Position IDs Matter: An Enhanced Position Layout for Efficient Context Compression in Large Language Models

Runsong Zhao[1,*], **Xin Liu**[1,*], **Xinyu Liu**[1], **Pengcheng Huang**[1], **Chunyang Xiao**, **Tong Xiao**[1,2,†] and **Jingbo Zhu**[1,2]

[1] NLP Lab, School of Computer Science and Engineering, Northeastern University, Shenyang, China

[2] NiuTrans Research, Shenyang, China

{fhe43694, lxy1051493182, chunyangx}@gmail.com, lx_meteor_code@163.com

pengcheng.neu@outlook.com, {xiaotong, zhujingbo}@mail.neu.edu.com

Abstract

Using special tokens (e.g., gist, memory, or compressed tokens) to compress context information is a common practice for large language models (LLMs). However, existing approaches often neglect that position encodings inherently induce local inductive biases in models, causing the compression process to ignore holistic contextual dependencies. We propose **Enhanced Position Layout (EPL)**, a simple yet effective method that improves the context compression capability of LLMs by only adjusting position IDs, the numerical identifiers that specify token positions. EPL minimizes the distance between context tokens and their corresponding special tokens and at the same time maintains the sequence order in position IDs between context tokens, special tokens, and the subsequent tokens. Integrating EPL into our best performing context compression model results in a 1.9 ROUGE-1 F1 improvement on out-of-domain question answering datasets on average. When extended to multimodal scenarios, EPL leads to an average accuracy gain of 2.6 points for vision compression LLMs. [1]

1 Introduction

In Transformer (Vaswani et al., 2017) architectures, special tokens have been widely adopted as compression carriers of contextual information across natural language processing (Devlin et al., 2019; Liu et al., 2019; Bulatov et al., 2022; Ge et al., 2024; Li et al., 2024b) and computer vision (Dosovitskiy et al., 2021; Ye et al., 2025). For context compression, the so-called soft prompt methods (Chang et al., 2024; Li et al., 2025) employ encoders to condense long contexts into few special tokens, enabling decoders to perform inference based on compressed representations rather than raw inputs, thereby significantly reducing memory consumption and inference latency in long-context scenarios (Jiang et al., 2024; Xu et al., 2024; Luo et al., 2025). We illustrate typical soft prompt architectures in Figure 1 where special tokens are appended at the end of the context, in order to capture the context semantics via causal attention mechanism in LLMs.

The design ensures full context visibility for special tokens. However, in Transformer architectures position IDs do not need to coincide with physical token positions and the model's perceived positional information is primarily determined by position IDs rather than physical token positions (Vaswani et al., 2017). From this viewpoint, the local inductive biases introduced by position encodings (Devlin et al., 2019; Vaswani et al., 2017; Su et al., 2023; Raffel et al., 2020; Press et al., 2022) weaken the efficacy of context compression under the default position layout (DPL), due to the substantial distance between the special tokens and the context tokens, as shown in Figure 1 (DPL). In this paper, we carefully examine position layout designs and propose **Enhanced Position Layout (EPL)** for soft prompt architectures, which comprises **Uniform Position Layout (UPL)** and **Consistent Position Layout (CPL)**.

UPL redistributes special tokens' position IDs to achieve uniform distribution in the context tokens' position ID space, as exemplified in Figure 1. By uniformly assigning position IDs among context token position IDs, *a priori*, most context tokens would have corresponding special tokens close to them. We assume that such a prior helps the special tokens compress the context. We formalize such intuitions, demonstrating the optimality of the UPL in Section 3.2.1. During compression, because special tokens are inserted and text chunks are reorganized, the position IDs between context, special tokens and subsequent tokens (e.g. reconstructed tokens or subsequent tokens such as QA pairs) can

[*]These authors contributed equally to this work.

[†]Corresponding author.

[1]Our code is available at `https://github.com/1azybug/EPL`.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 17715–17734

become inconsistent compared to their original positions before compression. Our proposed CPL in Section 3.2.2 guarantees to maintain the order of the position ID sequence for different tokens, between different text chunks in their natural causal order.

We empirically apply EPL to two dominant context compression frameworks: ICAE (Ge et al., 2024) and 500xCompressor (Li et al., 2024b). For the best model, on the autoencoding (AE) task, EPL yields a 1.8 BLEU gain and converges 9.7 times faster than DPL; on out-of-domain question answering (QA) tasks, EPL gives an average 1.9 ROUGE-1 F1 gain. To demonstrate the universality of EPL, we further conduct experiments by incorporating EPL into a more recent fine-grained compression framework (Zhang et al., 2025) and test our EPL equipped compressor on PwC dataset (Ge et al., 2024), an instruction-tuning dataset containing diverse context-related prompts. In both scenarios, incorporating EPL brings consistent improvement. When extending our application to multimodality with VoCo-LLaMA (Ye et al., 2025), EPL yields an average 2.6 accuracy gain on multimodal benchmarks. The improvement of EPL is consistent across base models of different scales. Further analysis shows that both UPL (which aims for better context compression) and CPL (which maintains causal sequence ordering) are essential for the final performance improvement across tasks. Finally, our UPL attention map visualization confirms the usefulness of our specified prior: UPL special tokens indeed focus more on tokens close to its assigned position IDs.

2 Background

2.1 Local Bias of Position Encodings

Transformer architectures (Vaswani et al., 2017) compute contextual token embeddings through position-invariant self-attention. Since natural language semantics crucially depend on token order, various position encodings (PEs) have been proposed to inject positional awareness including *Sinusoidal PE* (Vaswani et al., 2017), *RoPE* (Su et al., 2023), *Learnable PE* (Devlin et al., 2019), *T5 Bias* (Raffel et al., 2020) and *ALiBi* (Press et al., 2022), etc. All approaches share the inductive bias that adjacent tokens should correlate more strongly. Taking the PEs with the trigonometric encoding (e.g. Sinusoidal/RoPE) design as examples, the position embedding at a certain position is mostly

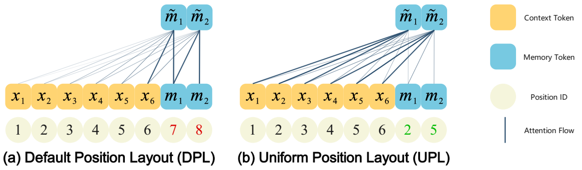

Figure 1: Comparison of UPL and DPL. In prior work (DPL), memory tokens are assigned position IDs 7 and 8. Our method (UPL) allocates them to position IDs 2 and 5. Tokens with IDs in close proximity tend to exhibit higher attention scores.

similar to its neighbors and the similarity decays as the distance increases, see Appendix H.1 for more details. ALiBi enforces this inductive bias by applying distance-sensitive penalties to attention scores. We further show in Section 5.3 that Learnable PE and T5 Bias learn similar local bias through pre-training on natural text.

2.2 Position Layout

While the local inductive bias for PEs is well known, less is known about the **position layout**. For any given token sequence and model, we refer to position layout as the actual position ID sequence that is assigned by the model to the token sequence. Notably, in this work, we remark that the local inductive bias applies to the position layout, not the physical token positions.

This nuance is important but hardly noticeable because the **default position layout (DPL)** often coincides with the physical token positions as illustrated in DPL in Figure 1 (see Table 14 for more details on DPL). Such position layout is helpful for language modeling (LM) task since the task depends largely on its recent context (Hu et al., 2024; Liu et al., 2024a). However, as LLMs are becoming a ubiquitous tool, we hypothesize that careful position layouts for some tasks can inject helpful inductive bias. In the rest of the paper, we focus on LLM compression tasks as our testbed.

3 Method

In Section 3.1, we review existing LLM compression frameworks and their DPLs; in Section 3.2 we describe our improved position layout.

3.1 Soft Prompt Methods

ICAE (Ge et al., 2024) is a widely used encoder-decoder soft prompt method. Its encoder compresses long context into a few memory tokens, after which the decoder performs inference conditioned only on the memory tokens to achieve faster inference speed. The left side of Figure 3 in the Appendix illustrates this process through an example. ICAE can be trained through two stages: continued pretraining and fine-tuning. Continued pretraining trains on a combination of AutoEncoding (AE) tasks and Language Modeling (LM) tasks, which trains the LLM encoder so that the encoded memory tokens enable a frozen LLM to losslessly reconstruct the original context and at the same time predict the subsequent tokens following the context to maintain ICAE's generation capability. The right side of Figure 3 illustrates the AE training process. Fine-tuning further trains the ICAE encoder to adapt to real-world applications such as question answering (QA) that we use in this work. We review the AE and LM pretraining as well as QA fine-tuning in the next subsections. To handle arbitrarily long contexts, we adopt the multi-chunk version of ICAE, which divides any long context into chunks for independent compression and then aggregates the resulting memory tokens to represent the complete long context. 500xCompressor (Ge et al., 2024) is similar to ICAE, with the difference of using the KV Cache of memory tokens as the compression carrier instead of the output of memory tokens.

3.1.1 Pretraining

During the pretraining stage, for a token sequence $X = \{x_0, x_1, \ldots, x_{|X|-1}\}$, we take the first p tokens as the context $X_{\text{context}} = \{x_0, x_1, \ldots, x_{p-1}\}$ and the subsequent $|X| - p$ tokens as the completion $X_{\text{completion}} = \{x_p, x_{p+1}, \ldots, x_{|X|-1}\}$. The AE task only uses X_{context}, while the LM task leverages both X_{context} and $X_{\text{completion}}$.

Compress X_{context} is partitioned into $k = \lceil p/L \rceil$ chunks (with chunk size L) where each chunk $S^{(i)} = \{x_{(i-1)L}, x_{(i-1)L+1}, \ldots, x_{iL-1}\}$ is appended with a set of learnable memory tokens $M^{(i)} = \{m_0^{(i)}, m_1^{(i)}, \ldots, m_{|M|-1}^{(i)}\}$. A LLM learns to encode each chunk into memory output tokens $\tilde{M}^{(i)}$ and key-value cache $KV^{(i)}$:

$$\tilde{M}^{(i)}, KV^{(i)} = \text{LLM}([S^{(i)}; M^{(i)}] \mid \theta_{\text{LoRA}}) \quad (1)$$

where $[;]$ denotes concatenation along the sequence dimension. All $M^{(i)}$ share the same learnable parameters M, and θ_{LoRA} denotes a set of low-rank adapter (Hu et al., 2022) parameters for the LLM. The final compressed representation is obtained by concatenating the results of each chunk: $\tilde{M} = [\tilde{M}^{(1)}; \tilde{M}^{(2)}; \ldots; \tilde{M}^{(k)}]$ or $KV = [KV^{(1)}; KV^{(2)}; \ldots; KV^{(k)}]$.

Pretraining AE loss in ICAE is given by:

$$\mathcal{L}_{\text{AE}} = -\log P\big(X_{\text{context}} \mid [\tilde{M}; \texttt{[AE]}]\big) \quad (2)$$

where [AE] is a learnable token prompting the frozen decoder to generate X_{context} as reconstruction. Similar to the AE task, the loss for the LM task in ICAE is given by:

$$\mathcal{L}_{\text{LM}} = -\log P\big(X_{\text{completion}} \mid [\tilde{M}; \texttt{[LM]}]\big) \quad (3)$$

where [LM] is a learnable token that prompts the LLM to perform completion. We employ a weighted loss function for joint training[2] :

$$\mathcal{L}_{\text{pretrain}} = \alpha \mathcal{L}_{\text{AE}} + (1-\alpha)\mathcal{L}_{\text{LM}}, \quad \alpha = 0.5$$

Position Layout Recall that for encoding, the token sequence starts with text chunk $S^{(i)}$ followed by memory tokens $M^{(i)}$. For decoding, the token sequence starts with memory token outputs $\tilde{M}$ for ICAE or KV for 500xCompressor followed by [LM] or [AE] and then subsequent tokens (i.e. X_{context} or $X_{\text{completion}}$).

The ICAE default position layout (DPL) always coincides with their physical token positions. For ICAE encoder for example, this means its DPL starts with position ID 0 and ranges till $|S^{(i)}| + |M^{(i)}| - 1$. For 500xCompressor, its encoder DPL also coincides with physical token positions. However, the position IDs of the decoder's $KV^{(i)}$ are the same as the position IDs of the encoder's $M^{(i)}$ (the memory tokens of the i-th chunk)[3], which implies that the decoder DPL for the KV[4] consists of k repeated ranges from $|S^{(i)}|$ to $|S^{(i)}| + |M^{(i)}| - 1$. During decoding, DPL for the rest of the tokens (e.g. [AE], X_{context} in AE task) still coincides with their physical token positions. Table 2 and Table 3 show examples of ICAE and 500xCompressor DPL with X_{context} having two chunks (k=2) under LM and AE task respectively.

[2]Unlike Ge et al. (2024)'s per-instance task allocation (AE with probability α /LM with $1-\alpha$), our joint training processes both tasks simultaneously through shared compressed representations, improving training efficiency by eliminating redundant context compression operations.

[3]This is because the KV Cache has already cached the position information of the key_state, see this code snippet.

[4]Recall that KV concatenates all $KV^{(i)}$ with $KV = [KV^{(1)}; KV^{(2)}; \ldots; KV^{(k)}]$.

$k=2, L=510, \lvert M \rvert=102, r=5, \lvert X \rvert=2040, p=1020$																
	$S^{(1)}$								$S^{(2)}$							
	x_0	x_1	…	x_{509}	$m_0^{(1)}$	$m_1^{(1)}$	…	$m_{101}^{(1)}$	x_{510}	x_{511}	…	x_{1019}	$m_0^{(2)}$	$m_1^{(2)}$	…	$m_{101}^{(2)}$
DPL(ICAE/500x)	0	1	…	509	510	511	…	611	0	1	…	509	510	511	…	611
EPL	1	2	…	510	3	8	…	508	511	512	…	1020	513	518	…	1018

Table 1: Position layout of the **encoder** for $S^{(1)}$ and $S^{(2)}$. We show DPL's problematic position IDs in red, EPL's UPL-adjusted IDs in green, and CPL-adjusted IDs in blue. For more details, see Appendix A, Table 15.

$k=2, L=510, \lvert M \rvert=102, r=5, \lvert X \rvert=2040, p=1020$											
	$\tilde{m}_0^{(1)}/KV_0^{(1)}$	…	$\tilde{m}_{101}^{(1)}/KV_{101}^{(1)}$	$\tilde{m}_0^{(2)}/KV_0^{(2)}$	…	$\tilde{m}_{101}^{(2)}/KV_{101}^{(2)}$	[AE]	x_0	x_1	…	x_{1019}
DPL(ICAE)	0	…	101	102	…	203	204	205	206	…	1224
DPL(500x)	510	…	611	510	…	611	204	205	206	…	1224
EPL	3	…	508	513	…	1018	0	1	2	…	1020

Table 2: Position layout example of the **decoder** in AE Task. For more details, see Appendix A, Table 16.

$k=2, L=510, \lvert M \rvert=102, r=5, \lvert X \rvert=2040, p=1020$											
	$\tilde{m}_0^{(1)}/KV_0^{(1)}$	…	$\tilde{m}_{101}^{(1)}/KV_{101}^{(1)}$	$\tilde{m}_0^{(2)}/KV_0^{(2)}$	…	$\tilde{m}_{101}^{(2)}/KV_{101}^{(2)}$	[LM]	x_{1020}	x_{1021}	…	x_{2039}
DPL(ICAE)	0	…	101	102	…	203	204	205	206	…	1224
DPL(500x)	510	…	611	510	…	611	204	205	206	…	1224
EPL	3	…	508	513	…	1018	1020	1021	1022	…	2040

Table 3: Position layout example of the **decoder** in LM Task. For more details, see Appendix A, Table 17.

3.1.2 Fine-tuning

ICAE (Ge et al., 2024) allows further fine-tuning to enhance downstream task performance by enabling memory tokens to learn to focus on the tokens that are most related to the task; the training process is similar to LM pretraining. Let each training instance consist of a triplet (C, Q, A), where context C is compressed into either $\tilde{M}$ (ICAE) or KV (500xCompressor). The answer $A = \{a_0, a_1, ..., a_{|A|-1}\}$ is generated conditioned on the compressed representation and the question $Q = \{q_0, q_1, ..., q_{|Q|-1}\}$. The loss for QA task in ICAE is given by:

$$\mathcal{L}_{\text{QA}} = -\log P\big(A \mid [\tilde{M}; \texttt{[LM]}; Q]\big) \qquad (4)$$

The QA loss for 500xCompressor is similar to Eq. (4), with the difference of conditioning on KV instead of $\tilde{M}$.

Position Layout The DPL for the QA task is similar to the DPL for the LM task and can be derived by replacing the $X_{\text{completion}}$ in the LM task with the concatenation $[Q; A]$. The resulting DPL for $[Q; A]$ coincides with their physical token positions. For the detailed DPL, see Table 18 in the appendix.

3.2 Enhanced Position Layout

For the DPL in soft prompt methods as described in Section 3.1, we identify two limitations:

Distant Memory Tokens Memory tokens in the soft prompt framework mainly aim to compress the context tokens so that the inference can be solely based on them to accelerate inference. However, the memory tokens in DPL consist of a continuous range (i.e. 510-611 in Table 1) and are all very distant from the context token's range (i.e. 0-509). Given the local inductive bias of PEs that we briefly review in Section 2.1, it would be advantageous to have memory token position IDs to be both close to context token position IDs and covering the context token ID range. In Section 3.2.1, we propose Uniform Position Layout (UPL) that has memory token position layout covering the context token ID range while achieving minimum ID distances between memory tokens and context tokens.

Inconsistent Layout Standard Transformer DPL coincides with physical token positions, which implies that tokens with larger position IDs follow the tokens with smaller position IDs, reflecting the causal relationship between the tokens through their assigned IDs. However, we observe that some DPL does not comply with such properties. For example, 500xCompressor's decoder DPL as shown in Table 3 starts with the KV position IDs $\{510, 511, \ldots, 611\}$ but is followed by position ID sequence $\{204, 205, \ldots, 1224\}$ representing $[\texttt{[LM]}; X_{\text{completion}}]$. In Section 3.2.2, we detail

Algorithm 1 Generate Uniformly Distributed Compression Position IDs

Require: Memory tokens count $|M|$, start/end IDs v_1, v_L

Ensure: Uniform positions U^*

$r \leftarrow (v_L - v_1 + 1)/|M|$

$o \leftarrow \frac{r-1}{2}$

$U^* \leftarrow$ torch.linspace$\big(v_1 + o, v_L - o,\ |M|\big)$

return torch.round(U^*)

our Consistent Position Layout (CPL) design which guarantees the resulting position layout to maintain the causal structure amongst position IDs. We hypothesize that aligning the causal structures for position IDs will benefit performance.

3.2.1 Uniform Position Layout

Recall that for memory token position layout, we aim to achieve two objectives: (1) the memory token position ID should be close to context tokens (2) for any context token, there is some memory token(s) whose IDs are close to it. Figure 1 (b) **Uniform Position Layout (UPL)** illustrates such design, for any context token position ID, the nearest memory token position ID does not deviate more than 1, which sets contrast to DPL where the position ID of the first context token x_1 is far from all memory token position IDs. In the following, we formalize the desiderata to derive analytically the optimal position layout UPL.

Given a sequence of context token position IDs $V = \{v_1, v_2, \ldots, v_L\}$, $v_i \in \mathbb{N}$ and $v_{i+1} - v_i = 1\ \forall i$, we aim to devise an algorithm to find position IDs for $|M|$ memory tokens $U = \{u_1, u_2, \ldots, u_{|M|}\}$, $u_j \in \mathbb{N}$ [5], that minimize the following function:

$$\max_{v_i \in V} \left(\min_{u_j \in U} |v_i - u_j| \right)$$

where $\min_{u_j \in U} |v_i - u_j|$ represents the distance from the i-th context token to its nearest memory token, and $\max_{v_i \in V}$ takes the maximum of all minimum distances across context tokens.

The optimal solution divides V evenly into $|M|$ groups, with each group containing at most $\lceil r \rceil$ tokens ($r = \frac{L}{|M|}$), and assigns each memory token the middle position of each group. In this case, the maximum distance from any context token to its nearest memory token is $\left\lfloor \frac{\lceil r \rceil}{2} \right\rfloor$ [6]. Intuitively, the solution spreads memory token position IDs uniformly in the range of context token position IDs to ensure that no context token position ID is too far away. We detail the memory token position layout algorithm in Algorithm 1 that we apply for each chunk to be compressed. Table 1 EPL row shows how the memory token position layout in UPL differs from DPL.

3.2.2 Consistent Position Layout

In this subsection, we propose consistent position layout (CPL) to ensure that the decoder position layout maintains the causal sequence order in position IDs between context tokens, `[LM]`/`[AE]`, and the subsequent tokens. As shown in Table 2 and 3 EPL rows, we keep memory token position layout unchanged compared to its encoding stage.

For tokens in X_{context} and $X_{\text{completion}}$, we simply assign their original sequence positions as their position IDs. For example, in the AE task, the token sequence $[$`[AE]`$; X_{\text{context}}]$ will be equipped with the position layout $\{0, 1, \ldots, p\}$ where p is the context length to reflect the tokens to be reconstructed from memory tokens.[7] For the LM task, the position layout is $\{p, p+1, \ldots, |X|\}$ for $[$`[LM]`$; X_{\text{completion}}]$ as $X_{\text{completion}}$ logically follows X_{context} in the physical token space. Table 2 and Table 3 EPL rows show concrete CPL during decoding through examples.

4 Experimental Results

4.1 Experimental Setup

Data For continued pretraining, we utilize the SlimPajama-6B (Soboleva et al., 2023) corpus. To evaluate model fine-tuning performance, we use the MRQA (Fisch et al., 2019a) dataset as our testbed. The dataset contains evaluation on both in-domain scenarios where the validation dataset has its training counterpart used during training and out-of-domain scenarios. We report results from both settings but mainly discuss results for out-of-domain scenarios as it assesses more critically the soft prompt compression effectiveness.

Model Configuration We evaluate our method on Llama-3.2-1B (Grattafiori et al., 2024). For efficient adaptation, we apply LoRA(Hu et al., 2022)

[5] Although non-integer position IDs are valid in RoPE (Su et al., 2023), they have not been encountered during pretraining, making it difficult for the model to effectively utilize these non-integer position IDs.

[6] Note that any position layout will have its maximum distance $\geq \left\lfloor \frac{\lceil r \rceil}{2} \right\rfloor$, proving the optimality.

[7] Remark that the procedure is the inverse of memory token construction presented in Table 1.

to the query and value projection matrices within the multi-head attention layers of the encoder. The LoRA rank is set to 128, and the LoRA alpha is set to 256. Following ICAE (Ge et al., 2024), we do not train the decoder. In our default configuration, the number of memory tokens is $|M| = 102$, the chunk size is $L = 510$ [8], implying a $r = 5$ compression ratio. All models are further pretrained and fine-tuned for 20k steps with a batch size of 16. Further hyperparameter details can be found in Appendix C, Table 11.

4.2 Fine-tuning Results

Following ICAE and 500xCompressor, we pretrain then fine-tune Llama-3.2-1B; we integrate our EPL changes described in Section 3.2 into different architectures respectively, then follow the same pretraining and fine-tuning steps. We evaluate the downstream performance using MRQA (Fisch et al., 2019b) for different experiments. We assess the quality of the model's answers using ROUGE-1 F1 (Lin, 2004) and Exact Match (EM) and report out-of-domain results in Table 4.

For both ICAE and 500xCompressor, incorporating EPL significantly improves the performance. The average ROUGE-1 F1 improves from 39.95 to 43.87 for ICAE and improves from 45.76 to 48.03 for 500xCompressor. The improvement is observed for most domains, suggesting that the method is overall effective. We observe a similar improvement for in-domain settings (see Table 19 in the appendix).

To further validate the effectiveness of EPL for general context compression, we test on the PwC dataset (Ge et al., 2023) which contains diverse scenarios and question types. The results in Table 5 show that EPL significantly improves downstream task performance compared to its counterpart, demonstrating its effectiveness.

4.3 Pretraining Results

Through pretraining, LLMs have learned to compress context into memory tokens, allowing evaluation over memory tokens for their reconstruction and language modeling capability. The evaluation methodology is widely adopted for soft prompting (Ge et al., 2024; Li et al., 2024a) and we expect EPL to bring a similar improvement to the fine-tuning settings since EPL incorporates useful prior to reconstruction and language modeling through its position layouts. The reconstruction quality and language modeling capability are evaluated using BLEU-4 (Papineni et al., 2002) and perplexity (PPL), respectively[9].

Table 4 confirms the EPL improvement. For both ICAE and 500xCompressor architectures, we observe better language modeling capability with lower perplexity as well as better reconstruction capability with higher BLEU. The improvement is more significant with the weaker ICAE model but significant for both architectures.

4.4 Applications to Multimodal Models

As EPL can be applied to all applications that compress context into special tokens, in this subsection, we showcase its application in multimodality. We follow VoCo-LLaMA (Ye et al., 2025) for visual question-answering tasks. Given a triplet (I, Q, A), the model encodes image I into a sequence of 576 visual tokens $V^t = \{vt_0, vt_1, ..., vt_{575}\}$, and subsequently compresses V^t into the KV values of **Vi**sion **Co**mpression (VoCo) tokens. VoCo-LLaMA adopts a single training stage akin to fine-tuning stage in 500xCompressor and employs a single-forward via an attention mask (see Figure 8 for the attention mask detail) to prevent Q and A from directly accessing V^t. VoCo-LLaMA uses DPL and we follow its experimental setup to examine the effect of changing DPL to EPL. The position layout changes are illustrated on top of Figure 8.

We evaluate VoCo-LLaMA with 128 VoCo tokens (i.e. 4.5x compression ratio) and report performance on multimodal benchmarks. As shown in Table 6, VoCo-LLaMA combined with EPL significantly outperforms both its DPL counterpart from our reproduction and the results reported by Ye et al. (2025). We observe improvement across all three tasks, validating the universality of EPL.

4.5 Ablation Studies

We conduct ablation studies to analyze the independent effects of UPL and CPL in EPL (see Section 3.2 for the detailed description of UPL and CPL). LM perplexity and AE BLEU are measured under the same experimental settings as in Section 4.3 while out-of-domain performance is measured in the same way as in Section 4.2. We show the results in Table 7 which confirm that:

[8] As context often exceeds the chunk size, we extensively evaluate multi-chunk settings, contrary to ICAE and 500xCompressor.

[9] Reconstruction texts are generated via greedy search.

	LM	AE	BioASQ		DROP		DuoRC		RACE		RE		TQA		Avg.	
	PPL	BLEU	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
ICAE(DPL)	12.18	31.80	38.39	26.99	**37.13**	**27.28**	27.90	18.32	21.67	4.30	**76.98**	**64.45**	37.62	22.55	39.95	27.32
ICAE(EPL)	**11.42**	**95.98**	**42.66**	**29.19**	37.03	26.95	**32.50**	**21.39**	**26.87**	**5.34**	76.90	64.31	**47.25**	**29.81**	**43.87**	**29.50**
500x(DPL)	11.22	93.73	44.22	31.38	40.47	29.34	32.20	21.19	27.67	5.93	**83.03**	**71.98**	47.00	28.88	45.76	31.45
500x(EPL)	**10.80**	**98.50**	**46.11**	**31.91**	**40.94**	**29.41**	**39.07**	**26.32**	**31.89**	**7.27**	80.39	67.71	**49.80**	**31.14**	**48.03**	**32.29**

Table 4: Pretraining and fine-tuning results. For the pretraining results, we report perplexity for completion and BLEU-4 (Papineni et al., 2002) score for reconstruction calculated on a held-out set of 1k examples. For the fine-tuning results, we report out-of-domain results, which include 6 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). F1 in all the tables is ROUGE-1 F1 (Lin, 2004).

	ROUGE-1 F1	BLEU4
ICAE-1B (DPL)	42.71	18.05
ICAE-1B (EPL)	**45.59**	**21.04**

Table 5: Results on PwC (Ge et al., 2024).

	SQA(OOD)	MMB(OOD)	VQAv2(ID)
DPL*	-	61.0	76.9
DPL	64.9	59.9	76.9
EPL	**66.5**	**64.6**	**78.4**

Table 6: Results of VoCo-LLaMA on SQA (Lu et al., 2022), MMB (Liu et al., 2024b), and VQAv2 (Goyal et al., 2017). Results marked with * are from (Ye et al., 2025).

UPL improves performance by injecting compression prior For all testing cases (AE and LM during pretraining and MRQA out-of-domain performance for fine-tuning) over all architectures (ICAE and 500xCompressor), UPL improves performance compared to its counterparts. We note that the improvement is more significant for the weaker ICAE model. For example, the MRQA out-of-domain ROUGE-1 F1 improves by 3.72 points while the improvement is 1.07 points for 500xCompressor. The results confirm that by carefully designing the memory position layout for compression tasks, the memory tokens obtain useful compression inductive prior, which improves the compression efficiency (i.e. AE task performance) and consequently downstream task performance as shown by MRQA out-of-domain performance.

CPL improves performance by maintaining token sequential orders Similarly, we observe consistent improvement across the board by incorporating CPL. We note that the performance improvement for CPL on top of UPL is more prominent for the stronger 500xCompressor model; MRQA out-of-domain ROUGE-1 F1 improves by 1.20 points while the improvement is only 0.20 for

	LM	AE	Out-of-Domain	
	PPL	BLEU	F1	EM
ICAE-1B	12.18	31.80	39.95	27.31
+UPL	11.50	72.35	43.67	29.11
+UPL & CPL (i.e. EPL)	**11.42**	**95.98**	**43.87**	**29.50**
500x-1B	11.22	93.73	45.76	31.45
+UPL	10.94	95.86	46.83	31.64
+UPL & CPL (i.e. EPL)	**10.80**	**98.50**	**48.03**	**32.29**

Table 7: Ablation study on EPL integration. *+UPL* applies only UPL to encoder; *UPL & CPL (i.e. EPL)* applies both UPL and CPL. Reported values are averages. More ablation results are in Tables 21 and Tables 22.

ICAE. We think this is because ICAE DPL maintains the token sequential orders between X_{context}, `[LM]`, and $X_{\text{completion}}$ while 500xCompressor does not. As illustrated in Table 3, while the ICAE DPL coincides with physical token sequence positions, the 500xCompressor DPL exhibits KV position IDs larger than the `[LM]` token position ID (and some $X_{\text{completion}}$ position IDs), breaking the causal relationship between $\{X_{\text{context}}, \texttt{[LM]}, X_{\text{completion}}\}$ in the position layout.

4.6 Scalability Results

We conduct experiments at 3B and 8B scales from the same model family to verify method scalability. Table 8 shows that EPL achieves performance improvements at all scales and the improvement does not attenuate with larger scale. For example, at 8B scale, the MRQA ROUGE-1 F1 increases by 14.03 points for ICAE and by 1.89 points for 500xCompressor, which is similar to the improvement observed in 1B case in Table 4, holding promise for the method's effectiveness on large-scale language models. More detailed results on MRQA can be found in Table 19 and 20.

4.7 Result of Fine-Grained Compression

To further validate the effectiveness of EPL, we conduct additional experiments on Fine-Grained

	LM	AE	Out-of-Domain	
	PPL	BLEU	F1	EM
Llama-3.2-3B				
ICAE(DPL)	10.33	48.12	42.49	29.87
ICAE(EPL)	**9.07**	**97.05**	**55.03**	**38.54**
500x(DPL)	9.44	96.86	51.37	36.58
500x(EPL)	**8.68**	**99.34**	**57.71**	**40.49**
Llama-3.1-8B				
ICAE(DPL)	9.22	49.08	43.09	30.01
ICAE(EPL)	**7.61**	**98.82**	**57.12**	**39.76**
500x(DPL)	7.61	97.68	57.42	40.32
500x(EPL)	**7.40**	**99.49**	**59.31**	**41.45**

Table 8: Results at Llama-3.2-3B (Grattafiori et al., 2024), and Llama-3.1-8B. Reported values are averages.

Compression (FGC) (Zhang et al., 2025). FGC shares a similar motivation with our UPL, as it divides the context into finer-grained units and interleaves memory tokens among them. Since FGC places memory tokens after these fine-grained units, EPL can be naturally applied to enhance its performance by setting each memory token's position ID to the middle position of the corresponding fine-grained unit. Given that FGC utilizes the KV cache as its compression carrier like the 500xCompressor, we incorporate FGC into the 500xCompressor architecture maintaining the same experimental setup as in Section 4.2.

	LM	AE	Out-of-Domain	
	PPL	BLEU	F1	EM
500x-1B	11.22	93.73	45.76	31.45
+EPL	**10.80**	98.50	**48.03**	32.29
+FGC	11.03	94.49	45.52	30.40
+FGC & EPL	**10.80**	**98.58**	47.54	**32.45**

Table 9: Results of fine-grained compression (Zhang et al., 2025).

EPL consistently enhances FGC performance As shown in Table 9, applying EPL to the FGC yields consistent performance improvements across all evaluation metrics. These results demonstrate that EPL's position layout adjustments provide a generalizable enhancement to existing compression methods.

4.8 Sensitivity Analysis for Compression Ratios

We conduct a sensitivity analysis of compression ratios to evaluate EPL's robustness with experimental settings aligned with Table 4.

As shown in Table 10, the results demonstrate

	LM	AE	Out-of-Domain	
	PPL	BLEU4	F1	EM
ICAE-1B (DPL) *5x*	12.18	31.80	39.95	27.32
ICAE-1B (EPL) *5x*	**11.42**	**95.98**	**43.87**	**29.50**
ICAE-1B (DPL) *15x*	**12.18**	8.12	34.46	23.33
ICAE-1B (EPL) *15x*	12.43	**52.90**	**36.71**	**23.90**
ICAE-1B (DPL) *31x*	**12.30**	6.83	**33.46**	**22.65**
ICAE-1B (EPL) *31x*	12.55	**18.09**	31.63	21.01
ICAE-1B (DPL) *51x*	**12.18**	3.81	29.99	20.39
ICAE-1B (EPL) *51x*	12.55	**10.19**	**31.46**	**21.01**

Table 10: Results of different compression ratios.

that EPL significantly outperforms DPL across compression ratios up to 15x, encompassing the majority of practical application scenarios, although improvement attenuates as the compression ratio goes up. This performance range is particularly relevant, as lossless encoders typically fail to achieve compression ratios exceeding 10x (Kuratov et al., 2025). EPL performance improvement becomes less consistent at higher compression ratios, which we discuss further in the Limitation Section.

5 Analysis

5.1 Training Curve

During pretraining, we observe that the adoption of EPL significantly accelerates the convergence speed of the AE loss. In the 8B-500xCompressor setting, for example, EPL reduces the training steps required to achieve an AE loss of 0.01 from 9.7k to 1.0k steps, while effectively mitigating AE loss fluctuations. This suggests that the prior information that EPL incorporates is well-suited for AE tasks. For more discussion, see Appendix E.

5.2 Attention Visualization

To verify whether the performance improvement is indeed due to more suitable attention patterns for compression tasks idealized in Figure 1, we visualize the summed attention matrices of all attention heads in the second and final layers of the ICAE-3B model (Figure 6 and Figure 7) after the model has been fine-tuned on MRQA tasks.

For the second layer, we observe that the memory token under **UPL** attends to its surrounding[10] context tokens, forming a slope (central top of Figure 6), in contrast to **DPL** that only exhibits self-attention among memory tokens, showing that the attention adheres to our specified prior through EPL

[10] In terms of position IDs, not physical token positions.

even after learning[11]. In the final layer (Figure 7), **UPL** maintains the slope pattern, while **DPL** overcomes the position encoding resistance and attends to distant context tokens through learning without a clear pattern.

5.3 Learnable PEs

Our results rely on the local bias of PEs that are explicitly defined in (Vaswani et al., 2017; Su et al., 2023; Press et al., 2022). We examine in this section empirically the properties of learnable position encodings (Devlin et al., 2019; Raffel et al., 2020).

Figure 9 shows the cosine similarity of BERT's (Devlin et al., 2019) position embeddings where we observe significantly higher similarity between adjacent position embeddings compared to distant tokens. Similarly, the bias values of T5 Bias (Raffel et al., 2020) decay with increasing relative distance as shown in Figure 2, indicating that attention is stronger among close tokens.

BERT's `[CLS]` token (position ID 0) shows a special pattern as shown in Figure 10: in contrast to other tokens, its cosine similarities are not higher with its adjacent tokens. Given that the `[CLS]` token behaves like a compressed memory token trained using the next sentence prediction, the phenomenon motivates our current work, suggesting that different priors should be given to special tokens for best performance.

6 Related Work

Soft Prompt Methods GIST (Mu et al., 2023) trains LLMs with modified attention mechanisms (similar to Figure 8) to compress prompt information into a few gist tokens. AutoCompressor (Chevalier et al., 2023) trains an LLM to recursively compress long prompts by combining compressed tokens with new sub-prompts in each iteration, ultimately collecting all compressed tokens to form a compact representation. ICAE (Ge et al., 2024) introduces an AE task enabling LLMs to pretrain compression capabilities on large-scale corpora, requiring only minimal parameter tuning for the encoder while freezing the decoder. 500xCompressor (Li et al., 2024b) builds upon ICAE by changing the information carrier from memory token outputs to memory tokens' KV values. UniICL (Gao et al., 2024b) and SelfCP (Gao et al., 2024a) freeze both the encoder and decoder and train only a connector module to transform the encoder's output memory tokens into decoder inputs. VoCo-LLaMA (Ye et al., 2025) is the first to use LLMs for compressing visual tokens. Similar to GIST, it compresses visual token information into VoCo tokens through modified attention masks, outperforming methods like Q-Former (Li et al., 2023) and average pooling with linear projection (Li et al., 2024a). None of these methods discuss the impact of position layout and our EPL can be applied to all these soft prompt methods.

Position Layout Although we are not aware of position layout work in the compression domain, we find related work in multimodality by considering memory tokens as another modality. Due to images being two-dimensional and text being one-dimensional, handling mixed image-text positional layout remains an open question. Current mainstream approaches in multimodality flatten 2D images into 1D sequences (Liu et al., 2023b,a; Bavishi et al., 2023; Lu et al., 2024; Sun et al., 2024). Seeking a more elegant solution, Su (2024) proposes RoPE-Tie by placing visual and text tokens along the diagonal $(y = x)$ in 2D space. Although Su (2024) does not thoroughly validate its design, as the approach maintains the sequence order between modalities and the internal locality of images and text respectively, our empirical results suggest that the design can result in better performance. Qwen2-VL (Wang et al., 2024) adopts similar designs in video domains.

7 Conclusion

We examine position layout, an understudied topic in context compression, and propose EPL for soft prompt methods. EPL improves over default position layout by bringing memory tokens close to its context and at the same time maintains the logical sequence token ordering among context, memory tokens and the subsequent tokens. Extensive experiments show EPL enhances compression efficiency and downstream performance across architectures and modalities.

Given LLM's widespread use, we believe examining position layout will benefit problems beyond context compression by adopting a prior tailored to tasks that can be expressed in the position layout. We hope EPL's success fosters research in this underexplored area.

[11]The angle of the slope is around $11°$ ($\arctan(1/\mathbf{5}) = 11.3°$ at a compression rate of **5**), which is consistent with our UPL conception illustrated in Figure 1.

Limitations

Throughout our experiments, we have tested and confirmed EPL's effectiveness under 5x, 15x, 51x compression ratios for text reconstruction and QA tasks and 4.5x compression ratio for visual QA. There remains a question as to whether our method is still effective at high compression ratio. We don't have a positive outlook on this question.

First, EPL, requiring memory tokens to achieve uniform coverage across the context, intrinsically aligns with lossless compression scenarios (i.e., autoencoding tasks) which assume that "all tokens in the context are equally important"; however, under high compression ratio (lossy compression), with information carrier capacity being limited, focus should be placed on important tokens, which violates our "equally important" assumption. Empirically, Figure 10 shows that the BERT's `[CLS]` would have higher attention to some tokens without a clear pattern. Our compression ratio analysis in Table 10 and results in Table 13 on VoCo-LLaMA also show that EPL does not demonstrate any significant gains when used under high compression ratios.

The analysis suggests that high-ratio lossy compression may require a compression mechanism that goes beyond our current work. At a higher level, this suggests that when one adapts position layout methods to different application scenarios, the success can highly depend on whether the conceived layout captures the underlying prior characteristics of specific tasks.

Acknowledgments

This work was supported in part by the National Science Foundation of China (Nos. 62276056 and U24A20334), the Yunnan Fundamental Research Projects (No.202401BC070021), the Yunnan Science and Technology Major Project (No. 202502AD080014), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009).

References

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. 2023. Introducing our multimodal models.

Aydar Bulatov, Yuri Kuratov, and Mikhail Burtsev. 2022. Recurrent memory transformer. In *Advances in Neural Information Processing Systems*.

Kaiyan Chang, Songcheng Xu, Chenglong Wang, Yingfeng Luo, Xiaoqian Liu, Tong Xiao, and Jingbo Zhu. 2024. Efficient prompting methods for large language models: A survey. *Preprint*, arXiv:2404.01077.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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

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

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

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019b. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

Jun Gao, Ziqiang Cao, and Wenjie Li. 2024a. Selfcp: Compressing over-limit prompt via the frozen large language model itself. *Inf. Process. Manag.*, 61:103873.

Jun Gao, Ziqiang Cao, and Wenjie Li. 2024b. Unifying demonstration selection and compression for in-context learning. *Preprint*, arXiv:2405.17062.

Tao Ge, Jing Hu, Li Dong, Shaoguang Mao, Yan Xia, Xun Wang, Si-Qing Chen, and Furu Wei. 2023. Extensible prompts for language models on zero-shot language style customization. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yutong Hu, Quzhe Huang, Mingxu Tao, Chen Zhang, and Yansong Feng. 2024. Can perplexity reflect large language model's ability in long text understanding? In *The Second Tiny Papers Track at ICLR 2024*.

Yichen Jiang, Marco Vecchio, Mohit Bansal, and Anders Johannsen. 2024. Hierarchical and dynamic prompt compression for efficient zero-shot API usage. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2162–2174, St. Julian's, Malta. Association for Computational Linguistics.

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

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. 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)*.

Yuri Kuratov, Mikhail Arkhipov, Aydar Bulatov, and Mikhail Burtsev. 2025. Cramming 1568 tokens into a single vector and back again: Exploring the limits of embedding space capacity. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19323–19339, Vienna, Austria. Association for Computational Linguistics.

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. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: bootstrapping language-image pretraining with frozen image encoders and large language models. In *ICML*.

Yanwei Li, Chengyao Wang, and Jiaya Jia. 2024a. Llama-vid: An image is worth 2 tokens in large language models.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. 2025. Prompt compression for large language models: A survey. In *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)*, pages 7182–7195, Albuquerque, New Mexico. Association for Computational Linguistics.

Zongqian Li, Yixuan Su, and Nigel Collier. 2024b. 500xcompressor: Generalized prompt compression for large language models. *Preprint*, arXiv:2408.03094.

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.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning.

Xinyu Liu, Runsong Zhao, Pengcheng Huang, Chunyang Xiao, Bei Li, Jingang Wang, Tong Xiao, and JingBo Zhu. 2024a. Forgetting curve: A reliable method for evaluating memorization capability for long-context models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4667–4682, Miami, Florida, USA. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Yike Yuan, Wangbo Zhao, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024b. MMBench: Is your multi-modal model an all-around player?

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. 2024. Deepseek-vl: Towards real-world vision-language understanding. *Preprint*, arXiv:2403.05525.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Yingfeng Luo, Tong Zheng, Yongyu Mu, Bei Li, Qinghong Zhang, Yongqi Gao, Ziqiang Xu, Peinan Feng, Xiaoqian Liu, Tong Xiao, and JingBo Zhu. 2025. Beyond decoder-only: Large language models can be good encoders for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9399–9431, Vienna, Austria. Association for Computational Linguistics.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. *Preprint*, arXiv:2108.12409.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693, Melbourne, Australia. Association for Computational Linguistics.

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

Jianlin Su. 2024. The evolution path of transformer: 17. simple thoughts on multimodal positional encoding.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. 2024. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14398–14409.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

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, and 3 others. 2015.

An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(1):138.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yang Xu, Yunlong Feng, Honglin Mu, Yutai Hou, Yitong Li, Xinghao Wang, Wanjun Zhong, Zhongyang Li, Dandan Tu, Qingfu Zhu, Min Zhang, and Wanxiang Che. 2024. Concise and precise context compression for tool-using language models. *Preprint*, arXiv:2407.02043.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. 2025. Voco-llama: Towards vision compression with large language models. *Preprint*, arXiv:2406.12275.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2025. Long context compression with activation beacon. In *The Thirteenth International Conference on Learning Representations*.

A Detailed Position Layout

Detailed position layouts are provided in this section. Table 14 illustrates the position layout for LLMs without compression under standard conditions, reflecting natural language priors. Table 15 details the encoder's position layout. For the decoder, position layouts are described in Tables 16, 17, and 18, with slight variations depending on the specific task (AE, LM, or QA). Each of these tables (Tables 15, 16, 17, and 18) includes a formula at the top representing the generalized layout. Below the formula, an illustrative example is given using a context length $|C| = p = 1020$, chunk size $L = 510$, compression ratio $r = 5$, total sequence length $|X| = 2040$, question length $|Q| = 50$, and answer length $|A| = 5$.

B Overview of ICAE

ICAE (Ge et al., 2024) is an autoencoder framework to compress long contexts into short compact memory slots. The method operates by concatenating designated memory tokens to the end of the input sequence before an encoder processes the entire combined sequence. Subsequently, a decoder reconstructs the original sequence using only the information contained within the memory tokens. ICAE is trained in two main phases. It is first pretrained on massive text data using a combination of autoencoding and language modeling objectives, enabling it to generate memory slots that represent the original context. Following pretraining, the model is fine-tuned on instruction data for the purpose of producing desirable responses to various prompts. An overview of the ICAE framework is shown in Figure 3.

C Hyperparameters

For the 1B and 3B models, we perform continued pre-training on sequences with lengths $|X|$ ranging from 510 to 2040. For the 8B model, the input sequence length $|X|$ ranges from 510 to 4080 during continued pre-training. We take the first $p = \lfloor |X|/2 \rfloor$ tokens as the context. Additional hyperparameters are listed in Table 11.

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

Table 11: Hyperparameters for training

D Detailed Results

In this section, we provide detailed results. Considering the potential risk of data leakage where LLMs may have encountered context information from evaluation datasets during the pretraining phase(Li et al., 2024b), we also report on **NoContext** and **FullContext** settings where **NoContext** performs inference solely based on the question and **FullContext** utilize the complete context and the question for inference. In both cases, we only train `[AE]` and `[LM]` tokens to guide the model executing corresponding tasks.

Table 12 fully presents the performance of ICAE and 500xCompressor of different scales on LM tasks and AE tasks. Table 19 and Table 20 respectively show all results of these models in-domain and out-of-domain in MRQA. Ablation results in-domain and out-of-domain in MRQA are presented in Table 21 and Table 22, respectively. Table 13 shows all results of VoCo-LLaMA on multimodal benchmarks.

	PPL(AE)	**PPL(LM)**	**BLEU(AE)**
Llama-3.2-1B			
NoContext	11.56	13.25	0.00
ICAE(DPL)	1.40	12.18	31.80
ICAE(EPL)	**1.04**	**11.42**	**95.98**
500x(DPL)	1.04	11.22	93.73
500x(EPL)	**1.01**	**10.80**	**98.50**
FullContext	1.02	9.90	33.49
Llama-3.2-3B			
NoContext	9.58	11.02	0.00
ICAE(DPL)	1.49	10.33	48.12
ICAE(EPL)	**1.02**	**9.07**	**97.05**
500x(DPL)	1.02	9.44	96.86
500x(EPL)	**1.00**	**8.68**	**99.34**
FullContext	1.06	8.25	60.16
Llama-3.1-8B			
NoContext	7.79	9.02	0.00
ICAE(DPL)	1.58	9.22	49.08
ICAE(EPL)	**1.00**	**7.61**	**98.82**
500x(DPL)	1.01	7.61	97.68
500x(EPL)	**1.00**	**7.40**	**99.49**
FullContext	1.00	7.30	98.00

Table 12: Perplexity of X_{context} and $X_{\text{completion}}$, and BLEU score for the reconstruction quality of X_{context}, calculated on a held-out set of 1k examples.

	voco_num	SQA(OOD)	MMB(OOD)	VQAv2(ID)
Lower Bound*	1	60.7	22.3	41.2
DPL*	2	-	60.1	73.5
DPL	2	66.1	**61.5**	**73.7**
EPL	2	**67.6**	61.4	69.5
DPL*	128	-	61.0	76.9
DPL	128	64.9	59.9	76.9
EPL	128	**66.5**	**64.6**	**78.4**
Upper Bound*	576	66.5	64.0	77.7

Table 13: Results of VoCo-LLaMA on SQA (Lu et al., 2022), MMB (Liu et al., 2024b), and VQAv2 (Goyal et al., 2017) from our experiments. Results marked with * are from (Ye et al., 2025).

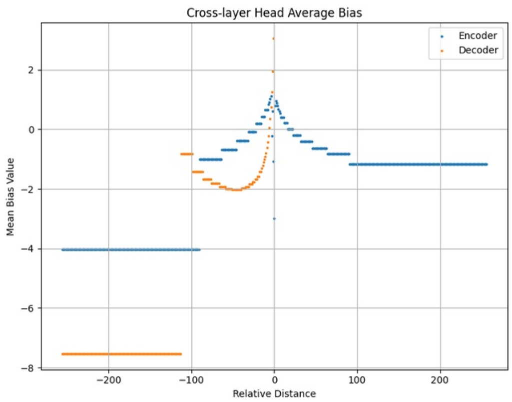

Figure 2: T5 bias.

E Training Curves

This section provides training curves. Figure 4 and Figure 5 respectively show the training curves of ICAE and 500xCompressor of different scales during pretraining. For AE loss, it can be observed that ICAE(DPL) struggles to decrease to 0, while 500xCompressor(DPL) requires a period of oscillation before converging near 0. When UPL is applied, their AE loss rapidly converges to around 0.

F VoCo-LLaMA

VoCo-LLaMA (Ye et al., 2025) employs a modified attention mask which restricts text tokens from attending to vision tokens. Figure 8 illustrates this mask, and we have indicated its position layout at the top of the figure.

G Attention Visualization

Attention maps are presented in this section. Due to the low attention values of memory tokens in the first layer (which appear almost empty in the figure), we present the attention map of the second layer. See Figure 6. We provide a magnified view of the self-attention of memory tokens and their attention to other tokens. It can be observed that in the second layer, memory tokens in DPL only attend to themselves, while in UPL, they are able to attend to the entire context. Additionally, we also present the attention map of the last layer, shown in Figure 7. We use grey dashed lines to indicate the special attention pattern of UPL.

H Local bias of Position Encodings

H.1 Sinusoidal Position Encoding

The sinusoidal position encoding (Vaswani et al., 2017; Su et al., 2023) is given by:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

Consider two nearby positions, pos and $pos + \delta$, where δ is a small number.

For any dimension i, the argument to the sine/cosine function changes from $\frac{pos}{10000^{2i/d_{model}}}$ to $\frac{pos+\delta}{10000^{2i/d_{model}}}$. The change in the argument is $\frac{\delta}{10000^{2i/d_{model}}}$.

For small δ, this change in the argument is small for all dimensions.

Since sine and cosine functions are continuous, a small change in their input argument results in a small change in their output value.

$$\sin(x+\epsilon) \approx \sin(x) \quad \text{for small } \epsilon$$

$$\cos(x+\epsilon) \approx \cos(x) \quad \text{for small } \epsilon$$

Therefore, each dimension of $PE_{(pos+\delta)}$ is very close to the corresponding dimension of $PE_{(pos)}$.

This means the entire vector $PE_{(pos+\delta)}$ is very similar to $PE_{(pos)}$ when δ is small.

This property injects a local inductive bias.

H.2 Learnable Position Encodings

In this section, we present figures related to learnable position embeddings. Figure 9 illustrates the cosine similarity between different positions of BERT (Devlin et al., 2019). It can be observed that the cosine similarity between nearby positions is significantly high. To illustrate the special behavior of the [CLS] token's position embedding, we show in Figure 10 the cosine similarity of position 0 (i.e., the position ID of [CLS]), position 100, position 200, and position 300 with other positions. Figure 2 illustrates that as the relative distance increases, the learnable bias added by T5 (Raffel et al., 2020) to the attention scores decreases.

	x_0	x_1	...	$x_{\lvert X\rvert-1}$	y_0	y_1	...	$y_{\lvert Y\rvert-1}$
Default Position ID	0	1	...	$\lvert X\rvert-1$	$\lvert X\rvert$	$\lvert X\rvert+1$	...	$\lvert X\rvert+\lvert Y\rvert-1$
	$KV(x_0)$	$KV(x_1)$	...	$KV(x_{\lvert X\rvert-1})$	y_0	y_1	...	$y_{\lvert Y\rvert-1}$
Default Position ID	0	1	...	$\lvert X\rvert-1$	$\lvert X\rvert$	$\lvert X\rvert+1$	...	$\lvert X\rvert+\lvert Y\rvert-1$

Table 14: Default position layout of Transformers.

	$x_{(i-1)L}$	$x_{(i-1)L+1}$	...	x_{iL-1}	m_0	m_1	...	$m_{\lvert M\rvert-1}$
DPL	0	1	...	$L-1$	L	$L+1$	...	$L+\lvert M\rvert-1$
EPL	$(i-1)L+1$	$(i-1)L+2$	...	iL	$\lfloor b\rceil$	$\lfloor b+r\rceil$	...	$\lfloor(b+(\lvert M\rvert-1)r\rceil$
$i=1, L=510, \lvert M\rvert=102, r=5$								
	x_0	x_1	...	x_{509}	m_0	m_1	...	m_{101}
DPL	0	1	...	509	510	511	...	611
EPL	1	2	...	510	3	8	...	508
$i=2, L=510, \lvert M\rvert=102, r=5$								
	x_{510}	x_{511}	...	x_{1019}	m_0	m_1	...	m_{101}
DPL	0	1	...	509	510	511	...	611
EPL	511	512	...	1020	513	518	...	1018

Table 15: Position Layout of Encoder for $S^{(i)}$. $r=\frac{L}{\lvert M\rvert}$; $b=(i-1)*L+1+\frac{r-1}{2}$. The notation $\lfloor\cdot\rceil$ indicates rounding to the nearest integer.

	$\tilde{m}_0^{(1)}/KV_0^{(1)}$	$\tilde{m}_1^{(1)}/KV_1^{(1)}$	...	$\tilde{m}_{\lvert M\rvert-1}^{(k)}/KV_{\lvert M\rvert-1}^{(k)}$	[AE]	x_0	x_1	...	x_{p-1}
DPL(ICAE)	0	1	...	$k\lvert M\rvert-1$	$k\lvert M\rvert$	$k\lvert M\rvert+1$	$k\lvert M\rvert+2$	...	$k\lvert M\rvert+p$
DPL(500x)	L	$L+1$	...	$L+\lvert M\rvert-1$	$k\lvert M\rvert$	$k\lvert M\rvert+1$	$k\lvert M\rvert+2$	...	$k\lvert M\rvert+p$
EPL	$\lfloor b\rceil$	$\lfloor b+r\rceil$	...	$\lfloor b+(k\lvert M\rvert-1)r\rceil$	0	1	2	...	p
$k=2, L=510, \lvert M\rvert=102, r=5, p=1020$									
	$\tilde{m}_0^{(1)}/KV_0^{(1)}$	$\tilde{m}_1^{(1)}/KV_1^{(1)}$	...	$\tilde{m}_{101}^{(2)}/KV_{101}^{(2)}$	[AE]	x_0	x_1	...	x_{1019}
DPL(ICAE)	0	1	...	203	204	205	206	...	1224
DPL(500x)	510	511	...	611	204	205	206	...	1224
EPL	3	8	...	1018	0	1	2	...	1020

Table 16: Position Layout of Decoder in AE Task. $r=\frac{L}{\lvert M\rvert}$; $b=1+\frac{r-1}{2}$. The notation $\lfloor\cdot\rceil$ indicates rounding to the nearest integer.

	$\tilde{m}_0^{(1)}/KV_0^{(1)}$	$\tilde{m}_1^{(1)}/KV_1^{(1)}$	...	$\tilde{m}_{\lvert M\rvert-1}^{(k)}/KV_{\lvert M\rvert-1}^{(k)}$	[LM]	x_p	x_{p+1}	...	$x_{\lvert X\rvert-1}$
DPL(ICAE)	0	1	...	$k\lvert M\rvert-1$	$k\lvert M\rvert$	$k\lvert M\rvert+1$	$k\lvert M\rvert+2$	...	$k\lvert M\rvert+\lvert X\rvert-p$
DPL(500x)	L	$L+1$	...	$L+\lvert M\rvert-1$	$k\lvert M\rvert$	$k\lvert M\rvert+1$	$k\lvert M\rvert+2$	...	$k\lvert M\rvert+\lvert X\rvert-p$
EPL	$\lfloor b\rceil$	$\lfloor b+r\rceil$	...	$\lfloor b+(k\lvert M\rvert-1)r\rceil$	p	$p+1$	$p+2$	...	$\lvert X\rvert$
$k=2, L=510, \lvert M\rvert=102, r=5, \lvert X\rvert=2040, p=1020$									
	$\tilde{m}_0^{(1)}/KV_0^{(1)}$	$\tilde{m}_1^{(1)}/KV_1^{(1)}$	...	$\tilde{m}_{101}^{(2)}/KV_{101}^{(2)}$	[LM]	x_{1020}	x_{1021}	...	x_{2039}
DPL(ICAE)	0	1	...	203	204	205	206	...	1224
DPL(500x)	510	511	...	611	204	205	206	...	1224
EPL	3	8	...	1018	1020	1021	1022	...	2040

Table 17: Position Layout of Decoder in LM Task. $r=\frac{L}{\lvert M\rvert}$; $b=1+\frac{r-1}{2}$. The notation $\lfloor\cdot\rceil$ indicates rounding to the nearest integer.

	$\tilde{m}_0^{(1)}/KV_0^{(1)}$	$\tilde{m}_1^{(1)}/KV_1^{(1)}$	...	$\tilde{m}_{\lvert M\rvert-1}^{(k)}/KV_{\lvert M\rvert-1}^{(k)}$	[LM]	q_0	...	$q_{\lvert Q\rvert-1}$	a_0	...	$a_{\lvert A\rvert-1}$
DPL(ICAE)	0	1	...	$k\lvert M\rvert-1$	$k\lvert M\rvert$	$k\lvert M\rvert+1$	...	t	$t+1$	...	$t+\lvert A\rvert$
DPL(500x)	L	$L+1$	...	$L+\lvert M\rvert-1$	$k\lvert M\rvert$	$k\lvert M\rvert+1$	...	t	$t+1$	...	$t+\lvert A\rvert$
EPL	$\lfloor b\rceil$	$\lfloor b+r\rceil$	...	$\lfloor b+(k\lvert M\rvert-1)r\rceil$	$\lvert C\rvert$	$\lvert C\rvert+1$	...	t'	$t'+1$	...	$t'+\lvert A\rvert$
$k=2, L=510, \lvert M\rvert=102, r=5, \lvert C\rvert=1020, \lvert Q\rvert=50, \lvert A\rvert=5$											
	$\tilde{m}_0^{(1)}/KV_0^{(1)}$	$\tilde{m}_1^{(1)}/KV_1^{(1)}$	...	$\tilde{m}_{101}^{(2)}/KV_{101}^{(2)}$	[LM]	q_0	...	q_{49}	a_0	...	a_4
DPL(ICAE)	0	1	...	203	204	205	...	254	255	...	259
DPL(500x)	510	511	...	611	204	205	...	254	255	...	259
EPL	3	8	...	1018	1020	1021	...	1070	1071	...	1075

Table 18: Position Layout of Decoder in QA Task. $r=\frac{L}{\lvert M\rvert}$; $b=1+\frac{r-1}{2}$; $t=k\lvert M\rvert+\lvert Q\rvert$; $t'=\lvert C\rvert+\lvert Q\rvert$. The notation $\lfloor\cdot\rceil$ indicates rounding to the nearest integer.

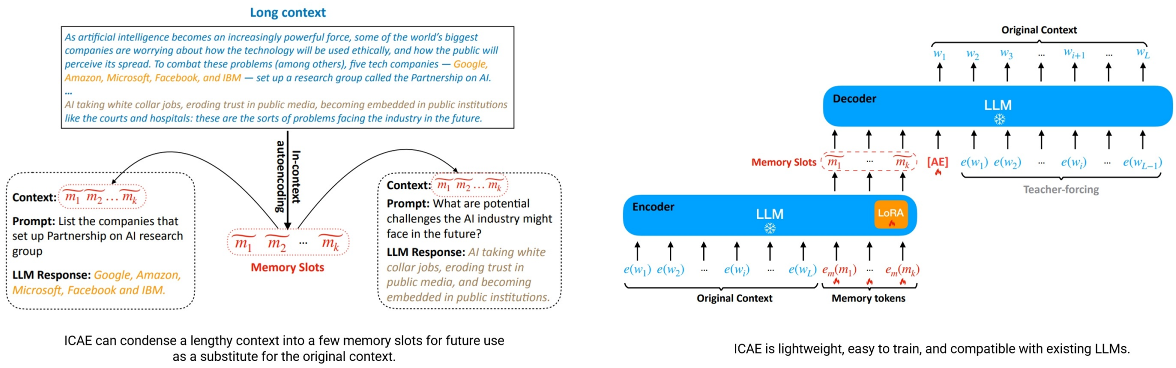

Figure 3: Overview of the ICAE framework proposed by (Ge et al., 2024). Source: ICAE official repository (CC0-1.0).

	SQuAD		NewsQA		TriQA		SearchQA		HQA		NQ		Avg.	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Llama-3.2-1B														
NoContext	9.34	1.92	4.80	0.59	3.04	0.92	14.99	8.71	9.47	3.27	10.88	3.91	8.75	3.22
ICAE(DPL)	54.58	36.99	37.57	21.25	57.96	48.85	68.90	56.68	59.61	43.01	59.40	42.01	56.34	41.47
ICAE(EPL)	**59.94**	**39.39**	**43.61**	**24.95**	**61.50**	**51.98**	**69.31**	**57.11**	**64.22**	**46.84**	**61.21**	**42.11**	**59.97**	**43.73**
500x(DPL)	**68.26**	**49.15**	43.17	25.31	60.30	51.03	70.33	57.96	66.44	49.45	64.69	46.78	62.20	46.61
500x(EPL)	67.87	47.54	**49.23**	**29.58**	**64.53**	**55.16**	**72.51**	**60.39**	**68.51**	**51.06**	**65.55**	**46.97**	**64.70**	**48.45**
FullContext	58.72	39.20	38.72	16.62	31.82	24.42	49.15	36.27	53.22	39.08	53.47	36.12	47.51	31.95
Llama-3.2-3B														
NoContext	15.65	7.09	6.15	1.26	8.68	6.38	42.54	31.83	16.31	9.02	16.43	8.18	17.63	10.63
ICAE(DPL)	58.95	41.53	39.70	23.12	60.49	51.64	66.35	53.81	58.68	42.55	58.55	41.94	57.12	42.43
ICAE(EPL)	**73.82**	**53.67**	**58.02**	**36.51**	**71.22**	**62.26**	**71.41**	**59.90**	**72.61**	**55.77**	**70.19**	**51.33**	**69.55**	**53.24**
500x(DPL)	68.28	50.15	50.98	32.15	68.10	59.54	74.16	62.11	70.63	53.57	67.58	50.19	66.62	51.28
500x(EPL)	**77.74**	**58.38**	**61.19**	**41.62**	**71.62**	**62.66**	**74.32**	**62.67**	**75.06**	**58.52**	**72.03**	**54.07**	**71.99**	**56.32**
FullContext	74.07	55.43	48.99	24.62	63.92	54.05	67.69	52.31	63.73	48.33	64.32	46.54	63.79	46.88
Llama-3.1-8B														
NoContext	21.84	11.74	9.52	3.28	31.58	26.37	59.66	45.49	20.34	12.71	29.62	17.94	28.76	19.59
ICAE(DPL)	56.56	38.87	36.99	20.06	64.54	55.58	71.35	58.92	57.04	41.20	58.04	41.19	57.42	42.64
ICAE(EPL)	**78.44**	**58.90**	**61.69**	**40.17**	**73.92**	**65.05**	**80.05**	**67.65**	**74.93**	**58.33**	**72.43**	**54.14**	**73.58**	**57.37**
500x(DPL)	80.56	**62.11**	60.31	40.65	74.00	65.39	**79.60**	67.51	76.18	59.62	74.17	**56.48**	74.14	58.63
500x(EPL)	**80.60**	61.37	**64.43**	**44.35**	**74.75**	**66.04**	79.39	**67.74**	**77.17**	**60.53**	**74.71**	56.16	**75.18**	**59.36**
FullContext	80.53	61.97	60.24	40.05	72.65	63.28	76.54	61.99	73.07	57.19	72.25	54.01	72.55	56.42

Table 19: Results on the in-domain validation set, including six QA datasets: SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), TriviaQA (TriQA) (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (HQA) (Yang et al., 2018), and NaturalQuestions (NQ) (Kwiatkowski et al., 2019).

	BioASQ		DROP		DuoRC		RACE		RE		TQA		Avg.	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Llama-3.2-1B														
NoContext	10.63	4.52	17.72	10.18	3.93	0.53	6.58	0.30	11.74	3.70	20.50	9.78	11.85	4.83
ICAE(DPL)	38.39	26.99	**37.13**	**27.28**	27.90	18.32	21.67	4.30	**76.98**	**64.45**	37.62	22.55	39.95	27.32
ICAE(EPL)	**42.66**	**29.19**	37.03	26.95	**32.50**	**21.39**	**26.87**	**5.34**	76.90	64.31	**47.25**	**29.81**	**43.87**	**29.50**
500x(DPL)	44.22	31.38	40.47	29.34	32.20	21.19	27.67	5.93	**83.03**	**71.98**	47.00	28.88	45.76	31.45
500x(EPL)	**46.11**	**31.91**	**40.94**	**29.41**	**39.07**	**26.32**	**31.89**	**7.27**	80.39	67.71	**49.80**	**31.14**	**48.03**	**32.29**
FullContext	45.72	31.45	40.74	30.27	39.20	27.85	29.57	7.72	74.85	64.01	53.31	34.60	47.23	32.65
Llama-3.2-3B														
NoContext	24.84	17.22	21.17	13.37	5.87	1.93	8.87	1.19	20.60	13.30	37.34	22.82	19.78	11.64
ICAE(DPL)	41.75	31.65	38.77	29.81	28.40	18.99	21.58	3.56	76.56	64.82	47.85	30.41	42.49	29.87
ICAE(EPL)	**50.83**	**36.70**	**54.10**	**43.38**	**45.66**	**33.24**	**38.78**	**9.35**	**82.71**	**71.64**	**58.13**	**36.93**	**55.03**	**38.54**
500x(DPL)	50.20	37.63	48.64	38.32	38.77	26.85	31.86	8.16	83.67	73.71	55.08	34.80	51.37	36.58
500x(EPL)	**53.15**	**39.03**	**57.00**	**45.84**	**50.35**	**36.51**	**42.45**	**10.83**	**85.20**	**75.20**	**58.09**	**35.53**	**57.71**	**40.49**
FullContext	59.20	42.82	50.30	35.53	39.09	27.51	36.72	9.20	81.73	73.27	67.50	43.65	55.76	38.66
Llama-3.2-8B														
NoContext	43.16	33.64	25.60	18.43	6.15	2.33	7.98	1.63	33.89	25.27	50.27	32.67	27.84	19.00
ICAE(DPL)	47.42	33.98	35.98	27.41	22.54	13.99	21.48	4.60	77.58	65.98	53.55	34.13	43.09	30.01
ICAE(EPL)	**53.21**	**37.90**	**58.83**	**47.50**	**47.26**	**33.44**	**40.65**	**9.64**	**84.35**	**73.91**	**58.39**	**36.13**	**57.12**	**39.76**
500x(DPL)	51.50	37.70	59.85	47.64	47.95	34.44	41.17	10.39	**85.86**	**75.85**	58.22	35.93	57.42	40.32
500x(EPL)	**54.80**	**39.43**	**62.46**	**50.70**	**51.33**	**37.38**	**43.40**	**11.13**	85.77	75.10	58.11	35.00	**59.31**	**41.45**
FullContext	59.80	42.95	60.64	47.64	26.91	16.86	42.80	10.53	85.11	74.08	71.68	47.57	57.83	39.94

Table 20: Results on the out-of-domain validation set, including six QA 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).

	SQuAD		NewsQA		TriQA		SearchQA		HQA		NQ		Avg.	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Llama-3.2-1B														
ICAE(DPL)	54.58	36.99	37.57	21.25	57.96	48.85	68.90	56.68	59.61	43.01	59.40	42.01	56.34	41.47
ICAE(UPL)	**62.22**	**42.25**	**44.49**	**26.16**	61.35	51.88	**70.33**	**57.73**	63.85	**46.89**	**61.71**	**43.00**	**60.66**	**44.65**
ICAE(EPL)	59.94	39.39	43.61	24.95	**61.50**	**51.98**	69.31	57.11	**64.22**	46.84	61.21	42.11	59.97	43.73
500x(DPL)	**68.26**	**49.15**	43.17	25.31	60.30	51.03	70.33	57.96	66.44	49.45	64.69	46.78	62.20	46.61
500x(UPL)	65.92	46.35	**49.33**	**29.89**	63.97	54.98	71.85	59.77	67.04	50.53	64.20	45.66	63.72	47.86
500x(EPL)	67.87	47.54	49.23	29.58	**64.53**	**55.16**	**72.51**	**60.39**	**68.51**	**51.06**	**65.55**	**46.97**	**64.70**	**48.45**

Table 21: Ablation results on the in-domain validation set, including six QA datasets: SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), TriviaQA (TriQA) (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (HQA) (Yang et al., 2018), and NaturalQuestions (NQ) (Kwiatkowski et al., 2019).

	BioASQ		DROP		DuoRC		RACE		RE		TQA		Avg.	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Llama-3.2-1B														
ICAE(DPL)	38.39	26.99	37.13	27.28	27.90	18.32	21.67	4.30	**76.98**	**64.45**	37.62	22.55	39.95	27.32
ICAE(UPL)	41.82	**29.32**	**37.77**	**27.41**	**33.73**	**22.45**	**26.90**	4.90	73.61	60.62	**48.19**	**29.94**	43.67	29.11
ICAE(EPL)	**42.66**	29.19	37.03	26.95	26.87	21.39	26.87	**5.34**	76.90	64.31	47.25	29.81	**43.87**	**29.50**
500x(DPL)	44.22	31.38	40.47	29.34	32.20	21.19	27.67	5.93	**83.03**	**71.98**	47.00	28.88	45.76	31.45
500x(UPL)	45.71	**32.38**	40.19	28.81	37.66	26.12	29.02	6.08	78.22	64.55	**50.19**	**31.87**	46.83	31.64
500x(EPL)	**46.11**	31.91	**40.94**	**29.41**	**39.07**	**26.32**	**31.89**	**7.27**	80.39	67.71	49.80	31.14	**48.03**	**32.29**

Table 22: Ablation results on the out-of-domain validation set, including six QA 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).

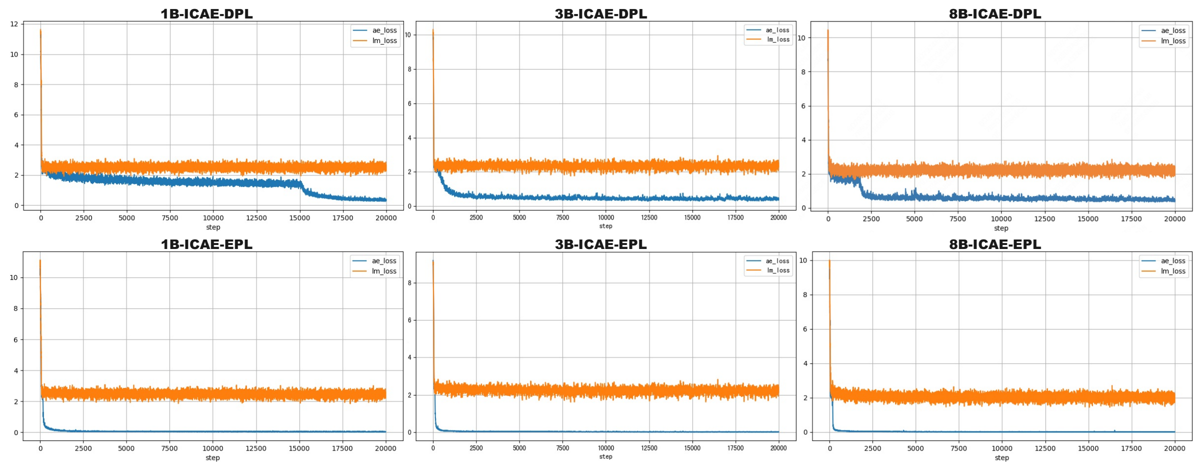

Figure 4: Training Loss of ICAE.

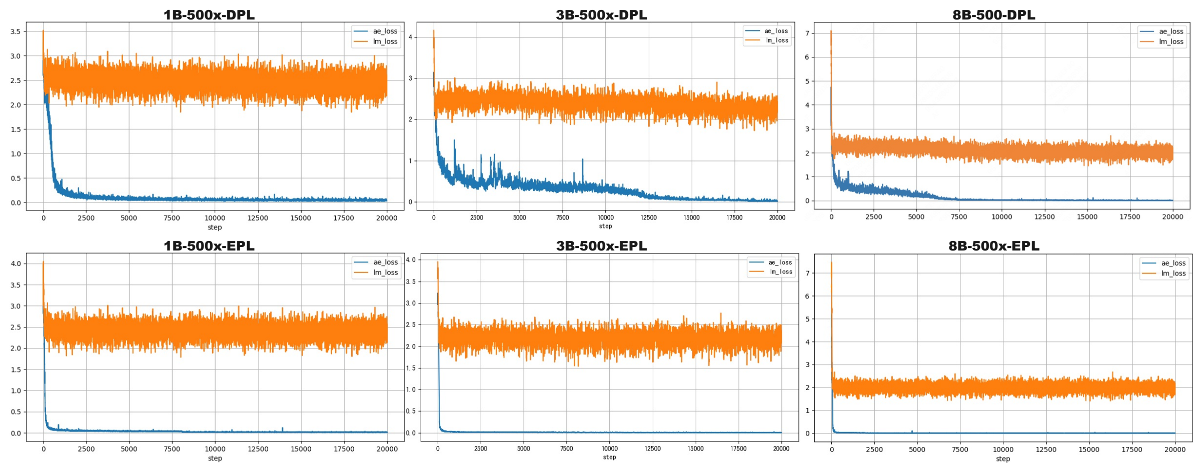

Figure 5: Training Loss of 500xCompressor.

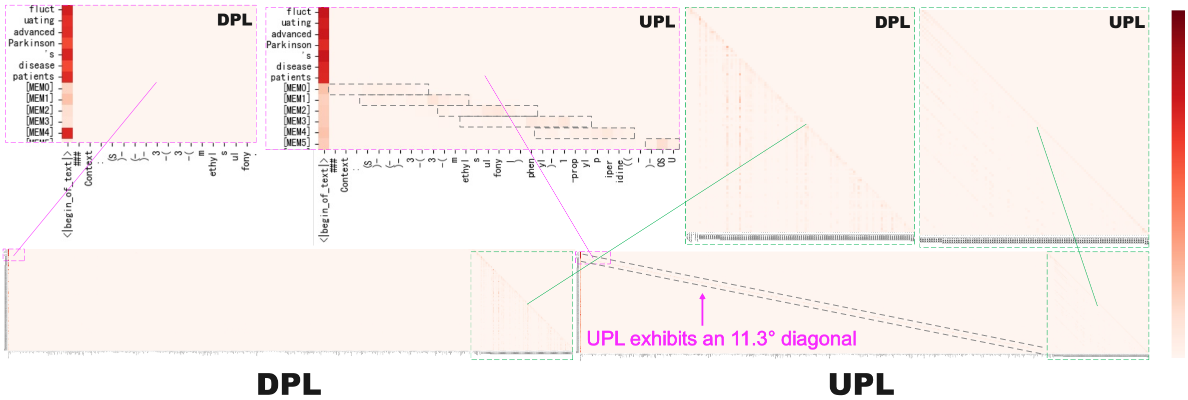

Figure 6: Attention Matrix in the 2nd layer of ICAE after MRQA fine-tuning.

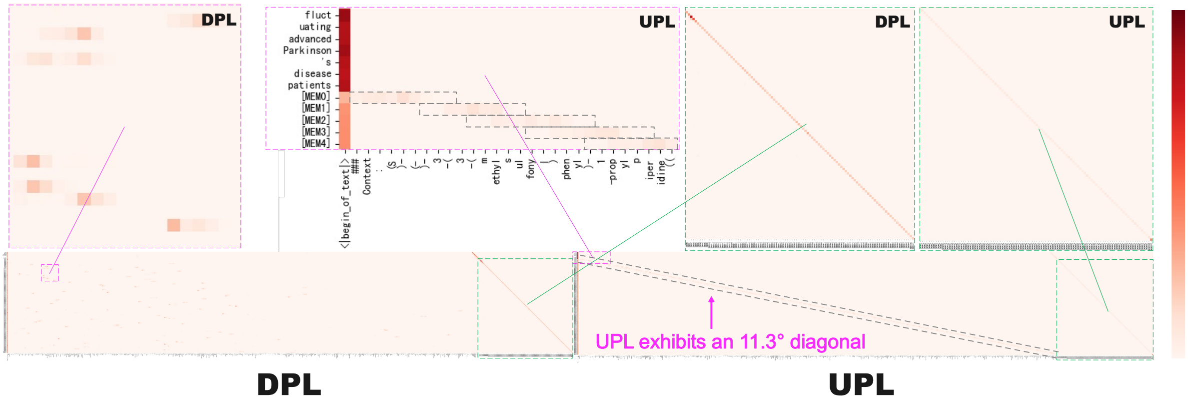

Figure 7: Attention Matrix in the last layer of ICAE after MRQA fine-tuning.

DPL	0	1	2	3	4	5	6	7	8	9	10	11	12	13
EPL	0	1	2	3	4	5	1	4	6	7	8	9	10	11
	vt_0	vt_1	vt_2	vt_3	vt_4	vt_5	voco	voco	q_0	q_1	q_2	q_3	a_0	a_1
vt_0	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
vt_1	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
vt_2	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
vt_3	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
vt_4	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
vt_5	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
voco	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
voco	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
q_0	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗
q_1	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗
q_2	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗
q_3	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✗	✗
a_0	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✗
a_1	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓

Figure 8: Attention mask and position layout of VoCo-LLaMA (Ye et al., 2025). The Position IDs modified by EPL are marked in green (UPL) and blue (CPL) on top of the figure.

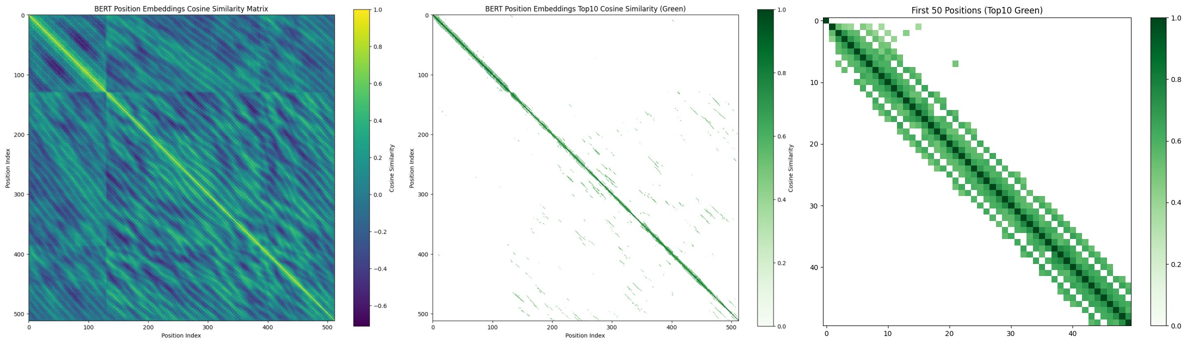

Figure 9: The cosine similarity of BERT's positional encodings. To improve readability, we have removed results beyond the top 10 and zoomed in on the first 50 positions.

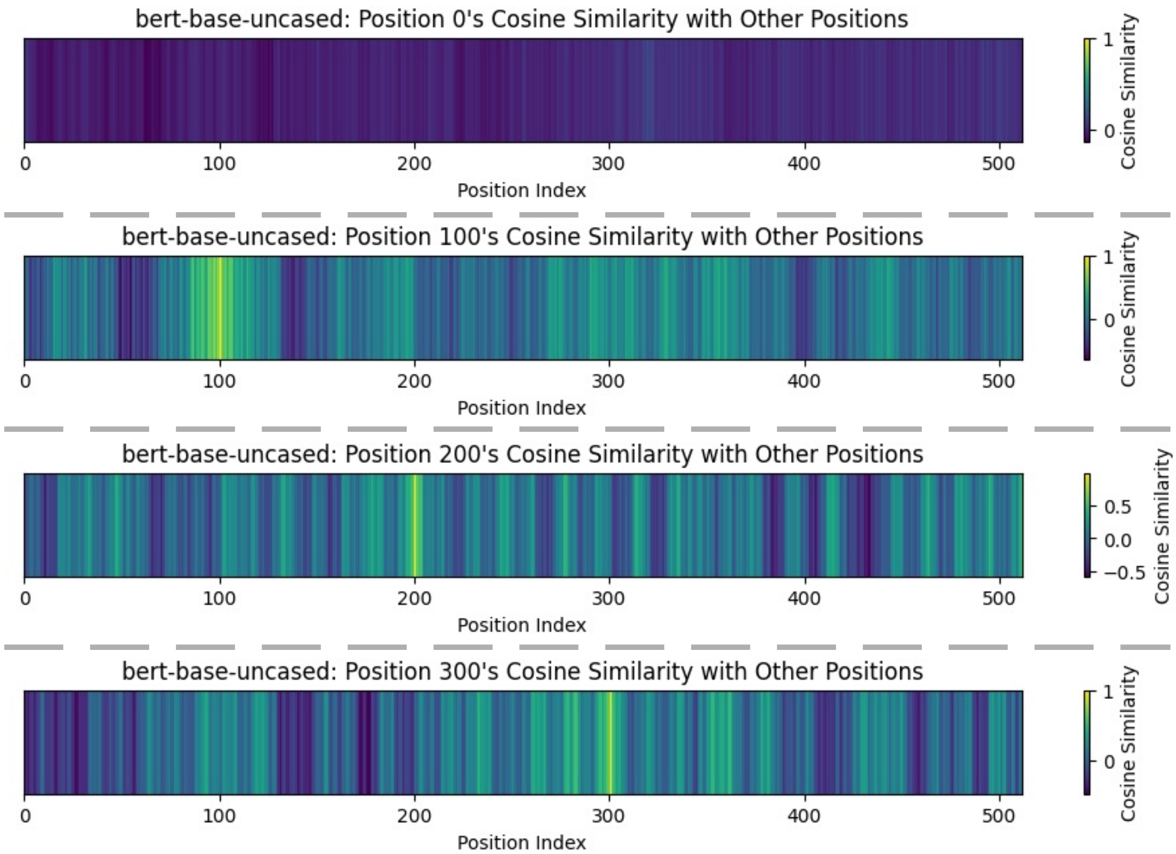

Figure 10: The cosine similarity between the positional encoding of the `[CLS]` token and other positions.