ChunkOut: Information-Sufficient Token Pruning for Efficient Prompt Compression

Anonymous ACL submission

Abstract

Large language models (LLMs) increasingly rely on long prompts or retrieved contexts, driving up inference latency and cost. We observe that tokens whose *forward probability* is already high given the preceding context contribute little additional information, as the model has effectively encoded their content in its hidden state. Leveraging this **information-sufficiency** insight, we introduce CHUNKOUT, a model-agnostic algorithm that scores each token with its next-token likelihood and simply drops those above a threshold CHUNKOUT requires no extra training, incurs $\mathcal{O}(n)$ overhead, and can be plugged into any frozen LLM. Across QA and summarization benchmarks, it trims **50**% of prompt tokens while maintaining (and occasionally improving) task accuracy, outperforming prior compression baselines by up to **5**% pp. CHUNKOUT offers a principled yet lightweight path toward faster, cheaper, and greener LLM inference.

1 Introduction

State-of-the-art language models now tackle tasks by ingesting ever-longer prompts—chat histories, retrieved documents, or chain-of-thought exemplars. While larger contexts improve quality, they also inflate computation, latency, and energy consumption. The question, therefore, is *how much of a prompt is actually needed* once the model has already internalised its essential content.

Transformer hidden states are predictive: a single vector often forecasts several upcoming tokens (Pal et al., 2023; Din et al., 2024). Complementing this, Li et al. (2024) show that, after training, an LLM "glues" certain tokens together, assigning uniformly high probabilities to entire multi-token chunks and creating plateau-shaped likelihood regions. If token x_i can be guessed with high probability p_i from its prefix $x_{<i}$, then the semantic content of x_i is already encoded in the context representation, so removing it incurs little information loss.

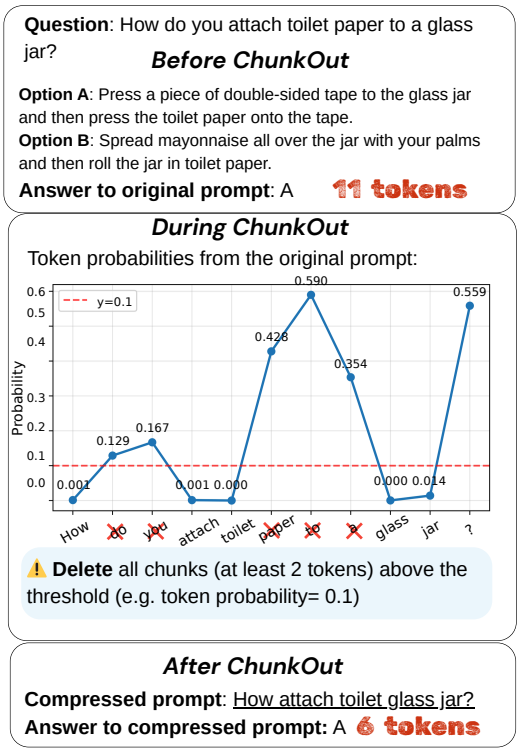

Figure 1: Overview of CHUNKOUT. Rather than deleting individual high-probability tokens, our approach targets contiguous chunks of high-probability tokens.

We operationalise this intuition with CHUNKOUT, a two-line procedure: First, run the frozen LLM once to collect next-token probabilities $\{p_i\}_{i=1}^{n}$. Second, remove every contiguous chunk of tokens where each token's next-token probability p_i exceeds a user-chosen threshold τ. No gradients, distillation, or auxiliary models are required; token scoring is parallel and linear in sequence length. The retained subsequence acts as a compressed prompt that the same solver model can process at full fidelity. Our results demonstrate that CHUNKOUT enables significant prompt reduc-

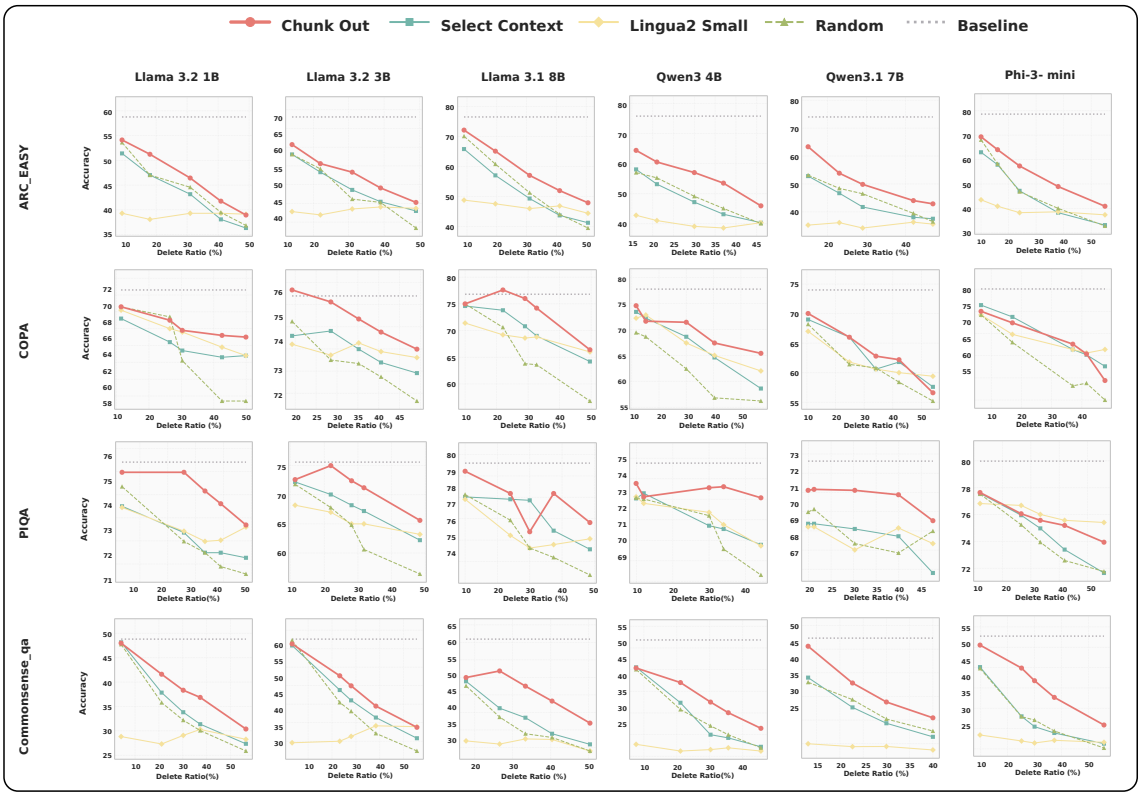

Figure 2: Question-Answering results across all models and tasks.

tion across diverse NLP benchmarks, preserving accuracy and summary quality without sacrificing performance. CHUNKOUT offers a plug-and-play route to faster, cheaper inference, complementing orthogonal advances in speculative decoding and model compression.

2 Related Work

Prompt Compression. Prompt compression methods aim to reduce input length while preserving task-relevant information, and can be broadly categorized into soft and hard approaches. Soft prompt compression methods replace the original input with learned representations. Some methods use learned vector tokens (Wingate et al., 2022) or task-specific gist prefixes (Mu et al., 2024) to encode prompt semantics, while others generate summary vectors from long contexts via recursive distillation (Chevalier et al., 2023) or encode input into memory slots with lightweight autoencoders (Ge et al., 2024). Hard prompt compression, the focus of this work, reduces prompt length by selectively removing tokens. Some methods score tokens by self-information to filter redundant content (Li et al., 2023), while others apply iterative entropy-based pruning with budget control and alignment techniques (Jiang et al., 2023). More recent methods frame token selection as a supervised task using LLM-distilled labels to train a compression classifier (Pan et al., 2024).

Future Token Prediction. While prompt compression focuses on removing tokens *after* observing their likelihoods, a complementary line of work shows that *hidden states themselves already encode information about multiple upcoming tokens*. Early "lens" probes revealed that intermediate activations can be linearly decoded into next-token distributions (nostalgebraist, 2020; Belrose et al., 2023). Going further, Pal et al. (2023) and Din et al. (2024) demonstrate that single hidden vectors often predict phrases several steps ahead, suggesting that transformers implicitly "plan" continuations long before generation. Mechanistic studies disentangle whether this foresight is deliberate caching or incidental "breadcrumbs": synthetic tasks show explicit pre-caching when future information is required, whereas natural text appears mostly myopic (Wu et al., 2024). Practical systems exploit these latent predictions to accelerate decoding: Medusa attaches parallel heads to output two–four tokens per step without quality loss (Cai et al., 2024), and lightweight re-ranking can push a frozen

model to emit coherent multi-token chunks (Samragh et al., 2025). Together, these findings motivate our method: if a token's content is already recoverable from its context's hidden state, it can be *pruned* without harming downstream performance, enabling more aggressive yet faithful compression.

3 Methodology

3.1 Problem Definition

Let a prompt be a sequence of n input tokens $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ fed to a frozen language model $g(\cdot)$. Given a downstream solver $f(\mathbf{x})$ (e.g. multiple-choice QA), our goal is to build the *shortest* subsequence $\tilde{\mathbf{x}} \subseteq \mathbf{x}$ such that

$$f(\tilde{\mathbf{x}}) = f(\mathbf{x}).$$

CHUNKOUT meets this target through a single forward pass that (a) scores every token and (b) deletes only *contiguous spans* whose tokens are all highly predictable.

3.2 Step 1: Token Scoring

For each position $i \in [1, n]$ we query the model once in the forward direction to obtain the next-token likelihood

$$p_i = g[x_i \mid x_{<i}]. \tag{1}$$

A large p_i means x_i carries little new information beyond its prefix $x_{<i}$; a small p_i signals that x_i is informative or surprising. This pass is embarrassingly parallel and costs $\mathcal{O}(n)$.

3.3 Step 2: Chunk Identification

Fix a user-chosen sparsity threshold $\tau \in (0, 1)$. We sweep the scored sequence left-to-right and group *maximal* contiguous runs of tokens whose probabilities all exceed τ:

$$S_j = \{ x_s, \ldots, x_e \mid p_k > \tau \text{ for } k \in [s, e], \\ p_{s-1} \leq \tau,\ p_{e+1} \leq \tau \}.$$

Only spans of length at least two are retained as deletion candidates:

$$|S_j| \geq 2.$$

3.4 Step 3: Prompt Reconstruction

The compressed prompt is the original sequence with every candidate span removed:

$$\tilde{\mathbf{x}} = \mathbf{x} \setminus \left(\bigcup_j S_j \right). \tag{2}$$

Because both the scoring and the linear scan are $\mathcal{O}(n)$, CHUNKOUT adds negligible overhead and introduces no additional parameters or training.

4 Results and Analysis

4.1 Experimental Setup

We evaluate our proposed CHUNKOUT method across a variety of pre-trained language models of different sizes: LLaMA-3.2 1.5B (Grattafiori et al., 2024), LLaMA-3.2 3B, LLaMA-3.1 8B, Qwen-3 4B (Yang et al., 2025), Qwen-3.1 7B, and Phi-3-mini (Abdin et al., 2024). Experiments cover both question answering and summarization benchmarks. For prompt deletion, we compare against three baselines: Selective Context (Li et al., 2023), which removes low-information tokens based on self-information computed by an LLM; LLMLingua2 (Pan et al., 2024), a supervised Transformer encoder trained for token retention; and random deletion.

4.2 Document-Summary

Main Results In the summary task setting, we conduct experiments on the CNN/DailyMail dataset (Nallapati et al., 2016), evaluating with ROUGE (Ganesan, 2018) and BERTScore (Zhang et al., 2020). As shown in Table 1, we set the threshold τ to 0.5 for all models and compare all baselines under the same compression ratio. **ChunkOut** achieves the best overall performance across all models and evaluation metrics.

Performance with Compression Rates CHUNKOUT consistently preserves summary quality as the compression ratio increases, outperforming baseline methods that suffer sharp drops in ROUGE and BERTScore with higher token removal. In some cases, CHUNKOUT even matches the uncompressed baseline, suggesting effective pruning of redundancy while retaining essential information. These results demonstrate the robustness of probability-guided chunk deletion, enabling substantial prompt compression with minimal impact on summarization quality across model architectures.

Model-Specific Observations While LLMLingua2 delivers competitive results on Phi-3-mini, its performance falls substantially behind CHUNKOUT and other baselines on larger models such as Llama and Qwen. This discrepancy may be partially explained by the shared development lineage of LLMLingua2 and Phi-3-mini, which both originate from the Microsoft research ecosystem and likely share similarities in data distribution.

Model	Method	Delete Ratio	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum	BertScore
Llama 3.2 1B	Baseline	–	36.51	15.24	23.55	30.59	86.41
	Chunk	24%	**33.89**	**12.24**	**21.56**	**28.00**	**85.33**
	Select	24%	31.37	9.83	19.95	25.77	84.40
	Lingua2	24%	15.58	6.86	11.10	13.71	82.56
	Random	24%	28.47	8.81	18.39	23.62	83.59
Llama 3.2 3B	Baseline	–	36.72	15.08	23.56	30.63	86.55
	Chunk	25%	**34.47**	**12.19**	**21.75**	**28.31**	**85.36**
	Select	25%	29.26	8.26	18.53	23.77	83.94
	Lingua2	25%	15.89	6.88	11.16	13.84	82.57
	Random	25%	30.53	9.53	19.50	25.11	84.15
Llama 3.1 8B	Baseline	–	37.44	15.56	24.31	31.29	86.79
	Chunk	26.5%	**35.56**	**12.93**	**22.58**	**29.24**	**85.98**
	Select	26.5%	28.64	8.45	18.19	23.11	83.93
	Lingua2	26.5%	16.72	6.71	11.74	14.51	81.96
	Random	26.5%	30.02	9.88	19.31	24.83	84.3
Qwen 3 4B	Baseline	–	37.20	13.58	23.35	30.25	86.06
	Chunk	28%	**35.09**	**11.12**	**21.69**	**28.37**	**86.32**
	Select	28%	34.42	10.88	21.45	28.08	85.19
	Lingua2	28%	21.59	8.77	13.81	18.33	83.66
	Random	28%	33.35	10.24	20.58	27.48	85.93
Qwen 3.1 7B	Baseline	–	36.56	13.05	22.86	29.75	85.91
	Chunk	26%	**34.54**	**10.66**	**21.29**	**27.86**	**86.25**
	Select	26%	32.22	9.26	19.89	26.22	85.10
	Lingua2	26%	19.80	8.12	12.71	16.92	83.28
	Random	26%	33.24	9.76	20.39	27.17	86
Phi-3-mini	Baseline	–	36.2	12.79	23.39	30.13	86.95
	Chunk	30%	32.12	9.69	20.43	26.67	85.86
	Select	30%	32.22	9.61	20.63	26.59	85.20
	Lingua2	30%	**34.89**	**12.35**	**22.56**	**29.23**	**86.20**
	Random	30%	30.79	8.74	19.53	25.67	85.51

Table 1: Summarization results (Rouge-1, Rouge-2, Rouge-L, Rouge-Lsum, BertScore) on CNN/DailyMail for all models and baselines. Threshold τ is set as **0.5**.

4.3 Question-Answering

Main Results We directly apply our CHUNKOUT method to four QA tasks: ARC_Easy (Clark et al., 2018), Copa (Gordon et al., 2012), PIQA (Bisk et al., 2019), and Commonsense_QA (Talmor et al., 2019), evaluating performance with zero-shot QA accuracy. For CHUNKOUT, we vary the probability threshold to control the degree of token deletion, and report results at the corresponding kept ratios. As shown in Figure 2, CHUNKOUT consistently outperforms all baselines across a wide range of compression ratios and tasks.

Performance with Compression Rates Across all evaluated QA tasks, CHUNKOUT demonstrates exceptional robustness to prompt compression, consistently maintaining accuracy close to the baseline even at high deletion ratios. For instance, on ARC_Easy and Copa, CHUNKOUT retains strong performance with up to 40–50% of tokens removed, while other methods show a much steeper drop in accuracy as more tokens are deleted. Furthermore, the performance gap between CHUNKOUT and alternative baselines widens as deletion rates increase, indicating that chunk-level removal is more effective at retaining critical information under aggressive compression. This pattern holds consistently across all tested models, highlighting the broad applicability and effectiveness of our approach.

5 Conclusion

We introduce CHUNKOUT, a simple yet effective prompt compression method that prunes contiguous chunks of tokens whose next-token probabilities exceed a fixed threshold. This chunk-based pruning strategy systematically identifies and removes highly predictable spans, achieving significant prompt length reduction with minimal information loss. Across both summarization and question answering benchmarks, CHUNKOUT consistently outperforms competitive baselines in terms of compression rate and downstream accuracy. The method is model-agnostic, requires no retraining, and incurs only negligible overhead, making it broadly applicable to diverse language models and practical for large-scale inference. Future work will explore extending chunk-based pruning to the generation phase and developing adaptive, task-aware compression schemes.

Limitations

While our CHUNKOUT method enables efficient prompt compression, performance inevitably degrades as deletion rates increase, constraining the extent of usable compression in practice. Like Select Context and Lingua baselines, our approach requires an initial full pass over the input to identify deletable tokens, limiting its suitability for scenarios demanding true online or incremental compression. At present, CHUNKOUT is used solely for prompt compression; extending token deletion to the generation phase remains an open direction. Future work could explore more lightweight, online, or dynamic deletion strategies that operate during generation.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *Preprint*, arXiv:2303.08112.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *Preprint*, arXiv:1911.11641.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *Preprint*, arXiv:2401.10774.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. *Preprint*, arXiv:2305.14788.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2024. Jump to conclusions: Short-cutting transformers with linear transformations. *Preprint*, arXiv:2303.09435.

Kavita Ganesan. 2018. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. *Preprint*, arXiv:1803.01937.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context autoencoder for context compression in a large language model. *Preprint*, arXiv:2307.06945.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada. Association for Computational Linguistics.

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

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Llmlingua: Compressing prompts for accelerated inference of large language models. *Preprint*, arXiv:2310.05736.

Yanhong Li, Karen Livescu, and Jiawei Zhou. 2024. Chunk-distilled language modeling. *Preprint*, arXiv:2501.00343.

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.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2024. Learning to compress prompts with gist tokens. *Preprint*, arXiv:2304.08467.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

nostalgebraist. 2020. Interpreting gpt: the logit lens. `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`. Blog post.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron Wallace, and David Bau. 2023. Future lens: Anticipating subsequent tokens from a single hidden state. In

Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL), pages 548–560, Singapore. Association for Computational Linguistics.

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. *Preprint*, arXiv:2403.12968.

Mohammad Samragh, Arnav Kundu, David Harrison, Kumari Nishu, Devang Naik, Minsik Cho, and Mehrdad Farajtabar. 2025. Your llm knows the future: Uncovering its multi-token prediction potential. *Preprint*, arXiv:2507.11851.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *Preprint*, arXiv:1811.00937.

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.

Wilson Wu, John X. Morris, and Lionel Levine. 2024. Do language models plan ahead for future tokens? *Preprint*, arXiv:2404.00859.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

A Implementation Appendix

A.1 Implementation Details

General Implementation Details All experiments were conducted on NVIDIA GPUs (a mix of RTX A6000, RTX 6000 Ada, L40S, each with 48GB memory, and A100 40G).

QA Task-Specific Implementation Details For all question answering (QA) tasks, we use the `lm-eval-harness` framework to ensure consistency and comparability with prior work. The evaluation process is as follows:For each example in the test split, the framework constructs a prompt by concatenating the question with each answer option, using dataset-specific templates. For multiple-choice datasets (e.g., ARC, PIQA, CommonsenseQA, COPA), the model is prompted with each candidate answer in turn.The model computes the log-likelihood of the answer tokens conditioned on the given prompt. Specifically, the log-probabilities are obtained from the model outputs, and summed over all tokens in the answer span. This sum corresponds to the total log-likelihood of generating the answer given the context.For each example, the answer option with the highest total log-likelihood (i.e., the lowest negative log-likelihood, or "loss") is selected as the predicted answer. This is equivalent to greedy maximum-likelihood selection across all candidates. The final accuracy is reported as the proportion of questions for which the predicted answer matches the ground-truth label. In line With lm-eval-harness defaults, all results are computed on the official test split of each QA dataset. For all baselines and ablation variants, we follow the same evaluation pipeline, applying the respective prompt compression or token deletion method prior to model inference. To ensure statistical robustness, results for all methods are averaged over three random seeds.

For our CHUNKOUT method, we systematically sweep the probability threshold, recording the proportion of tokens deleted at each setting and measuring the corresponding QA accuracy. For baseline methods, we set their deletion ratios to match the target compression rates as closely as possible; however, due to intrinsic algorithmic differences, baselines may not achieve the exact target ratio, and the actual number of deleted tokens may fluctuate around the intended value.

Summarization Task-Specific Implementation Details For document summarization experiments, we evaluate models on the CNN/DailyMail test split. We use the prompt "Summarize the following article:\n\n{article}\n\nSummary:" to elicit model-generated summaries. For Qwen-series models, we additionally apply the official chat template as recommended by the model authors.

During evaluation, each article is tokenized and—depending on the compression method—may undergo prompt compression prior to inference. The prompt is re-tokenized after compression, and the model generates a summary in a deterministic or controlled sampling mode, depending on the model family. For most models, we use greedy decoding (i.e., `do_sample=False`), whereas for

Qwen-series models, we follow the official recommendations and enable sampling with `top-p=0.8`, `top-k=20`, `temperature=0.7`, and a maximum of 80 new tokens.

All experiments are conducted with a batch size of one, and for distributed settings, results are aggregated across processes. Summaries are decoded from generated tokens with special tokens removed. We report standard automatic evaluation metrics, including ROUGE, BERTScore, computed on all valid (non-empty) summaries. For each method, we record the average proportion of tokens deleted and the corresponding evaluation scores.

For our CHUNKOUT method, we fix the probability threshold at 0.5 to determine which tokens to prune, and compute the resulting compression rate. The same target compression ratio is then applied to all baseline methods for a fair comparison.This procedure ensures fair comparison across compression methods and model architectures. Hyperparameters (e.g., decoding strategy and prompt format) are chosen according to official model documentation to maximize performance for each model.

Data and Model Sources All datasets and pretrained models used in this work are publicly available on the Hugging Face Hub. We list their names, repositories, and official licenses below:

Datasets:

- **CNN/DailyMail** (`abisee/cnn_dailymail`): https://huggingface.co/datasets/abisee/cnn_dailymail
  License: Non-commercial, for research use only.
- **ARC** (`ai2_arc`): https://huggingface.co/datasets/allenai/ai2_arc
- **CommonsenseQA** (`taucommonsense_qa`): https://huggingface.co/datasets/tau/commonsense_qa
- **PIQA** (`baberpiqa`): https://huggingface.co/datasets/baber/piqa
- **COPA** (`pkavumba/balanced-copa`): https://huggingface.co/datasets/pkavumba/balanced-copa

Models:

- **Llama-3.1 8B** : https://huggingface.co/meta-llama/Llama-3.1-8B
- **Llama-3.2 1B** : https://huggingface.co/meta-llama/Llama-3.2-1B
- **Llama-3.2 3B** : https://huggingface.co/meta-llama/Llama-3.2-3B
  License: Meta Llama 3 Community License.
- **Qwen-3.1 7B**: https://huggingface.co/Qwen/Qwen3-1.7B
- **Qwen-3 4B** : https://huggingface.co/Qwen/Qwen3-4B
  License: Qwen License.
- **Phi-3-mini**: https://huggingface.co/microsoft/Phi-3-mini-128k-instruct
  License: Microsoft Research License.

All resources were used in accordance with their respective licenses and intended for research purposes only. We refer readers to the Hugging Face model and dataset pages for detailed license texts and citation formats.

A.2 ChunkOut Algorithm

Algorithm 1 ChunkOut Token Pruning

Require: Frozen language model $g(\cdot)$, tokenized input sequence $\mathbf{x} = [x_1, \ldots, x_n]$, probability threshold $\tau \in (0, 1)$
Ensure: Compressed token sequence $\tilde{\mathbf{x}}$

1: Compute next-token probabilities for all positions:
$$p_i = P(x_i \mid x_{<i}; g) \text{ for } i = 1 \text{ to } n$$
2: Initialize deletion set $D \leftarrow \emptyset$
3: Identify all maximal consecutive spans S_j where $p_k > \tau$ for all $k \in S_j$ and $|S_j| \geq 2$
4: **for** each span S_j **do**
5: $\quad D \leftarrow D \cup S_j$
6: **end for**
7: Output compressed sequence: $\tilde{\mathbf{x}} = [x_i \mid i \notin D]$

Description. Given a frozen language model and a tokenized input, the CHUNKOUT algorithm proceeds as follows: (1) Compute the next-token probability for each position using a single model forward pass. (2) Sweep through the sequence to identify maximal consecutive runs of tokens whose probabilities all exceed the user-chosen threshold τ; only spans of length at least 2 are eligible for removal. (3) Delete all such spans to form a compressed prompt. This procedure is linear in sequence length and model-agnostic. The threshold τ controls the trade-off between compression and information retention.

B Example Appendix

To provide qualitative insight into the behavior of CHUNKOUT, we present several representative QA samples. For each, we show the original and compressed prompt, with answer options and correct label.

- **Sample 1**
  Question: To cream butter and sugar together, you can
  Options: (A) Place it in a bowl and use a hand warmer. (B) Place in a bowl and use a hand mixer.
  Correct answer: B
  Original prompt: `Question: To cream butter and sugar together, you can Answer:`
  Compressed prompt: `To cream butter together, you can Answer:`

- **Sample 2**
  Question: How do I fill holes and tiny gaps in the concrete when making a concrete countertop?
  Options: (A) Use a concrete slurry. (B) Use a concrete brush.
  Correct answer: A
  Original prompt: `Question: How do I fill holes and tiny gaps in the concrete when making a concrete countertop? Answer:`
  Compressed prompt: `How do I fill holes and tiny the concrete when making Answer:`

- **Sample 3**
  Question: How do you remove gum from being stuck in hair?
  Options: (A) Apply an ice cube and gently remove the hardened gum. (B) Use a blow dryer and gently remove melted gum.
  Correct answer: A
  Original prompt: `Question: How do you remove gum from being stuck in hair? Answer:`
  Compressed prompt: `How remove gum from being Answer:`

- **Sample 4**
  Question: How to light a candle with a deep seated wick?
  Options: (A) invert the candle upside down and pull the wick until the lighter can reach it. (B) invert the candle upside down and use the lighter to reach into the wick to light it.
  Correct answer: B
  Original prompt: `Question: How to light a candle with a deep seated wick? Answer:`
  Compressed prompt: `How to light a candle with a deep seated Answer:`

- **Sample 5**
  Question: What materials are needed to hand sew an article of clothing?
  Options: (A) Thread, needle, scissors, material and ruler. (B) Thread, needle, knife, material and ruler.
  Correct answer: A
  Original prompt: `Question: What materials are needed to hand sew an article of clothing? Answer:`
  Compressed prompt: `What materials are needed to hand sew an Answer:`