DAC: A Dynamic Attention-aware Approach for Task-Agnostic Prompt Compression

Anonymous ACL submission

Abstract

Task-agnostic prompt compression leverages the redundancy in natural language to reduce computational overhead and enhance information density within prompts, especially in long-context scenarios. Existing methods predominantly rely on information entropy as the metric to compress lexical units, aiming to achieve minimal information loss. However, these approaches overlook two critical aspects: (i) the importance of attention-critical tokens at the algorithmic level, and (ii) shifts in information entropy during the compression process. Motivated by these challenges, we propose a dynamic attention-aware approach for task-agnostic prompt compression (DAC). This approach effectively integrates entropy and attention information, dynamically sensing entropy shifts during compression to achieve fine-grained prompt compression. Extensive experiments across various domains, including LongBench, GSM8K, and BBH, show that DAC consistently yields robust and substantial improvements across a diverse range of tasks and LLMs, offering compelling evidence of its efficacy.

1 Introduction

Recent advent of In-Context Learning (ICL) (Brown, 2020; Dong et al., 2024), Chain-of-Thought (CoT) (Wei et al., 2022; Yao et al., 2024a,b), Retrieval Augmented Generation (RAG) (Lewis et al., 2020), and Autonomous Agent (Xi et al., 2023) technologies has significantly invigorated the landscape of applications based on Large Language Models (LLMs). While these methodologies have expanded the capabilities of LLMs by activating domain-specific knowledge or enhancing memory capacities, they also introduce the challenge of exceedingly long context lengths, which leads to a substantial increase in computation and memory consumption due to inherent self-attention mechanism.

Efficient LLM is the technology that aims to achieve computational efficiency while retaining performance, which is accomplished through various methods such as modifying model architecture (Sun et al., 2024), parameter quantization (Lin et al., 2024), key-value (KV) cache compression (Yang et al., 2024), and the utilization of soft prompts (Mu et al., 2024), among others. Despite the effectiveness of these methods, they often require modifications to the model, which is not feasible for black-box LLMs, such as those accessible only through APIs. In such cases, prompt compression, which seeks to shorten the prompt while preserving essential information, represents the most direct way.

Several studies consider essential information to be the parts most relevant to the question or task, thus proposing a task-aware manner for prompt compression (Jiang et al., 2024; Xu et al., 2023; Huang et al., 2024; Jung and Kim, 2024). Benefiting from the sparsity of question-related information in original prompts, these methods achieve significant performance on specific benchmarks with high compression ratios, even surpass the performance of the original prompts in some cases. However, these approaches are highly dependent on the type of downstream task, leading to the following limitations: i) the prompt needs to be repeatedly compressed in scenarios involving multiple questions or tasks, ii) it is challenging to define user's intent or questions in extended dialogic engagements.

Task-agnostic prompt compression aims to compress prompts relying on the self-information of the language without any additional clues (Li et al., 2023; Jiang et al., 2023; Pan et al., 2024). Previous works primarily utilize coarse-grained information entropy output by the logits layer of entire model for compression. They do not delve into the inner layers of LLMs for gathering finer-grained attention scores to enhance the compression process. In

addition, these entropy-based methods treat information entropy as static while ignoring the dynamic shift during the compression process. Some low-entropy tokens may become high-entropy tokens after their dependent tokens are compressed.

To address these challenges, we propose a dynamic attention-aware approach for task-agnostic prompt compression (DAC). Specifically, we propose a novel metric for prompt compression that effectively integrates information from both entropy and the attention scores, enabling the compression process to be conducted in an attention-aware manner. Additionally, we propose a dynamic approach to iteratively identify tokens with significant entropy shifts during the compression. This method aims to minimize information loss more precisely, thereby preserving the information density in the compressed prompts.

We evaluate the effectiveness of our approach across different domains, including LongBench for contextual understanding and GSM8K, BBH for reasoning and in-context learning. The experimental results demonstrate that our method significantly outperforms other entropy-based compression methods, achieving an improvement of 4.03 points on LongBench. Moreover, it surpasses current state-of-the-art (SOTA) approaches by an average score of 1.33 on LongBench. Our method also exhibits robust generalization ability, as evidenced by consistent performance improvements across a range of model series, from Qwen2 to LLaMA3.

2 Related Work

There are two primary approaches in prompt compression, which can be categorized based on the form of the compressed output: soft prompts and hard prompts. The soft prompts methods typically compress the original prompt into non-linguistic forms, such as special tokens or embeddings. Wingate et al. (2022) proposed optimizing the KL divergence between the answers generated by the original and compressed prompts to achieve soft prompt compression. Mu et al. (2024) addressed this challenge by introducing GIST tokens. Ge et al. (2023) propose ICAE, which pre-trained a compression model to transform prompts into compact memory slots that can be conditioned on by LLM. However, soft prompts often result in non-human-readable formats, leading to difficulties in interpretability of compressed content. Otherwise, these methods frequently require modifications of the model (pre-training or fine-tuning), making them unsuitable for black-box LLMs.

The hard prompts methods achieve compression by identifying and dropping low-information content from the original prompt. Li et al. (2023) first introduced using information entropy to measure the information content. They also considered the most effective lexical units for compression and determining using phrases through experiments. Similarly, LLMLingua (Jiang et al., 2023) employs information entropy as the metric, proposing the use of a budget controller to assign different compression rates to various parts of the prompt (instruction, demonstrations, and the question). Additionally, they introduced iterative token-level prompt compression, which segments the original prompt and then performs fine-grained token-level compression. These entropy-based methods effectively identify and compress low-information content in the original prompt, achieving satisfactory results in downstream tasks. However, they do not fully account for the information contained in the attention mechanism and the entropy shifts occurring during the compression process. In contrast, LLMLingua2 (Pan et al., 2024) takes a different approach by training a specialized classification model dedicated to prompt compression. The compression data used for training is distillated from a more powerful LLM (i.e. GPT4). However, powerful LLMs may not good at compression task, and training on a specific dataset may not generalize well to other tasks.

Our proposed DAC method falls into this category as well and more simular to entropy-based methods. Differently, DAC employs a novel metric that integrates information from both information entropy and the attention mechanism, and further introduces a dynamic approach for accurately identify low-information content. It is worth noting that KV cache compression yields comparable effects to prompt compression. For a comprehensive discussion of KV cache compression techniques, please refer to Appendix C, as these approaches are not directly related to our DAC methodology.

3 Preliminaries

3.1 Problem Formulation

We first formally define the compression process with target budget. Denote original input tokens as $\boldsymbol{x} = \{x_i\}_{i=1}^{L}$ and tokens after compression as

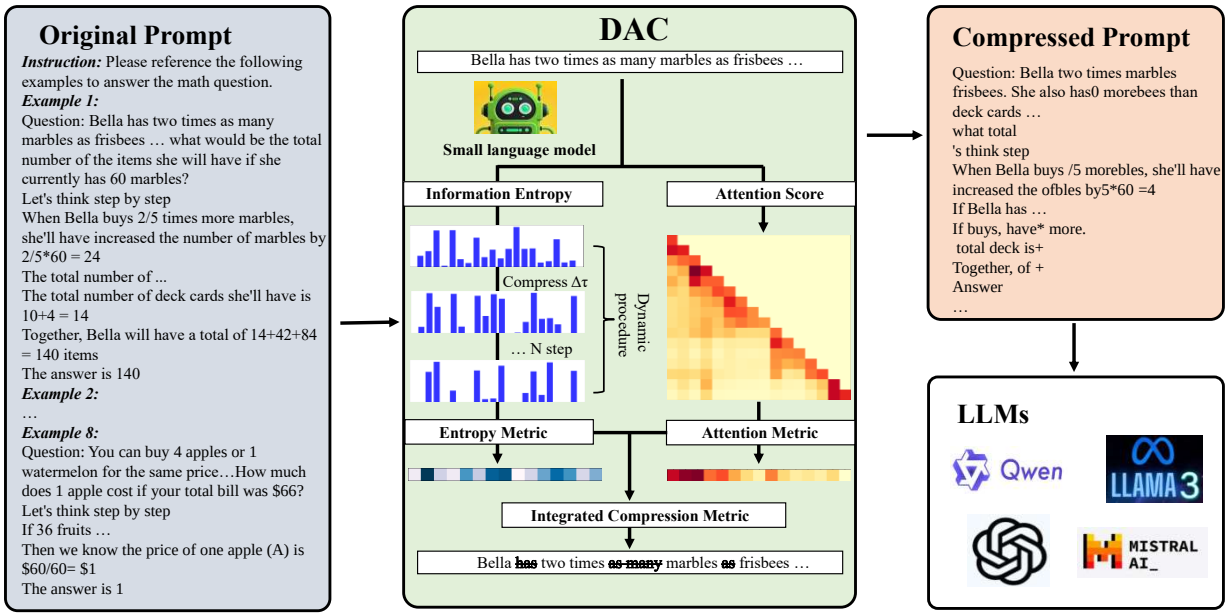

Figure 1: Framework of our proposed DAC for task-agnostic prompt compression.

$\widetilde{\boldsymbol{x}} = \{\widetilde{x}_i\}_{i=1}^{\widetilde{L}}$, where $\widetilde{\boldsymbol{x}}$ is a subset of $\boldsymbol{x}$. The compression rate can be derived from $\tau = \widetilde{L}/L$. Our goal is to find a subset chosen policy that the output of the generative process is comparable to the original prompt, which can be expressed by the formula:

$$\min_{x,\tau} \mathcal{D}(P(\widetilde{y}|\widetilde{x}), P(y|x)) \tag{1}$$

where function D(,) denotes the distance between two distributions such as KL divergence. $\widetilde{y}$ represents LLM generate results from the compressed context $\widetilde{\boldsymbol{x}}$, and y represents the original output derived from $\boldsymbol{x}$.

3.2 Information Entropy

From an information-theoretic perspective, an effective compression algorithm should strive to minimize the loss of information. The information entropy of the token can be quantified as the output distribution during its autoregressive generation. This can be expressed as follow:

$$I_t(x) = -\log_2 P(x_t|x_0, x_1, ..., x_{t-1}) \tag{2}$$

where $I_t(x)$ represents the information entropy of token x_t and $P(x)$ denotes the output probability while generating token x_t. Consequently, a token with a higher certainty in its probability distribution indicates a lower information entropy and thus conveys less information, which can be the guidance during compression.

3.3 Attention Scores

The attention mechanism is essential in the Transformer architecture, as it enables the model to focus on critical segments of the input sequence, thereby enhancing its capability to manage long-range dependencies. We denote query matrix as $Q \in R^{n\times d}$ and key matrix as $K \in R^{n\times d}$ in attention mechanism. Then the normalized attention matrix of the i-th layer and the j-th head can be expressed as $Softmax\left(\frac{Q_{ij}K_{ij}^{\top}}{\sqrt{d_h}}\right) \in R^{n\times n}$. Suppose that each element in this matrix is denoted as q_{uv}, then the accumulated attention score vector of this matrix can be calculated by:

$$F_{score}^{ij} = (s_1^{ij}, s_2^{ij}, \ldots, s_n^{ij}), s_v^{ij} = \sum_{u=1}^{n} q_{uv} \tag{3}$$

where s_v^{ij} denotes the accumulated attention score of v-th token in the i-th layer and the j-th head. All accumulated attention score vectors are aggregated into the final one by computing the mean across all layers and heads:

$$\overline{F_{score}} = \frac{1}{MN} \cdot \sum_{i=1}^{N}\sum_{j=1}^{M} F_{score}^{ij} = (\overline{s_1}, \overline{s_2}, \ldots, \overline{s_n}) \tag{4}$$

4 Methodology

In this section, we introduce the Dynamic Attention-aware Compression framework, as shown in Figure 1. Prior to describing the framework in detail, we first present two key observations that motivated our approach.

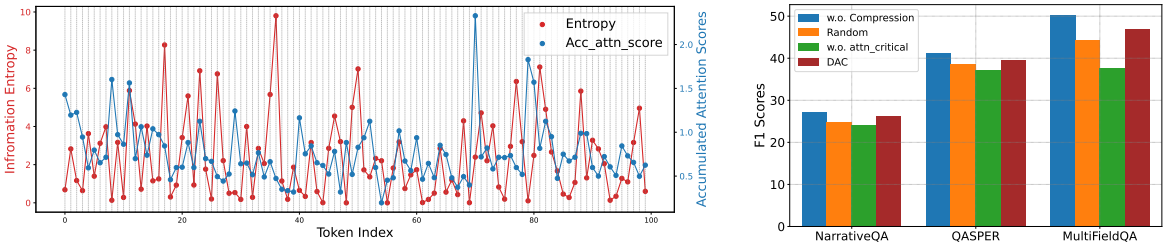

Figure 2: Left: The information entropy (red) and corresponding accumulated attention scores (blue) on one sequence. It reveals that attention-critical tokens (with high accumulated attention scores) do not necessarily possess high entropy. Right: The performance comparison across four methods: w.o. compression, random, w.o. attention-critical tokens and our DAC method. Without attention-critical tokens, the model's performance on all three datasets significantly deterioratesevenperforming worse than random selection methods. Our DAC method effectively addresses this issue by identifying the attention-critical tokens.

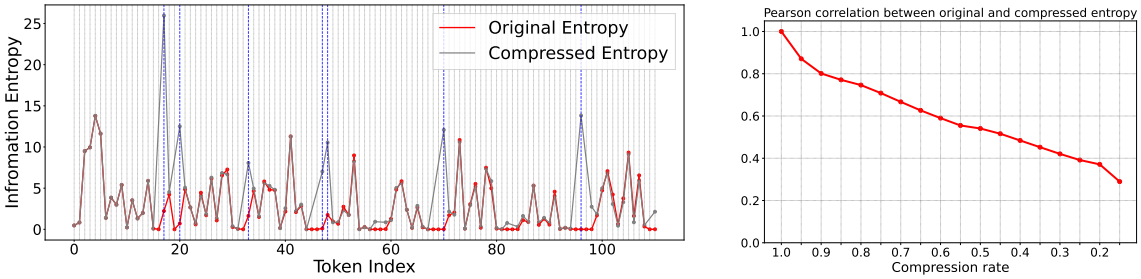

Figure 3: Left: Entropy differences highlight significant shifts in tokens whose preceding tokens were removed during compression. Right: Compression ratio inversely correlates with entropy differences.

4.1 Observations

Observation 1: Attention-critical Tokens Matter. We first observe the relationship between information entropy and accumulated attention score of each token throughout the inference process. In detail, we select Qwen2-7B as the base model, and employ a popular NarrativeQA benchmark. The visualization results on one of sequences are presented in the left of Figure 2. We initially identify tokens with higher accumulated attention scores (e.g., those exceeding a threshold of 1.0) as attention-critical tokens. Our observations reveal that a substantial number of these attention-critical tokens do not exhibit correspondingly high information entropy, suggesting the absence of a linear relationship between the two metrics. Quantitative analysis of 200 sequences from the NarrativeQA dataset also indicates an average Pearson correlation coefficient of 0.095 between the accumulated attention scores and information entropy. This finding implies that relying solely on information entropy as a criterion for compression might result in the exclusion of numerous attention-critical tokens.

We further analyzed the implications of ignoring attention-critical tokens. We conducted experiments on three single-document QA benchmarks: NarrativeQA, QASPER, and MultiFieldQA and compared four different methods using the F1 score as the evaluation metric. These methods included: w.o. compression, random select compression, w.o. attention-critical tokens and our DAC method. The compression rate was set to $\tau = 0.9$. The results are shown in the right of Figure 2. Our findings revealed that w.o. attention-critical tokens in prompt significantly degraded the model's answering performance, performing even worse than random select compression. A more effective perceptual compression scheme should integrate both information entropy and accumulated attention scores. The proposed DAC method achieves superior performance by implementing this integrated scheme.

Observation 2: Entropy Shift during Compression. Information entropy serves as a fundamental metric in past prompt compression methods. Consequently, it is imperative to analyze characteristics of the entropy changes during the compression process. To this end, Figure 3 (left) shows the entropy before and after compression with the ratio

of 0.9. The figure reveals that even with a relatively low compression ratio, a significant proportion of tokens exhibit substantial shifts in entropy (marked by blue vertical lines). This phenomenon implies that past non-dynamic approach may ignore these critical entropy shifts, thereby compromising the effectiveness of the compression method. It is also noted that, for the majority of these tokens exhibiting substantial shifts, their preceding tokens have been dropped during compression. This observation also inspires an efficient way to address the entropy shift.

Figure 3 (right) presents the Pearson correlation between the original and compressed entropy across varying compression ratios. The results demonstrate that as the compression ratio increases, these shifts become more pronounced. This finding illustrates the necessity of developing a dynamic compression method that can adaptively respond to entropy changes, thus enabling a higher compression rate while maintaining LLM performance.

4.2 Dynamic Attention-aware Compression

Based on observation 1, we first propose a metric that integrates both information entropy and accumulated attention scores for prompt compression. We explore two fusion strategies: additive fusion (Eq.(5)) and multiplicative fusion (Eq.(6)). In the additive fusion, a parameter α is introduced to balance the contributions of the information from both sides, and the optimal α is determined through experimentation.

$$M_t^a = (1-\alpha)\cdot I_t(x) + \alpha \cdot \overline{s_t} \tag{5}$$

$$M_t^m = I_t(x) \cdot \overline{s_t} \tag{6}$$

Based on the insights derived from observation 2, we propose a dynamic method for prompt compression. Specifically, rather than completing the compression in a single step based on the metric, we divide the process into multiple stages. At each stage, we recalculate the information entropy for usage in the current stage to reduce the impact of the significant entropy shifts observed in observation 2.

Additionally, since observation 2 revealed that most tokens with significant entropy shifts have their preceding tokens compressed, we address this issue by limiting the compression of consecutive tokens. This strategy offers two primary benefits: it helps prevent unexpected compression due to the change of entropy within the same compression stage, and it provides a dynamically adjusted compression rate. That is, the compression rate for each round is dynamically adjusted based on the compression pattern from the previous round, expressed as:

$$\triangle\tau = \tau^{1/D} + \triangle P \tag{7}$$

$\triangle\tau$ and D in the equation represents the compression rate in current stage and the dynamic iterations respectively, while $\triangle P$ denotes the percentage of tokens retained in the previous stage due to the limiting of consecutive compression.

After getting the compression rate in current stage, we first use a percentile-based filtering approach for compression. In detail, the threshold of integrated metric can be calculated by:

$$T_{\triangle\tau} = np.percentile([M_0], \ldots, [M_n], \triangle\tau) \tag{8}$$

Next, We filter out those tokens whose fusion metric exceeds the threshold or whose preceding token has already been compressed, and add them to the set $\widetilde{\boldsymbol{x}}$ as the compression result in the current stage, denoted as:

$$\widetilde{\boldsymbol{x}} = \{\widetilde{x}_i \mid M(\widetilde{x}_i) \geq T_{\triangle\tau} \vee \widetilde{x}_{i-1} \notin \widetilde{\boldsymbol{x}}\} \tag{9}$$

The compression procedure is completed by performing above steps in multiple stage upon a preset dynamic iterations. The overall procedure of DAC can be referred to Algorithm 1.

5 Experiments

5.1 Experiments Setup

Datasets We comprehensively evaluate the utility preservation of LLM with different prompt compression methods from two different aspects: (i) Contextual Understanding: we utilize four types of tasks from the LongBench (Bai et al., 2024): Single-document QA, Multi-document QA, Summarization, and Few-shot Learning. Each task category includes three specific benchmarks. (ii) Reasoning and In-context Learning: We employ GSM8K (Cobbe et al., 2021) and Big Bench Hard (BBH) (Suzgun et al., 2023) datasets. Consistent with prior studies, we adopt Exact Match (EM) as evaluation metric.

Baselines We take three most effective prior studies as baselines:

Selective-Context (Li et al., 2023): Selective-Context employs the information entropy of the

Algorithm 1 Dynamic Attention-aware Prompt Compression (DAC).

Input: Prompt to compress $\boldsymbol{x} = \{x_i\}_{i=1}^{L}$; Target compression rate τ; Dynamic iterations D; A small language model SLM;

Calculate the accumulated attention score of each token via Eq.(3) and Eq.(4)
for $i = 1, 2, \ldots, D$ **do**
  Calculate the compression rate $\triangle\tau$ via Eq.(7)
  Update the information entropy by SLM and Eq.(2)
  Calculate the integrated metric M_i for each token via Eq.(5) or Eq.(6)
  Find the $\triangle\tau$-percentile threshold $T_{\triangle\tau}$ via Eq.(8)
  for j in range($\widetilde{\boldsymbol{x}}$) **do**
    if $M(\widetilde{x}_j) < T_{\triangle\tau}$ **then**
      if $\widetilde{x}_{j-1} \notin \widetilde{\boldsymbol{x}}$ **then**
        $P = P + 1$
      else
        $\widetilde{\boldsymbol{x}}$.delete($\widetilde{x}_j$)
      end if
    end if
  end for
  Update differential compression ratio $\triangle P$ by $P/\text{len}(\boldsymbol{x})$
end for

Output: The compressed prompt $\widetilde{\boldsymbol{x}}$.

most effective lexical units (phrases) as the mertic for compression.

LLMLingua (Jiang et al., 2023): LLMLingua first involves a budget controller that assigns different compression strategies to various components of the prompt (instruction, demonstrations, and the question). Subsequently, it performs iterative token-level prompt compression, by which the prompt is segmented, and each segment is compressed sequentially based on the information entropy of the tokens.

LLMLingua2 (Pan et al., 2024): LLMLingua2 first distills compressed data from GPT4 using MeetingBank dataset. It then trains a specialize small model for prompt compression based on a transformer encoder architecture.

To ensure a fair comparison, all compression rates in experiments are actual compression rates. For those methods that dynamically adjust the compression rate based on the input prompt, we adjust the target compression rate to ensure that the final number of compressed tokens is approximately consistent across different methods.

Implementation Details Our experiments are conducted on two kinds of LLMs. The main results are performed on the Qwen2 series models[1]. Specifically, the SLM used for compression is Qwen2-0.5B, and the LLM used for evaluation is Qwen2-7B. Additionally, to study DAC's adaptation to various LLMs, we also use the LLaMA3 series models[2] to measure the impact of different models (see Section 5.5). For these experiments, the SLM is LLaMA 3.2-1B, and the LLM is LLaMA 3.1-8B. The Dynamic times D in algorithm is set by $D = L_{input}/100$, with the maximum value of 15. All experiments are conducted on an NVIDIA A800 GPU. We use greedy decoding to ensure the stability of the experimental results. The experimental environment includes the following configurations: CUDA Version 12.0, PyTorch version 2.4.0, and HuggingFace's Transformers[3] with version 4.45.1.

[1]https://github.com/QwenLM/Qwen

5.2 Fusion Strategies

In section 4.2, we discussed two fusion strategies for integrating information entropy and accumulated attention scores: additive fusion with parameter α and multiplicative fusion. In this section, we conduct experiments to find the appropriate fusion strategy. We consider five situations: additive with $\alpha = 0.2, 0.4, 0.6, 0.8$ and multiplicative. The experiments are conducted on single-document QA task from the Longbench, and the results are shown in Figure 4. The results show that the best performance is achieved with additive fusion with $\alpha = 0.8$. Therefore, subsequent experiments will adopt this fusion strategy.

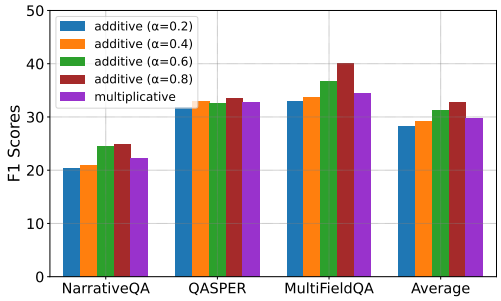

Figure 4: The performance of DAC method with different fusion strategies.

5.3 Main Results

Table 1 shows the comparative performance of proposed DAC method against the baselines on Longbench with two compression rates ($\tau = 0.2$ and $\tau = 0.5$). Overall, DAC method outperforms the baselines in the task types of single-document QA, summarization, and few-shot learning, as well as in the overall average score.

[2]https://github.com/meta-llama/llama3
[3]https://github.com/huggingface/transformers

Methods	LongBench																
	Single-Doc QA				Multi-Doc QA				Summarization				Few-shot Learning				All
	Nar.QA	Qasper	Mul.QA	AVG	Hot.QA	2Wi.QA	Musique	AVG	GovRe.	QMSum	M.News	AVG	TREC	Tri.QA	SAMSum	AVG	AVG
Compression rate τ = 0.5																	
Selective-Context	18.75	35.07	28.90	27.57	38.30	36.07	21.28	31.88	25.83	24.11	24.77	24.90	29	81.80	38.07	49.62	33.50
LLMLingua	20.36	28.26	27.69	25.44	40.11	35.69	21.00	32.27	26.19	23.5	24.71	24.80	40	78.07	39.15	52.41	33.73
LLMLingua-2	24.25	35.22	38.70	32.72	43.61	38.11	26.80	**36.17**	26.15	25.54	25.78	25.82	35	77.6	40.37	50.99	36.43
DAC	24.85	33.46	40.12	**32.81**	42.37	39.57	22.44	34.79	30.4	25.95	25.77	**27.37**	50	80.00	38.14	**56.05**	**37.76**
Compression rate τ = 0.2																	
Selective-Context	16.83	29.25	25.57	23.88	36.58	33.09	18.08	29.25	22.3	23.57	21.77	22.55	21.5	75.92	37.67	45.03	30.18
LLMLingua	18.35	21.78	24.65	21.59	38.43	32.85	21.95	31.08	22.91	22.38	22.5	22.60	30	77.83	36.07	47.97	30.81
LLMLingua-2	19.47	30.45	25.86	25.26	40.26	33.32	21.85	**31.81**	21.56	24.32	22.48	22.79	24	78.92	33.23	45.38	31.31
DAC	18.16	27.84	30.18	**25.39**	38.68	31.72	22.49	30.96	25.73	23.32	23.59	**24.21**	31	80.41	38.47	**49.96**	**32.63**
Original Prompt	27.08	41.1	50.23	39.47	56.92	55.65	36.56	49.71	35.2	26.02	24.35	28.52	78	87.19	45.03	70.07	46.94
Zero-Shot	14.92	18.09	19.11	17.37	17.11	25.88	6.35	16.45	19.92	10.27	8.55	12.91	3	73.43	30.89	21.59	17.08

Table 1: Performance of different methods under different compression rates on the LongBench (Qwen2-7B). The results of original prompt and zero-shot experiments are also shown at the bottom.

Table 2 presents the comparative results on the GSM8K and BBH datasets. It can be seen that the DAC method achieves the best performance in most cases. With the compression rate $\tau = 0.5$, the DAC method's performance on GSM8K decreases by only 0.84 compared to the original prompt, demonstrating its performance retention in reasoning tasks. We also notice that although the DAC method is behind LLMLingua by 0.67 on the BBH dataset with $\tau = 0.5$, it outperforms LLMLingua by 0.8 with $\tau = 0.2$. This demonstrates that the introduction of the dynamic mechanism of information entropy enables DAC to perform better with high compression rates.

The experimental results also suggest that compression methods based on information entropy may be more generalized. Training specialized small models for compression, such as LLMLingua2, may suffer from reduced generalization due to limitations in the training datasets. For instance, while LLMLingua2 performs very well on LongBench, especially for multi-document QA tasks, it falls short compared to entropy-based method on GSM8K and BBH datasets.

Methods	GSM8K	BBH
Compression rate τ = 0.5		
Selective-Context	61.33	50.07
LLMLingua	72.86	**54.98**
LLMLingua-2	67.85	47.74
DAC	**74.37**	54.31
Compression rate τ = 0.2		
Selective-Context	60.12	46.66
LLMLingua	65.73	49.61
LLMLingua-2	66.49	44.16
DAC	**67.85**	**50.41**
Original Prompt	75.21	60.70
Zero-Shot	42.61	37.75

Table 2: Performance of different methods under different compression rates on the GSM8K and BBH dataset (Qwen2-7B).

5.4 Ablation Study

To validate the effectiveness of each component in the proposed DAC method, we conducted ablation studies with three different configurations: *DAC w/o Attention-aware Metric* indicates that the accumulated attention score is not involved, and only information entropy is used as the metric for compression. *DAC w/o Dynamic Procedure* indicates that the dynamical observation of entropy shift is not used, and the entropy is calculated only once in the beginning instead. *DAC w/o Limiting Consecutive Compression* indicates that consecutive tokens are allowed to be compressed in same stage of dynamic procedure. The results are shown in Table 3.

First, it can be observed that the F1 score of Single-Doc QA drops the most when DAC is w/o Attention-aware Metric. This significant decline occurs when only information entropy is used, without considering attention between tokens. This is consistent with our observation 1, which states that not only should information entropy be used as a guideline for compression, but it is also crucial to retain tokens that are important in the attention mechanism. A metric that can integrate both high-level information entropy and low-level information in algorithm itself can significantly enhance

	Single-Doc QA
DAC	32.81
- w/o Attention-aware Metric	28.16
- w/o Dynamic Procedure	29.84
- w/o Limiting Consecutive Compression	31.88

Table 3: Ablation study on single-document QA with compression rates $\tau = 0.5$.

Methods	LongBench				ALL
	Sin.QA	Mul.QA	Summ.	Few.	AVG
Selective-Context	26.40	24.78	21.98	55.47	32.16
LLMLingua	27.81	24.68	**23.13**	57.63	33.31
LLMLingua-2	32.51	27.93	22.74	56.08	34.82
DAC	**32.68**	**28.08**	22.92	**58.01**	**35.42**
Original Prompt	37.35	36.36	27.39	71.01	43.03
Zero-Shot	14.83	17.23	13.61	42.29	21.99

Table 4: The comparision results on LongBench using the LLaMA3 series models (compression rates $\tau = 0.5$)

the performance. Then compared with DAC w/o Dynamic Procedure, it reveals that the introduced dynamic procedure can identify essential information during the compression process, which would be missed without dynamic detection of entropy. Finally, it also can be found that DAC w/o Limiting Consecutive Compression will degrade the performance slightly. We conjecture that this could be due to the inappropriate compression of a subsequent token, which was caused by the dropping of its preceding token and then resulting entropy shift.

5.5 Different Models

To ensure the effectiveness of our method across different model types, we conducted experiments on other models. Specifically, here we use LLaMA 3.2-1B as the SLM for compression and LLaMA 3.1-8B as the LLM for evaluation. The experimental results are shown in Table 4. For simplicity, we report the average scores across different task types. The experimental results are shown in Table 4. It can be observed that DAC also demonstrates excellent performance on LLaMA3 series models, achieving state-of-the-art results in most task types and in the overall average score.

5.6 Overhead Analysis

We analyzed the overhead introduced by compression using different methods. Specifically, the profilling of overhead is conducted on a random sample from the GovReport benchmark, which contains 12,908 tokens of prompt. The length of the output tokens is set to 500, and the compression rate is 0.2. We followed the experimental setup for section 5.3, where the SLM is Qwen2-0.5B and the LLM is Qwen2-7B.

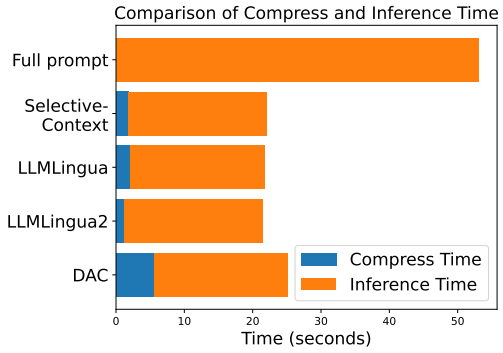

Figure 5: The comparison of compress and inference time using different methods

For each compression method, we plot the time taken for compression and the actual inference time. We also record the time required for using the full prompt. The results are shown in Figure 5. As can be seen, the DAC method, due to the introduction of the dynamic procedure, takes longer time in compression compared to other methods, as indicated by blue in the bar chart. However, this additional time is quite small when compared to the time saved by compression (compared to the full prompt). Additionally, it is worth noting that the parameter of LLM in production is often much larger than 7B, which further amplify the benefits of compression in terms of reduced inference time and memory.

6 Conclusion

This paper addresses task-agnostic prompt compression by proposing a dynamic attention-aware method. This approach aims to overcome the limitations identified in existing work, achieving fine-grained compression through the integration of information across different levels. We conduct extensive experiments and analyses across various domains, including contextual understanding, reasoning, and in-context learning. Our approach outperforms strong baselines across various domains and different series of LLMs, while introducing only acceptable additional overhead. The results indicate significant practical implications of our method for enabling LLMs to save computational costs and handle longer contexts effectively.

Limitations

There are also some limitations in our approach: (1) The current implementation of DAC requires obtaining attention matrices from all layers and heads, which necessitates the development of a method to identify the most representative attention matrices for more efficient information fusion. Additionally, DAC is not compatible with high-efficiency attention methods (e.g., Flash Attention) as it does not require calculating attention scores. The application of DAC on such methods will result in additional attention score calculations. (2) While the existing dynamic procedure supports adjusting the number of dynamic iterations based on context length, it hits an upper limit when the context becomes excessively long. This can lead to performance degradation as the granularity of perception becomes coarser. A potential solution could involve developing a method that senses information density to adaptively adjust the number of dynamic iterations, thereby maintaining performance even with very long contexts.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128.

Tao Ge, Jing Hu, Haixun Wang, Si-Qing Chen, and Furu Wei. 2023. In-context autoencoder for context compression in a large language model. In *ICLR 2024*.

Xijie Huang, Li Lyna Zhang, Kwang-Ting Cheng, Fan Yang, and Mao Yang. 2024. Fewer is more: Boosting math reasoning with reinforced context pruning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13674–13695, Miami, Florida, USA. Association for Computational Linguistics.

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.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Hoyoun Jung and Kyung-Joong Kim. 2024. Discrete prompt compression with reinforcement learning. *IEEE Access*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

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.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023. Scissorhands: Exploiting the persistence of importance hypothesis for llm kv cache compression at test time. In *Advances in Neural Information Processing Systems*, volume 36, pages 52342–52364. Curran Associates, Inc.

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

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. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981, Bangkok, Thailand. Association for Computational Linguistics.

Luohe Shi, Hongyi Zhang, Yao Yao, Zuchao Li, and Hai Zhao. 2024. Keep the cost down: A review on methods to optimize llm' s kv-cache consumption. *Preprint*, arXiv:2407.18003.

Yutao Sun, Li Dong, Yi Zhu, Shaohan Huang, Wenhui Wang, Shuming Ma, Quanlu Zhang, Jianyong Wang, and Furu Wei. 2024. You only cache once: Decoder-decoder architectures for language models. *arXiv preprint arXiv:2405.05254*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *Preprint*, arXiv:2310.04408.

Dongjie Yang, Xiaodong Han, Yan Gao, Yao Hu, Shilin Zhang, and Hai Zhao. 2024. PyramidInfer: Pyramid KV cache compression for high-throughput LLM inference. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3258–3270, Bangkok, Thailand. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024a. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Yao Yao, Zuchao Li, and Hai Zhao. 2024b. GoT: Effective graph-of-thought reasoning in language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2901–2921, Mexico City, Mexico. Association for Computational Linguistics.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang "Atlas" Wang, and Beidi Chen. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 34661–34710. Curran Associates, Inc.

A Why Small Models Can Be Used for Compression

Previous works have typically employed a small language model for compression but has not extensively discussed the underlying reasons. In this section, we provide empirical evidence to support why the entropy from a smaller model can effectively aid a larger model in identifying salient information.

We conducted experiments on the ArXiv corpus (Li et al., 2023) using various sizes of the Qwen2 models, ranging from 0.5B to 72B parameters, to analyze their entropy outputs. For clearer visualization, we focused on the initial tokens, as illustrated in Figure 8. Despite substantial differences in model size, all models exhibit remarkable consistency in their entropy across different texts. Figure 6 presents the quantitative results of entropy similarities using Pearson correlation. Here, we consider the entropy output from Qwen2-72B as the ground truth (GT) and compare it with the entropy outputs from smaller models. The x-axis represents different articles. The data show that while the similarity to GT increases with model size, even the smallest 0.5B model achieves an average similarity of 0.835. The consistent entropy patterns across different model sizes demonstrate that the critical information captured by large models can be efficiently approximated by smaller models.

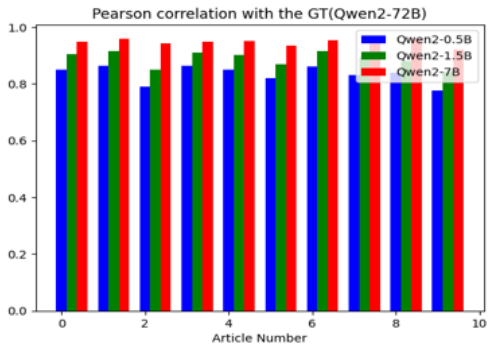

Figure 6: Quantitative analysis of the entropy similarities between different model parameter sizes (Qwen2 series models).

B Impact of compression rate

To assist users in achieving a balanced trade-off between utility preserving and efficiency, we present a detailed analysis of how various compression rates affect the performance of our DAC method. The experiments are conducted on the GSM8K dataset,

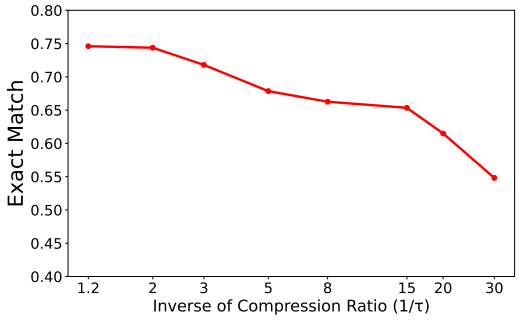

Figure 7: The performance of DAC at various compression rates on GSM8K (Qwen2-7B).

and the results are shown in the Figure 7. It can be observed that when the compression rate exceeds 0.5 ($1/\tau < 2$), the model's utility preservation is relatively good. For scenarios that prioritize low compression rates and can tolerate a certain degree of performance degradation, setting $1/\tau$ to around 8 offers a feasible solution.

C KV Cache Compression

Another line of related work is KV cache compression (Shi et al., 2024), as we consider attention mechanism in DAC method. Many previous studies about KV cache compression have analyzed the characteristics in attention matrices of LLM. H_2O (Zhang et al., 2023) observed that the accumulated attention scores of all tokens follow a power-law distribution, indicating that only a small subset of tokens is highly significant in the generation. Scissorhands (Liu et al., 2023) revealed the 'persistence of importance', indicating that tokens identified as important in initial remain significant throughout subsequent stages of inference. PyramidInfer (Yang et al., 2024) further explores the distinct attention characteristics across different layers within LLM, and identified that deeper layers exhibit greater redundancy.

The insights from these studies have inspired us to integrate information from the attention mechanism into our prompt compression methods. Nevertheless, the main objectives of these prior works are distinct from those of our research.

D Compression Case Study

We present various compresstion examples in Figure 9 and Figure 10 using DAC. In each example, tokens preserved at a compression rate of 0.2 are highlighted in dark red, while those preserved at a compression rate of 0.5 are shown in light red.

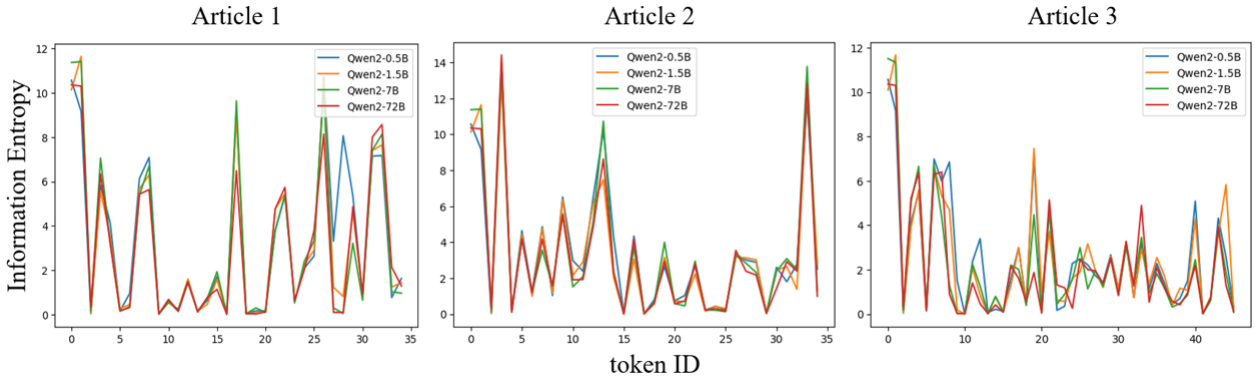

Figure 8: Visualization of the entropy similarities between different model parameter sizes (Qwen2 series models).

Original Prompt

Question: A group of 4 fruit baskets contains 9 apples, 15 oranges, and 14 bananas in the first three baskets and 2 less of each fruit in the fourth basket. How many fruits are there?
Let's think step by step
For the first three baskets, the number of apples and oranges in one basket is 9+15=24
In total, together with bananas, the number of fruits in one basket is 24+14=38 for the first three baskets.
Since there are three baskets each having 38 fruits, there are 3*38=114 fruits in the first three baskets.
The number of apples in the fourth basket is 9-2=7
There are also 15-2=13 oranges in the fourth basket
The combined number of oranges and apples in the fourth basket is 13+7=20
The fourth basket also contains 14-2=12 bananas.
In total, the fourth basket has 20+12=32 fruits.
The four baskets together have 32+114=146 fruits.
The answer is 146

Figure 9: Cases study on GSM8K dataset.

Original Prompt

Distinguish deductively valid arguments from formal fallacies.
Q: "It is not always easy to see who is related to whom -- and in which ways. The following argument pertains to this question: To begin with, Lesley is a close friend of Fernando. Moreover, being a close friend of Fernando or a schoolmate of Lowell is sufficient for being a great-grandfather of Leroy. It follows that Lesley is a great-grandfather of Leroy."
Is the argument, given the explicitly stated premises, deductively valid or invalid?
Options:
- valid
- invalid
A: Let's think step by step.
(1) Lesley is a close friend of Fernando: Lesley = friend(Fernando).
(2) Being a close friend of Fernando or a schoolmate of Lowell is sufficient for being a great-grandfather of Leroy: If X = friend(Fernando) OR SCHOOLMATE(Lowell), then X = great-grandfather(Leroy).
Hypothesis: Does it follow that Lesley is a great-grandfather of Leroy: Lesley = great-grandfather(Leroy)?
Let's see whether the Hypothesis can be deduced from the arguments (1) and (2) by logical reasoning?
By (1), we have Lesley = friend(Fernando). By (2), we have if Lesley = friend(Fernando), then Lesley = great-grandfather(Leroy).
So, it is true that Lesley is a great-grandfather of Leroy. So the answer is valid.

Figure 10: Cases study on formal_fallacies of BBH dataset.