Better Prompt Compression Without Multi-Layer Perceptrons

Edouardo Honig[1]
e.honig@ucla.edu

Andrew Lizarraga[1]
andrewlizarraga@g.ucla.edu

Zijun Frank Zhang[2]
fzhang@natera.com

Ying Nian Wu[1]
ywu@stat.ucla.edu

[1]University of California, Los Angeles: Department of Statistics & Data Science
[2]Natera

Abstract

Prompt compression is a promising approach to speeding up language model inference without altering the generative model. Prior works compress prompts into smaller sequences of learned tokens using an encoder that is trained as a Low-Rank Adaptation (LoRA) of the inference language model. However, we show that the encoder does not need to keep the original language model's architecture to achieve useful compression. We introduce the Attention-Only Compressor (AOC), which learns a prompt compression encoder after removing the multi-layer perceptron (MLP) layers in the Transformer blocks of a language model, resulting in an encoder with roughly 67% less parameters compared to the original model. Intriguingly we find that, across a range of compression ratios up to $480\times$, AOC can better regenerate prompts and outperform a baseline compression encoder that is a LoRA of the inference language model without removing MLP layers. These results demonstrate that the architecture of prompt compression encoders does not need to be identical to that of the original decoder language model, paving the way for further research into architectures and approaches for prompt compression.

1 Introduction

Large language models (LLMs) display incredible usefulness across many natural language tasks, and generally have increased utility with increasingly long and complex prompts [Agarwal et al., 2024, Bertsch et al., 2024]. The downside of lengthier prompts is increased computational load and response time, motivating research into compressing prompts into a smaller number of tokens, known as prompt compression.

While some methods focus on compressing prompts by pruning information in the prompt/text space [Li et al., 2023, Jiang et al., 2023a,b], one can also consider compressing prompts into a lower dimensional latent space [Wingate et al., 2022, Mu et al., 2023, Chevalier et al., 2023]. The In-context Autoencoder (ICAE) [Ge et al., 2024] exemplifies this approach by training a LLM encoder to compress prompts into a shorter sequence of learned memory tokens and uses a learned **[AE]** autoencoder token for decoding the original prompt. This latent representation retains the information of the prompt and is used with the original frozen (meaning not further trained) LLM decoder to reduce the number of tokens at inference time. 500xCompressor [Li et al., 2024] works similarly, but compresses prompts into neural attention [Vaswani et al., 2017] key-value pairs instead of explicit tokens, and uses a pretrained **[BOS]** token instead of a learned **[AE]** token. Notably, both ICAE and 500xCompresson use Low-Rank Adaptation (LoRA) [Hu et al., 2021] to train encoders from the

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

frozen decoder LLM used for inference, which requires more computational resources to perform compression than may be necessary.

We demonstrate that using the entire decoder LLM as an encoder is unnecessary and introduce an alternative in the Attention-Only Compressor (AOC). Instead of learning the encoder as a LoRA of the decoder LLM, we first remove the multi-layer perceptron (MLP) layers before training the entire encoder. By removing MLPs, AOC's prompt compression encoder has roughly 67% less parameters compared to previous methods' encoders, while improving or maintaining similar compression ability. These results emphasize that prompt compression encoders do not need identical architecture to their decoders and that there exist compression models that with higher performance and lower inference-time computational requirements compared to recent approaches using frozen-LLM-based compressors.

Our contributions can be summarized as follows:

- We introduce the Attention-Only Compressor (AOC), a novel prompt compression encoder that removes the MLP layers from a LLM, resulting in an encoder that performs comparably to baseline compression encoders that are roughly three times larger.
- Preliminary experimental results on regeneration demonstrate that compression encoders do not need architecture identical to their decoders, which motivates further research into more efficient compressors.
- To further study compression encoders, we present examples of interpolating between the embeddings of two compressed prompts, showcasing a novel classifier-free approach to merging separate prompts and understanding the latent space of compressed prompts.

2 Methods

Model. Our proposed model consists of a learned prompt compression encoder $\mathbf{E}$ and a pretrained LLM decoder $\mathbf{D}$ that is always frozen throughout training and inference. The encoder is architecturally identical to the decoder as in 500xCompressor and ICAE, with the key exception that the MLP layers have been replaced with the identity operation within each block of the Transformer [Vaswani et al., 2017]:

$$h_\ell = \mathrm{LN}_{\mathrm{pre}}(h_{\ell-1}) \qquad h_\ell = \mathrm{LN}_{\mathrm{pre}}(h_{\ell-1}) \tag{1}$$

$$h_\ell = \mathrm{MHA}(h_\ell) + h_\ell \qquad h_\ell = \mathrm{MHA}(h_\ell) + h_\ell \tag{2}$$

$$h_\ell = \mathrm{MLP}(\mathrm{LN}_{\mathrm{post}}(h_\ell)) + h_\ell \qquad h_\ell = \mathrm{LN}_{\mathrm{post}}(h_\ell) + h_\ell \tag{3}$$

$h_{\ell-1}$ denotes the input hidden state to the ℓth Transformer block, $\mathrm{LN}_{\mathrm{pre}}$ and $\mathrm{LN}_{\mathrm{post}}$ are layer norms [Jimmy Lei Ba and Hinton, 2016], and MHA denotes multi-headed attention [Vaswani et al., 2017].

Let the input for the encoder be represented by the concatenation of n prompt tokens $\mathbf{X}_n = (x_1, \ldots, x_n)$ with the encoder's m learned memory tokens $\mathbf{Y}_m = (y_1, \ldots, y_m)$. $\mathbf{Z} = \mathbf{E}([\mathbf{X}_n, \mathbf{Y}_m])$ is the latent representation from the encoder output. For 500xCompressor, $\mathbf{Z} = \{\mathbf{KV}(h_\ell^{\mathbf{Y}_m}) \forall \ell\}$: the encoder's per-layer attention key-value pairs corresponding to $\mathbf{Y}_m$. The input to the decoder is $\mathbf{Z}$ concatenated with a regeneration token **[REGEN]**, which is used to regenerate $\mathbf{X}$ using the latent information from $\mathbf{E}$. For both 500xCompressor and AOC **[REGEN]** is the **[BOS]** token. Therefore, the regeneration of $\mathbf{X}_n$ from the latent representation $\mathbf{Z}$ is given by

$$\hat{\mathbf{X}}_n = \mathbf{D}\left([\mathbf{Z}, [\mathbf{REGEN}]]\right) = \mathbf{D}\left([\mathbf{E}([\mathbf{X}_n, \mathbf{Y}_m]), [\mathbf{BOS}]]\right) \tag{4}$$

The standard cross-entropy loss between the decoder logits and the input $\mathbf{X}$ is used to train the encoder via backpropagation [LeCun et al., 1989]. For all experiments, we use Llama 3.2 1B Instruct [Llama Team, 2024] as the pretrained LLM in bfloat16 [Wang and Kanwar, 2019] precision, AdamW [Kingma and Ba, 2015] with a 300-step warmup to a learning rate 2×10^{-4} as the optimizer in PyTorch [Paszke et al., 2019] conducting training using Transformers [Wolf et al., 2020] on a single NVIDIA A6000 GPU. LoRAs are trained on the queries, keys, values, and output projections in the multi-headed attention components for 500xCompressor and LoRA ablations on AOC.

LoRA Ablations. Due to the lower number of total parameters in the encoder for AOC, we perform full training instead of LoRA to learn a strong prompt compressor. However, this causes the total number of parameters in memory at both training and inference time to be slightly larger with AOC compared to the baseline 500xCompressor which use a LoRA of the decoder LLM. This trade-off of

increased memory for decreased compression time motivates ablations on learning the AOC encoder using LoRA (LoRA-AOC) instead of training the entire encoder.

Compressed Prompt Interpolation. The latent information $\mathbf{Z}$ from compressing a prompt has not been extensively studied beyond classifier-guided generation by Wingate et al. [2022]. As an initial step toward better understanding the compressed forms of prompts, we conduct linear interpolations between compressed prompts and qualitatively inspect the intermediary output. The interpolation between $\mathbf{Z}_0$ and $\mathbf{Z}_1$ with a given weight w is given by:

$$\mathbf{Z}_{\text{interp}} = \mathbf{Z}_0 + w(\mathbf{Z}_1 - \mathbf{Z}_0) \tag{5}$$

Data. Experiments are performed using random samples from the arXiv dataset [arXiv.org submitters, 2024]. AOC is trained on 300,000 abstracts from the arXiv dataset first submitted before July 1, 2023 and validated on 3,000 abstracts first submitted after January 4, 2024. Final evaluations were conducted on a held-out test set of 3,000 abstracts from after January 4, 2024. These dates were chosen based on the Llama 3.2 training cutoff of December 2023, and are identical to the cutoffs presented in [Li et al., 2024]. The amount of training data was determined while accounting for limited computational resources.

Metrics. We evaluate AOC on text regeneration as performed using Equation 4. Following [Ge et al., 2024], we report the Bilingual Evaluation Understudy (BLEU) [Papineni et al., 2002] and Exact-Match (EM) scores. Notably, the EM metric defined by [Ge et al., 2024] is the proportion of identical prefix length to total target length. Given a regenerated sequence of length n', this proportional EM metric is defined as:

$$\text{EM}(\mathbf{X}_n, \hat{\mathbf{X}}_{n'}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\mathbf{X}_i = \hat{\mathbf{X}}_i}(\mathbf{X}_i, \hat{\mathbf{X}}_i) \tag{6}$$

In contrast, the EM metric defined by [Li et al., 2024] is a binary metric equal to 1 when the regeneration $\hat{\mathbf{X}}_{n'}$ is identical to $\mathbf{X}_n$ and 0 otherwise, introducing a discrepancy in notation. We report the EM metric as defined in Equation 6 since it is more informative. Additionally, we report the Recall-Oriented Understudy for Gisting Evaluation Longest Common Subsequence (ROUGE-L) [Lin and Och, 2004] F1 scores which evaluate overall sequence similarity, following [Li et al., 2024].

3 Results

Baseline Comparison. To demonstrate the benefits of AOC, we compare to 500xCompressor with a variety of input prompt lengths $n \in \{96, 192, 288, 384, 480\}$ and number of memory tokens $m \in \{1, 4, 16\}$.

Table 1: Evaluation results for models trained with $m = 16$ memory tokens.

Prompt Length	Model	BLEU (↑)	EM (↑)	ROUGE-L F1 (↑)
$n = 96$	500xCompressor	0.981	0.740	0.990
	LoRA-AOC	0.740	0.197	0.856
	AOC	**0.984**	**0.889**	**0.991**
$n = 192$	500xCompressor	0.850	0.109	0.915
	LoRA-AOC	0.284	0.065	0.510
	AOC	**0.868**	**0.454**	**0.924**
$n = 288$	500xCompressor	0.685	0.130	0.816
	LoRA-AOC	0.319	0.068	0.548
	AOC	**0.839**	**0.465**	**0.901**
$n = 384$	500xCompressor	0.662	0.106	0.799
	LoRA-AOC	0.255	0.068	0.478
	AOC	**0.801**	**0.386**	**0.880**
$n = 480$	500xCompressor	0.588	0.082	0.746
	LoRA-AOC	0.201	0.053	0.421
	AOC	**0.823**	**0.483**	**0.893**

As seen in Table 1 and Table 2, AOC outperforms 500xCompressor across all prompt lengths with 4 or 16 memory tokens despite having 67% less encoder parameters. Based on the results in Table 1 we find that AOC and 500xCompressor only perform similarly when restricted to a single memory token. The large variance in EM between models can be attributed to differences in early parts of the regeneration, as the EM metric is based on the proportion of identical prefix matching. Interestingly, LoRA-AOC tends to perform worse than AOC and the baseline 500xCompressor across all metrics, which suggests that the effectiveness of LoRA in Transformers relies in part on the frozen MLPs, in line with prior work on freezing Transformer components Lu et al. [2022].

Table 2: Evaluation results for models trained with $m = 4$ memory tokens.

Prompt Length	Model	BLEU (↑)	EM (↑)	ROUGE-L F1 (↑)
$n = 96$	500xCompressor	0.669	0.073	0.815
	LoRA-AOC	0.342	0.069	0.599
	AOC	**0.711**	**0.221**	**0.843**
$n = 192$	500xCompressor	0.302	0.015	0.561
	LoRA-AOC	0.136	0.021	0.399
	AOC	**0.374**	**0.064**	**0.615**
$n = 288$	500xCompressor	0.218	0.035	0.484
	LoRA-AOC	0.126	0.019	0.387
	AOC	**0.339**	**0.056**	**0.578**
$n = 384$	500xCompressor	0.236	0.013	0.507
	LoRA-AOC	0.117	0.015	0.378
	AOC	**0.300**	**0.040**	**0.558**
$n = 480$	500xCompressor	0.241	0.027	0.508
	LoRA-AOC	0.068	0.011	0.288
	AOC	**0.343**	**0.058**	**0.587**

It can be seen in Table 3 that for $m = 1$, AOC performs on-par with 500xCompressor, although both display poor regeneration abilities for some of the largest compression ratios in our experiments. Upon inspection of the loss curves from training the $m = 1$ models in Table 3, we discover that they are likely under-trained due to computational budget constraints. Based on these results, it appears that increasing the amount of memory tokens m may allow for a smaller training data set.

Table 3: Evaluation results for models trained with $m = 1$ memory token.

Prompt Length	Model	BLEU (↑)	EM (↑)	ROUGE-L F1 (↑)
$n = 96$	500xCompressor	0.122	0.013	**0.382**
	LoRA-AOC	0.092	0.022	0.355
	AOC	**0.129**	**0.037**	0.369
$n = 192$	500xCompressor	**0.102**	0.015	**0.352**
	LoRA-AOC	0.074	0.013	0.308
	AOC	0.095	**0.017**	0.327
$n = 288$	500xCompressor	**0.090**	0.007	**0.337**
	LoRA-AOC	0.061	0.009	0.278
	AOC	0.089	**0.016**	0.317
$n = 384$	500xCompressor	0.089	0.009	**0.337**
	LoRA-AOC	0.068	0.010	0.302
	AOC	**0.094**	**0.019**	0.330
$n = 480$	500xCompressor	0.094	0.004	**0.355**
	LoRA-AOC	0.056	0.008	0.273
	AOC	**0.097**	**0.015**	0.341

Latent Space Inspection. In Table 4 we show the result of linearly interpolating between the compressed information $\mathbf{Z}$ from the prompt p_0 ="`We present an awesome new idea.`" and the prompt p_1 ="`Large planets may have many moons.`" for AOC, color-coding by similarity to p_0 or p_1. As can be observed, the interpolation of the two latent representations results in a regenerated mixture of prompts, such as when the interpolation weight $w = 0.5$ `planet` which is more closely related to `planets` from p_1 than `idea` from p_0. For $w = 0.53$, `many moons` from p_1 appears in a regeneration that shares the same prefix as p_0. Similarly, for interpolation weights $w = 0.55$ and $w = 0.6$, `amazing` and `wonderful`, which are more closely related to `awesome` from p_0, appear in a regeneration almost identical to p_1 with the same two-word prefix. We also note that Table 4 shows both p_0 and p_1 were perfectly regenerated from their unaltered compressed states with zero information loss.

Table 4: Regeneration of linearly interpolated latent information.

Interpolation Weight	Regeneration
$w = 0.00$	We present an awesome new idea.
$w = 0.40$	We present an amazing new idea.
$w = 0.50$	We present an amazing new planet.
$w = 0.53$	We present an amazing many moons.
$w = 0.55$	Large planets have many amazing.
$w = 0.60$	Large planets have many wonderful.
$w = 1.00$	Large planets may have many moons.

4 Conclusion

We introduce AOC, a prompt compression encoder using only attention layers from a decoder LLM that demonstrably achieves comparable or better compression to LoRA baselines with identical architecture to the decoder LLM. Experiments show that the memory tokens learned with AOC can encode similar amounts of information to baselines with $3\times$ the amount of parameters. In future work, we hope to further explore encoder architectures, as our results indicate that a prompt compression encoder need not have the same architecture as the decoder LLM. Additionally, we seek to better understand the latent space formed by compressed prompts and extend the use of compressed prompts beyond the interpolation example presented in this work. While this work was performed with limited computational resources, we aim to study more diverse and larger datasets, model architectures, and compression ratios in the future.

References

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Stephanie Chan, Ankesh Anand, Zaheer Abbas, Azade Nova, John D Co-Reyes, Eric Chu, et al. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*, 2024.

Amanda Bertsch, Maor Ivgi, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*, 2024.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, 2023.

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

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

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. 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, 2022.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36, 2023.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. *International conference on learning representations*, 2024.

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

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

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

Jamie Ryan Kiros Jimmy Lei Ba and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

AI @ Meta Llama Team. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Shibo Wang and Pankaj Kanwar. Bfloat16: The secret to high performance on cloud tpus. `https://cloud.google.com/blog/products/ai-machine-learning/bfloat16-the-secret-to-high-performance-on-cloud-tpus`, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

arXiv.org submitters. arxiv dataset, 2024. URL `https://www.kaggle.com/dsv/7548853`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*, pages 605–612, 2004.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Pretrained transformers as universal computation engines. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7628–7636, 2022.