens lack importance, but also due to their positional randomness, which leads to insufficient global coverage and fails to effectively represent the context. In contrast, both information-based selection (*Lingua-2*) and our default strategy achieve near-optimal results, and both substantially outperform existing baselines in Tables 1 and 2. This demonstrates that the SAC architecture can effectively leverage and enhance any high-quality token selection strategy, rather than relying on a specific choice, highlighting the generality and robustness of the SAC framework.

Table 4: Token selection ablation results. This table demonstrates how different token selection strategies affect model performance, comparing Random selection, Lingua-2-based selection (Pan et al., 2024b), and our uniform selection (Zhao et al., 2025). We report average F1/EM scores across in-domain (ID) and out-of-domain (OOD) tasks, with comprehensive results for all individual tasks presented in the Appendix B.2.

Methods	ID						OOD					
	TriviaQA		HotpotQA		Avg		BioASQ		TextbookQA		Avg	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
SAC	**65.06**	**55.93**	**67.41**	**50.28**	**66.24**	**53.11**	**44.66**	31.45	**52.24**	**32.93**	**48.45**	**32.19**
SAC(Random)	59.24	50.22	58.84	41.86	59.04	46.04	43.18	30.59	45.36	29.27	44.27	29.93
SAC(Lingua-2)	64.55	55.13	67.05	49.74	65.80	52.44	44.49	**31.91**	51.46	32.07	47.98	31.99

AE Effect. As shown in Table 5, we compare the effect of introducing an autoencoding (AE) objective during training on the performance of SAC. Traditional context compression methods employ AE tasks to force independent compression tokens to attend to the original context for reconstruction. However, we find that the AE objective itself has inherent limitations, as its reconstruction target is misaligned with downstream tasks. The experimental results validate this observation: training with only the AE objective leads to a substantial performance drop, and even when combined with the LM objective, the performance still lags behind the full SAC model. This highlights the architectural advantage of SAC: since anchor tokens are naturally semantically aligned with the original context, our method does not require AE objectives to force learning. Instead, SAC effectively aggregates contextual information into anchor token representations solely through anchor embeddings and bidirectional attention. It is worth noting that the ablation experiments in ICAE demonstrate that combining autoencoding tasks with language modeling tasks yields better results (Ge et al., 2024). However, our reproduction on 500xCompressor does not fully support this finding. Specifically, under 15x and 51x compression ratios, 500xCompressor achieves better in-distribution (ID) results when using language modeling tasks alone, with other scenarios being exceptions. This observation raises questions regarding the necessity of autoencoding tasks and suggests that autoencoding may not be entirely essential for context compression methods.

Table 5: Ablation study on the effects of autoencoding (AE) and language modeling (LM) objectives.

Methods	ID						OOD					
	5x		15x		51x		5x		15x		51x	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
SAC	**63.63**	**46.95**	**54.95**	**39.67**	**46.37**	**33.08**	**47.72**	**32.30**	**39.26**	**26.02**	**32.24**	**21.44**
500x(w/ LM only)	53.23	38.70	49.76	35.71	44.46	31.41	38.22	25.73	33.99	22.05	30.09	18.99
500x(w/ AE+LM)	55.26	40.14	49.35	34.98	43.19	30.26	38.46	25.40	34.04	22.09	30.43	20.09
SAC(w/ AE only)	56.55	40.34	49.93	35.33	43.95	30.64	42.08	27.98	35.50	23.29	28.77	18.32
SAC(w/ AE+LM)	62.04	45.80	51.73	36.67	44.69	31.37	47.26	32.25	37.23	23.96	31.01	19.90

5 ANALYSIS

5.1 ATTENTION VISUALIZATION

To understand the unique behavior of compressed models, we analyzed the attention patterns of the final layer at a 5x compression rate. Attention maps for other compression rates can be found in Appendix C.2.

As observed in Figure 4, The attention map for the 500xCompressor exhibits a distinct anti-diagonal trend. To complete the autoencoding task, the model must condense the entire original sequence into these remaining compressed tokens. This forces later compressed tokens to break locality constraints and actively seek out and attend to distant but important tokens in the sequence. In contrast, the SAC model demonstrates a clear diagonal pattern, where its anchor tokens effectively attend to their neighboring original context tokens, showing a better ability to focus on local information.

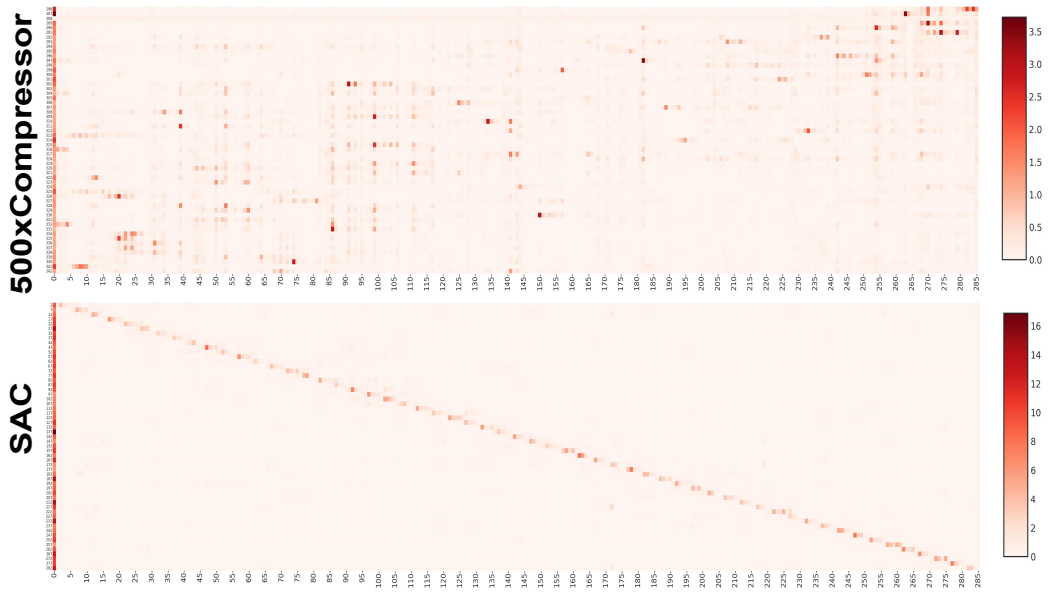

Figure 4: Attention maps of different models finetuned under a 5x compression rate. From top to bottom, the figure displays the final layer attention maps for the 500xCompressor and SAC models, respectively. The x-axis represents the original context tokens, and the y-axis represents the compression/anchor tokens.

5.2 REPRESENTATION ANALYSIS

Key Representation Analysis. In the Key representation space (see Figure 5), the compression tokens (orange) of SAC and EPL are distributed relatively close to the context tokens (blue), while the compression tokens of 500xCompressor are clearly separated from the context tokens. This discrepancy arises from the architectural design of each method. Specifically, although the positional

IDs of 500xCompressor's additional compression tokens are contiguous, their semantics are not aligned, leading to a complete separation in the Key space. In contrast, EPL modifies the positional IDs of its additional compression tokens to share the same rotational angle (RoPE) as the original context tokens, thereby reducing the distance between them. However, in SAC, the anchor tokens are directly embedded within the original context, and their representations maintain close semantic ties with the context tokens from the outset, naturally preventing significant representational divergence.

Value Representation Analysis. In the Value representation space (see Figure 5), the anchor tokens of SAC are uniformly distributed across all regions with the Value representations of the context tokens, without forming independent sub-clusters. This suggests that SAC's anchor embedding strategy allows for compressed Value representations that more closely match the distribution of the original Value space. In contrast, although EPL's compression tokens also overlap with the context tokens, their distribution is less complete than SAC's: they appear relatively sparse in the core regions and show a slight clustering tendency at the boundaries. This indicates that EPL's Value representations still exhibit a degree of semantic shift relative to the original Value space, which is even more pronounced in the 500xCompressor.

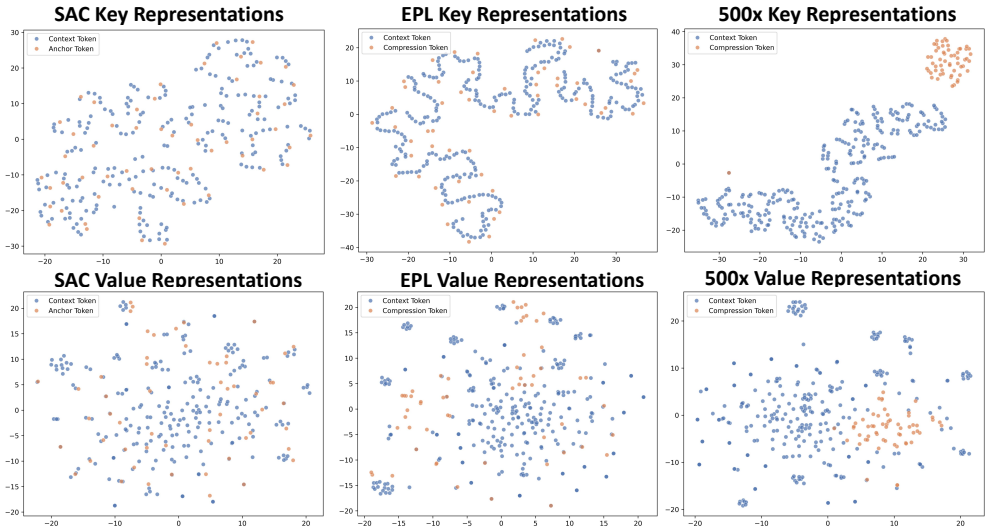

Figure 5: The t-SNE visualization shows the key representations of the final layer KV values for SAC, 500xCompressor (Li et al., 2025b), and EPL (Zhao et al., 2025), respectively.

6 Conclusion

This paper proposes a novel, autoencoding-free context compression method, **Semantic-Anchor Compression (SAC)**, designed to address the performance degradation in downstream tasks caused by context-agnostic compression tokens and autoencoding objectives in existing context compression methods. Unlike traditional context compression approaches, SAC does not rely on training compression tokens to reconstruct the original input. Instead, it directly selects representative *anchor* tokens from the context and aggregates contextual information into their key-value (KV) representations via a bidirectional attention mechanism. This approach effectively compresses lengthy contexts while avoiding any impairment to the language model's original language modeling capabilities. Experiments on multiple question answering tasks demonstrate that SAC achieves a high compression ratio and significantly outperforms existing compression methods, highlighting its superiority in balancing compression efficiency and model performance.

7 Reproducibility Statement

We declare that the work presented in this paper is reproducible. We provide a link to our anonymous source code as supplementary material: https://anonymous.4open.science/r/SAC-E32C. This code

can be used to reproduce the experimental results. The repository includes detailed instructions for environment setup, running experiments, data processing, and result evaluation.

REFERENCES

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders, 2024. URL https://arxiv.org/abs/2404.05961.

Kaiyan Chang, Songcheng Xu, Chenglong Wang, Yingfeng Luo, Xiaoqian Liu, Tong Xiao, and Jingbo Zhu. Efficient prompting methods for large language models: A survey, 2024. URL https://arxiv.org/abs/2404.01077.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.232. URL https://aclanthology.org/2023.emnlp-main.232/.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246/.

Shaohua Duan, Xinze Li, Zhenghao Liu, Xiaoyuan Yi, Yukun Yan, Shuo Wang, Yu Gu, Ge Yu, and Maosong Sun. Chunks as arms: Multi-armed bandit-guided sampling for long-context llm preference optimization. *arXiv preprint arXiv:2508.13993*, 2025.

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. URL http://arxiv.org/abs/1704.05179.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen (eds.), *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pp. 1–13, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-5801. URL https://aclanthology.org/D19-5801/.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP*, 2019b.

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, 2024. URL https://arxiv.org/abs/2307.06945.

Aaron Grattafiori and Abhimanyu Dubey. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Shengyue Guan, Haoyi Xiong, Jindong Wang, Jiang Bian, Bin Zhu, and Jian guang Lou. Evaluating llm-based agents for multi-turn conversations: A survey, 2025. URL https://arxiv.org/abs/2503.22458.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Pengcheng Huang, Zhenghao Liu, Yukun Yan, Xiaoyuan Yi, Hao Chen, Zhiyuan Liu, Maosong Sun, Tong Xiao, Ge Yu, and Chenyan Xiong. Pip-kag: Mitigating knowledge conflicts in knowledge-augmented generation via parametric pruning. *arXiv preprint arXiv:2502.15543*, 2025.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.825. URL https://aclanthology.org/2023.emnlp-main.825/.

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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.91. URL https://aclanthology.org/2024.acl-long.91/.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Aniruddha Kembhavi, Minjoon 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082/.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models, 2025. URL https://arxiv.org/abs/2405.17428.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In Roger Levy and Lucia Specia (eds.), *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1034. URL https://aclanthology.org/K17-1034/.

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 *Proceedings of the 34th*

International Conference on Neural Information Processing Systems, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. LooGLE: Can long-context language models understand long contexts? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16304–16333, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.859. URL https://aclanthology.org/2024.acl-long.859/.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6342–6353, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.391. URL https://aclanthology.org/2023.emnlp-main.391/.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7182–7195, Albuquerque, New Mexico, April 2025a. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.368. URL https://aclanthology.org/2025.naacl-long.368/.

Zongqian Li, Yixuan Su, and Nigel Collier. 500xCompressor: Generalized prompt compression for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25081–25091, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1219. URL https://aclanthology.org/2025.acl-long.1219/.

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. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024a. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

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. *arXiv preprint arXiv:2410.04727*, 2024b.

Nicolas El Maalouly. Exact matching: Algorithms and related problems, 2022. URL https://arxiv.org/abs/2203.13899.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=2DtxPCL3T5.

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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.57. URL https://aclanthology.org/2024.findings-acl.57/.

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, 2024b. URL https://arxiv.org/abs/2403.12968.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1683–1693, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1156. URL https://aclanthology.org/P18-1156/.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama, 2023. URL https://huggingface.co/datasets/cerebras/SlimPajama-627B.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. In Phil Blunsom, Antoine Bordes, Kyunghyun Cho, Shay Cohen, Chris Dyer, Edward Grefenstette, Karl Moritz Hermann, Laura Rimell, Jason Weston, and Scott Yih (eds.), *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 191–200, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2623. URL https://aclanthology.org/W17-2623/.

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, Eric 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 Bioinformatics*, 16(1):138, 2015. ISSN 1471-2105. doi: 10.1186/s12859-015-0564-6. URL https://doi.org/10.1186/s12859-015-0564-6.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems, 2020. URL https://arxiv.org/abs/1905.00537.

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5621–5634, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.412. URL https://aclanthology.org/2022.findings-emnlp.412/.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Zihao Yi, Jiarui Ouyang, Zhe Xu, Yuwen Liu, Tianhao Liao, Haohao Luo, and Ying Shen. A survey on recent advances in llm-based multi-turn dialogue systems, 2025. URL https://arxiv.org/abs/2402.18013.

Chen Zhang, Xinyi Dai, Yaxiong Wu, Qu Yang, Yasheng Wang, Ruiming Tang, and Yong Liu. A survey on multi-turn interaction capabilities of large language models, 2025. URL https://arxiv.org/abs/2501.09959.

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, 2025. URL https://arxiv.org/abs/2409.14364.

A Experiment Details

We perform pretraining and finetuning using bf16 precision on 8 NVIDIA RTX 3090 GPUs (24GB). For pretraining, we randomly sample data from the SlimPajama-6B dataset with a token length ranging from 510 to 2040. This data is then split into two halves: one for the auto-encoding (AE) task and the other for the language modeling (LM) task (the AE half is discarded for models without the AE objective). For downstream tasks, we process the MRQA dataset into a (Context, Question, Answer) format for finetuning. Detailed hyperparameters can be found in Table 6.

Table 6: Hyperparameters for training

Hyperparameter	**Value**
Optimizer	AdamW
Betas	(0.9, 0.95)
Weight decay	0.1
Learning rate	1e-4 (pretrain) 5e-5 (finetuning)
Scheduler	Constant
Batch size	16
Warmup	300
Training steps	20k (pretrain) 20k (finetuning)
Clip norm	2.0

B Detailed Results

B.1 Pretraining Results

As shown in Table 7, our method, SAC, achieves the lowest perplexity (10.79) among all baseline models. This suggests that removing the autoencoding (AE) objective in SAC allows the model to better focus on the language modeling task, thereby improving its predictive capability. Furthermore, since SAC avoids the additional computational overhead from independent compression tokens and the AE task, its training is approximately 31% faster than ICAE and 26% faster than 500xCompressor and EPL.

Table 7: Pretraining comparison of SAC and existing context compression methods, results on LM perplexity and training time.

Methods	**LM-PPL**	**Training Time(h)**
ICAE	12.35	3.85
500xCompress	11.83	3.60
EPL	10.88	3.60
SAC	**10.79**	**2.66**

B.2 Ablation Results

In the main text, we have discussed the significant performance gains of SAC over all baseline methods. To provide more detailed evidence, we present the full ablation study results here. As shown in Table 8 and Table 9, our conclusion holds not only in terms of average performance but is also consistently validated on each individual dataset.

Table 8: Ablation studies for SAC under a 5x compression rate on the in-domain dataset are conducted in three sets: component ablation, token selection, and the influence of the auto-encoding (AE) task.

Methods	SQuAD		NewsQA		TriviaQA		SearchQA		HotpotQA		NQ		Avg	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Component Ablation														
SAC	65.37	44.83	49.39	27.14	65.06	55.93	69.99	58.06	67.41	50.28	64.56	45.44	63.63	46.95
SAC(w/o mask)	60.21	39.93	45.93	25.74	62.60	53.27	66.66	54.82	64.63	47.43	61.53	42.55	60.26	43.96
SAC(w/o anchor)	61.69	41.72	46.52	25.45	63.90	54.81	68.03	56.17	65.25	48.31	62.21	43.88	61.27	45.06
Token Selection														
SAC(Random)	52.27	33.41	39.51	19.90	59.24	50.22	68.06	55.87	58.84	41.86	56.57	37.86	55.75	39.85
SAC(Lingua-2)	64.89	44.28	48.92	27.11	64.55	55.13	69.89	58.04	67.05	49.74	64.23	44.93	63.26	46.54
AE Effect														
500x(w/ LM only)	44.71	28.89	37.24	20.39	58.97	50.19	65.67	53.74	56.74	40.52	56.07	38.45	53.23	38.70
500x(w/ AE+LM)	51.62	33.63	39.70	22.63	57.62	48.76	66.43	54.38	59.10	42.20	57.11	39.26	55.26	40.14
SAC(w/ AE only)	56.98	37.60	41.09	20.61	58.19	49.08	64.02	51.65	61.58	44.13	57.23	38.98	56.55	40.34
SAC(w/ AE+LM)	64.68	44.62	46.64	25.62	63.34	54.27	68.40	56.48	66.61	49.72	62.56	44.06	62.04	45.80

Table 9: Ablation studies for SAC under a 5x compression rate on the out-of-domain dataset are conducted in three sets: component ablation, token selection, and the influence of the auto-encoding (AE) task.

Methods	BioASQ		DROP		DouRC		RACE		RE		TQA		Avg	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Component Ablation														
SAC	44.66	31.45	41.55	30.87	39.48	26.92	30.53	6.23	77.87	65.40	52.24	32.93	47.72	32.30
SAC(w/o mask)	41.93	30.65	40.24	28.48	36.48	23.58	28.21	5.49	69.09	55.63	48.29	29.67	44.04	28.92
SAC(w/o anchor)	43.70	31.78	40.55	30.34	36.97	25.58	30.05	6.82	75.88	62.35	51.59	32.20	46.46	31.51
Token Selection														
SAC(Random)	43.18	30.59	37.67	27.21	23.44	14.59	22.48	5.64	66.79	51.39	45.36	29.27	39.82	26.45
SAC(Lingua-2)	44.49	31.91	41.50	29.61	39.47	26.58	29.96	7.12	77.67	65.47	51.46	32.07	47.43	32.13
AE Effect														
500x(w/ LM only)	43.54	33.11	35.40	25.82	27.71	17.59	19.73	3.86	62.31	48.27	40.60	25.75	38.22	25.73
500x(w/ AE+LM)	40.30	28.99	35.40	25.55	29.43	19.32	21.57	4.90	65.43	50.88	38.62	22.75	38.46	25.40
SAC(w/ AE only)	40.85	29.39	35.32	25.28	31.55	21.32	25.86	4.90	72.29	57.90	46.61	29.08	42.08	27.98
SAC(w/ AE+LM)	44.84	32.31	41.47	31.14	39.29	27.58	30.11	6.23	77.12	64.42	50.74	31.87	47.26	32.26

C Visualization Analysis

C.1 Training Curves Analysis

Figure 6 shows the training loss curves at different compression ratios on the MRQA dataset. The training loss of our SAC model consistently converges better than other baseline methods across all compression ratios, which demonstrates that the compressed representations obtained from the SAC architecture are more beneficial for language modeling tasks. Notably, as the compression ratio increases appropriately, the difference in convergence between SAC and the other baselines becomes more significant.

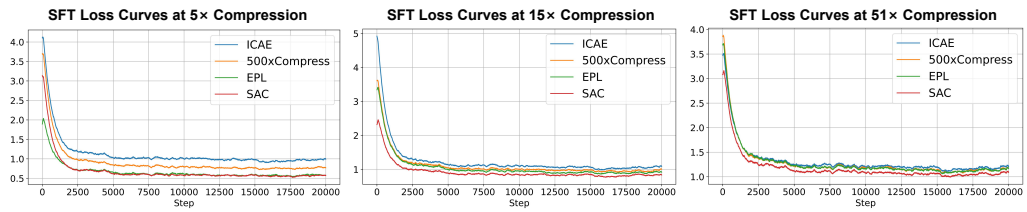

Figure 6: Supervised finetuning loss curves. The figure illustrates the training loss trajectories of different models under three compression ratios: 5x, 15x, and 51x.

C.2 Attention Analysis

At a lower 5x compression rate, as shown in Figure 7, the attention map of EPL presents a clear positive diagonal, indicating that its compressed tokens primarily attend to local tokens. In contrast, the attention map of 500xCompressor appears more diffused, while our SAC model exhibits a sparse and highly focused attention pattern, with its anchor tokens attending to only a few key original context tokens. This phenomenon becomes more pronounced with increasing compression rates, being most evident at the 51x compression rate in Figure 9, which strongly demonstrates the robustness of SAC in extreme compression environments.

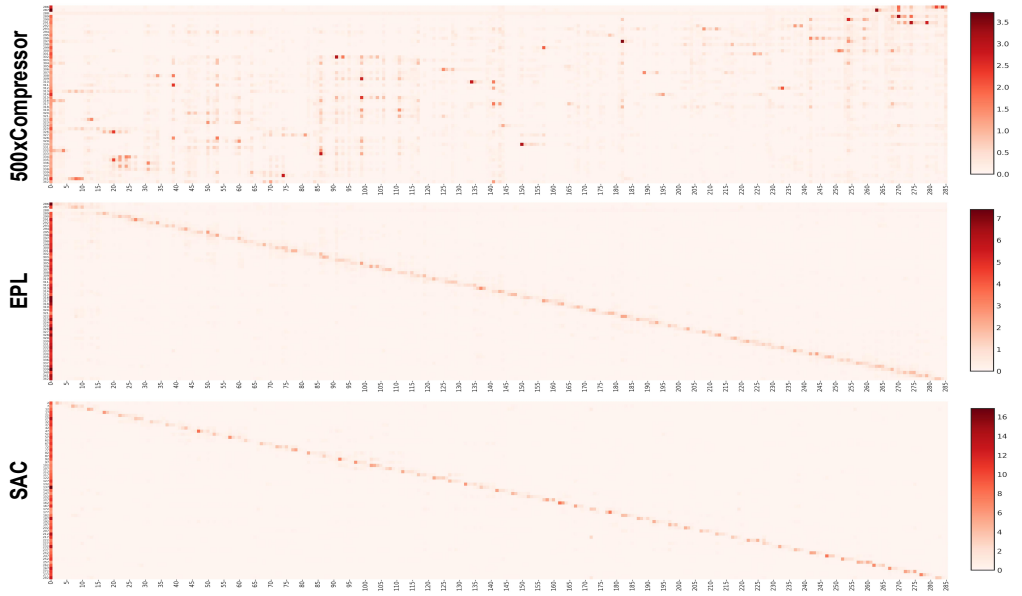

Figure 7: Attention maps of different models finetuned under a 5x compression rate. From top to bottom, the figure displays the final layer attention maps for the 500xCompressor, EPL, and SAC models, respectively. The x-axis represents the original context tokens, and the y-axis represents the compression tokens.

D The Use of Large Language Models

We used a large language model (LLM) as a general-purpose assist tool. The LLM's primary role was in assisting with writing and text editing, such as refining prose and correcting grammar and spelling to ensure the paper's professionalism and fluency. We explicitly state that the LLM was not involved in the core ideation or methodological design of this research. All core contributions of the paper, including the proposal of the methodology, the construction and execution of experiments, and the analysis of results, were performed independently by the authors.

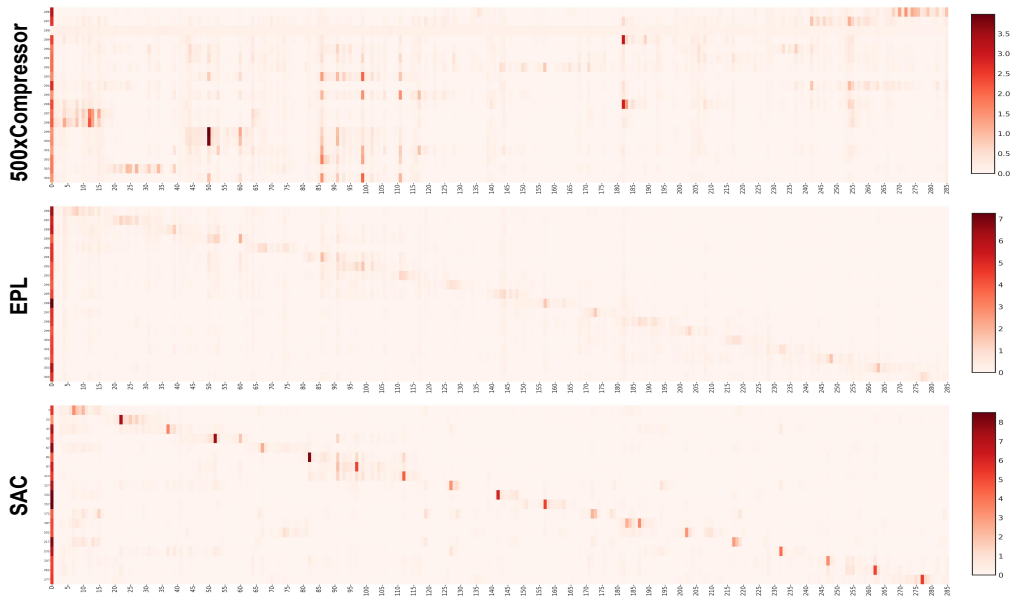

Figure 8: Attention maps of different models finetuned under a 15x compression rate. From top to bottom, the figure displays the final layer attention maps for the 500xCompressor, EPL, and SAC models, respectively. The x-axis represents the original context tokens, and the y-axis represents the compression/anchor tokens.

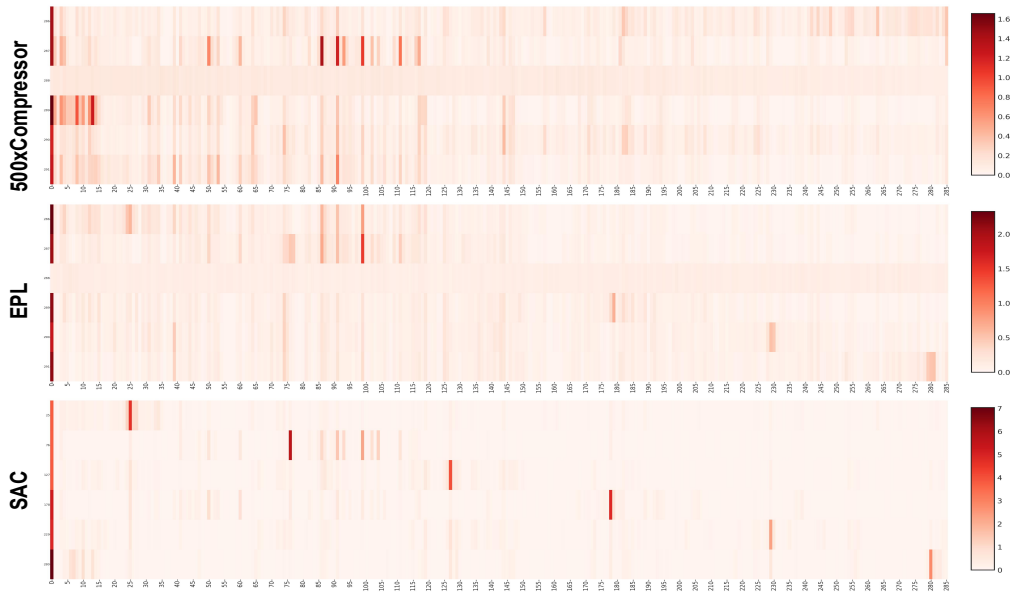

Figure 9: Attention maps of different models finetuned under a 51x compression rate. From top to bottom, the figure displays the final layer attention maps for the 500xCompressor, EPL, and SAC models, respectively. The x-axis represents the original context tokens, and the y-axis represents the compression tokens.

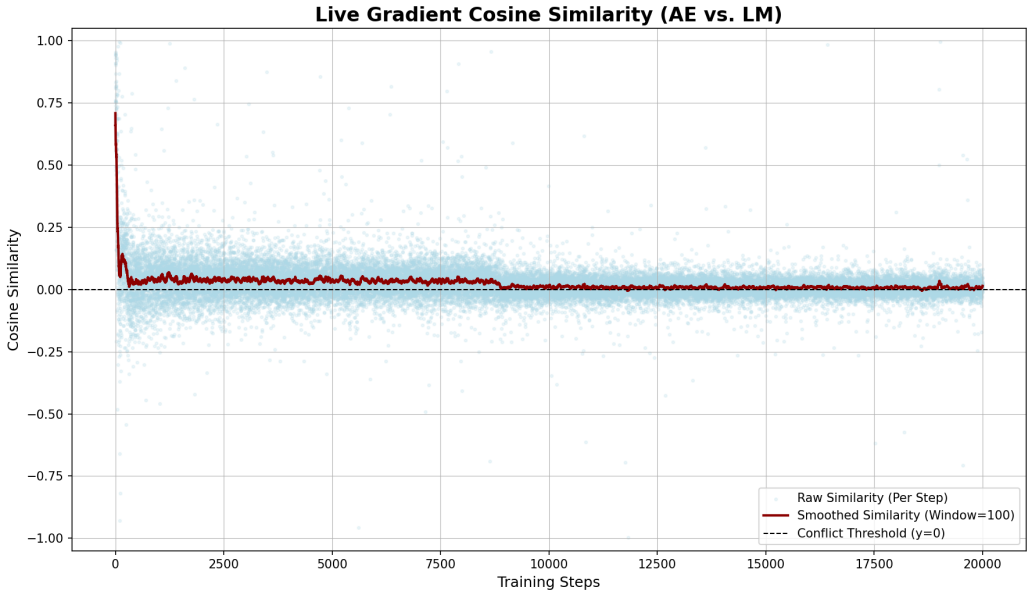

Figure 10: Gradient Cosine Similarity between AutoEncoder (AE) Loss and Language Modeling (LM) Loss.

Figure 11: Efficiency and Performance Trade-off Curves on In-Domain (ID) Tasks.