In-Context Former: Lightning-fast Compressing Context for Large Language Model

Anonymous ACL submission

Abstract

With the rising popularity of Transformer-based large language models (LLMs), reducing their high inference costs has become a significant research focus. One effective approach is to compress the long input contexts. Existing methods typically leverage the self-attention mechanism of the LLM itself for context compression. While these methods have achieved notable results, the compression process still involves quadratic time complexity, which limits their applicability. To mitigate this limitation, we propose the In-Context Former (IC-Former). Unlike previous methods, IC-Former does not depend on the target LLMs. Instead, it leverages the cross-attention mechanism and a small number of learnable digest tokens to directly condense information from the contextual word embeddings. This approach significantly reduces inference time, which achieves linear growth in time complexity within the compression range. Experimental results indicate that our method requires only 1/32 of the floating-point operations of the baseline during compression and improves processing speed by 68 to 112 times while achieving over 90% of the baseline performance on evaluation metrics. Overall, our model effectively reduces compression costs and makes real-time compression scenarios feasible.

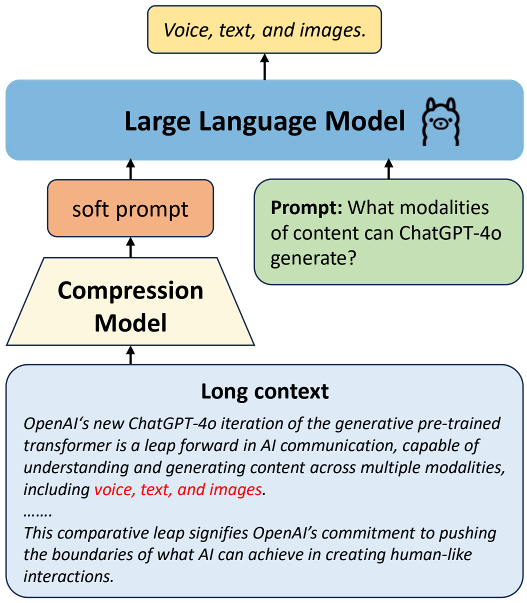

Figure 1: Compressing long contexts into short soft prompts to improve inference efficiency.

1 Introduction

In recent years, transformer-based (Vaswani et al., 2017) language models especially large language models (LLMs) have made significant strides in the field of natural language processing, demonstrating exceptional performance across a wide range of tasks. However, the self-attention mechanism in LLMs leads to high inference costs. Previous work (Child et al., 2019; Beltagy et al., 2020; Bulatov et al., 2022; Zheng et al., 2022; Wu et al., 2022; Ding et al., 2023; Dai et al., 2019; Choromanski et al., 2020; Borgeaud et al., 2022) has explored various approaches to reduce computational complexity by improving the self-attention mechanism of language models. Although these strategies mitigate the overhead of long context processing, they inevitably introduce modifications to the original structure of LLMs, potentially impacting the capabilities of the original model (Liu et al., 2024).

To better avoid modifications to the LLM structure, a more intuitive approach is to introduce a preliminary context compression process. These methods are based on a core assumption: most natural language texts contain redundant information, which makes context compression feasible. In early exploration, Mu et al. (2024) have attempted to compress the instructions into short soft prompts. This method offers a novel perspective but still has limitations in long context compression. Later works (Chevalier et al., 2023; Ge et al., 2024) aim to further extend compression abilities for document-level long contexts, and achieved considerable results. As illustrated in Figure 1, these methods design compression models to condense lengthy contexts into short, context-rich soft prompts, which then serve as substitutes for the original context when input into the LLM. However, these methods still suffer the issue of expensive time costs during the compression process. This limitation restricts their application in real-time compression scenarios, such as compressing retrieved (Guu et al., 2020) or real-time Internet documents (Asai et al., 2023) immediately.

By reviewing previous works on compressors, we find that existing methods typically utilize the LLM as the encoder. While these methods fully utilize the powerful semantic understanding capabilities of LLM, they also suffer from rapidly increasing quadratic time complexity as the context lengthens. So is there a way to significantly reduce the theoretical complexity of compressors, with an acceptable decrease in performance?

Driven by this motivation, we design an efficient context compression model, the In-Context Former (IC-Former), which aims at optimizing resource consumption during the compression of long context in existing models. This model is based on two assumptions regarding semantic content compression: (1) Word embeddings already contain sufficient semantic information (Mikolov et al., 2013; Tache et al., 2021), suggesting that additional interactions may not be necessary prior to the extraction process. (2) Learnable tokens within an elaborate structure can effectively aggregate information to a certain extent (Chevalier et al., 2023; Ge et al., 2024). Based on these assumptions, we creatively discard the costly self-attention interaction of text content in previous models. Instead, we leverage the efficiency of the cross-attention mechanism for information extraction. This innovative strategy ensures that the computational overhead of compression grows linearly with the context length within the compression range, significantly enhancing compression efficiency compared to the previous methods.

Specifically, our IC-Former consists of a few cross-attention blocks and some learnable digest tokens. Through this structure, the IC-Former leverages the digest tokens to extract information from lengthy contextual content and refine it into compact digest vectors. Subsequently, these digest vectors directly replace the original, verbose context and serve as input to LLMs while ensuring that the generated texts are faithful to the original context. In the training phase, to effectively compress context, we follow the previous training paradigm (Ge et al., 2024), employing a strategy that combines pre-training and fine-tuning to optimize the IC-Former. During the pre-training phase, the IC-Former engages in a context reconstruction task. It generates digest vectors from which an LLM can reconstruct the original context. In the fine-tuning phase, we train the IC-Former on instruction data to ensure the generated digest vectors correctly respond to various context-related prompts.

Additionally, through theoretical calculations, we demonstrate that at a compression ratio of 4x, our IC-Former achieves only 1/32 of the floating-point operations required by the baseline. Experimental results further show that our method achieves a compression speed that is 68 to 112 times faster than the baseline while maintaining over 90% of the baseline performance on evaluation metrics. This indicates a higher cost-effectiveness.

Overall, our contributions can be summarized in the following three points:

- We propose the In-Context Former (IC-Former), a novel context compression model that can compress context to a quarter of its original length as a soft prompt while preserving most of original contextual information.

- The IC-Former is lightweight and efficient, with a parameter size that is 9% of the target LLM. It achieves compression speeds 68 to 112 times faster than the baseline while maintaining more than 90% of the baseline performance.

- We analyze the interaction between the IC-Former and the context, enhancing the interpretability of the IC-Former's compression process.

2 Related Work

Soft prompt compression Wingate et al. (2022) propose to learn a compact soft prompt (Lester et al., 2021) to represent the original natural language prompt. They align the model predictions that are based on the original prompt and those conditioned on the soft prompt by optimizing KL divergence (Hershey and Olsen, 2007). As a result, Wingate et al. (2022) discover that the trained soft prompt retain high-level semantic information and can be utilized to control generation. However, this approach suffers high computational costs as

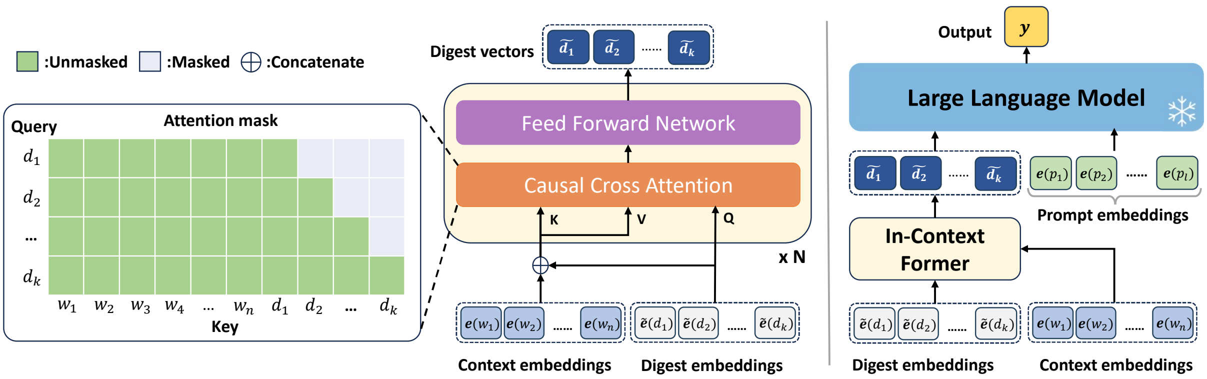

Figure 2: **Left:** Model architecture of In-Context Former. In-Context Former utilizes a set of learnable digest embeddings to condense the information of context and generates digest vectors. And we apply causal attention masks for digest tokens. **Right:** Overview of In-Context Former's framework.

it requires retraining a new soft prompt for each new context. In contrast, our method can predict the soft prompt corresponding to the input context.

Context distillation Another related work (Snell et al., 2022; Askell et al., 2021) focuses on distilling the contextual information such as instruction into a student model without prompting. Mu et al. (2024) propose GIST to compress prompts into gist tokens, which can be viewed as key-value attention prefixes. Nonetheless, this approach did not address the long context issue as it is limited to compressing short prompts. In addition, this method requires updating the parameters of language model, which differs from our method. Our method keeps the language model fixed and therefore preserves its capability.

Context compression Chevalier et al. (2023) propose AutoCompressors to compress long text into summary vectors recursively. However, the compression procedure is sophisticated and LLMs are still required to be fine-tuned to generate summary vectors. ICAE (Ge et al., 2024) is the most closely related study to our research. ICAE compresses context into short memory slots, with a small number of additional parameters by the LoRA (Hu et al., 2021) approach with a fixed LLM. However, both AutoCompressors and ICAE employ self-attention to integrate contextual information, resulting in a quadratic complexity with respect to the length of context. Instead, our model does not incorporate contextual interactions and reduces both time and space complexities, striking a balance between efficiency and performance.

3 Method

3.1 Task Formulation

Context compression aims to transform lengthy contexts into brief, compact representations while endeavoring to preserve the fundamental semantics and integrity of the original contexts.

Formally, we define the original context that is to be compressed as $w = (w_1, w_2, ..., w_n)$, where w_i represents the i-th token of context and n is the number of tokens in context. Then, we denote $\boldsymbol{e}(\cdot)$ as the word embedding lookup in the LLM and $\tilde{\boldsymbol{e}}(\cdot)$ as the learnable embeddings of soft tokens. A context compressor model Θ utilizes the embeddings of soft tokens $\widetilde{\boldsymbol{e}}(d) = (\tilde{\boldsymbol{e}}(d_1), \tilde{\boldsymbol{e}}(d_2), ..., \tilde{\boldsymbol{e}}(d_k))$ and context embeddings $\boldsymbol{e}(w) = (\boldsymbol{e}(w_1), \boldsymbol{e}(w_2), ..., \boldsymbol{e}(w_n))$ to generate compact representations $\widetilde{\boldsymbol{d}} = (\widetilde{\boldsymbol{d}_1}, \widetilde{\boldsymbol{d}_2}, ..., \widetilde{\boldsymbol{d}_k})$ of context, where k is the length of compressed context and $k \ll n$.

The condensed vectors $\widetilde{\boldsymbol{d}}$ can substitute the original context and be combined with other prompt $\boldsymbol{e}(p) = (\boldsymbol{e}(p_1), ..., \boldsymbol{e}(p_l))$ for input to an LLM Φ. The output $y = (y_1, ..., y_m)$ remains faithful to the content of the original context w.

3.2 In-Context Former

As illustrated in Figure 2, IC-Former consists of a few cross-attention layers and a set of learnable soft tokens, which are named digest tokens. The IC-Former utilizes context tokens and digest tokens as inputs, leveraging a causal cross-attention mechanism to condense the context information into digest vectors. Subsequent sections will detail the attention computation process, attention masks, and positional embeddings.

Attention computation When compressing a long context, the context tokens are concatenated with digest tokens and subsequently mapped into embeddings, which serve as key and value in the cross-attention layer. Meanwhile, the embeddings of digest tokens serve as query to interact with both

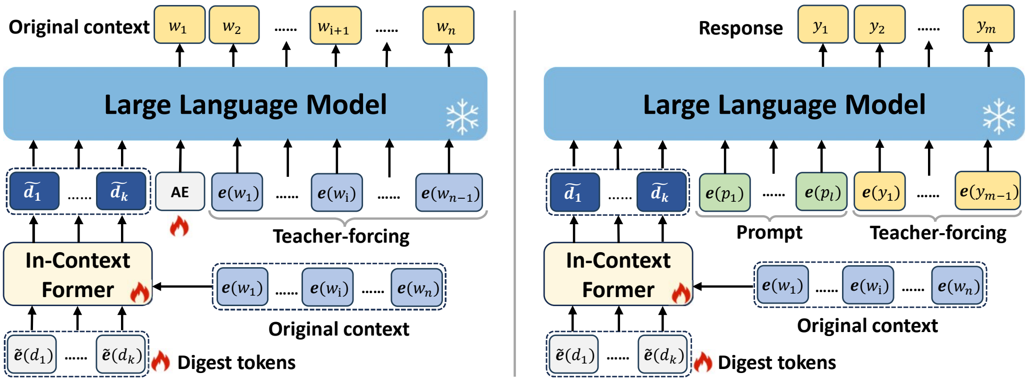

Figure 3: **Left:** Pretraining stage. IC-Former learns to generate digest vectors such that, when these vectors and a special token AE are jointly fed into an LLM, the LLM reproduces the original context. **Right:** Instruction fine-tuning stage. Training IC-Former to generate digest vectors capable of correctly responding to prompts.

context embeddings and digest embeddings. To be specific, the Q, K and V in IC-Former can be computed as:

$$Q = W_Q \, \tilde{\mathbf{e}}(d)^T \tag{1}$$

$$K = W_K \, [\mathbf{e}(w); \tilde{\mathbf{e}}(d)]^T \tag{2}$$

$$V = W_V \, [\mathbf{e}(w); \tilde{\mathbf{e}}(d)]^T \tag{3}$$

Then we employ the cross-attention mechanism to condense contextual information, as this approach has been empirically validated effective. (Li et al., 2023; Ye et al., 2023; Zhu et al., 2023).

Attention masks As depicted in Figure 2, our design for attention masks allows digest tokens to attend to all context tokens as well as preceding digest tokens, thereby mitigating the deficiency of interaction among context tokens.

Additionally, it can be observed from the attention matrix that given a context length of n and a target compression length of k, the time complexity and space complexity of our method are both $\mathcal{O}(kn + k^2) \sim \mathcal{O}(kn)$. This indicates that the complexity of this model grows linearly with the increase of context.

Positional embeddings We recognize that the pure cross-attention mechanism does not capture the relative positional relationships among tokens within the context. This implies swapping any two tokens in the context results in an identical digest vector, which does not align with our expectations. To address this, we applied RoPE (Su et al., 2024) to represent the relative positional relations within the context tokens.

We denote the positional embeddings of the nth token in the sequence as $\mathrm{RoPE}(n)$ and is abbreviated as R_n.

$$\mathrm{RoPE}(n) = \begin{bmatrix} R_n^{(0)} & & & \\ & R_n^{(1)} & & \\ & & \ddots & \\ & & & R_n^{(\frac{h}{2}-1)} \end{bmatrix},$$

$$\text{where } R_n^{(i)} = \begin{bmatrix} \cos(n\theta^i) & -\sin(n\theta^i) \\ \sin(n\theta^i) & \cos(n\theta^i) \end{bmatrix} \tag{4}$$

In the Eq.4, $\theta = \theta_{base}^{-\frac{2}{h}}$ where θ_{base} is a hyperparameter and h is the hidden size and assumed to be even. We restate Eq.1 & 2 as follows:

$$Q = (\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_k) \tag{5}$$

$$K = (\boldsymbol{k}_1, ..., \boldsymbol{k}_n, \boldsymbol{k}_{n+1}, ..., \boldsymbol{k}_{n+k}) \tag{6}$$

We allocate positional embeddings as if placing the digest tokens subsequent to the context tokens as demonstrated in Eq.7 & 8.

$$Q_{\mathrm{RoPE}} = (\mathrm{R}_{n+1}\boldsymbol{q}_1, \mathrm{R}_{n+2}\boldsymbol{q}_2, ..., \mathrm{R}_{n+k}\boldsymbol{q}_k) \tag{7}$$

$$K_{\mathrm{RoPE}} = (\mathrm{R}_1\boldsymbol{k}_1, ..., \mathrm{R}_n\boldsymbol{k}_n, ..., \mathrm{R}_{n+k}\boldsymbol{k}_{n+k}) \tag{8}$$

The RoPE manifests the relative positional relationships through the inner product between Q_{RoPE} and K_{RoPE}:

$$(\mathrm{R}_i\boldsymbol{q})^T(\mathrm{R}_j\boldsymbol{k}) = \boldsymbol{q}^T\mathrm{R}_i^T\mathrm{R}_j\boldsymbol{k} = \boldsymbol{q}^T\mathrm{R}_{j-i}\boldsymbol{k} \tag{9}$$

In this manner, each digest token is capable of perceiving the relative positions of both context tokens and other digest tokens.

3.3 Training process

This section introduces the training objectives of IC-Former, including pretraining and instruction fine-tuning, and a divide-and-conquer training strategy when dealing with too long contexts.

Pretraining Previous works (Rumelhart et al., 1986; Kramer, 1991; Van Den Oord et al., 2017; Ge et al., 2024) have demonstrated that autoencoding tasks can benefit models to effectively condense and encode information. We adopt this approach to pretrain our IC-Former by using a text reconstruction task. The objective of this task is to leverage digest vectors, which are extracted from compressed contexts, to reconstruct the original contexts. As illustrated in Figure 3, the context tokens are compressed into digest vectors by IC-Former and then serve as input to LLM with a special token "[AE]" to indicate the autoencoding task.

To make LLM reconstruct the original context w conditioned on the digest vectors $\widetilde{\boldsymbol{d}}$, we optimize IC-Former Θ and digest embeddings $\widetilde{\boldsymbol{e}}(d)$ by minimizing negative log-likelihood of context w. The pretraining objective can be written as:

$$\begin{aligned}\mathcal{L}_{\mathrm{AE}} &= -\log p\left(w|\widetilde{\boldsymbol{d}_1}, ..., \widetilde{\boldsymbol{d}_k}; \Phi\right) \\ &= -\log p\left(w|d_1, ..., d_k; \widetilde{\boldsymbol{e}}; \Theta; \Phi\right) \end{aligned} \tag{10}$$

This reconstruction task forces IC-Former to focus on each token in context, thereby preserving all context information. The analysis on pretraining in Section 4.3 demonstrates that this task can help IC-Former learn to aggregate contextual information.

Instruction fine-tuning After the pretraining phase, IC-Former has effectively learned to meticulously attend to context. However, to ensure that the compressed digest vectors appropriately respond to various prompts, further instruction fine-tuning (Zhang et al., 2023) of IC-Former is necessary. As shown in Figure 3, we input the digest vectors generated from IC-Former along with the prompt embeddings into the LLM. Similarly, by optimizing IC-Former Θ and digest embeddings $\widetilde{\boldsymbol{e}}(d)$, we minimize the negative log-likelihood of the expected output y:

$$\begin{aligned}\mathcal{L}_{\mathrm{FT}} &= -\log p(y|\widetilde{\boldsymbol{d}_1}, ..., \widetilde{\boldsymbol{d}_k}; p_1, ..., p_l; \Theta; \Phi) \\ &= -\log p(y|d_1, ..., d_k; p_1, ..., p_l; \widetilde{\boldsymbol{e}}; \Theta; \Phi) \end{aligned} \tag{11}$$

Divide and conquer When the context length exceeds the compression limit, a divide-and-conquer strategy (Bertsch et al., 2024; Song et al., 2024; Chen et al., 2023) proves to be effective. We first uniformly split the context into several chunks of acceptable length. Each of these chunks is then compressed individually to obtain local vectors. As illustrated in Figure 4, we subsequently concatenate all these local vectors to form the global vectors. This strategy is applied in both the training and inference phases.

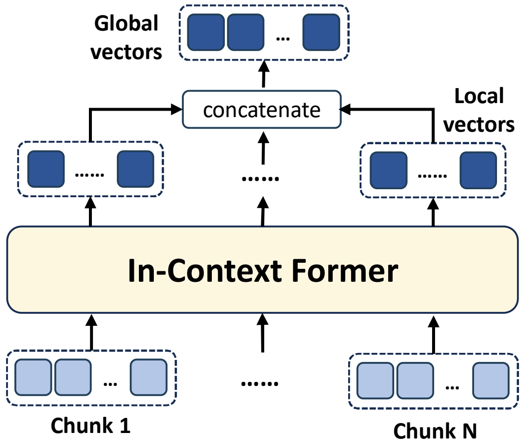

Figure 4: The excessively long contexts are broken into chunks, which are then compressed and concatenated.

4 Experiments

4.1 Experimental setting

This section introduces the experimental setting including data, baseline, and model configuration.

Data Due to resource constraints, we pretrain IC-Former using a subset of the Pile (Gao et al., 2020) dataset, comprising approximately 2.29 million text entries. In the fine-tuning phase, we employed the PwC (Prompt-with-Context) dataset (Ge et al., 2024), which includes contexts accompanied by corresponding questions. This dataset is suitable for evaluating the compressor's ability to preserve contextual information. For each context, the dataset provides ten specific and five general questions. For evaluation convenience, we select the ten specific questions to evaluate as their answers are relatively more definitive.

Baseline We select ICAE as our baseline for comparison, because the motivations behind other related works are distinct from ours. For instance, AutoCompressors fine-tune LLMs and focus on stability in long-context modeling rather than on restoring details in compressed text. Likewise, GIST also modifies model parameters, and its strength lies in compressing instruction information rather than long context. We replicate ICAE on this dataset.

Model configuration We use Llama2-7b-chat (Touvron et al., 2023) as the target LLM for evaluation. Both attention and feed-forward network modules of IC-Former have the same hidden size as Llama2-7b-chat. The default number of digest tokens k is set to 128 unless otherwise specified. Furthermore, IC-Former consists of only three trans-

Input (Batchsize×Length)	Method	Memory (GB)	Compression Time (s)	Inference Time (s)	Total Time (s)
8×2048	LLM	35.96	-	1.845	1.845
	LLM+ICAE	19.76	3.268	0.314	3.582
	LLM+IC-Former	**15.96 / 2.38**	**0.029 (112×)**	0.314	**0.343 (5.3×)**
8×512	LLM	17.46	-	0.318	0.318
	LLM+ICAE	19.76	0.476	0.079	0.555
	LLM+IC-Former	**15.82 / 2.28**	**0.007 (68×)**	0.079	**0.086 (3.7×)**
32×512	LLM	29.07	-	1.186	1.186
	LLM+ICAE	38.74	1.848	0.288	2.136
	LLM+IC-Former	**18.98 / 3.52**	**0.017 (108×)**	0.289	**0.306 (3.8×)**

Table 1: Compression and inference overhead. Inference time refers to the period required to perform a forward pass, utilizing either original context embeddings or compressed vectors as input to the LLM. Memory denotes the peak GPU memory usage during the compression and inference processes. Additionally, we quantify the memory utilization when employing IC-Former for compression independently (right of the /).

Method	Time&Space Complexity	Theoretical FLOPs
ICAE	$\mathcal{O}(n^2 + 2kn)$	8.50×10^{12}
IC-Former	$\mathcal{O}(kn)$	$2.62 \times 10^{11} (\sim \frac{1}{32})$

Table 2: Complexity analysis. The theoretical FLOPs represent the computational cost incurred when compressing a context of length 512 into 128 vectors for the Llama2-7b-chat model. For further details, see the Appendix C.

Length	BLEU-4		Loss	
	ICAE	IC-Former	ICAE	IC-Former
100	**0.9967**	0.9965	**0.1461**	0.1789
200	0.9969	**0.9972**	0.0971	**0.0851**
300	**0.9974**	0.9971	0.0602	**0.0558**
400	0.9889	**0.9892**	0.0499	**0.0483**
500	0.9654	**0.9689**	0.1116	**0.1078**

Table 3: Results of BLEU-4 scores and cross-entropy loss between reconstructed context and original context across different context lengths.

former layers and includes approximately 607M parameters, encompassing the digest embeddings.

4.2 Experiment Results

4.2.1 Compression & Inference Efficiency

Firstly we analyze the theoretical time-space complexity of the IC-Former and baseline method and the floating point operations (FLOPs) required to compress 512 tokens to a length of 128. As illustrated in Table 2, our approach significantly reduces both the temporal and spatial overhead compared to the baseline. In experiments involving compression of contexts with a length of 512, the required FLOPs are merely 1/32 of those needed by the baseline method.

We further assess and compare the compression time and memory utilization of IC-Former during actual compression processes with the baseline model. Experimental results indicate that our IC-Former significantly outperforms existing methods in terms of both temporal efficiency and spatial occupancy.

As shown in Table 1, our IC-Former has the lowest memory usage during compression among the compared models. Additionally, IC-Former's compression process does not depend on the target LLM, enabling it to perform compression independently and achieve over 88% memory savings relative to the baseline. In terms of compression time, our method is 68 to 112 times faster than the baseline, rendering the compression overhead negligible compared to the inference time of the target LLM. In scenarios where compression is followed by inference, our method achieves approximately four times faster processing than directly inferring using the original context, whereas the baseline method consumes even more time. Our approach thus offers a viable solution for real-time compression scenarios.

4.2.2 Pretraining: Context Reconstruction

We evaluate the pretraining performance of IC-Former, focusing on its ability to reconstruct the original context. To measure the discrepancies between the reconstructed text and the original, we utilize BLEU (Papineni et al., 2002) and cross-entropy loss as metrics.

As shown in Table 3, the reconstructed context

Input content	ROUGE-1			ROUGE-2			ROUGE-L		
	P	R	F1	P	R	F1	P	R	F1
512 original context tokens	0.456	**0.635**	0.501	0.300	**0.438**	0.331	0.426	**0.594**	0.468
128 memory slots (ICAE)	**0.592**	0.561	**0.555**	**0.404**	0.385	**0.377**	**0.553**	0.525	**0.519**
128 digest vectors (IC-Former)	0.554	0.520	0.516	0.374	0.355	0.348	0.517	0.487	0.482
(performance ratio)	93.6%	92.7%	93.0%	92.6%	92.2%	92.3%	93.5%	92.8%	92.9%
64 digest vectors	0.384	0.412	0.377	0.211	0.234	0.209	0.349	0.375	0.343
64+64 digest vectors	0.545	0.498	0.500	0.358	0.330	0.327	0.507	0.464	0.465
128 digest vectors	**0.554**	**0.520**	**0.516**	**0.374**	**0.355**	**0.348**	**0.517**	**0.487**	**0.482**
128 digest vectors (w/o pretrain)	0.431	0.381	0.389	0.234	0.211	0.212	0.393	0.349	0.355

Table 4: Evaluation results on PwC test set. The first row of the table compares the performance of our method with other baseline models, and the performance ratio means the ratio of our IC-Former to the ICAE. The second row demonstrates the performance variations when different compression strategies are implemented, where "64+64" represents a divide-and-conquer approach. The third row reveals the impact of ablation pre-training on performance.

Text type	BLEU	Loss
Normal text	0.9006	0.125
Reversed text	0.6652	1.803
Patterned random text	0.1347	4.401
Completely random text	0.0080	8.137

Table 5: Reconstruction results for texts with varying degrees of randomness, with randomness increasing from top to bottom. The patterned text is generated by adding 1 to each token_id of normal text. All texts above are compressed from length of 512 to 128.

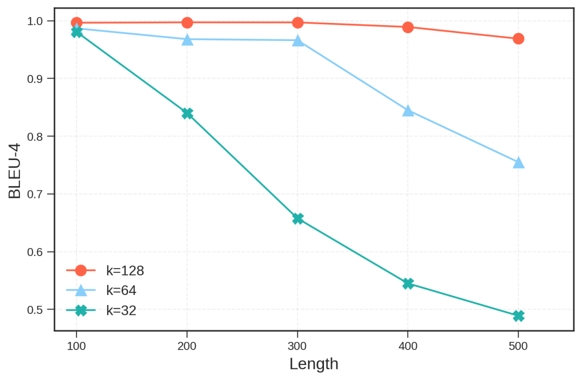

Figure 5: BLEU-4 for different digest token lengths k.

by IC-Former exhibits minimal discrepancies when compared to the original context. For a context length of less than 400, the BLEU-4 score reaches 0.99, and the cross-entropy loss hovers around 0.05. When the context length is extended to 500, the BLEU score maintains a high value of 0.96, and the cross-entropy loss is approximately 0.1. These results suggest that IC-Former effectively captures the contextual information, achieving a 4x compression ratio while maintaining performance comparable to the baseline.

Then we explore the impact of digest tokens length k on the reconstruction task. As shown in Figure 5, it is not surprising that the quality of the reconstructed text deteriorates as k decreases.

Additionally, we attempt to use IC-Former to compress texts with various levels of randomness and analyze the reconstruction results. As observed from Table 5, the reconstruction performance of IC-Former progressively declines as the randomness of the text increases. This phenomenon may suggest that IC-Former primarily achieves information compression through semantic understanding rather than mere rote memorization. Further analysis is conducted in Section 4.3.

4.2.3 Performance on Downstream Task

In this section, we evaluate the model's performance on the PwC dataset. Although our model can achieve good results based on the BLEU metric, considering that BLEU is more susceptible to response length, we ultimately choose the ROUGE metric (Lin, 2004) to evaluate the performance of our model, which more faithfully reflects the original content of the text. We compare the performance of various context compression models by keeping the target LLM frozen and substituting the context with different vectors.

As illustrated in the first row of Table 4, our method achieves over 92% of the baseline performance while significantly surpassing the baseline model in terms of compression speed. The second row of the table compares the performance of digest vectors of varying lengths, including the compression of 512 context tokens into 64 digest vectors and their subsequent division and compression into two sets of 64 digest vectors each, as discussed in Section 3.3 under the strategy of divide-and-conquer. It can be observed that compared to directly compressing 512 context tokens into 128 di-

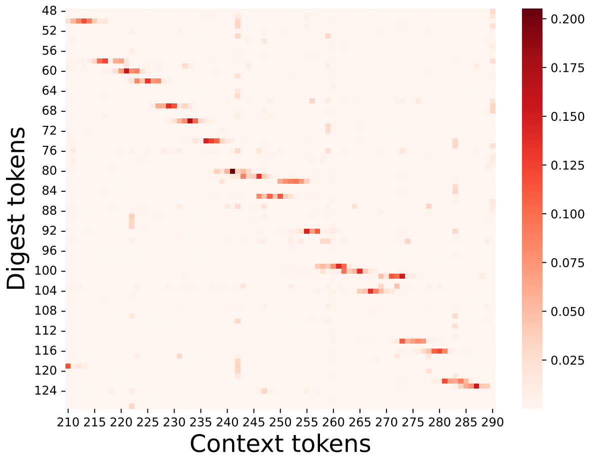

Figure 6: Local attention map in the last layer of IC-Former. The horizontal axis represents context tokens acting as key and the vertical axis represents digest tokens acting as query. For complete attention map, see Appendix E.

An example of context

A large language model (LLM) is a computational model notable for its ability to achieve general-purpose language generation and other natural language processing tasks such as classification. Based on language models, LLMs acquire these abilities by learning statistical relationships from vast amounts of text during a computationally intensive self-supervised and semi-supervised training process. LLMs can be used for text generation, a form of generative AI, by taking an input text and repeatedly predicting the next token or word. LLMs are artificial neural networks that utilize the transformer architecture, invented in 2017. The largest and most capable LLMs, as of June 2024, are built with a decoder-only transformer-based architecture, which enables efficient processing and generation of large-scale text data. Larger models such as GPT-3 have demonstrated the ability to achieve similar results through prompt engineering, which involves crafting specific input prompts to guide the model's responses.

Table 6: The context tokens that are most attended to by digest tokens across layers. The color of each token is determined by the layer when it is **initially** attended to. Red, cyan, and blue denote the first, second and third layer respectively. Gray indicates tokens that are never attended to.

gest vectors, the approach of divide-and-conquer results in a slight performance degradation. However, this performance loss is acceptable when compared to the costs associated with retraining a model to accommodate longer digest embeddings. Additionally, we utilize an ablation study to demonstrate the efficacy of pretraining. IC-Former without pretraining performs poorly in capturing contextual information and is more prone to generating hallucinations. (See examples in Appendix D).

4.3 Analysis

To better understand the working principles of IC-Former, we conducted further visualization analysis based on the attention map.

Neighbourhood information aggregation We average the attention scores of all attention heads in the third layer (last layer) of the IC-Former to obtain an attention map. It can be observed from Figure 6 that each digest token attends to 3 to 5 consecutive context tokens, and digest tokens focus on the context tokens in accordance with their sequential order, which presents a backslash shape pattern. It is worth mentioning that the non-pretrained IC-Former does not exhibit this phenomenon (See examples in Appendix E). These phenomena indicate that IC-Former compresses context by aggregating information from adjacent tokens and integrating it into digest vectors. Moreover, the application of positional embeddings ensures that digest tokens attend to context in a sequential manner.

Layer-wise semantic diversification Thanks to IC-Former being composed of merely three layers, we are able to conduct a detailed analysis of each layer. We examine each layer of the IC-Former to identify the top five context tokens with the highest attention scores for each digest token.

As illustrated in Table 6, it can be observed that in the first layer, digest tokens mainly focus on prepositions, articles, be-verb, and punctuation marks. As we proceed to the second layer, digest tokens start to extend their focus to verbs, nouns, adjectives, and adverbs. The third layer continues this trend based on the second layer, further broadening the range of grammatical categories of tokens covered, encompassing a more extensive context. This implies that IC-Former might rely on semantic structures to understand and compress context effectively.

5 Conclusion

In this paper, we propose the In-Context Former (IC-Former), a novel context compression model, which can efficiently condense contextual information into digest vectors in a linear complexity by removing irrelevant interaction processing. Moreover, our proposed IC-Former utilizes the cross-attention mechanism to enhance the extraction ability of digest tokens. Our experimental results demonstrate that IC-Former significantly reduces time and space complexity while preserving contextual semantics, thereby supporting broader applications requiring extensive context.

Limitations

1. We only apply IC-Former to the Llama2-7b-chat model. Future efforts will involve conducting experiments on larger-scale models to explore further potential. It is anticipated that the increased hidden size in larger models will continue to enhance the performance of the IC-Former.

2. Although our method is capable of handling longer texts in implementation, we did not conduct compression experiments on longer contextual content to more comprehensively validate the method's performance due to resource constraints.

3. Despite our model significantly outperforming the baseline in terms of efficiency, it has not surpassed the baseline's performance in downstream tasks. Our future work will aim to enhance performance in scenarios that are less sensitive to real-time requirements.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew Gormley. 2024. Unlimiformer: Long-range transformers with unlimited length input. *Advances in Neural Information Processing Systems*, 36.

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

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. 2022. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 35:11079–11091.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023. Walking down the memory maze: Beyond context limit through interactive reading. *arXiv preprint arXiv:2310.05029*.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846, Singapore. Association for Computational Linguistics.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. 2020. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. 2023. Longnet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*.

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

Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

John R Hershey and Peder A Olsen. 2007. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Mark A Kramer. 1991. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dongxu Li, Junnan Li, Hung Le, Guangsen Wang, Silvio Savarese, and Steven C.H. Hoi. 2023. LAVIS: A one-stop library for language-vision intelligence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 31–41, Toronto, Canada. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jesse Mu, Xiang Li, and Noah Goodman. 2024. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71:599–607.

Charlie Snell, Dan Klein, and Ruiqi Zhong. 2022. Learning by distilling context. *arXiv preprint arXiv:2209.15189*.

Woomin Song, Seunghyuk Oh, Sangwoo Mo, Jaehyung Kim, Sukmin Yun, Jung-Woo Ha, and Jinwoo Shin. 2024. Hierarchical context merging: Better long context understanding for pre-trained llms. *arXiv preprint arXiv:2404.10308*.

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

Anca Maria Tache, Mihaela Gaman, and Radu Tudor Ionescu. 2021. Clustering word embeddings with self-organizing maps. application on laroseda–a large romanian sentiment data set. *arXiv preprint arXiv:2101.04197*.

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

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. 2022. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5621–5634, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuhuai Wu, Markus N Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing transformers. *arXiv preprint arXiv:2203.08913*.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chaoya Jiang, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. mplug-owl: Modularization empowers large language models with multimodality. *Preprint*, arXiv:2304.14178.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Lin Zheng, Chong Wang, and Lingpeng Kong. 2022. Linear complexity randomized self-attention mechanism. In *International conference on machine learning*, pages 27011–27041. PMLR.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A Experiment Details

A.1 Model Configuration

We show the detailed configuration of our IC-Former model in Table 7.

Hyperparameter	Value
theta base	10000.0
hidden size	4096
layer number	3
rms norm eps	1e-6
initializer range	0.02
activate function	silu
intermediate size	11008
digest tokens number	128
attention heads number	32
max position embeddings	2048

Table 7: Detailed configuration of IC-Former.

A.2 Training Configuration

We show the detailed configuration of pretraining and fine-tuning in Table 8 & 9.

Hyperparameter	Value
optimizer	AdamW
learning rate	1e-4
batch size	1
gradient accumulation	16
clip norm	2.0
training steps	9.3k
dtype	bfloat16

Table 8: Detailed configuration of pretraining.

Hyperparameter	Value
optimizer	AdamW
learning rate	5e-5
batch size	1
gradient accumulation	256
clip norm	2.0
training steps	7.9k
dtype	bfloat16

Table 9: Detailed configuration of fine-tuning.

A.3 Prompt Template on Evaluation

The prompt template we used for evaluation is as follows:

```
Response the Prompt based on the below text:\n\n {context}\n\n Prompt:{prompt}
```

B Profiling Setup

We use a single Nvidia RTX A6000 GPU (48GB) for pretraining, fine-tuning, and efficiency tests (Table 1). The CPU of our machine is Intel(R) Xeon(R) Gold 6326 with 16 cores and 1007GB RAM. The runtime configuration is python=3.8.18, pytorch=1.13.1, cuda=11.7, cudnn=8.5.

C Theoretical Analysis

C.1 Complexity Analysis

In Table 2 we assert that the time and space complexity of ICAE is $\mathcal{O}(n^2 + 2kn)$. This conclusion can be easily drawn by comparing the attention maps of the IC-Former and ICAE. As illustrated in Figure 7, ICAE utilizes memory tokens and context for causal self-attention interaction, resulting in a complexity of $\mathcal{O}\left((n+k)^2\right) \sim \mathcal{O}(n^2 + 2kn)$.

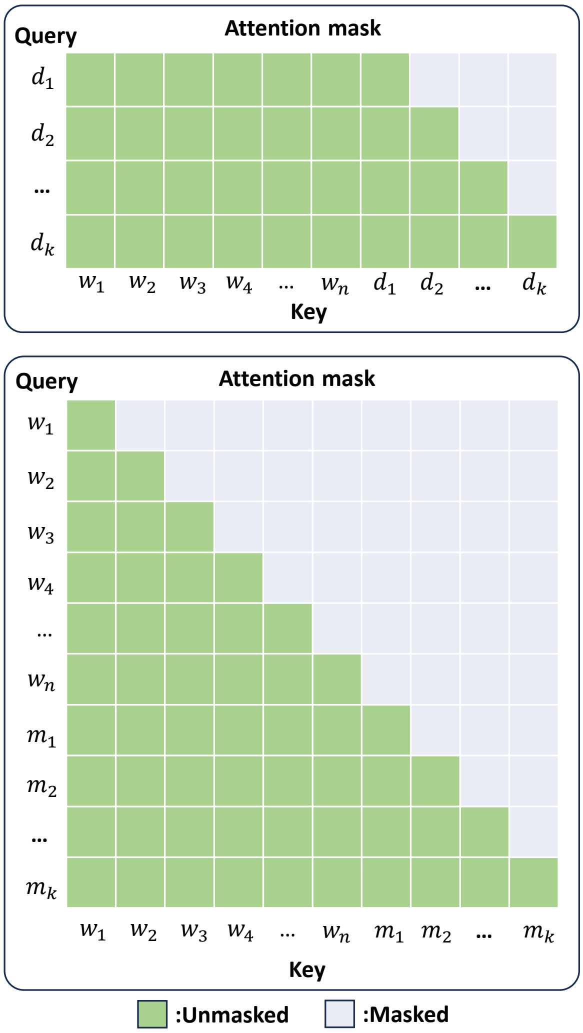

Figure 7: **Top:** Attention mask in IC-Former. **Bottom:** Attention mask in ICAE. The d_i represents digest tokens in IC-Former and the m_i represents the memory tokens in ICAE's encoder.

C.2 Floating Point Operations Calculation

When calculating the floating-point operations, we considered only the matrix multiplication computations involved in the attention and feed-forward network (FFN) modules, while ignoring the relatively smaller computational overhead of modules such as normalization and softmax.

Given context embedding with shape of $[b, s, h]$ where b represents batch size, s represents sequence length and h represents hidden size, the theoretical calculation of the FLOPs for ICAE and IC-Former required to compress it into vectors of length k are shown in Tables 10 & 11:

Modules	FLOPs
$xW_Q/W_K/W_V$	$3 \cdot 2b(s+k)h^2$
QK^T	$2b(s+k)^2h$
AV	$2b(s+k)^2h$
xW_O	$2b(s+k)h^2$
$x_{out}W_{up}$	$2b(s+k)hm$
$x_{out}W_{gate}$	$2b(s+k)hm$
$x_{out}W_{down}$	$2b(s+k)hm$
SUM	$4bh(s+k)(2h+s+k)$ $+6bhm(s+k)$

Table 10: Theoretical complexity in each layer of ICAE's encoder. A represents the attention scores matrix, m represents the intermediate size of FFN.

Modules	FLOPs
xW_Q	$2bkh^2$
xW_K/W_V	$2 \cdot 2b(s+k)h^2$
QK^T	$2bk(s+k)h$
AV	$2bk(s+k)h$
xW_O	$2bkh^2$
$x_{out}W_{up}$	$2bkhm$
$x_{out}W_{gate}$	$2bkhm$
$x_{out}W_{down}$	$2bkhm$
SUM	$4bkh^2 + 2bh(s+k)(h+2k)$ $+6bkhm$

Table 11: Theoretical complexity in each layer of IC-Former. A represents the attention scores matrix, m represents the intermediate size of FFN.

The ratio of FLOPs between ICAE and IC-Former R can be calculated as follows:

$$R = \frac{l_1 \cdot [2(s+k)(2h+s+k) + 3m(s+k)]}{l_2 \cdot [2kh + (s+k)(h+2k) + 3mk]}, \tag{12}$$

where l_1 is the layers of ICAE and l_2 is the layers of IC-Former.

In our experimental settings, $l_1 = 32$, $l_2 = 3$, $s = 512$, $k = 128$, $h = 4096$, $m = 11004$, thus

$$R \approx 32.39 \tag{13}$$

D Case Study

In Table 12, we present several cases to compare the outputs of Llama2-7b-chat based on the 128 digest vectors generated from the pretrained and non-pretrained IC-Former. The results indicate that the IC-Former without pre-training has a poor ability to capture contextual information and thus is more prone to hallucinating.

E Attention Maps in IC-Former

Additionally, by comparing the attention maps of the pretrained and non-pretrained IC-Former models (Figure 8 & 9), it is observable that the non-pretrained IC-Former does not exhibit the phenomenon of neighborhood information aggregation. Furthermore, the words captured by each layer do not demonstrate distinct grammatical patterns, which underscores the necessity of pretraining in enhancing model performance.

Context 1
French senior civil servant arrested on suspicion of spying for North Korea November 27, 2018 by Joseph Fitsanakis A senior civil servant in the upper house of the French parliament has been arrested on suspicion of spying for North Korea, according to prosecutors. The news of the suspected spy's arrest was first reported on Monday by Quotidien, a daily politics and culture show on the Monaco-based television channel TMC. The show cited "a judicial source in Paris" and said that France's domestic security and counterintelligence agency, the General Directorate for Internal Security (DGSI), was in charge of the espionage case. **(omitted 155 tokens)** Quennedey is believed to be the president of the Franco-Korean Friendship Association, the French branch of a Spanish-based organization that lobbies in favor of international support for North Korea.
Prompt: Mention the organization Benoit Quennedey is believed to be the president of.
Output (with pretrained IC-Former): Benoit Quennedey is believed to be the president of the Franco-Korean Friendship Association.
Output (with non-pretrianed IC-Former): Benoit Quennedey is believed to be the president of the French intelligence agency, the Directorate General for External Security (DGSE).
Label (by GPT-4): Franco-Korean Friendship Association
Context 2
One of NewsBusters' classification categories is "Covert Liberal Activists," and my antennae immediately went up when Lauren Leader was introduced on today's Morning Joe as "co-founder and CEO of All In Together, "a non-partisan, women's civic and political organization." **(omitted 462 tokens)** Another clue as to Leader's liberal leanings. Questioning whether the immigrants "actually threw rocks," she cites . . . NPR reporting that things were "relatively peaceful except that, obviously, they were trying to break through." When people are trying to illegally "break through" a border, there is nothing "relatively peaceful" about it.
Prompt: What source does Lauren Leader use to question the accuracy of the immigrants throwing rocks?
Output (with pretrained IC-Former): Lauren Leader uses NPR to question the accuracy of the immigrants throwing rocks.
Output (with non-pretrianed IC-Former): The source is a Fox News segment.
Label (by GPT-4): Lauren Leader cites NPR reporting as a source to question the accuracy of the immigrants throwing rocks.

Table 12: Examples of output results from Llama2-7b-chat model utilizing digest vectors generated by pretrained and non-pretrained IC-Former models. The evidence of prompt is marked in blue and red denote the outputs that are not faith to the original context.

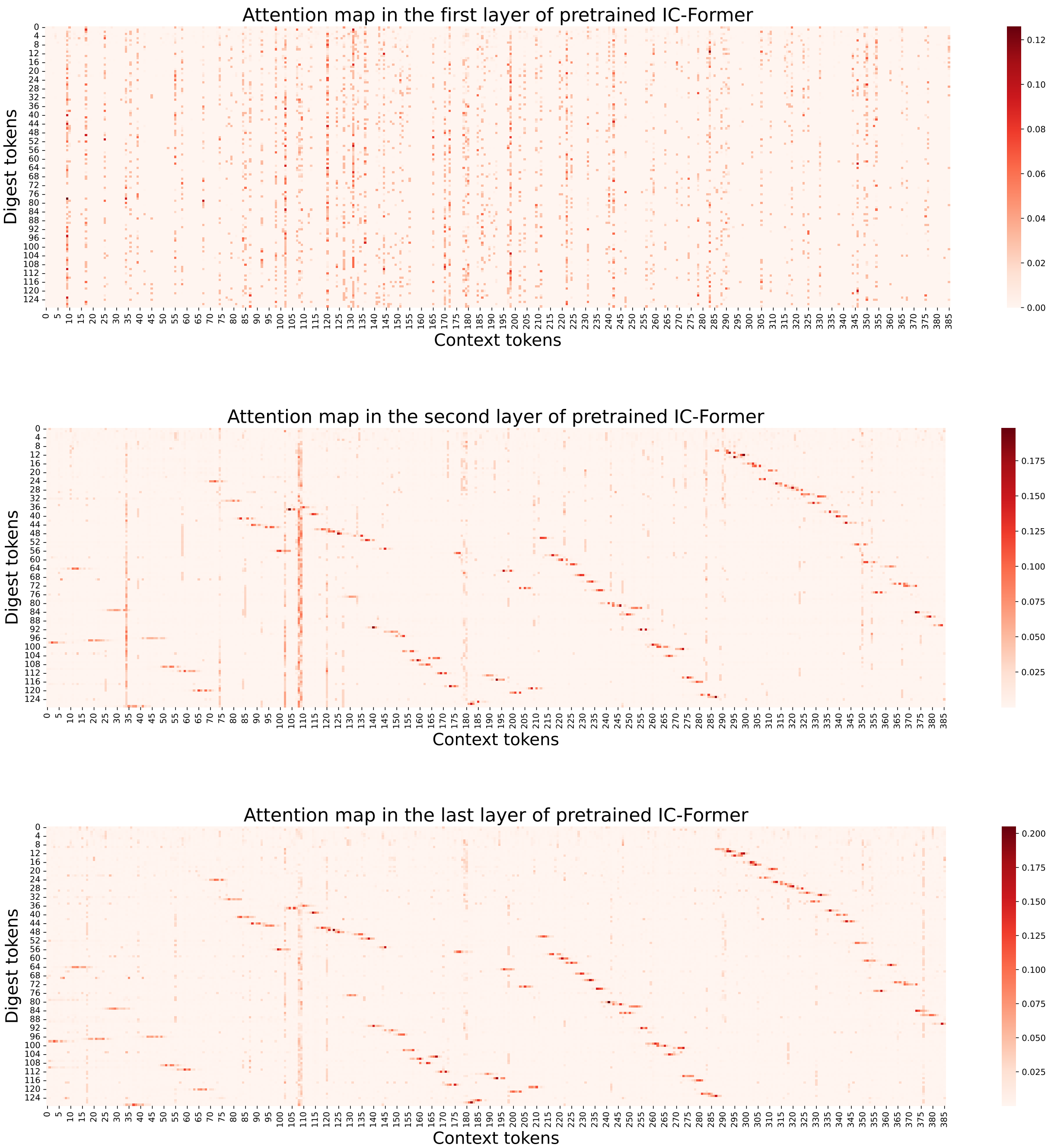

Figure 8: Complete attention maps of pretrained IC-Former. From top to bottom are attention maps of the first, second, and third layers of IC-Former.

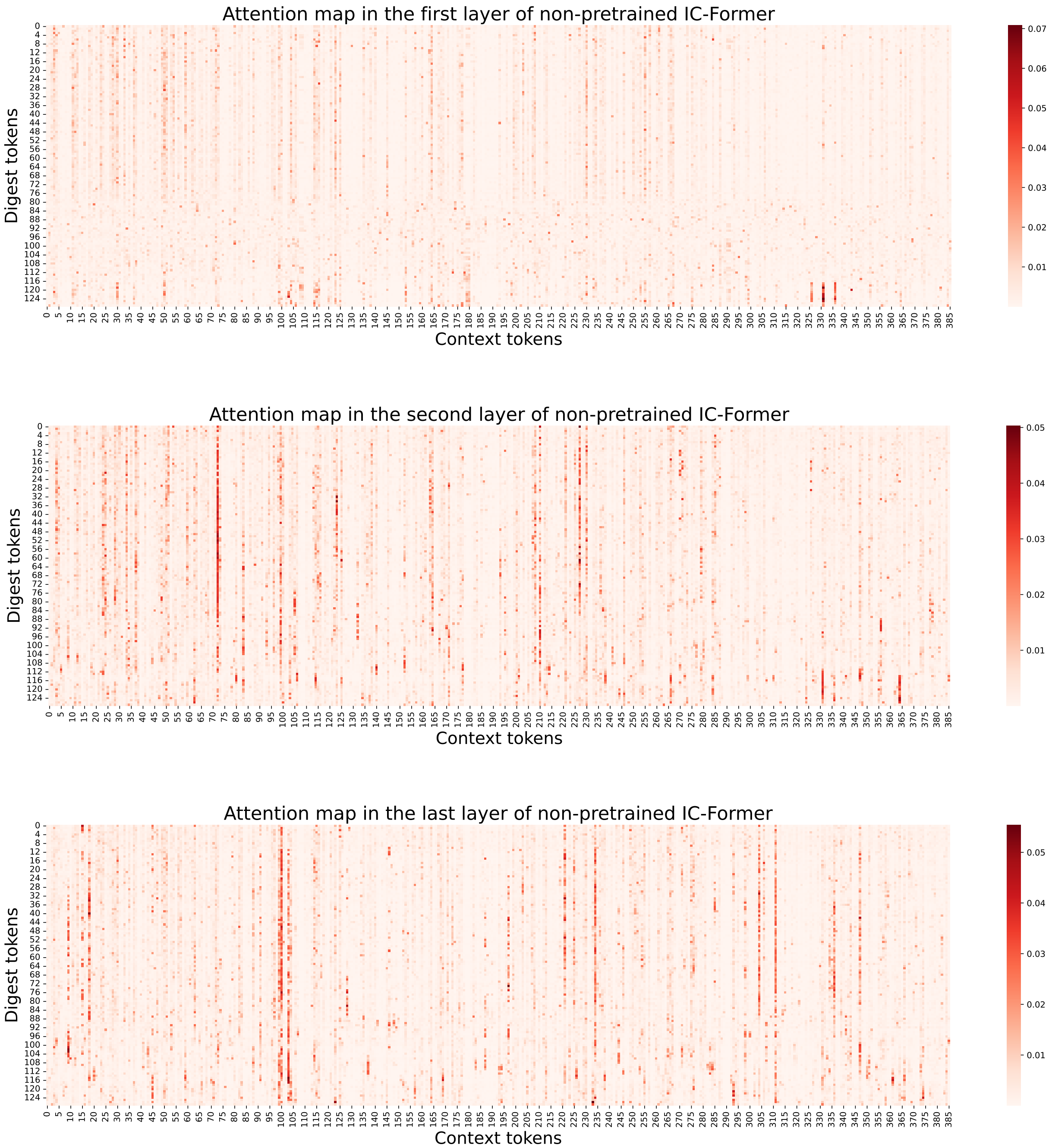

Figure 9: Complete attention maps of non-pretrained IC-Former. From top to bottom are attention maps of the first, second, and third layers of IC-Former.