Selection-p: Self-Supervised Task-Agnostic Prompt Compression for Faithfulness and Transferability

Tsz Ting Chung[1] **Leyang Cui**[2] **Lemao Liu**[2] **Xinting Huang**[2]
Shuming Shi[2] **Dit-Yan Yeung**[1]
[1]The Hong Kong University of Science and Technology [2]Tencent AILab
{ttchungc, nealcly.nlp, lemaoliu, shuming}@gmail.com
timxthuang@tencent.com dyyeung@cse.ust.hk

Abstract

Large Language Models (LLMs) have demonstrated impressive capabilities in a wide range of natural language processing tasks when leveraging in-context learning. To mitigate the additional computational and financial costs associated with in-context learning, several prompt compression methods have been proposed to compress the in-context learning prompts. Despite their success, these methods face challenges with transferability due to model-specific compression, or rely on external training data, such as GPT-4. In this paper, we investigate the ability of LLMs to develop a unified compression method that discretizes uninformative tokens, utilizing a self-supervised pre-training technique. By introducing a small number of parameters during the continual pre-training, the proposed Selection-p produces a probability for each input token, indicating whether to preserve or discard it. Experiments show Selection-p achieves state-of-the-art performance across numerous classification tasks, achieving compression rates of up to 10 times while experiencing only a marginal 0.8% decrease in performance. Moreover, it exhibits superior transferability to different models compared to prior work. Additionally, we further analyze how Selection-p helps maintain performance on in-context learning with long contexts.

1 Introduction

In-context learning has shown remarkable success in various natural language processing tasks (Brown et al., 2020), such as classification task (Min et al., 2022), and mathematical reasoning task (Wei et al., 2023), enabling Large Language Models (LLMs) to tackle complex and diverse tasks using only few-shot samples. However, in-context learning also significantly extends the length of prompts, resulting in increased computational and financial costs. Recently, a line of work has been focusing on prompt compression, which aims to compress the original prompts while minimizing information loss. They are categorized into discrete compression (Li et al., 2023; Jiang et al., 2023; Pan et al., 2024) and continuous compression (Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2024); the former compresses the context into discrete tokens, while the latter compresses it into a short sequence of continuous vectors.

	Transferable	Single Run	¬External
AutoCompressor	✗	✓	✓
LLMLingua	✓	✗	✓
LLMLingua-2	✓	✓	✗
Selection-p	✓	✓	✓

Table 1: **Comparison between the proposed Selection-p and existing content compression approaches.** Selection-p exhibits great transferability (Transferable), does not require multi-round iterative decoding (Single Run), and does not rely on costly external resources for training (¬External).

Observing redundant and repetitive content in a given input, discrete compression methods aim to eliminate less informative context without significantly compromising the model's performance. For example, LLMLingua (Jiang et al., 2023) proposes to perform iterative token truncation based on the content perplexity, requiring multi-round decoding (Jiang et al., 2023). LLMLingua-2 (Pan et al., 2024), distilled from GPT-4 (OpenAI, 2023), addresses the potential misalignment between entropy and the compression objective, as well as the distribution gap between the perplexity of the compression model and the target model. High costs are still involved in the training data construction. Meanwhile, optimizing the distribution for specific LLMs (GPT-4) may, on the other hand, hinder the transferability of the compressed content to other LLMs. More details are discussed in Section 4.4.

Continuous compression (Bulatov et al., 2022;

Wingate et al., 2022) teaches pre-trained LMs the ability to compress text into a short sequence of continuous vectors. AutoCompressors (Chevalier et al., 2023) uses an unsupervised learning objective, which motivates the model to cache crucial information within the summary vectors. Despite their success, these methods have poor generalization as they can only compress to the length specified during training. Additionally, since the continuous vector cannot be transferred between models, a separate compressor must be trained for each model.

This raises an interesting research question:

"Can LLMs learn to identify less informative tokens within a given context without external annotated signals?"

To answer this question, we propose a pre-training strategy with a self-supervised signal, which enables the model to autonomously learn to predict the next token based on compressed context. With a small additional number of parameters, a forward pass on the proposed selection-p creates the probability vector p corresponding to each input token, indicating whether to preserve or discard the token. During inference, we can apply the detokenized compressed tokens to any downstream LLMs with only single-turn decoding and without reliance on any costly external resources as depicted in Table 1.

The main contributions of this work are fourfold:

- We present Selection-p which achieves only 0.8% drops in performance under 10x compression rate across nine traditional classification tasks, surpassing the performance of the existing compression models. Under this setting, a speedup of 5.3x can be achieved during inference with in-context learning.
- Selection-p demonstrated great transferability, which surpasses the performance of prior work in performing hard compression for both open-source and close-source models.
- We further analyze how Selection-p helps in in-context learning in long-context settings, presenting a potential solution to address the performance declination of long-context models in ICL.
- We connect in-domain prior works and make comparisons with these state-of-the-art compression models, providing a complete picture.

2 Related Work

2.1 Hard Compression

Some studies focus on token pruning (Goyal et al., 2020; Kim and Cho, 2021; Rao et al., 2021; Kim et al., 2022; Modarressi et al., 2022) and token merging (Bolya et al., 2023) but they are designed primarily for smaller models like BERT. More recently, Selective Context (Li et al., 2023) is the first to propose to prune less important tokens based on information entropy. Subsequently, LLMLingua (Jiang et al., 2023) refined the approach by integrating the selection of demonstrations and the allocation of compression budgets for various segments of the input prompt. No training is required in these models but their efficacy in downstream tasks with compression applied to in-context demonstrations remains limited. Pan et al. (2024) extended the idea and addressed the potential misalignment between entropy and the compression objective, leveraging full bidirectional context by training on their proposed GPT-4 distilled compression dataset. Our simple yet effective approaches outperform previous studies.

2.2 Soft Compression

Gist (Mu et al., 2023) is first proposed to compress prompt with soft tokens. Subsequently, Autocompressor (Chevalier et al., 2023) and ICAE (Ge et al., 2024) extend the idea to handle long contexts with different pretraining approaches. ICAE further conducts instruction tuning to enhance model performance. The downstream performance of these models heavily relies on the tuned compression model, with a fixed compression rate. Additionally, retraining is necessary for different versions of LLMs. Compared to these approaches, our work offers greater flexibility and transferability, while simultaneously surpassing the performance of existing compression models.

3 Methodology

Under the intuition that redundant texts often exist and their removal does not hinder human understanding of the text, we assume that LLMs behave in a similar manner. To efficiently identify less informative tokens within a given context, we propose a simple pre-training objective encouraging the model to predict the same token both before and after discarding less informative tokens.

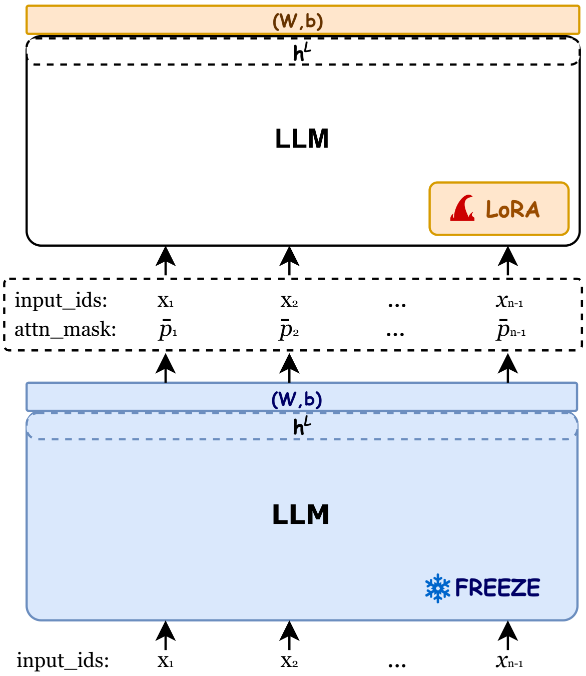
(W,b)
h^L
LLM
LoRA
input_ids: x₁ x₂ ... x_{n-1}
attn_mask: p̄₁ p̄₂ ... p̄_{n-1}
(W,b)
h^L
LLM
FREEZE
input_ids: x₁ x₂ ... x_{n-1}

Figure 1: **Illustration with the training process.** Areas in orange are learnable parameters. For the input context $[x_1, x_2, \ldots, x_{n-1}]$, inference without parameters update is performed first to create the attention mask $\bar{p}$. These subsequently form the model input for LoRA training and updating the parameters of the additional linear layer.

3.1 Preliminary

Language Model. Given a context $[x_1, x_2, \ldots, x_{n-1}]$, the objective of the language model is to predict the next token x_n, formed as $P(x_n|x_1, x_2, \ldots, x_{n-1})$. In case of training with the causal language modeling (CLM) loss, we will have,

$$\mathcal{L}_{\text{CLM}} = -\sum \log P(x_i \mid x_1, x_2, \ldots, x_{i-1}; \theta)$$

Tokens Selection Models. LLMLingua (Jiang et al., 2023) and its variant (Li et al., 2023; Pan et al., 2024) select tokens according to the computed distribution of the targeted LLM. Specifically, suppose x_i is a token in the prompt, if the probability of a token x_i is less than a threshold, then the token is selected to be compressed.

3.2 Selection-p

Following the tokens selection models (Li et al., 2023; Jiang et al., 2023; Pan et al., 2024), Selection-p includes the two steps for testing, i.e. the selection (compression) step and the inference step. In the *selection* step, unlike LLMLingua and its variant, we instead define a selection model to select less informative tokens within the context in a discriminative way. In the *inference* step, the tokens selected via the selection model are passed to our targeted inference model.

Selection. Assume $p_i \in [0, 1]$ denote a measure of the informativeness of token x_i. Theoretically, we can customize a deep neural model to instantiate p_i. In practice, we directly take a pre-trained language model and adopt the last layer of hidden representation $\mathbf{h}^L$ from a pre-trained LM to the new linear projection layer. Formally, suppose $\mathbf{h}^L = \{h_1, h_2, \cdots, h_n\}$ denotes the sequence of hidden states for all tokens x_i at the last layer of a pre-trained language model. The selection model $\mathbf{p} = \{p_1, p_2, \cdots, p_n\}$ is defined as follows:

$$\mathbf{p} = \sigma(\mathbf{W}\mathbf{h}^L + b) \tag{1}$$

where σ is the sigmoid function, $\mathbf{W}$ and b are parameters of the projection matrix and bias vector, respectively.

By using the selection model $\mathbf{p}$, it is straightforward to compress the context for inference: we directly prune the corresponding tokens according to our desired compression rate in retaining the top $k\%$ of tokens in the context. To ensure the efficiency in performing compression, a single forward pass on our model creates the token preservation probability for all input tokens for simplicity.

Training. Our training criterion for the selection model aims to preserve the language modeling ability of the LLM while also learning to discard tokens effectively. To this end, we employ the self-supervised approach to optimize the selection model and therefore we do not need external resources to train the selection model compared with Pan et al. (2024).

To keep the training process consistent with the inference process, we first discretize the selection model $\mathbf{p}$. Let $\bar{p}_i$ denote the discretized binary model of p_i. In other words, $\bar{p}_i$ is 1 if it ranks the top $k\%$ of tokens with the highest p values and 0 otherwise. Then we use the discretized model as a mask to define the CLM loss function as follows:

$$\hat{\mathcal{L}}_{\text{CLM}} = -\sum \log P(x_i \mid \bar{p}_1 x_1, \ldots, \bar{p}_{i-1} x_{i-1}; \theta) \tag{2}$$

where $\bar{p}_i x_i$ denotes whether the token x_i is masked or not depending on the value of $\bar{p}_i$, where the above language model P is set as the same model as that used in the selection model in Eq. 1.

Training Details for Transferability. One of our goals is to achieve a transferable compression method. Therefore, the parameters that achieve the best loss on the language models in Eq. 1 and Eq. 2 in training may not be transferred to the targeted language model in inference since the language models in training and inference can be different. As a result, to ensure the better transferability of the optimized selection model, we freeze the pre-trained language model in Eq. 1 and employ LoRA (Hu et al., 2021) to train a partial parameter in the language model in Eq. 2. [1] In summary, the CLM-based training loss is illustrated in Figure 1.

4 Experiment

4.1 Setting

We finetune a LLaMA-2-7b model (Touvron et al., 2023) on 100M tokens from RedPajama (TogetherAI, 2023) on split segments of 1,024 tokens via LoRA (Hu et al., 2022). To provide a comprehensive analysis of the model capabilities, our evaluation is conducted on traditional classification tasks as well as the long-context classification task.

For each task, we randomly sample from the training set to construct the demonstration set for In-Context Learning (ICL), which also serves as our compression target for token selection. During inference, the compression process only needs to be computed once for all subsequent inferences on the testing instances.

Traditional Classification Tasks. Following Chevalier et al. (2023), we evaluate and compare different compression models on nine classification tasks, including six tasks from SuperGlue (Wang et al., 2019). The predictions by LLMs are determined by iterating through all possible answer options for the instance and selecting the option with the minimum negative log-likelihood. The in-context demonstrations have been carefully selected to approximate a size of 750 tokens, and the complete demonstration is employed. This is referred to as the "full-shot". Since the average token length for a single demonstration varies for different tasks (e.g., a single demonstration in RTE averages about 75 tokens, resulting in a 10-shot setup under the full-shot setting), the exact number of shots differs depending on the task.

To ensure a fair comparison among different compression models, a compression rate of 0.1 is used. For each task, four sets of demonstrations are selected, and the average result across these four trials is presented in Table 2. Additionally, the average accuracy of the nine tasks is presented for a clearer performance comparison.

To achieve a compression rate of 0.1, a simpler approach is to directly retain one-tenth of the full-shot demonstration (e.g., using 1-shot instead of 10-shot for the RTE task) instead of performing selection at the token level. We also include this method as a baseline to further demonstrate the effectiveness of our approach.

Long Context Classification Tasks. Recent research by Li et al. (2024) shows the failure of in-context learning tasks when applied to long-context scenarios. To investigate whether compression models can serve as a viable solution in long-context settings, we compare Selection-p with long-context models, including LLaMA-2-7B-LongLora (Chen et al., 2023) and Long-LLaMA-code-7B (Tworkowski et al., 2023) on the BANKING77 dataset (Casanueva et al., 2020). The dataset contains 77 classes where traversing all the instances with unique labels requires approximately two thousand tokens. Evaluation is conducted at 2K, 4K, and 7K token levels, and we adopt a compression rate of 10x for Selection-p and LLMLingua-2 among all levels. Since the long in-context demonstration is used, chunking is performed for every 2,048 tokens in Selection-p. The compressed results are concatenated together with a space token between each pair of chunks. We again follow the evaluation setting by Chevalier et al. (2023) on the models' prediction, with the result presented in Table 3.

4.2 Baselines

We compare the Selection-p with the following state-of-the-art compression models.

- **LLMLingua** (Jiang et al., 2023) employs an iterative compression algorithm to filter less informative tokens based on the token-level perplexity. To further boost the performance, Jiang et al. (2023) also conducts a budget controller to allocate varying budgets across different demonstrations and questions. We find that there is also a significant discrepancy observed between the prescribed compression rate and the actual compression rate through

[1] In our preliminary experiments, we tried the Gumbel-softmax trick to optimize the loss in Eq. 2 but we did not observe gains over the direct optimization implemented in our paper.

LLMLingua API calls. For a fair comparison, we exclusively utilize the token-level prompt compression algorithm from LLMLingua in Table 2. We additionally compare LLMLingua with Selection-p equipped with Budget Controller in Section 5.5.

- **LLMLingua-2** (Pan et al., 2024) is derived from data distillation obtained by instructing GPT-4 to perform compression. Similar to ours, the model is trained as a binary classifier on each token, determining whether each token should be preserved.
- **AutoCompressor** (Chevalier et al., 2023) is constructed based on the RMT architecture (Bulatov et al., 2022). It compresses text into summary vectors that can be reused in subsequent segments. It is the only soft compression model adopted for comparison, considering that we followed the experiment setting for assessing the in-context learning ability of LLMs.

4.3 Evaluation Result

Traditional Classification Tasks. None of the compression models can achieve superior performance compared to the full-shot demonstration setting, which is in line with our expectations given the information loss during compression. However, certain tasks show a notable improvement when compared to both the zero-shot and full-shot approaches, e.g., all the hard compression models surpass zero-shot and full-shot by approximately 20% in the WSC task. Among all compression models, Selection-p demonstrates the highest performance in conducting ICL, with an average accuracy of 67.4% across all 10 tasks as presented in Table 2. Examples of in-context demonstration before and after compression are shown in Appendix A. To demonstrate the effectiveness of our model, we have included one-tenth of the original demonstration set as the baseline. Our model significantly outperforms the baseline with a comparable number of tokens, therefore highlighting the effect of performing compression at the token level.

Long Context Classification Tasks. Our model outperforms LLaMA-2-7B-LongLora, Long-LLaMA-code-7B and LLMLingua-2 at all token size levels, and achieves similar results to Li et al. (2024)'s findings on the long-context models. In addition, a growing trend with variations is observed with increasing compressed demonstrations, indicating that our model can successfully learn from additional information after compression. Examples of in-context demonstration before and after compression are presented in Appendix A.

4.4 Transferability

Compression is first performed on the demonstration set for ICL with Selection-p. Subsequently, the compressed tokens are passed to a separate downstream model (i.e. LLaMA-2-13B or the black-box models) as the compressed demonstration prompt for evaluation.

To LLaMA-2-13B. We follow the same setting of evaluation across different classification tasks in Section 4.3. To assess the transferability of the compression models, we compress demonstrations with Selection-p and input the compressed demonstration tokens into LLaMA-2-13B. In comparing different compression models, since retraining is required for soft compression methods, no results can be obtained for AutoCompressor (Chevalier et al., 2023). In the case of LLMLingua, token-level perplexity is calculated with LLaMA-2-13B instead of LLaMaA-2-7B in this experiment.

Our approach outperforms all other compression models as shown in Table 4. In addition, a small deviation is observed between the 10x compression rate and the full shot setting, demonstrating the great transferability of our models. Notably, with Selection-p, the tasks that outperform the full-shot setting in LLaMA-2-7b also exhibit similar patterns in LLaMA-2-13B.

To Black-box Models. Taking cost into consideration, we select ChatGPT (OpenAI, 2023) and Gemini (Team, 2023) for evaluation to examine its transferability to LLMs. Traditional classification tasks often have a simple nature and the potential issue of data contamination, leading to high accuracy and causing an insignificant evaluation. Therefore, we use BANKING77 (Casanueva et al., 2020) for evaluation. Following a similar setup as described in Section 4.3, we adopt a token size of 750 for examination. However, in case the compression rate is too high, ChatGPT and Gemini are more likely to deviate from the instructions and provide task-irrelevant responses. Therefore, we adopt a compression rate of 3x and use the EM metric for this experiment given their black-box nature. Note that there may be variation in the result of Gemini

	Subj	RTE	WSC	BoolQ	MultiRC	SST-2	WIC	COPA	AG News	AVG
Zero-shot	49.3	58.8	43.4	67.4	52.5	67.7	50.8	52.5	63.3	56.2
"One-tenth"-shot	48.6	65.3	52.6	68.3	49.2	84.0	53.6	83.9	55.5	62.3
Full-shot	80.7	70.7	41.8	62.8	46.8	92.5	56.4	85.6	76.3	68.2
AutoCompressor	57.9	56.1	39.4	66.5	51.8	92.8	53.0	84.4	80.9	64.7
LLMLingua	56.7	60.9	63.0	69.4	50.3	69.9	51.6	76.4	61.5	62.2
LLMLingua-2	57.3	67.8	63.7	69.8	52.8	66.2	50.3	71.4	61.9	62.3
Selection-p	68.5	68.5	61.1	69.7	54.4	90.7	50.3	76.9	66.8	**67.4**

Table 2: **Evaluation result on traditional classification tasks.** Four sets of random demonstrations were selected, with the average result being presented. The average result across different classification tasks is presented under AVG.

	2K	4K	7K
LLaMA-2-7B-LongLora	0.0	0.0	0.0
Long-LLaMA-code-7B	0.0	0.0	0.0
LLMLingua-2	41.2	31.6	35.2
Selection-p	**46.9**	**50.9**	**51.6**

Table 3: Evaluation result on BANKING77 with increasing in-content demonstrations tokens length.

since the discrete compressed tokens sometimes trigger the SAFETY error. The prompt used is presented in Appendix B.

Though the performance of our model still deviates from the full-shot setting, it achieved the best performance compared to the existing works as presented in Table 4, demonstrating fair transferability even in LLMs like ChatGPT. Surprisingly, though LLMLingua-2 is distilled from GPT-4, it exhibits poor generalization compared to other compression models.

5 Analysis

5.1 Flexibility

Performance with Different Number of Initial Tokens. The result in long context classification tasks in Section 4.3 shows the effectiveness of chunk-wise compression in long context. We further analyze if compression models work well in normal few-shot settings in classification tasks. In this experiment, the in-context demonstrations are selected with an approximate size of 250 tokens. The comparison to the result with the token size of 750 in Section 4.3 is presented in Table 6.

Selection-p shows the best performance under the constraint of 250 tokens when compared to other compression models. Additionally, it also follows the full-shot (i.e. 750 tokens level) trend, the average performance across all classification tasks increases along with the number of provided demonstrations. On the contrary, other compression models didn't achieve an improvement in accuracy with more demonstrations. For instance, there is a drop of 2% recorded with an additional 500 tokens of information for AutoCompressor.

5.2 Latency Analysis

We analyze end-to-end latency on A100-80G GPU with the WSC task, illustrated in Table 7. Our method can achieve 5.3x speed up on 10x compressed in-context demonstration. Compared to the inference time, negligible time is required for compression on the ICL task setting, demonstrating high efficiency in adopting our models for compression. We also compared LLMLingua with the disabled content Budget Controller. It requires iterative decoding on the segmented context while Selection-p only requires a single inference on all tokens and demonstrates a good performance.

5.3 Correlation with Attention and Perplexity

With the p-value ranging between 0 and 1 for each token, we further study whether any correlations exist among p, the mean attention value during the forward pass, and the tokens level perplexity (i.e. a core component in LLMLingua (Jiang et al., 2023)). Since the value of p is derived from the last hidden state of the model, we only consider the last layer mean attention of our tuned model. We employed Spearman's Rank Correlation Coefficient (Spearman, 1904) to compute the correlation between the three variables. It is calculated for different traditional classification tasks and the averaged value across tasks. The result presented in Figure 2 indicates only a weak correlation observed between the p value and the other two variables while the correlation between the last layer mean attention and perplexity is more significant. Among all tasks,

	LLaMA-2-13B									
	Subj	RTE	WSC	BoolQ	MultiRC	SST-2	WIC	COPA	AG News	AVG
Full-shot	91.6	74.8	46.9	67.7	45.7	94.7	54.6	77.6	79.2	70.3
LLMLingua	53.6	61.0	63.2	70.6	51.5	64.3	50.0	78.1	58.0	61.4
LLMLingua-2	48.3	68.7	41.3	75.8	52.2	51.3	49.0	48.2	71.1	56.2
Selection-p	69.3	69.5	65.4	74.7	50.7	81.1	50.5	87.5	63.2	**68.0**

Table 4: **Analysis of transferability to open-source model LLaMA-2-13B.** The experiment is performed on 750 tokens in-context demonstrations with a 10x compression rate.

	ChatGPT GPT-3.5-Turbo	**Gemini** Gemini-1.0-Pro
Full-shot	74.2	73.3
LLMLingua	58.6	40.2
LLMLingua-2	55.7	51.9
Selection-p	**62.9**	**58.9**

Table 5: **Analysis of transferability to blackbox models (i.e. ChatGPT and Gemini).** The experiment is performed on 750 tokens in-context demonstrations with a 3x compression rate.

WIC demonstrates a prominently high value compared to others, this may explain the small variation in accuracy across different compression models and different experimental settings.

5.4 Tokens Level Part-of-Speech Analysis

To further interpret the rationale behind our compression models, we analyze what kinds of words are likely preserved by Selection-p. Under the discreteness of our compression result, we locate the corresponding words from the compressed tokens and obtain the Part-of-Speech (PoS) tags with an NLTK tagger. For each type of PoS tag, we compute the token preservation percentage with

$$\frac{|\text{compressed\_token}_{\text{tag}_i}|}{|\text{total\_token}_{\text{tag}_i}|}$$

for each PoS tag tag$_i$. The experiment is conducted between the compressed result and the original demonstrations among the nine traditional classification tasks with four demonstration sets per task. We analyze tags with a frequency of appearance greater than 1%.

From the result presented in Figure 3, PRP and punctuations (i.e. indicating the start of the next sentence or phrase) are more likely preserved. The potential reason for the high preservation ratio on PRP (personal pronoun) likely corresponds to the pronoun resolution task of WSC. Under the task setting, it can be useful hints for the answer derivation.

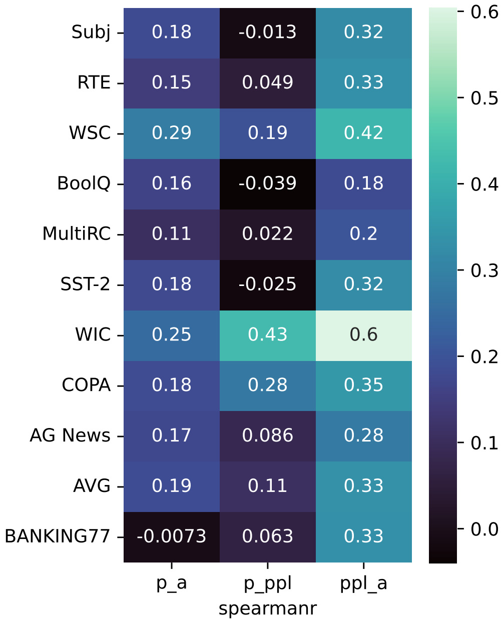

Figure 2: Spearman's Rank Correlation Coefficient (spearmanr) between p (p) value, mean attention (a) and token-level perplexity (ppl) across different traditional classification tasks.

The high preservation ratio of punctuation may indicate a large redundancy in a sentence, and truncating sentence separation tokens is undesirable. Additionally, as highlighted by Wang et al. (2023), formatting information (i.e. structure of the demonstrations) matters a lot in in-context learning. However, formatting tokens (i.e. ":") are unlikely to be preserved with Selection-p compared to the original distribution in our case. In general, we also observe a higher degree of preservation of noun phrases compared to verbs.

5.5 On Fair Comparison with LLMLingua

As described in Section 4.2, LLMLingua conducts demonstration selection prior to compression at the token level, while other methods compress directly on the token level. Since the demonstration selection process can also be incorporated into other

	Subj	RTE	WSC	BoolQ	MultiRC	SST-2	WIC	COPA	AG News	AVG
Zero-shot	49.3	58.8	43.4	67.4	52.5	67.7	50.8	52.5	63.3	56.2
Full-shot	81.3	69.9	51.2	62.7	46.8	89.3	51.8	85.1	67.9	67.3
$\Delta_{\text{Full-shot}}$	-0.6	+0.8	-9.4	+0.1	±0	+3.2	+4.6	+0.5	+8.4	+0.9
AutoCompressor	56.2	61.5	44.2	68.3	52.7	93.0	51.5	83.6	76.1	65.2
$\Delta_{\text{AutoCompressor}}$	+1.7	-5.4	-4.8	-1.8	-0.9	-0.2	+1.5	+0.8	+4.8	-0.5
LLMLingua*	55.6	61.4	61.3	68.2	53.1	81.1	50.2	75.8	70.9	64.2
$\Delta_{\text{LLMLingua}}$	+1.1	-0.5	+1.7	+1.2	-2.8	-11.2	+1.4	+0.6	-9.4	-2.0
LLMLingua-2	52.4	65.3	63.9	66.1	50.9	82.9	50.8	77.8	56.3	62.9
$\Delta_{\text{LLMLingua-2}}$	+4.9	+2.5	-0.2	+3.7	+1.9	-16.7	-0.5	-6.4	+5.6	-0.6
Selection-p	65.7	65.5	58.7	67.5	54.3	81.3	50.4	77.9	68.3	**65.5**
$\Delta_{\text{Selection-}p}$	+2.8	+3.0	+2.4	+2.3	+0.1	+9.4	-0.1	-1.0	-1.5	+1.9

Table 6: **Performance with different number of demonstrations from about 250 tokens to about 750 tokens.** Δ refers to the performance enhancement that can be achieved by increasing the demonstration tokens size to 750.

	1x	2x	5x	10x
End-to-End without compression	298.6			
End-to-End with Selection-p		167.0 (1.8x)	81.6 (3.7x)	55.6 (5.3x)
LLMLingua per demonstrations set	-	0.82	0.82	0.81
Selection-p per demonstrations set	-	**0.68**	**0.67**	**0.67**

Table 7: **Latency(s) comparison on WSC in 750 tokens level with about 16 demonstrations.** We present the averaged complete end-to-end inference with and without Selection-p among four sets of demonstrations. Comparison is conducted with LLMLingua which also builds upon the LLaMA-2-7B backbone, with the averaged compression time of the in-context demonstrations being presented.

compression models, we only utilize the modified version of LLMLingua in the previous experiments to ensure a fair comparison.

In this section, we further analyze the performance by comparing our proposed method, equipped with the Budget Controller, with the whole LLMLingua to provide a comprehensive analysis. We follow the setting described in Section 4.3 and select the WSC task for our experiment. To illustrate, in the original demonstration set consisting of 16 demonstrations, the LLMLingua API retains only four demonstrations. This leads to two options with Selection-p: (1) continuously applying the 10x compression directly to the filtered set of four demonstrations and resulting in a final compression rate of 38x, and (2) adjusting the compression rate of Selection-p to achieve a final compression rate of 10x.

The result presented in Table 8 demonstrates the significant impact of the Budget Controller. Similar trends in performance for both Selection-p and LLMLingua are observed (i.e., a decrease in performance on the WSC task). Notably, the performance of Selection-p surpasses LLMLingua after equipping with the Budget Controller.

	WSC	rate
Selection-p	61.1	10x
LLMLingua	**63.0**	10x
Selection-p (+ Budget Controller)	**47.6**	10x
Selection-p (+ Budget Controller)	**57.0**	38x
LLMLingua (whole)	44.7	10x

Table 8: **Comparison with LLMLingua on Budget Controller.** Adopting different strategies in equipping Selection-p with Budget Controller, leads to the two different compression rates (rate) of 10x and 38x.

Furthermore, there is a disparity between the instructed compression rate and the actual compression rate in LLMLingua. The target size for compressed tokens is 75, while LLMLingua typically achieves an average compressed token size of around 192.1, which is more than 1.5 times higher than the desired rate across all classification tasks.

6 Conclusion

We introduce a simple yet effective self-supervised approach in context compression and conduct evaluation across 10 classification tasks in both few-shot and long-context settings. Our approach also demonstrated great transferability to both the open-

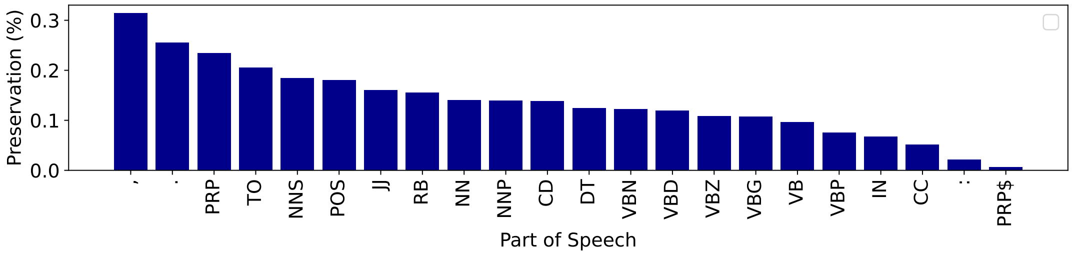

Figure 3: Analysis of the token preservation percentage with respect to different types of Part-of-Speech tags under 10x compression rate.

source (i.e. LLaMA-2-13B) and black-box models (i.e. ChatGPT and Gemini), with performance surpassing the existing state-of-the-art compression models. Analysis is also conducted among different compression rates and demonstration lengths. With a 10x compression rate, our model only shows a 0.8-point drop in performance across different traditional classification tasks with a 5.3x speedup. Through experiments in long-context settings, our work also presents the possibility of addressing the in-context learning issue of the recent long-context models. Both efficiency enhancement as well as performance preservation are shown in our model.

Acknowledgement This research has been made possible by the Hong Kong PhD Fellowship provided to Tsz Ting Chung and the Research Impact Fund project R6003-21 provided by the Research Grants Council of Hong Kong to Dit-Yan Yeung.

Limitations

Under the consideration of cost, we did not perform further analysis on other LLMs apart from ChatGPT and Gemini. In addition, our model which builds up LLaMA-2-7B does not achieve better latency than models like LLMLingua-2 and AutoCompressor. Under the ICL setting, minimal time is required for compression, leading to insignificance in end-to-end inference time. While AutoCompressor offers better latency, its soft compression nature limits its applicability to other LLMs. Overall, our experiments across various tasks and settings demonstrate better performance and transferability, with the benefits outweighing the latency issue.

References

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Longlora: Efficient fine-tuning of long-context large language models. *ArXiv preprint*, abs/2309.12307.

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.

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*.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan T. Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. Power-bert: Accelerating BERT inference via progressive word-vector elimination. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3690–3699. PMLR.

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

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.

Gyuwan Kim and Kyunghyun Cho. 2021. Length-adaptive transformer: Train once with length drop, use anytime with search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6501–6511, Online. Association for Computational Linguistics.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. 2022. Learned token pruning for transformers.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. 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, Singapore. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. 2022. AdapLeR: Speeding up inference by adaptive length reduction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–15, Dublin, Ireland. Association for Computational Linguistics.

Jesse Mu, Xiang Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, volume 36, pages 19327–19352. Curran Associates, Inc.

OpenAI. 2023. GPT-4 technical report. *ArXiv preprint*, abs/2303.08774.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 13937–13949.

C. Spearman. 1904. The proof and measurement of association between two things. *American Journal of Psychology*, 15:88–103.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models.

TogetherAI. 2023. Redpajama: An open dataset for training large language models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. 2023. Focused transformer: Contrastive training for context scaling.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, Singapore. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

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

A Examples Illustration of Compression Results

Figure 4 shows examples of two traditional classification tasks (i.e. Subj and WSC) with Selection-p under 10x compression, displaying both the compressed results and the original demonstration sets before compression. Additionally, Figure 5 illustrates an example for BANKING77 under 10x compression.

B Prompt of Evaluation on BANKING77

In our prompt to ChatGPT, we first list out all 77 labels and then provide a list of demonstrations. The template is detailed below,

Answer in activate_my_card or age_limit or apple_pay_or_google_pay or atm_support or automatic_top_up or balance_not_updated_after_bank_transfer or ... or wrong_amount_of_cash_received or wrong_exchange_rate_for_cash_withdrawal.

Context: <context>
Answer: <answer>

C Training Details

We use LLaMA-2-7B for compression (i.e. token selection). It takes roughly 50 hours on a single A100 GPU to train on 100M tokens from RedPajama.

D Detailed Latency Analysis

For the token-based compression models, the time needed for compression per demonstration set on the WSC task is presented in Table 9.

LLMLingua-2	Selection-p	LLMLingua
0.15	0.67	0.81

Table 9: Time needed for compression per demonstration set on the WSC task.

Under the ICL setting, the same demonstration set is used consistently, and compression on the demonstration set only needs to be computed once for all subsequent inferences. This contributed to the short compression time in the complete end-to-end process.

Considering that the time taken for compression on the demonstration set is negligible when compared to the time required for inference (approximately at a ratio of 0.01), the end-to-end inference time for the token-selection-based compression models is roughly the same. The end-to-end inference time result for the WSC task on LLaMA-2-7B is shown in Table 10. The table is presented in order of the inference time for clarity.

AutoCompressor	LLMLingua-2	Selection-p	LLMLingua
22.73	55.6	55.6	55.6

Table 10: End-to-end inference time on the WSC task.

[Subj] Original Demonstrations:
input: each of the principals has a radically different way of dealing with it .
type: objective

input: well-intentioned though it may be , its soap-opera morality tales have the antiseptic , preprogrammed feel of an after-school special .
type: subjective
...
input: an astonishing feat for a major star let alone a 27 year old from pickum , south carolina who only two years ago was sleeping in a cardboard box in the back alleys of detroit with her mother , connie , and her uncle clutch , while playing guitar on the streets for spare change .
type: objective

input: may is a young strange girl who had a very disturbed childhood and does not still know the meaning of true friendship or love .
type: objective

[Subj] Compressed demonstrations (10x):
input:-int it may- mor exc lord treasure planet- the ste moments- it thr en a r with he and l crowd gru a-erm own – into imposibly ris gar “ir wh sh de prere prede it mag m ro philosophvag rede thes v de spcer twz-passer aston fe two with cl for dist and still

---

[WSC] Original Demonstrations:
Question: In the sentence "James asked Robert for a favor but he was refused.", does the pronoun ’he’ refer to Robert?
Answer: no

Question: In the sentence "What about the time you cut up tulip bulbs in the hamburgers because you thought they were onions?", does the pronoun ’they’ refer to tulip bulbs?
Answer: yes
...
Question: In the sentence "When Mr. Bond , the veterinarian, came to look at the black horse that lay groaning on the grass, he felt him all over, and shook his head; one of his legs was broken.", does the pronoun ’his’ refer to the black horse?
Answer: no

Question: In the sentence "Sam took French classes from Adam , because he was eager to speak it fluently.", does the pronoun ’he’ refer to Adam?
Answer: no

[WSC] Compressed demonstrations (10x):
" pron:: " tul pron ’ tul: he pron ’ told P which P. He have pron ’ P a the would only Gru une pron pronI put cfr It pron refriger man pronJohn wheng. He very im Wainws d Fol he pron veterin gro his pron the repa pron w t win gro " he pron ’

Figure 4: Illustration of the compression result by Selection-p for Subj and WSC tasks under 10x compression rate. Compression is performed with 19 demonstrations for Subj while it is performed with 16 demonstrations for WSC with total sum of about 750 tokens respectively.

[BANKING77] Original Demonstrations:
Context: Why did using an ATM cause me to be charged an additional fee?
Answer: cash_withdrawal_charge

Context: I asked for a refund but its not here yet
Answer: Refund_not_showing_up

Context: is there a reason i need to verify top up
Answer: verify_top_up
...
Context: There is a payment on my card that I do not recognize. I’ve never seen the name on the transaction before.
Answer: card_payment_not_recognised

Context: I happened to forget my passcode
Answer: passcode_forgotten

Context: I made a cash withdrawal and it is still listed as a pending transaction.
Answer: pending_cash_withdrawal

[BANKING77] Compressed demonstrations (10x):
c_with asked ref Refnoting_up__recogn_____c wrong_rece_ transaction_chargtw unblock activ activ_not__fe_charg_tim: ex sho Please revert__ the card__wr__: I: pending__: the card_not_recogn: passf:_c

Figure 5: Illustration of the compression result by Selection-p for BANKING77 under 10x compression rate. Compression is performed with 27 demonstrations with total sum of about 750 tokens.