

Attention Score is not All You Need for Token Importance Indicator in KV Cache Reduction: Value Also Matters

Zhiyu Guo, Hidetaka Kamigaito, Taro Watanabe

Nara Institute of Science and Technology

{guo.zhiyu.fy1, kamigaito.h, taro}@is.naist.jp

Abstract

Scaling the context size of large language models (LLMs) enables them to perform various new tasks, e.g., book summarization. However, the memory cost of the Key and Value (KV) cache in attention significantly limits the practical applications of LLMs. Recent works have explored token pruning for KV cache reduction in LLMs, relying solely on attention scores as a token importance indicator. However, our investigation into value vector norms revealed a notably non-uniform pattern questioning their reliance only on attention scores. Inspired by this, we propose a new method: Value-Aware Token Pruning (VATP) which uses both attention scores and the ℓ_1 norm of value vectors to evaluate token importance. Extensive experiments on LLaMA2-7B-chat and Vicuna-v1.5-7B across 16 LongBench tasks demonstrate that VATP outperforms attention-score-only baselines in over 12 tasks, confirming the effectiveness of incorporating value vector norms into token importance evaluation of LLMs.¹

1 Introduction

Recent studies have focused on scaling the context sizes of Transformer-based (Vaswani et al., 2017) large language models (LLMs) in addition to scaling data, compute, and model size. For example, the context size has increased from 2048 tokens in GPT-3 (Brown et al., 2020) and LLaMA1 (Touvron et al., 2023a) to 2 million tokens in Gemini 1.5 Pro (Reid et al., 2024). Longer context sizes enable LLMs to address tasks that extend beyond conventional capabilities, such as book-length summarization (Chang et al., 2024), SWE-agent (Yang et al., 2024), and many-shot in-context learning (Agarwal et al., 2024). However, the enormous inference costs of LLMs limit their applications. Therefore, in addition to model weight compression (Dettmers et al., 2022; Sun et al., 2024b), en-

hancing the efficiency of long-context inference has become increasingly important.

LLMs utilize an auto-regressive framework in which tokens are produced sequentially. The generation of each token relies on the tokens generated before it. During generation, the key and value tensors of previously generated tokens, known as the KV cache, have to be preserved in memory throughout the generation process for attention computation. The memory cost of the KV cache scales linearly to the batch size and sequence length. This prohibitive memory cost has become a critical bottleneck limiting the applications of long-context LLMs.

One of the approaches for improving long-context inference efficiency is token pruning, which has been extensively explored for BERT (Goyal et al., 2020; Zhao et al., 2022; Guan et al., 2022). However, these methods necessitate a complicated fine-tuning process to restore optimal performance. Given the extensive text corpora and the considerable size of LLMs, such fine-tuning becomes exceptionally challenging and less preferred. Fortunately, recent studies (Zhang et al., 2023; Liu et al., 2023; Ge et al., 2024; Xiao et al., 2024) have explored token pruning for KV cache reduction without the need for fine-tuning, indicating that a significant number of tokens can be pruned with minimal impact on performance during token generation. It is notable that these studies unanimously chose to rely solely on the attention score as the token importance indicator in LLMs. This choice is reasonable for LLMs, as training additional token importance predictor (Guan et al., 2022) is computationally expensive.

In the pre-LLM era, however, the reliability of attention score as indicator of token importance was widely questioned (Wiegrefe and Pinter, 2019; Clark et al., 2019; Hassid et al., 2022).² Despite

¹Code is available at: <https://github.com/guozhiyu/vatp>

²For example, Clark et al. (2019) observed that special

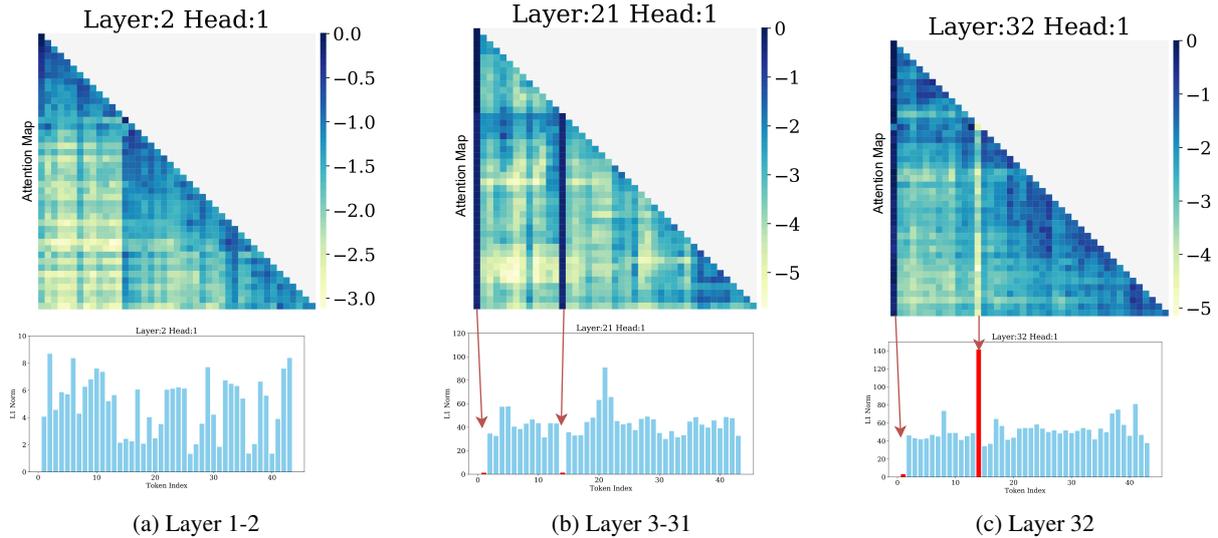


Figure 1: Typical attention map (logarithmic) and value vector norm patterns in LLaMA2-7B-chat. Key observations include: (1) The ℓ_1 norms are non-uniformly distributed across tokens in all layers and heads. (2) In figure (b), for most heads in layers 3-31, regardless of the input text, there are two attention sink (Xiao et al., 2024) tokens at the beginning of the text. Contrary to their massive attention scores, their ℓ_1 norms are close to 0 (highlighted in red). (3) In some heads of the last layer, the second attention sink token in figure (b) has a smaller attention score than other tokens, while its ℓ_1 norm is significantly larger than those of other tokens.

these concerns, in the context of KV cache reduction for LLMs, attention score appears to be highly indicative of token importance. Recent studies have demonstrated that removing even a very small number of tokens with large attention scores can significantly degrade the model’s performance (Zhang et al., 2023; Xiao et al., 2024). Nevertheless, before establishing attention score as the default choice for the token importance indicator in LLMs, we pose a timely question: *Are there any essential elements that may have been accidentally omitted when considering pivotal tokens for KV cache reduction?*

Since the output of the attention mechanism is the result of the multiplication of each token’s attention score with its corresponding value vector, we investigated the value vectors of LLMs. We found the ℓ_1 norm of each token is highly non-uniformly distributed, showing distinct differences in magnitude. Previous study (Xiao et al., 2024) identifies the attention sink tokens with massive attention scores. We find, in contrast to the attention scores, the value vector norms of the attention sink tokens are much smaller than other tokens. Such a phenomenon is similar to the finding in small Transformer models (Kobayashi et al., 2020). When considering each token’s effects on the at-

tokens tend to receive disproportionately large attention scores in BERT, yet those scores can often be significantly changed without impacting the model’s predictions.

tention output, their value vector should also be considered.

Building upon this observation, we introduce a new approach termed Value-Aware Token Pruning (VATP). Unlike traditional methods that rely solely on attention score, VATP augments the attention score with the norm of the value vector, providing a robust metric for evaluating token importance. Specifically, we propose a novel token pruning metric, where the KV cache of each token is assessed based on the product of its attention score and the ℓ_1 norm of the corresponding value vector. We conduct extensive experiments on the LLaMA2-7B-chat and Vicuna-v1.5-7B models, evaluating VATP across 16 long-context tasks from the LongBench (Bai et al., 2023) benchmark. The results demonstrate that VATP outperforms attention-score-only baselines across a wide variety of tasks. *Our research clearly reveals the critical, yet previously overlooked, role of the value vector norms in KV cache reduction, challenging the prevailing belief that attention score is all you need for evaluating token importance in LLMs.*

2 Related Work

Many works have explored improving the inference efficiency of Transformer via token pruning. Goyal et al. (2020); Zhao et al. (2022) accelerate BERT by eliminating redundant word vectors based on

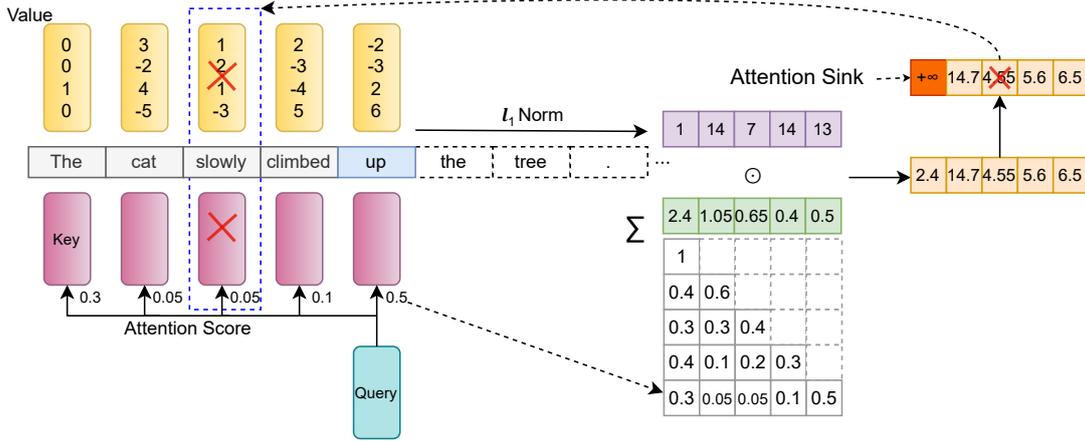


Figure 2: An illustration example of Value-Aware Token Pruning (VATP) method. VATP uses the product of value vector norm and attention score to evaluate the importance of each token’s KV cache. The KV cache of the token with the lowest importance score is removed. Here we use the accumulated attention score proposed by H₂O (Zhang et al., 2023), resulting in H₂O w/ VATP variant.

attention scores. Zhao et al. (2022) also reduces information loss in token pruning by using pooling strategies to compress the pruned tokens. To decide which tokens to prune, Transkimmer (Guan et al., 2022) adds an extra prediction module before each layer. Nevertheless, these methods are primarily designed for models similar to BERT (Devlin et al., 2019), which are encoder-only and typically smaller in size. They also generally require a complex fine-tuning process, making them less suitable for the currently prevalent larger auto-regressive language models.

Recent works investigate reducing the KV cache of unimportant tokens during auto-regressive generation. H₂O (Zhang et al., 2023) dynamically retain a combination of recent and heavy-hitter tokens in the KV cache, which are identified based on the accumulated attention scores. Scissorhands (Liu et al., 2023) uses the attention scores from a history window as the token importance indicator. FastGen (Ge et al., 2024) adjusts its compression strategies to align with the attention structure of each head. Xiao et al. (2024) observe attention sink tokens with massive attention scores, they simply keep the attention sink tokens together with the sliding window tokens to anchor the attention computation and stabilize the model’s performance. Notably, these methods primarily focus on attention scores as the basis for determining which tokens to prune, our work for the first time explores the critical role of value vector norm in token pruning.

3 Value-Aware Token Pruning

This section introduces Value-Aware Token Pruning (VATP), starting from observations and concluding with our algorithm. A visual example of VATP is shown in Figure 2.

3.1 Observations

The output of an attention head at step t is defined as follows:

$$\text{Attention}(Q, K, V)_t = \sum_{i \leq t} a_i^t v_i \quad (1)$$

where a_i^t is the attention score of query token t to token i , and v_i is the value state of token i . The attention output for the token t is thus a weighted sum of the value states v_i of all preceding tokens $i \leq t$, where the weights are the corresponding attention scores a_i^t . The goal of token pruning is to remove tokens that have a minimal impact on the attention output. From Equation (1), each token’s influence on the attention output is determined by both the attention score a_i^t and the value vector v_i .

Here, we jointly analyze the attention maps and the corresponding value vectors. In Figure 1, the ℓ_1 norm of value vector exhibits a highly non-uniform distribution across all layers and heads. Notably, the two attention sink tokens³ often show a striking

³The first token is the starting word token, the second token is often the token representing the first period (.) or new-line token (\n) in the text, there are corresponding to massive activations as discussed in (Sun et al., 2024a).

contrast between their attention scores and value vector norms. This observation is similar to the study of small Transformer models (Kobayashi et al., 2020).

3.2 Methodology

The above observation highlights the importance of considering both the attention score and the value vector norm together. Such a dual consideration provides a more comprehensive understanding of each token’s influence on the attention output. Consequently, it becomes obvious to implement token pruning strategies that take into account attention score and value vector norm simultaneously.

Attention Score H₂O (Zhang et al., 2023) uses the accumulated attention score as token importance indicator. Specially, the token importance score for a given token k at decoding step t is calculated as:

$$S_k^t = \sum_{k \leq j \leq t} a_k^j \quad (2)$$

Scissorhands (Liu et al., 2023) use the attention score based on the history window with size w .

$$S_k^t = \sum_{\max(t-w, k) \leq j \leq t} a_k^j \quad (3)$$

Value-aware Pruning Metric Motivated by the success of LLM weight pruning work Wanda (Sun et al., 2024b), which evaluates model weight importance by the product of its magnitude and the corresponding input feature norm, we propose a new metric to evaluate token importance. For each token in an attention head, its importance is evaluated by the product of its attention score S_k^t and the corresponding value vector norm. Specifically, the score for the token k at decoding step t is defined by:

$$I_k^t = S_k^t \cdot \|\mathbf{v}_k\|_1 \quad (4)$$

where $\|\mathbf{v}_k\|_1$ is the ℓ_1 norm of token k ’s value vector. The attention score S_k^t can be either Eq. (2) or Eq. (3). We empirically find ℓ_1 norm performs better than ℓ_2 norm in Appendix B. The computation of VATP metric is straightforward by jointly considering the attention score and value vector.

Attention Sink Tokens From the previous observations, the attention sink tokens have very small ℓ_1 norm. Based on our metric, the importance scores of attention sink tokens are significantly downgraded, and they could be accidentally removed. While the value updates from those tokens may be

small, the attention distribution of the rest tokens will be largely shifted after the removal, leading to deteriorated performance (Xiao et al., 2024). Thus we intentionally keep the first F tokens.

4 Experiments

4.1 Settings

Models We use two open LLMs, LLaMA2-7B-chat (Touvron et al., 2023b) and Vicuna-v1.5-7B-16k (Zheng et al., 2023). For LLaMA2-7B-chat, we set the max sequence length as 4K. For Vicuna-v1.5-7B-16k, we set the max sequence length as 8K due to GPU memory limitation. We conduct all experiments using one A6000 GPU.

Dataset To extensively assess the effectiveness of our method in real-world scenarios, we select all the English tasks in LongBench (Bai et al., 2023) as our evaluation benchmark. The LongBench benchmark consists of 16 English tasks, each containing between 150 and 500 samples. This benchmark encompasses a diverse array of long-text tasks, including question answering, text summarization, few-shot learning, synthetic tasks, and code completion. The detailed information about the dataset is in Table 5 in Appendix D. We use the official task-specific prompts to evaluate task-wise performance of instruction-tuned LLMs.

Baselines We choose several token pruning works: StreamLLM (Xiao et al., 2024), H₂O (Zhang et al., 2023), Scissorhands (Liu et al., 2023). The full KV cache is used for assessing the performance degradation. VATP needs to use both attention score and value vector norm. The attention score calculation method can be derived from either H₂O or Scissorhands. Using the accumulated attention score results in the “H₂O w/ VATP”, while using the attention score based on the history window results in the “Scissorhands w/ VATP”. Implementation details are in Appendix A.

4.2 Results

Main results While individual task results may exhibit variability, the aggregate results presented in Table 1 are more stable and reliable. For the LLaMA2-7B-chat model, the VATP method surpasses H₂O in 12 out of 16 tasks and outperforms Scissorhands in 13 out of 16 tasks. Similarly, for the Vicuna-v1.5-7B-16k model, VATP exceeds H₂O in 12 out of 16 tasks and Scissorhands in 14 out of 16 tasks. Note that for the tasks where VATP

Method	Single-Document QA			Multi-Document QA			Summarization			Few-shot Learning			Synthetic		Code		
	1-1	1-2	1-3	2-1	2-2	2-3	3-1	3-2	3-3	4-1	4-2	4-3	5-1	5-2	6-1	6-2	
LLaMA2-7B-chat	All Budget	19.12	20.99	37.55	30.55	27.44	8.31	27.77	20.67	24.39	58.33	86.22	39.14	3.89	9.67	59.88	48.61
	StreamLLM	15.4	18.6	25.96	28.19	23.59	7.08	23.87	19.97	22.52	56.67	86.45	38.72	3.87	2.62	58.55	48.28
	H ₂ O	18.4	18.83	33.67	30.18	25.74	7.85	26.18	21.12	23.44	58.67	85.35	39.0	4.37	7.0	59.4	49.09
	w/ VATP	18.77	19.6	35.31	29.95	27.15	8.44	26.08	21.14	23.76	58.33	86.09	38.74	4.39	8.33	59.56	49.46
	Scissorhands	18.5	19.32	36.35	29.5	25.51	8.59	25.42	20.35	23.86	57.33	85.55	38.77	4.38	6.0	58.33	48.86
	w/ VATP	19.4	19.53	36.58	29.57	27.71	9.66	26.17	20.46	23.63	58.0	85.98	38.9	4.18	10.0	59.39	48.71
vicuna-v1.5-7B	All Budget	18.67	23.39	39.25	27.48	19.62	8.09	30.84	22.85	24.7	64.33	86.53	39.69	4.33	13.0	50.15	36.52
	StreamLLM	16.97	21.55	26.01	23.79	16.94	5.83	26.6	21.94	22.48	62.67	86.4	39.54	2.0	11.33	49.56	37.79
	H ₂ O	18.46	21.84	32.99	26.86	19.79	6.04	27.92	23.2	23.78	64.0	79.06	39.19	4.33	11.67	51.38	36.86
	w/ VATP	18.86	21.89	36.94	28.23	19.47	7.72	28.57	23.21	23.74	64.33	86.57	40.02	4.33	13.0	50.18	37.54
	Scissorhands	18.19	20.56	33.83	26.08	18.93	5.99	26.14	22.62	23.24	61.67	80.26	39.7	4.33	9.0	49.91	35.05
	w/ VATP	18.99	21.95	37.63	28.22	20.3	7.98	27.82	23.45	23.44	62.33	86.36	39.89	4.33	13.0	48.73	36.11

Table 1: Performance of different token pruning methods on LongBench at 50% KV cache budget. To streamline the text, following (Bai et al., 2023), we refer to the dataset as ID (eg., 1-1 map to NarrativeQA, 2-2 map to 2WikiMultihopQA); the mapping from ID to the dataset and evaluation metrics are available in Table 5 of Appendix.

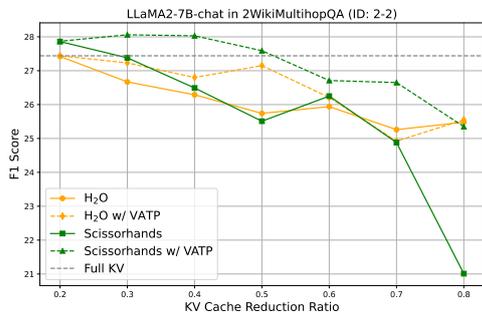


Figure 3: Performance on 2WikiMultihopQA of the LLaMA2-7B-chat with varying KV Cache Ratios.

does not surpass the baseline, its performance is still very comparable to the baseline. However, in certain tasks (e.g., 1-3 in Vicuna), VATP significantly outperforms the baseline. The overall performance gains demonstrate the effectiveness of our proposed method.

KV budget ratio variation. In Figure 3, we test the performance on 2WikiMultihopQA at different KV cache reduction ratio. Scissorhands w. VATP yields the best performance in nearly all the KV budget ratios, in the high reduction ratio, it outperforms Scissorhands significantly. The improvement of H₂O w/VATP over H₂O is mainly at the less aggressive reduction ratio.

Inference Efficiency. VATP maintains the inherent simplicity of baseline methods by introducing negligible computation and memory overhead compared with H₂O and Scissorhands, since the size of the value vector norm is $\frac{1}{2d_{\text{head}}}$ of the KV cache size, $d_{\text{head}} = 128$ for a 7B model. When integrating with H₂O, we need to calculate the accumulated attention score, this makes H₂O incompatible

with FlashAttention (Dao et al., 2022). We implemented the integration of FlashAttention based on Scissorhands. As shown in Table 3 in Appendix C, VATP introduces no significant difference in generation throughput or memory usage compared to Scissorhands at the same KV budget. Detailed throughput improvements at different KV budgets are presented in Table 4.

5 Conclusion

This study addresses a critical yet previously overlooked aspect of token pruning in LLMs—the value vectors. Motivated by the observed highly non-uniform distribution of value vector norms, we propose a simple and easy-to-implement token pruning method called Value-Aware Token Pruning (VATP). VATP jointly considers both attention scores and value vector norms to assess token importance, introducing negligible computational overhead and requiring no additional fine-tuning. Extensive experiments demonstrate that VATP consistently outperforms attention-score-only approaches across a variety of long-context tasks. These findings provide fresh insights into the significant role of value vector norms in the context of KV cache reduction. This work paves the way for developing more advanced KV cache reduction algorithms, potentially leading to more efficient and scalable deployment of long-context LLMs in practical applications.

Limitations

Our work has the following limitations:

FlashAttention Support for H₂O: When integrating with H₂O, we need to calculate the accumu-

lated attention score. However, the current implementation of FlashAttention (Dao et al., 2022) does not return the attention matrix. Without integrating FlashAttention, the memory cost of prompt pre-filling remains $O(n^2)$. Although it’s unnecessary to store the attention matrix for all layers simultaneously, handling extensive context significantly increases the memory cost during prompt prefilling. However, when integrating into Scissorhands, FlashAttention can be used. Since we don’t need to materialize the full attention matrix, Scissorhands only calculates the attention score based on last w tokens. Scissorhands generally achieve better performance than H₂O in QA tasks. In addition, concurrent work (Devoto et al., 2024) shows a clear correlation between the ℓ_2 norms of the key vector and the attention scores. This suggests the feasibility of reducing the KV cache without calculating attention scores, relying solely on the ℓ_2 norms of key vectors. Pruning the KV cache jointly using the norms of both key and value vectors is an intriguing direction for future research.

Compatibility with grouped-query attention:

Similar to Scissorhands and H₂O, our method is currently not applicable to grouped-query attention (GQA) (Ainslie et al., 2023). Token pruning and grouped-query attention are orthogonal in principle: grouped-query attention reduces the number of KV heads, while token pruning reduces the number of tokens. Ren and Zhu (2024) use the group-wise averaged attention score as the token importance score. Exploring the combination of VATP and GQA represents a promising research direction.

References

- Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Stephanie Chan, Ankesh Anand, Zaheer Abbas, Azade Nova, John D Co-Reyes, Eric Chu, et al. 2024. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*.
- Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. **GQA: Training generalized multi-query transformer models from multi-head checkpoints**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. **Boookscore: A systematic exploration of book-length summarization in the era of LLMs**. In *The Twelfth International Conference on Learning Representations*.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. **What does BERT look at? an analysis of BERT’s attention**. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.
- Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.
- Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. **GPT3.int8(): 8-bit matrix multiplication for transformers at scale**. In *Advances in Neural Information Processing Systems*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. 2024. A simple and effective ℓ_2 norm-based strategy for kv cache compression. *arXiv preprint arXiv:2406.11430*.
- Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2024. **Model tells you what to discard: Adaptive KV cache compression for LLMs**. In *The Twelfth International Conference on Learning Representations*.
- Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *International Conference on Machine Learning*, pages 3690–3699. PMLR.
- Yue Guan, Zhengyi Li, Jingwen Leng, Zhouhan Lin, and Minyi Guo. 2022. **Transkimmer: Transformer learns to layer-wise skim**. In *Proceedings of the 60th*

- Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7275–7286, Dublin, Ireland. Association for Computational Linguistics.
- Michael Hassid, Hao Peng, Daniel Rotem, Jungo Kasai, Ivan Montero, Noah A Smith, and Roy Schwartz. 2022. How much does attention actually attend? questioning the importance of attention in pretrained transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1403–1416.
- Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. [Attention is not only a weight: Analyzing transformers with vector norms](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.
- Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhao Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023. [Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Siyu Ren and Kenny Q Zhu. 2024. On the efficacy of eviction policy for key-value constrained generative language model inference. *arXiv preprint arXiv:2402.06262*.
- Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024a. [Massive activations in large language models](#). In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*.
- Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2024b. [A simple and effective pruning approach for large language models](#). In *The Twelfth International Conference on Learning Representations*.
- 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. 2023a. [Llama: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, and et al. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Sarah Wiegrefe and Yuval Pinter. 2019. [Attention is not not explanation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. [Efficient streaming language models with attention sinks](#). In *The Twelfth International Conference on Learning Representations*.
- John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. [Swe-agent: Agent-computer interfaces enable automated software engineering](#). *Preprint*, arXiv:2405.15793.
- Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Re, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. [H2o: Heavy-hitter oracle for efficient generative inference of large language models](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Jing Zhao, Yifan Wang, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022. [Fine- and coarse-granularity hybrid self-attention for efficient BERT](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4811–4820, Dublin, Ireland. Association for Computational Linguistics.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). *Preprint*, arXiv:2306.05685.

A Implementation Details

For each input sequence, we set the KV cache budget to 50% of the input prompt length. We assign uniform KV budgets across different heads and layers, as a uniform strategy is more practical to achieve actual inference improvements in hardware. FastGen (Ge et al., 2024) uses a non-uniform strategy, thus we haven’t chosen it as a baseline. For StreamLLM, we set the number of attention sink tokens to 20 for LLaMA2-7B-chat and 40 for Vicuna-v1.5-7B-16k. For VATP, we intentionally keep the first F tokens, where $F = 20$ for LLaMA2-7B-chat and $F = 40$ for Vicuna-v1.5-7B-16k. Given a KV budget of k tokens, the number of tokens selected

by importance score in H₂O is $\frac{k}{2}$, with a local window size also of $\frac{k}{2}$. In Scissorhands, following Liu et al. (2023), the number of tokens selected by importance score is $k - 10$, with a local window size of 10 and a history window size of $w = 400$. When integrating with Scissorhands and H₂O, the only differences are the token importance score and intentionally keeping attention sink tokens.

B Ablation Study

Table 2 shows the F1 scores for 4 QA tasks under different types of norm for value vector: l_1 , l_2 , and l_∞ . Overall, the l_1 norm achieves the highest average performance with an average F1 score of 28.00, indicating that l_1 norm performs better across the evaluated tasks compared to l_2 and l_∞ norms. Thus we use l_1 norm in all the experiments.

Task	l_1 Norm	l_2 Norm	l_∞ Norm
Qasper	19.60	18.47	18.76
MultifieldQA (en)	35.31	35.48	35.11
HotpotQA	29.95	30.30	30.20
2WikiMQA	27.15	26.55	26.76
Average	28.00	27.70	27.71

Table 2: F1 Scores under Different Norms for 4 QA Tasks

C Inference Efficiency

Computation Overhead of VATP For long-context sequences, the computation and memory overhead of the full attention matrix becomes a significant challenge, making the integration of FlashAttention essential. Since H₂O requires materializing the entire attention matrix to compute the accumulated attention score, it is incompatible with FlashAttention. In contrast, Scissorhands only requires the attention scores of the last w tokens, allowing us to avoid materializing the full attention matrix. We integrated FlashAttention with Scissorhands to maximize the efficiency gains of VATP.

To evaluate the computational overhead introduced by VATP, we measured the generation throughput on an A6000 Ada GPU with an input sequence length of 4096, an output length of 128, and a batch size of 12. The KV cache budget was set at 50%. As shown in Table 3, VATP introduces negligible computation and memory overhead compared to Scissorhands alone, with no significant difference in generation throughput or peak memory usage.

	throughput (token/s)	Peak Memory
Scissorhand	104.28	48380 MiB
w. VATP	103.93	48380 MiB

Table 3: Comparison of throughput and peak memory

GPU	KV budget	Throughput (tok/s)	Speedup
A100	100%	79.78	1.0×
	50%	126.45	1.58×
	25%	179.97	2.26×
A6000	100%	68.02	1.0×
	50%	103.93	1.53×
	25%	141.74	2.08×

Table 4: Generation throughput improvement of VATP at different KV budgets.

Throughput improvement of VATP We evaluate the generation throughput (measured in tokens per second, tok/s) of Scissorhands with VATP across various KV cache budgets to analyze the performance benefits of our approach. The experiments are conducted on both A100-80G and A6000 Ada (49GB) GPUs to account for different hardware configurations. For these tests, we use an input sequence length of 4096 tokens and an output sequence length of 128 tokens. Here we use LLaMA2-7B-chat model.

As shown in Table 4, the improvements in throughput across different hardware configurations underscore the practicality of VATP, particularly for real-world applications where long input sequences and memory constraints can be major bottlenecks. By integrating VATP, we can achieve more efficient token generation without sacrificing model accuracy, even when operating under constrained KV cache budgets.

D Dataset Details

We select the English subset from Longbench (Bai et al., 2023). Table 5 shows the information of 16 tasks we use in the experiments.

ID	Dataset	Avg len	Metric	#data
1-1	NarrativeQA	18,409	F1	200
1-2	Qasper	3,619	F1	200
1-3	MultiFieldQA-en	4,559	F1	150
2-1	HotpotQA	9,151	F1	200
2-2	2WikiMultihopQA	4,887	F1	200
2-3	MuSiQue	11,214	F1	200
3-1	GovReport	8,734	Rouge-L	200
3-2	QMSum	10,614	Rouge-L	200
3-3	MultiNews	2,113	Rouge-L	200
4-1	TREC	5,177	Accuracy (CLS)	200
4-2	TriviaQA	8,209	F1	200
4-3	SAMSum	6,258	Rouge-L	200
5-1	PassageCount	11,141	Accuracy (EM)	200
5-2	PassageRetrieval-en	9,289	Accuracy (EM)	200
6-1	LCC	1,235	Edit Sim	500
6-2	RepoBench-P	4,206	Edit Sim	500

Table 5: The dataset statistics in LongBench include several key metrics. 'Source' indicates where the context originates. The 'Avg len' (average length) is measured by the number of words for datasets in English (or code). 'Accuracy (CLS)' represents classification accuracy, while 'Accuracy (EM)' denotes exact match accuracy.