CO2: Precise Attention Score Observation for improving KV Cache Replacement in Large Language Models

Meguru Yamazaki [1 2] Shivaram Venkataraman [2]

Abstract

The widespread adoption of Large Language Models (LLMs) such as ChatGPT has highlighted significant challenges in inference cost management due to their autoregressive nature, requiring sequential token generation. KV cache has been introduced to mitigate recomputation costs during inference but at the expense of increased GPU memory usage, especially as input and output lengths grow. We introduce the Cumulative Observation Oracle (CO2), a novel approach to optimize KV cache replacement based on a sophisticated scoring system. Our method leverages an extended observation period, a decay mechanism for attention scores, and optimizing FIFO cache size adjustment to efficiently manage cache space and reduce overall memory demands. Evaluation on models such as the OPT-6.7B and the Llama2-7B demonstrates that CO2 significantly reduces memory usage while maintaining output quality, leading to 1.44x and 1.32x faster token generation throughput in the OPT-6.7B and the Llama2-7B, respectively.

1. Introduction

Since the release of ChatGPT (OpenAI, 2022), Large Language Models (LLMs) have gained increasing popularity and usage. This popularity has brought new challenges, especially in the inference cost; as LLMs generate only one token in each generation step, and we have to run N steps to get N tokens (Pope et al., 2022). A KV cache (Pope et al., 2022) is commonly used to reduce the recomputation cost in the above autoregressive inference step and KV caches are widely used in common frameworks (Wolf et al., 2020).

However, KV caches also present new challenges, as they consume significant GPU memory. Memory consumption grows with increase in input prompt lengths and output tokens. Moreover, in order to accommodate more complex tasks, the maximum token lengths of recently released LLMs, such as Gemini 1.5 (Team et al., 2024) and Claude 3 (Anthropic, 2024), are now up to 1 million tokens. Thus, longer inputs increase the KV cache size, reduce the batch size that can be used for inference, which in turn diminishes the token generation throughput.

A number of prior works including (Zhang et al., 2023) and (Liu et al., 2023a) have proposed techniques to reduce the memory usage of the KV cache (more details in Section 2). Both methods (Zhang et al., 2023; Liu et al., 2023a) reduce the size of the KV cache while maintaining the accuracy as they only hold the more important KVs. They utilize an attention score, which is calculated as $Softmax(QK^T)$, to select the important KVs because KVs with high attention scores are frequently referenced by other KVs, including themselves.

While these methods effectively reduce the KV cache size, our experiments (Section 3) indicate that they also introduce several problems. First, these methods are not able to handle dynamic changes in the importance of KVs, since KVs cannot be reused once they are discarded. For example, a KV that is discarded as unimportant in the first generation step cannot be reused even if it becomes important in subsequent steps. Second, once KVs receive a high attention score, they tend to stay in the cache for a long time because the scores are accumulated. This prevents new KVs from entering the cache. Third, these methods might cache unimportant KVs because they use suboptimal setting for a FIFO cache, which is used for newly generated KVs.

To address these problems, we propose a novel approach, termed Cumulative Observation Oracle (CO2), which leverages more precise observation of the attention scores. CO2 realizes this by (1) allowing for a prolonged assessment period to accumulate the attention scores over several steps to more accurately determine KV importance, (2) introducing a decay factor into the attention score accumulation, CO2 ensures that the significance of KVs is periodically reassessed, thereby preventing outdated data from monopolizing cache space, and (3) optimizing the size of the FIFO cache, which

[1] Fujitsu Limited, Japan [2] University of Wisconsin Madison. Correspondence to: Meguru Yamazaki <yamazaki.meguru@fujitsu.com>.

Presented in Efficient Systems for Foundation Models workshop in the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

is crucial for caching newly generated KVs, thus extending the cache managed by the accumulated attention score. Our evaluations show that CO2 can reduce the cache size from 60%to 20% for the OPT-6.7B (Zhang et al., 2022) and from 20% to 10% for the Llama2-7B (Touvron et al., 2023), while maintaining perplexity. These changes result in 1.44x and 1.32x faster token generation throughput in the OPT-6.7B and the Llama2-7B, respectively.

2. Related work

Reducing KV: (Zhang et al., 2023), (Liu et al., 2023a), and (Adnan et al., 2024) represent the mainstream approaches to KV cache replacement policies. These models remove less essential KVs from the cache based on specific metrics. Both (Zhang et al., 2023) and (Liu et al., 2023a) rely solely on the attention score as their metric. In contrast, (Adnan et al., 2024) enhances this approach by incorporating the Gumbel distribution (Cooray, 2010) to manage variability and improve the robustness of token selection.

(Shazeer, 2019) and (Ainslie et al., 2023) share KVs either entirely or by group. These methods were originally proposed to reduce the computational costs of attention calculations. Additionally, they can also decrease the number of KVs. (Liu et al., 2023b) prunes the ineffective attention head by training to determine which head is activated in each layer. (Munkhdalai et al., 2024) proposes Infini-attention, which compresses the KVs outside the current window into an additional hidden state. This hidden state, along with the KVs within the current window, is used for inference. As a result, Infini-attention maintains the number of KVs within a specified window size. (Mu et al., 2024) compresses task-specific prompts into shorter representations by leveraging learned compact embeddings that encapsulate key prompt information. (Kwon et al., 2023) is a KV cache management method inspired by the virtual memory management of an operating system in which the KVs are managed by small units (pages). This fine-grained KV cache management can eliminate extra pre-allocation and reduce redundant KVs in the same prompt which is caused by techniques like beam search.

Analysis of LLM: (Xiao et al., 2024) find some tokens have high attention scores that decrease the accuracy of window attention. (Sun et al., 2024) clarify that this phenomenon is caused by large values in certain dimensions of the hidden state, which acts as a bias. (Dettmers et al., 2022) find Outlier Features, which are unusually large values within specific dimensions of the model's hidden states. These significantly affect quantization performance by introducing bias and disrupting the standard quantization process.

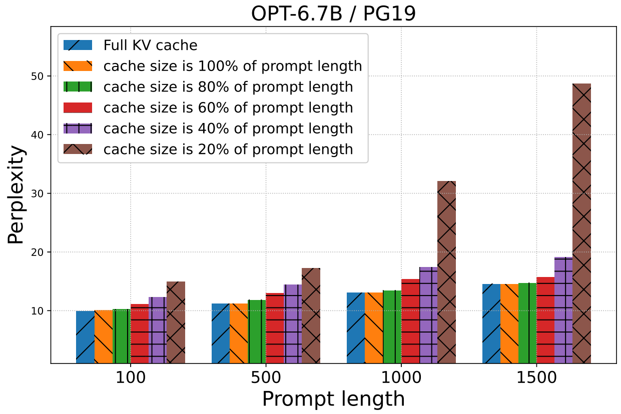


Figure 1. Evaluation of H2O: This figure shows the performance of H2O, measured by perplexity, across different cache sizes and prompt lengths using the OPT-6.7B model and PG19 dataset.

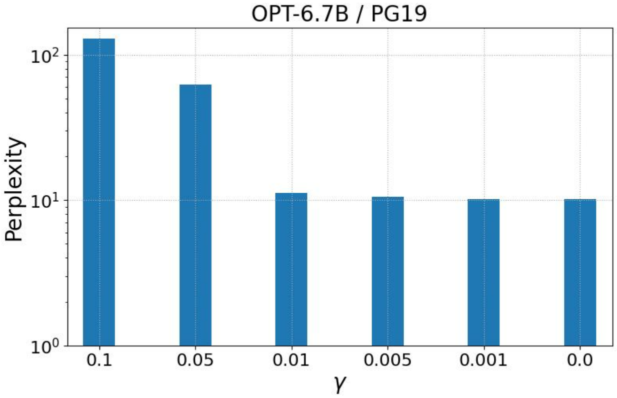


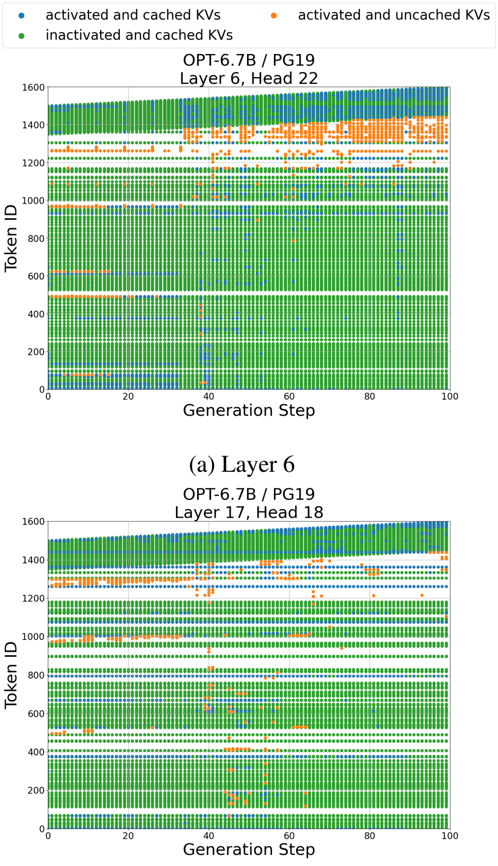
Figure 2. Perplexity with different γ values with full KV cache using the OPT-6.7B model and PG19 dataset. The result for $\gamma = 0.01$ is almost equal to the result for $\gamma = 0$ where all KVs are activated. Therefore we use $\gamma = 0.01$ for later examination.

3. Analysis

3.1. Preliminary evaluation

(Zhang et al., 2023) and (Liu et al., 2023a) utilized datasets with short prompt lengths, such as MathQA (Amini et al., 2019), and observed a trade-off between accuracy and compression ratio. However, the performance on datasets with longer prompt lengths has not been investigated. We begin by examining the compression performance of an existing KV cache replacement policy on datasets with longer prompts.

We use the OPT-6.7B (Zhang et al., 2022) as our baseline model. Our dataset is PG19 (Rae et al., 2019), which consists of old books published before 1919 and thus has long textual data. We evaluate perplexity by extracting prompts



(a) Layer 6

(b) Layer 17

Figure 3. KVs activated and cached using the OPT-6.7B and PG19: blue points are activated and cached KVs which indicates important KVs that are cached. The orange points are activated but uncached KVs which indicates important KVs that are not cached. The green points are inactive and cached KVs which indicates unimportant KVs that are cached. The blank spaces represent unimportant and uncached KVs.

of varying lengths from the dataset and generating the subsequent 100 tokens for each. The prompt lengths are set between 100 and 1500 tokens. We compare the Full KV cache, in which all KVs were cached, with H2O (Zhang et al., 2023) as the KV cache replacement policy where the cache ratio is set between 20% and 100% of the prompt length.

Figure 1 shows that the perplexity remains low when the cache ratio is between 80% and 100% across all prompt lengths. However, performance deteriorates as the cache ratio decreases below 60%. The magnitude of the performance degradation increases with longer prompt lengths. Notably, at a cache ratio of 20%, the perplexity for prompt lengths of 1000 and 1500 increases by 2.45x and 3.34x, respectively. This is our motivation for improving the KV cache replacement policy. We also investigate the reason for this performance drop in the remainder of this section. For additional results using another model, Llama2-7B (Touvron et al., 2023), please refer to Appendix A, where we present the same figures as in this section.

3.2. Which KV is important and cached

To understand why the performance of H2O is degraded, we show which KVs are cached and which KVs are important to maintain the performance. Although determining which KVs are cached is straightforward, finding which KVs are important is not trivial. We investigate which KVs are important in following manner. First, we set a threshold where if the value of the attention score exceeds this threshold, the corresponding KV is considered activated. Then, we evaluate the performance using only these activated KVs while varying the threshold. The KVs that remain activated at the threshold where the performance is stable are deemed important. In this paper, we set the threshold to LAS $\times\ \gamma$, where LAS represents the largest attention score at each layer and γ ranges from 0.001 to 1.0. We consider the results stable if the increase in perplexity is less than 10% of the original result ($\gamma = 0.0$). Figure 2 shows performance across different γ values, using the same model and dataset as in the preliminary evaluation. At $\gamma = 0.01$, the error is less than 10% compared to the original results, hence we use $\gamma = 0.01$ for all subsequent evaluations.

We analyzed which KVs are important and which are cached with $\gamma = 0.01$. Using the OPT-6.7B and PG19 dataset with a prompt length of 1500 and an H2O cache size set at 20% of the prompt length (where 10% is managed by the attention score and the rest of 10% is allocated to the FIFO cache for newly generated KVs), we generated 100 tokens. We then extracted data from two easily recognizable mid-layers, specifically layers 6 and 17, because these layers exhibit a more balanced number of activated KVs, unlike the initial layers which have too many and the final layers which have too few. In Figure 3, blue points represent activated and cached KVs (important and retained), orange points show activated but uncached KVs (important but not retained), and green points depict inactivated but cached KVs (unnecessary yet retained).

We first confirm that there are indeed many green points, indicating wasteful caching, and there are also many orange points, indicating important KVs which are missing from the cache. Therefore, there is room for improving the performance by replacing cache space used green points to instead cache orange points. We also notice that the orange points for the prompt tokens are not activated in the first step in both Figure 3a and 3b. This means the KVs that are not important to generate the first token are discarded after the first layer and are not recovered again even though they are important in following steps. We also notice that

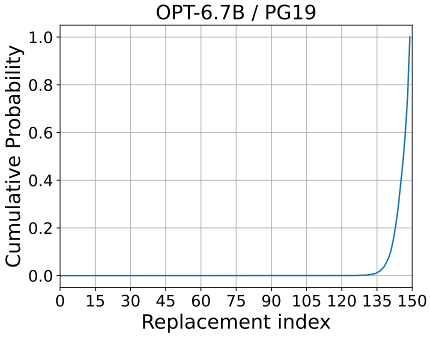


Figure 4. Replacement index of H2O with 20% cache size using the OPT-6.7B model and PG19 dataset. The cache size managed by the attention score is 150 and the index range is 0-149. More than 90% replaced index is larger than 135. This means many newly arrived KVs from the FIFO cache are replaced in few steps.

there are many green points among the newly generated tokens which are cached in FIFO cache. The FIFO cache is an auxiliary cache and helps evaluate the attention score for newly generated KVs and is by default set to half of the overall cache size in H2O. Our results show that using a simple fixed size FIFO cache is suboptimal and there is room for improvement.

3.3. Which KV is replaced

We also observed many orange points around a diagonal band at the top of Figure 3a, representing KVs initially cached in the FIFO cache. To investigate in detail, we show the replacement index across all layers in Figure 4. Since the cache size managed by the attention score is 150 (which is 10% of the prompt length), the replacement index ranges from 0 to 149 with larger numbers indicating newer entries. We confirm that for more than 90% of the entries the replacement index is greater than 135. This indicates that many newly arrived KVs from the FIFO cache are replaced in a few steps. Therefore, the cache managed by the attention score is occupied by the tokens in the prompt, and newly generated KVs are frequently replaced, despite many of them being important.

4. Cumulative Observation Oracle

Based on our results in the previous section, the issues with existing KV cache replacement policies are as follows:

- KVs that are not important for generating the first token are discarded after the first generation step and are not recovered, even though they become important in subsequent steps.
- The cache managed by the attention score is filled with the tokens from the prompt, resulting in many newly generated KVs being replaced, despite the presence of important KVs among them.
- Setting the FIFO cache size to half of the total cache size has been found to be suboptimal, suggesting that there is room for optimization.

To address these problems, we propose the Cumulative Observation Oracle (CO2), a novel approach that enhances the KV cache replacement policy by leveraging more precise observation of the attention scores. CO2 also considers changes in importance when calculating the accumulated attention scores and accordingly adjusts the size of the FIFO cache. CO2 comprises three main components, each designed to overcome the identified problems and optimize KV cache replacement.

Long Measurement Step: The Long Measurement Step in CO2 specifically addresses the issue where important KVs not activated in the first step are discarded during the caching process. The root of this problem lies in the determination of important KVs from prompt tokens based solely on the initial attention scores. To solve this, CO2 defers the decision-making process by N generation steps, enabling us to take into account the cumulative attention scores accumulated by that point. This delayed approach thus facilitates more accurate identification of critical KVs.

Decay of the Accumulated Attention Score: We next address the problem of newly generated KVs being immediately replaced after entering the cache managed by the attention score. This problem arises because the existing method continuously accumulates attention scores without any decay, meaning once a KV reaches a high attention score, it tends to remain in the cache regardless of its subsequent relevance. To address this, CO2 utilizes the Exponentially Weighted Moving Average (EWMA) to calculate the accumulated attention scores as follows:

$$\text{AAS}_t = \text{AAS}_{t-1} \cdot (1-\alpha) + \text{New Attention Score} \cdot \alpha, \quad (1)$$

where AAS_t represents the accumulated attention score at generation step t, and α denotes the decay parameter.

Adjusting FIFO Cache Size: Finally, we address the issue of suboptimal FIFO cache sizing which results in numerous inactive KVs remaining in the FIFO cache. To rectify this, CO2 simply adjusts the FIFO cache size based on the prompt length and the characteristics of the datasets. We introduce ρ as a parameter, with the actual FIFO cache size calculated as Cache Size $\times$ ρ.

5. Evaluation results

We evaluate the performance of CO2 using the same methodology as in our preliminary experiments (Section 3). In ad-

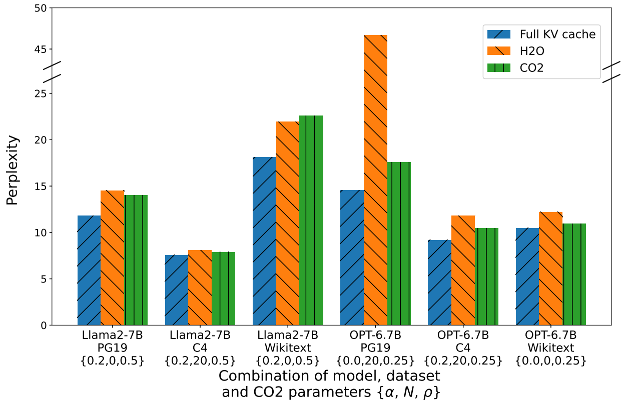


Figure 5. Comparing perplexity for the Full KV Cache, H2O, and CO2, across different combinations of models and datasets. CO2 parameters shown are cases where CO2 achieves best perplexity and are noted as $\{\alpha, \mathrm{N}, \rho\}$.

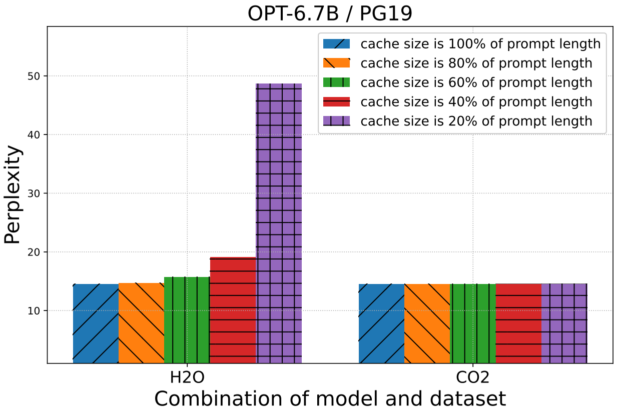


Figure 6. Perplexity of H2O and CO2 with different cache size using the OPT-6.7B and PG19.

dition to the OPT-6.7B (Zhang et al., 2022), we also utilize the Llama2-7B (Touvron et al., 2023). The datasets we use include PG19 (Rae et al., 2019), C4 (Raffel et al., 2019), and Wikitext (Merity et al., 2016). Our prompt length is set to 1500 tokens with 100 generating tokens. The cache size is configured as 20% for the OPT-6.7B and 10% for the Llama2-7B. For comparison, we used the Full KV cache in which all KVs were cached, and H2O (Zhang et al., 2023), which utilizes attention scores for cache replacement. The parameters for CO2 were as follows: $\alpha = 0.0, 0.2$, $N = 0, 20$, and $\rho = 0.25, 0.5$.

Figure 5 shows the performance of CO2 compared to our two baselines and includes the parameters used for CO2's results. We find that CO2 can improve the performance across almost all combinations of the models and datasets. CO2 maintains perplexity within 20% of the Full KV cache's performance. The only exception is the combination of the Llama2-7B and Wikitext, where CO2 does not improve performance as expected. We suspect this underperformance is due to the limited range of parameter settings available for CO2 in this evaluation. For the complete results with all parameters for CO2, please refer to Appendix B.

Additionally, we evaluate perplexity with different cache ratios using the OPT-6.7B model and PG19 dataset, as shown in Figure 6. The perplexity for 100% and 80% cache sizes remained stable for both H2O and CO2. However, while H2O's performance deteriorates with cache sizes below 60%, CO2 maintains consistent performance, even at a 20% cache size. This indicates that our caching strategy effectively compensates for the smaller cache sizes by prioritizing the retention of more significant KVs. We show results with different cache ratios using another dataset in Appendix B.

Furthermore, we validate the throughput improvements achieved by CO2. Using the same prompt settings as the previous evaluation, we adjust the batch size to the maximum value that does not result in out of memory on a single NVIDIA V100 GPU (16GB model). Table 1 shows that CO2's throughput outperforms both the Full KV cache and H2O. Specifically, CO2 achieves 1.44x and 1.32x higher throughput for OPT-6.7B and the Llama2-7B, respectively. This is because CO2 can accept larger batch sizes while maintaining low perplexity.

6. Conclusion

In this paper, we analyze existing KV cache replacement methods and derive a set of key issues that result in worse performance. To address these issues, we propose CO2, a novel approach to optimize KV cache replacement in large language models. CO2 demonstrates its effectiveness by reducing the cache size by 2x while maintaining accuracy and achieving 1.44x and 1.32x faster token generation throughput in the OPT-6.7B and the Llama2-7B, respectively. In future work, we plan to automate the tuning of CO2's parameters by elucidating the relationships between datasets, models, and the parameters themselves, thereby improving real-time performance.

References

Adnan, M., Arunkumar, A., Jain, G., Nair, P. J., Soloveychik, I. and Kamath, P. Keyformer: Kv cache reduction through key tokens selection for efficient generative inference, 2024.

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Le-

Method	Model	Cache size	Batch size	Perplexity	Throughput
Full KV cache	OPT-6.7B	Full	2	9.20	6.20
H2O	OPT-6.7B	20%	5	11.81	20.01
CO2	OPT-6.7B	10%	6	11.12	**28.98**
Full KV cache	Llama2-7B	Full	2	7.58	2.46
H2O	Llama2-7B	10%	4	8.10	11.39
CO2	Llama2-7B	5%	5	9.97	**15.02**

Table 1. Throughput of generating tokens per second on a single NVIDIA V100 GPU (16GB model) with a prompt length of 1500 and 100 generating tokens. ”FULL” in the Cache size column indicates that all KVs are cached, while X% cache size means the cache size is X% of the prompt length.

bron, F. and Sanghai, S. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In Bouamor, H., Pino, J. and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298.

Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y. and Hajishirzi, H. MathQA: Towards interpretable math word problem solving with operation-based formalisms. 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)*, pp. 2357–2367, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245. URL https://aclanthology.org/N19-1245.

Anthropic. Introducing the claude 3 family. https://www.anthropic.com/news/claude-3-family, 2024. Accessed: 2024-05-27.

Cooray, K. Generalized gumbel distribution. *Journal of Applied Statistics*, 37:171–179, 01 2010. doi: 10.1080/02664760802698995.

Dettmers, T., Lewis, M., Belkada, Y. and Zettlemoyer, L. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K. and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 30318–30332. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c3ba4962c05c49636d4c6206a97e9c8a-Paper-Conference.pdf.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H. and Stoica, I. Efficient memory management for large language model serving with pagedattention. SOSP ’23, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Liu, Z., Desai, A., Liao, F., Wang, W., Xie, V., Xu, Z., Kyrillidis, A. and Shrivastava, A. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=JZfg6wGi6g.

Liu, Z., Wang, J., Dao, T., Zhou, T., Yuan, B., Song, Z., Shrivastava, A., Zhang, C., Tian, Y., Ré, C. and Chen, B. Deja vu: contextual sparsity for efficient llms at inference time. In *Proceedings of the 40th International Conference on Machine Learning*, ICML’23. JMLR.org, 2023b.

Merity, S., Xiong, C., Bradbury, J. and Socher, R. Pointer sentinel mixture models, 2016.

Mu, J., Li, X. L. and Goodman, N. Learning to compress prompts with gist tokens, 2024.

Munkhdalai, T., Faruqui, M. and Gopal, S. Leave no context behind: Efficient infinite context transformers with infini-attention, 2024.

OpenAI. Chatgpt. https://openai.com/chatgpt/, 2022.

Pope, R., Douglas, S., Chowdhery, A., Devlin, J., Bradbury, J., Levskaya, A., Heek, J., Xiao, K., Agrawal, S. and Dean, J. Efficiently scaling transformer inference, 2022.

Rae, J. W., Potapenko, A., Jayakumar, S. M., Hillier, C. and Lillicrap, T. P. Compressive transformers for long-range sequence modelling. *arXiv preprint*, 2019. URL https://arxiv.org/abs/1911.05507.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

Shazeer, N. Fast transformer decoding: One write-head is all you need, 2019.

Sun, M., Chen, X., Kolter, J. Z. and Liu, Z. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

Team, G., Reid, M., Savinov, N., Teplyashin, D., Dmitry, Lepikhin, Lillicrap, T., baptiste Alayrac, J., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., Antonoglou, I., Anil, R., Borgeaud, S., Dai, A., Millican, K., Dyer, E., Glaese, M., et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Xiao, G., Tian, Y., Chen, B., Han, S. and Lewis, M. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T. and Zettlemoyer, L. Opt: Open pre-trained transformer language models, 2022.

Zhang, Z., Sheng, Y., Zhou, T., Chen, T., Zheng, L., Cai, R., Song, Z., Tian, Y., Re, C., Barrett, C., Wang, Z. and Chen, B. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=RkRrPp7GKO.

A. Analysis with Llama2

In this section, we present additional analysis as described in section 3 using the Llama2-7B (Touvron et al., 2023). Figure 7 shows the same results as Figure 2. Like OPT-6.7B, H2O also increases the perplexity in the Llama2-7B when using a small cache ratio. However, while H2O causes more than 3x increase in perplexity compared to the Full KV cache with 1500 prompt length and 20% cache ratio. in the OPT-6.7B, in the Llama2-7B with 1500 prompt length and 10% cache ratio, the increase is only 1.23x larger than the Full KV cache. Therefore, H2O with the Llama2-7B demonstrates greater resilience than with the OPT-6.7B.

We replicate the Figure 2 as Figure 8. $\gamma = 0.01$ maintains perplexity in the Llama2-7B as well as in the OPT-6.7B. Consequently, we adopt $\gamma = 0.01$ for subsequent figures.

Figure 9 shows the same information as Figure 3. We observe many green points which are cached and unimportant, although there are many orange points which are uncached and important. Many orange points are not activated in the first step, and many green points appear in the FIFO cache, similar to the results with the OPT-6.7B. Additionally, Figure 10 reveals that over 90% of the keys arriving in the cache managed by the attention score are immediately replaced, with a replacement index of 70. Therefore, we totally confirm the same problem we show in Figure3, with the Llama2-7B.

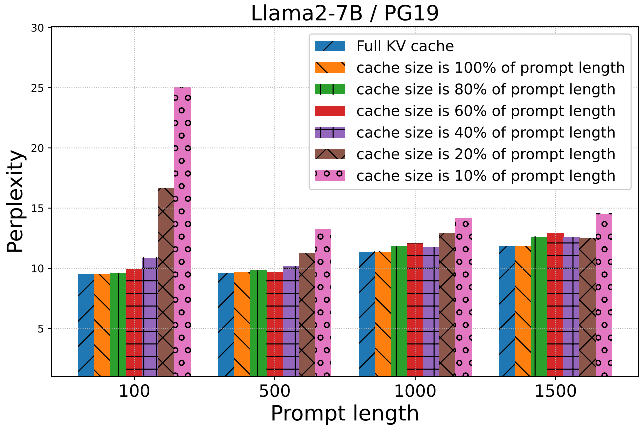


Figure 7. Preliminary evaluation results of H2O. This figure shows the performance of H2O, measured by perplexity, across different cache sizes and prompt lengths using the Llama2-7B model and PG19 dataset.

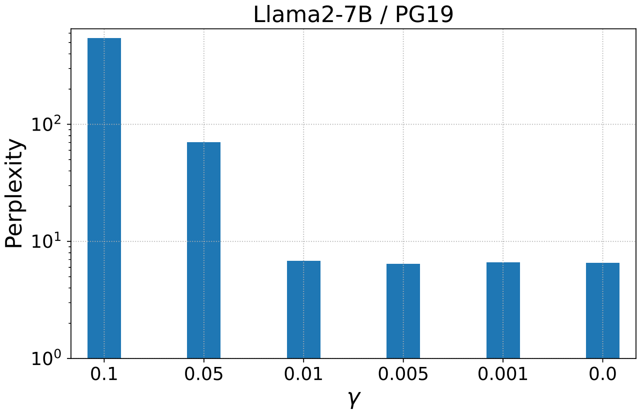
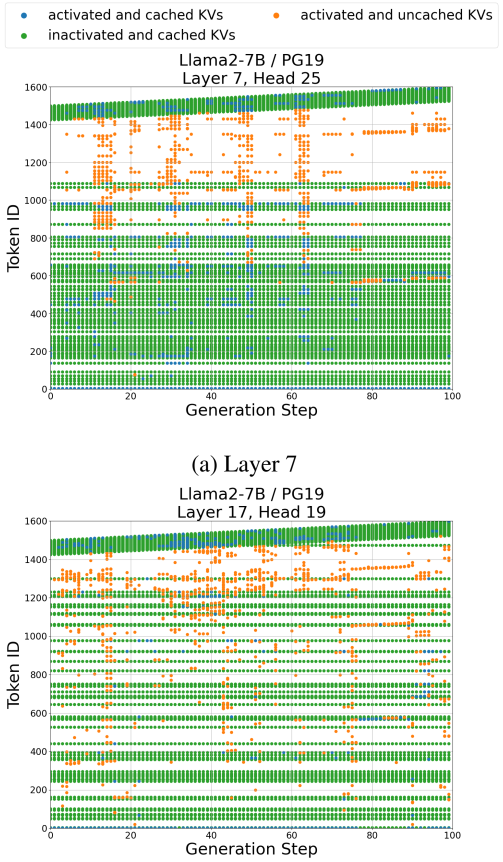


Figure 8. Perplexity with different γ values with full KV cache using the Llama2-7B model and PG19 dataset. The result for $\gamma = 0.01$ is almost equal to the result for $\gamma = 0$ where all KVs are activated. Therefore we use $\gamma = 0.01$ for later examination.



(a) Layer 7

(b) Layer 17

Figure 9. Which KV is activated and cached using the Llama2-7B and PG19. The Blue points are activated and cached KVs which means important KVs are cached. The orange points are activated but uncached KVs which means, important KVs are not cached. The green points are inactivated and cached KVs which means unimportant KVs are cached. The blank spaces represent unimportant and uncached KVs.

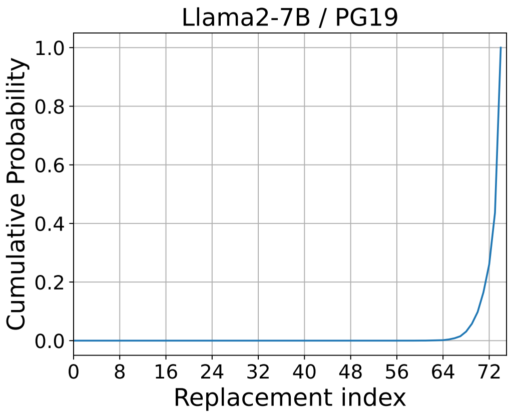


Figure 10. Replacement index of H2O with 10% cache size using the Llama2-7B model and PG19 dataset. The cache size managed by the attention score is 75 and the index range is 0-74. More than 90% replaced index is larger than 70.

B. Additional results

We present the results of all parameter combinations for CO2 in Table 2 and Table 3. We observe that the optimal combination of CO2 parameters varies depending on the model and dataset used. Therefore, we propose employing an online parameter tuning method as future work.

We also present the same results as those shown in Figure 6 in Figure 11. The perplexity remained stable for both H2O and CO2 at cache sizes from 100% to 60%. However, while the perplexity of H2O starts to deteriorate at a 40% cache size and increases by 28% at 20% cache size, CO2 shows only a 9% increase in perplexity at a 20% cache size.

Model	**Dataset**	**Cache Ratio**	α	N	ρ	**Perplexity**
OPT-6.7b	pg19	0.2	0	0	0.5	48.69
OPT-6.7b	pg19	0.2	0	0	0.25	17.91
OPT-6.7b	pg19	0.2	0	20	0.5	41.41
OPT-6.7b	pg19	0.2	0	20	0.25	**17.59**
OPT-6.7b	pg19	0.2	0.2	0	0.5	52.34
OPT-6.7b	pg19	0.2	0.2	0	0.25	19.55
OPT-6.7b	pg19	0.2	0.2	20	0.5	46.38
OPT-6.7b	pg19	0.2	0.2	20	0.25	19.03
OPT-6.7b	pg19	1	0	0	0	14.56
OPT-6.7b	c4	0.2	0	0	0.5	11.81
OPT-6.7b	c4	0.2	0	0	0.25	10.16
OPT-6.7b	c4	0.2	0	20	0.5	11.79
OPT-6.7b	c4	0.2	0	20	0.25	10.48
OPT-6.7b	c4	0.2	0.2	0	0.5	12.07
OPT-6.7b	c4	0.2	0.2	0	0.25	10.09
OPT-6.7b	c4	0.2	0.2	20	0.5	12.36
OPT-6.7b	c4	0.2	0.2	20	0.25	**10.09**
OPT-6.7b	c4	1	0	0	0	9.20
OPT-6.7b	wikitext	0.2	0	0	0.5	12.21
OPT-6.7b	wikitext	0.2	0	0	0.25	**10.95**
OPT-6.7b	wikitext	0.2	0	20	0.5	12.14
OPT-6.7b	wikitext	0.2	0	20	0.25	11.19
OPT-6.7b	wikitext	0.2	0.2	0	0.5	12.70
OPT-6.7b	wikitext	0.2	0.2	0	0.25	11.09
OPT-6.7b	wikitext	0.2	0.2	20	0.5	12.58
OPT-6.7b	wikitext	0.2	0.2	20	0.25	11.19
OPT-6.7b	wikitext	1	0	0	0	10.51

Table 2. Perplexity results of CO2 with OPT-6.7B across different configurations.

C. Additional visualization of Which KV is important and cached

In this section, we present additional visualizations similar to Figure 3, as the earlier figures only show specific layers and heads. First, we show the number of the activated and cached KVs, activated and uncached KVs and inactivate and cached KVs at each layer in Figure 12. In the early layers (before layer 8), we observe many activated and cached KVs, as well as activated and uncached KVs. In contrast, in the later layers (after layer 24), both types of KVs decrease. Since the layer trend changes roughly every 8 layers, we provide figures for each of these 8-layer groups. For the heads, we present figures for three heads, from head 0 to head 2. These are shown in Figures 13 and 14 for the OPT-6.7B and Llama2-7B models, respectively. We first confirm that the layer 1 figures show unique characteristics in both the OPT-6.7B and Llama2-7B models. As shown in Figure 12, layer 1 has a high number of activated and uncached KVs compared to other layers. These KVs in layer 1 are activated at certain generation steps where many KVs become active. We think addressing these characteristics is future work. In the other layers, we observe similar patterns. Some heads show the presence of orange points, while others do not. As discussed in Section 3, many of the orange points are not activated at the first generation step or are replaced soon after leaving the FIFO cache. Additionally, we see many green points within the FIFO cache.

Model	Dataset	Cache Ratio	α	N	ρ	Perplexity
Llama2-7b	pg19	0.1	0	0	0.5	14.53
Llama2-7b	pg19	0.1	0	0	0.25	28.67
Llama2-7b	pg19	0.1	0	20	0.5	14.22
Llama2-7b	pg19	0.1	0	20	0.25	24.19
Llama2-7b	pg19	0.1	0.2	0	0.5	14.34
Llama2-7b	pg19	0.1	0.2	0	0.25	19.36
Llama2-7b	pg19	0.1	0.2	20	0.5	**14.03**
Llama2-7b	pg19	0.1	0.2	20	0.25	16.08
Llama2-7b	pg19	1	0	0	0	11.81
Llama2-7b	c4	0.1	0	0	0.5	8.10
Llama2-7b	c4	0.1	0	0	0.25	13.70
Llama2-7b	c4	0.1	0	20	0.5	8.04
Llama2-7b	c4	0.1	0	20	0.25	11.47
Llama2-7b	c4	0.1	0.2	0	0.5	8.01
Llama2-7b	c4	0.1	0.2	0	0.25	9.30
Llama2-7b	c4	0.1	0.2	20	0.5	**7.91**
Llama2-7b	c4	0.1	0.2	20	0.25	8.54
Llama2-7b	c4	1	0	0	0	7.58
Llama2-7b	wikitext	0.1	0	0	0.5	21.94
Llama2-7b	wikitext	0.1	0	0	0.25	40.59
Llama2-7b	wikitext	0.1	0	20	0.5	25.91
Llama2-7b	wikitext	0.1	0	20	0.25	30.34
Llama2-7b	wikitext	0.1	0.2	0	0.5	**22.59**
Llama2-7b	wikitext	0.1	0.2	0	0.25	26.22
Llama2-7b	wikitext	0.1	0.2	20	0.5	25.16
Llama2-7b	wikitext	0.1	0.2	20	0.25	25.81
Llama2-7b	wikitext	1	0	0	0	18.13

Table 3. Perplexity results of CO2 with Llama2-7B across different configurations.

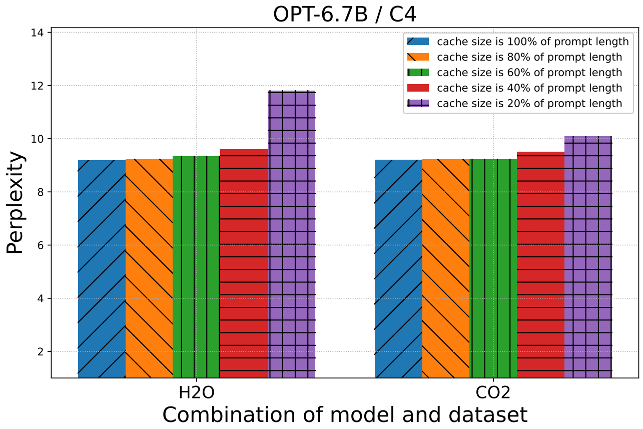


Figure 11. Perplexity of H2O and CO2 with different cache size using the OPT-6.7B and C4.

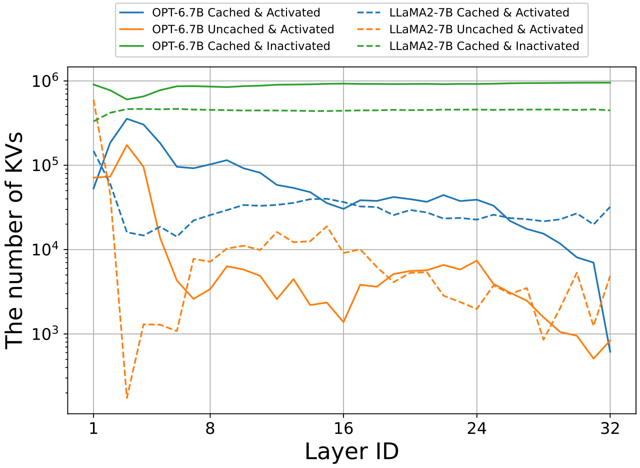


Figure 12. The number of KVs classified as Activated & Cached, Activated & Uncached, Inactivated & Cached.

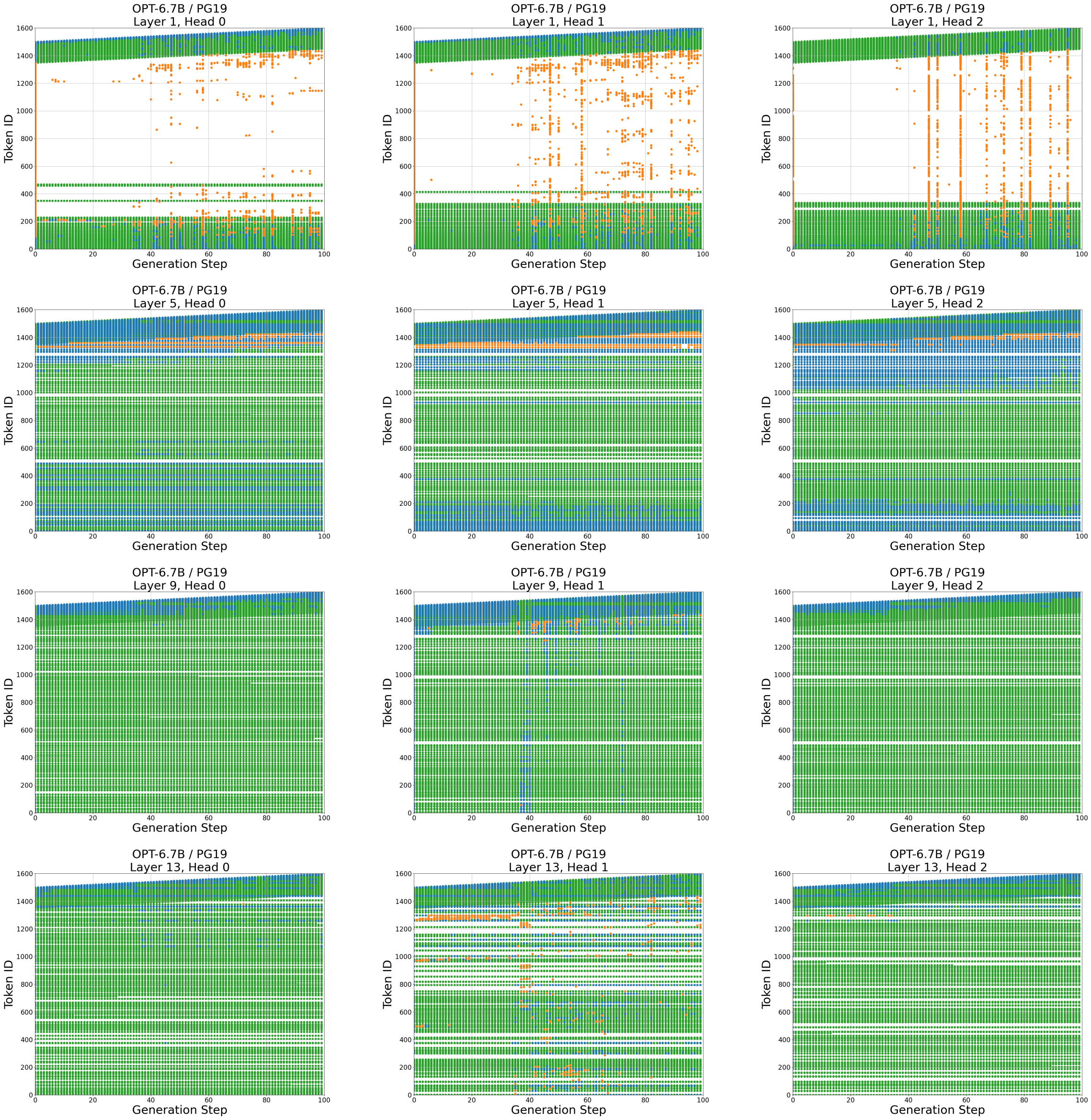


Figure 13. Additional which KV activated and cache with the OPT-6.7B and PG19.

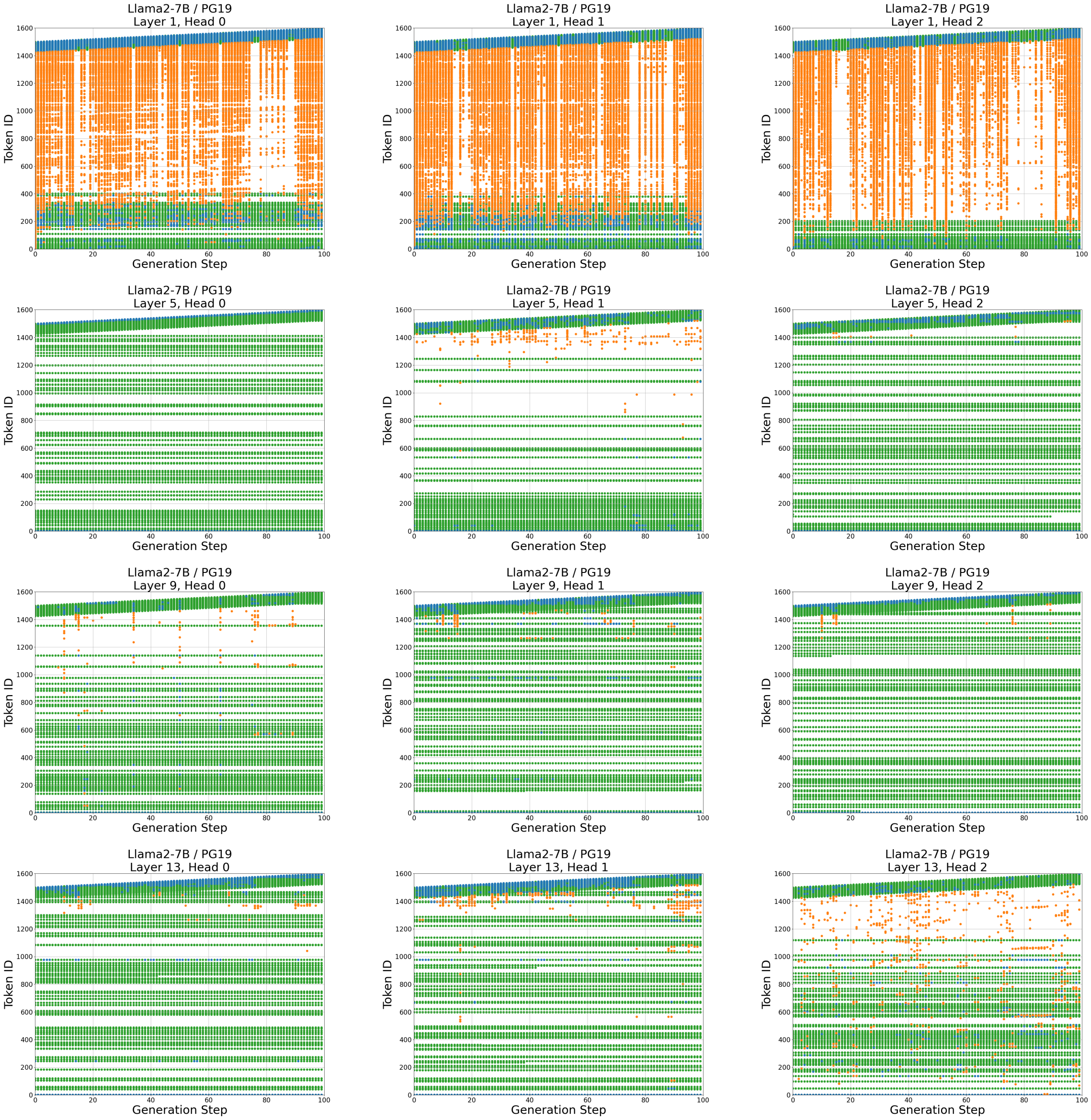


Figure 14. Additional which KV activated and cache with the Llama2-7B and PG19.