Attention is Not Static: EMA-Attention for Dynamic KV Cache Pruning in LLMs

Anonymous ACL submission

Abstract

Static KV cache pruning estimates token importance early and rarely updates it. Stale scores diverge from current attention as decoding proceeds, causing premature eviction of later-critical tokens—a phenomenon we term attention drift. We propose EMA-Attention, a training-free, plug-and-play method that updates token importance via exponential moving average of per-step attention scores. It dynamically re-ranks cached tokens as queries evolve. Theoretically and empirically, it mitigates long-horizon tracking error from static methods. On LLaMA-3-8B with 40% KV budget, it achieves 10.90 perplexity on WikiText-2 and outperforms H2O, SnapKV, and PyramidKV on LongBench QA. Notably, it matches Full KV performance with <50% cache, with largest gains under 20% retention budgets.

1 Introduction

The key-value (KV) cache has become a major bottleneck for long-context Transformer inference (Ashish, 2017). Modern decoder-only LLMs such as LLaMA (Touvron et al., 2023) and LLaMA 3 (Grattafiori et al., 2024) cache all previous keys and values during autoregressive decoding, so memory grows linearly with context length. Efficient attention kernels reduce compute cost (Dao et al., 2022), but they do not remove the need to store KV states. This is problematic because long-context behavior is dynamic: Transformers can exhibit multi-state recurrent dynamics (Oren et al., 2024), rely on concentrated activations (Sun et al., 2024), and require mechanisms for length generalization (Han et al., 2023).

Existing KV-efficient methods either quantize cached states (Hooper et al., 2024; Liu et al., 2024) or prune tokens using attention-derived importance scores. Representative pruning methods include StreamingLLM (Xiao et al., 2024), H2O (Zhang et al., 2023), ScissorHands (Liu et al., 2023), SnapKV (Li et al., 2024), PyramidKV (Cai et al.). Despite different designs, these methods largely assume that once token importance is estimated, it remains a reliable guide for later decoding steps.

This assumption is fragile in long-context generation. Attention in autoregressive LLMs is *non-stationary*: as decoding proceeds, the query state changes, and so does the set of cached tokens that receive high attention. Figure 1 illustrates this effect: the cosine similarity between attention weight vectors at steps t and $t+k$ drops rapidly as k grows. We call this **attention drift**. It makes static pruning brittle, since stale scores can evict tokens that become important only after the cache has already been compressed.

We address attention drift by treating token importance as a *time-series tracking problem*. EMA-Attention updates each token's score with an exponentially weighted moving average of its received attention, allowing cache rankings to evolve with the query distribution. The method is training-free, parameter-free, requires only $O(1)$ extra storage per cached entry, and applies to standard decoder-only Transformers. We evaluate on WikiText-2 (Merity et al., 2016), LongBench (Bai et al., 2024), NarrativeQA (Kočiskỳ et al., 2018), SciQAG-style scientific QA (Wan et al., 2024), and HotpotQA (Yang et al., 2018), following common LM evaluation practice (Gao et al., 2021). In this paper, we propose three contributions:

- **Attention drift.** We identify and characterize attention non-stationarity as a source of KV pruning error in long-context decoding.
- **Tracking analysis.** We formalize static pruning as a tracking problem and show why fixed importance scores accumulate error as the context grows.
- **EMA-Attention.** We introduce a simple dy-

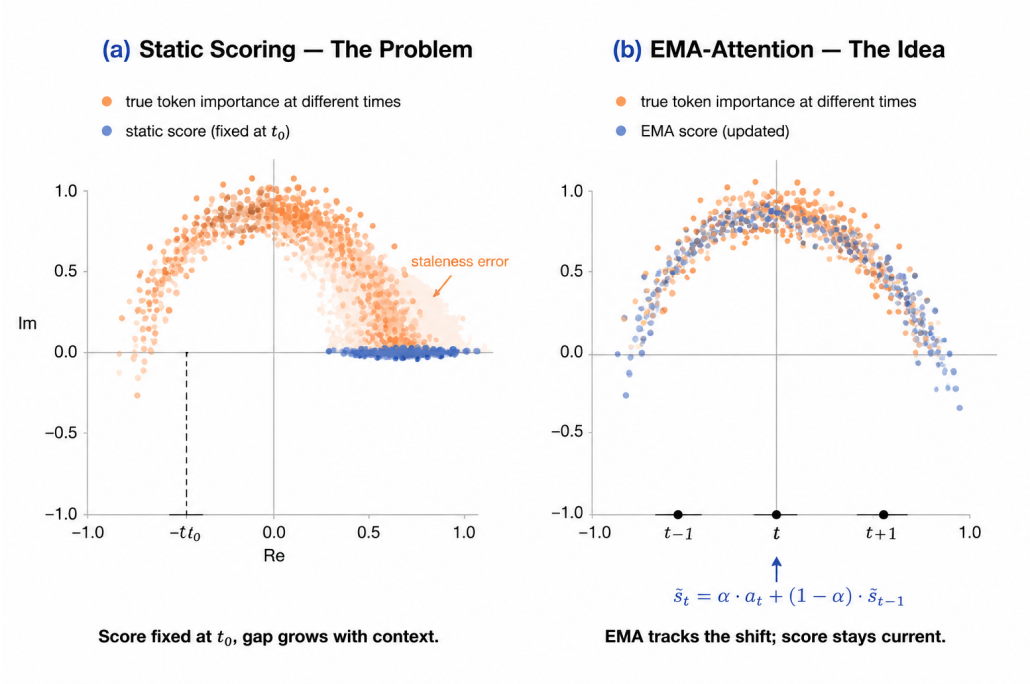

Figure 1: Static scores become stale as attention drifts, while EMA scores track true token importance.

namic pruning method that consistently improves over H2O, SnapKV, and PyramidKV under matching cache budgets.

2 Method

2.1 Notation and Problem Setup

We assume that $\mathbf{x} = (x_1, \ldots, x_n)$ be an input sequence of n tokens. At decoding step t, the model maintains a KV cache $\mathcal{C}_t = \{(k_i, v_i)\}_{i \in \mathcal{S}_t}$ over the retained positions $\mathcal{S}_t \subseteq [n+t]$. We summarize the contribution of position i to the current prediction by its head-averaged attention weight

$$a_t(i) = \frac{1}{H} \sum_{h=1}^{H} \mathrm{Attn}_h(q_t, K)_i,$$
$$\mathrm{Attn}_h(q_t, K) = \mathrm{softmax}\left(\frac{q_t^h (K^h)^\top}{\sqrt{d_h}} \right), \quad (1)$$
$$\sum_{i \in \mathcal{S}_t} a_t(i) = 1.$$

where H is the number of heads and d_h is the head dimension. Given a cache budget $B \ll |\mathcal{C}_t|$, a pruning policy assigns each retained position an importance score $s_t(i)$ and keeps the top-scoring entries:

$$\mathcal{S}_{t+1} = \text{top-}B\big(\{s_t(i)\}_{i \in \mathcal{S}_t}\big). \quad (2)$$

Attention-based pruning methods can be viewed through how they update importance scores. Purely static methods such as SnapKV (Li et al., 2024) compute scores once during prefill, $s^{\mathrm{Snap}}(i) = \sum_{\tau=t_0-w}^{t_0} a_\tau(i)$, and never revise them afterward. Dynamic methods such as H2O (Zhang et al., 2023) update scores at every step using cumulative attention, $s_t^{\mathrm{H2O}}(i) = \sum_{\tau=1}^{t} a_\tau(i)$, but weight all historical steps equally, regardless of recency. Neither approach accounts for the fact that recent attention can better predict future importance as queries drift.

This lack of recency bias is costly when attention is non-stationary. If the instantaneous importance at step t is given by $a_t(i)$, then an estimator that weights old and new observations equally will accumulate tracking error as the attention distribution shifts:

$$\mathrm{Err}(t) = E\big[\big\|s_{t_0}(\cdot) - a_t(\cdot)\big\|_1\big]. \quad (3)$$

For a non-stationary process with per-step variance σ^2, consistent with the drift observed in Figure 1, the equal-weight cumulative estimator used by H2O has tracking error that grows linearly with time:

$$\mathrm{Err}^{\mathrm{H2O}}(t) = O\big(\sigma^2 \cdot t\big), \quad (4)$$

as outdated observations continue to influence the score indefinitely. Purely static methods such as SnapKV can suffer even more severe error growth, as their scores are never updated after prefill. Thus the error grows with the very quantity long-context

inference seeks to scale: the number of generated tokens.

2.2 EMA-Attention

EMA-Attention replaces the static score with a running estimate that follows the current attention distribution. For every cached position, we update the score after each decoding step by

$$\begin{aligned} \tilde{s}_t(i) &= \alpha a_t(i) + (1-\alpha)\tilde{s}_{t-1}(i), \\ \tilde{s}_0(i) &= a_0(i). \end{aligned} \tag{5}$$

where $\alpha \in (0,1)$ controls how quickly the estimate reacts to new attention evidence. Unrolling the recursion yields

$$\tilde{s}_t(i) = \alpha \sum_{\tau=0}^{t} (1-\alpha)^{t-\tau}\, a_\tau(i), \tag{6}$$

so observations from recent steps receive exponentially larger weight than older observations. The score therefore remains smooth enough to suppress step-level noise, while still adapting when the model attends to a different part of the context.

Tracking error of EMA-Attention. Under the same non-stationary model, the EMA estimator satisfies

$$\mathrm{Err}^{\mathrm{EMA}}(t) = O\left(\frac{\sigma^2}{\alpha}\right), \tag{7}$$

which is independent of t. The bound depends on the drift scale and the decay factor, but not on the context length. This contrasts with the $O(\sigma^2 t)$ growth of static scoring rules and captures the main advantage of updating importance online.

Relation to prior methods. EMA-Attention unifies and improves upon existing dynamic scoring approaches. At $\alpha = 1$, the policy becomes purely myopic, ranking tokens only by the latest attention vector. As α decreases, the estimator incorporates more historical information, with $\alpha \to 0$ approaching the equal-weight cumulative behavior of H2O. Intermediate values of α provide a principled trade-off between responsiveness to new attention patterns and stability against step-level noise; our ablations (Table 2) show that this middle regime is consistently best.

Complexity. The update in Equation 5 adds one multiply-add per cached position, or $O(B)$ work per decoding step. This overhead is small relative to the attention computation, and the method introduces no trainable parameters.

2.3 Dynamic Pruning Algorithm

At each decoding step, EMA-Attention first updates the score of every cached position using the newly computed attention weights, then retains the top B positions under the updated scores:

$$\begin{aligned} \tilde{s}_t(i) &\leftarrow \alpha a_t(i) + (1-\alpha)\tilde{s}_{t-1}(i), \\ &\qquad \forall\, i \in \mathcal{S}_t, \\ \mathcal{S}_{t+1} &\leftarrow \text{top-}B\big(\tilde{s}_t\big). \end{aligned} \tag{8}$$

Since pruning is re-evaluated after every update, the retained set reflects the current attention pattern rather than a ranking frozen at prefill. The procedure wraps standard decoder attention and requires no changes to model weights, tokenization, or training.

3 Experiments

3.1 Experimental Setup

Models and baselines. We evaluate EMA-Attention on LLaMA-3-8B (Grattafiori et al., 2024) using greedy decoding. We compare against four representative KV cache methods: Full KV without compression, H2O (Zhang et al., 2023), SnapKV (Li et al., 2024), and PyramidKV (Cai et al.). All methods are evaluated under KV retention budgets of 20% and 40%, defined as the percentage of tokens retained from the full context (input + generated tokens). Unless otherwise specified, EMA-Attention uses $\alpha = 0.5$ (see Table 2).

Datasets. We report perplexity on WikiText-2 (Merity et al., 2016) and evaluate long-context reasoning performance on three representative LongBench (Bai et al., 2024) tasks: NarrativeQA (long-document QA), Qasper (scientific document QA), and HotpotQA (multi-hop reasoning). These tasks cover narrative understanding, scientific reasoning, and multi-step retrieval under contexts ranging from 3.6k to 18k tokens.

Implementation details. Experiments are primarily conducted on a single NVIDIA B200 GPU, with additional validation on 8×H20 and RTX 5090 platforms to verify scalability and consumer-grade deployability. All runs use `bfloat16` precision. EMA updates require only one additional fused multiply-add per cached token, introducing negligible computational overhead relative to standard attention. In practice, throughput overhead remains below 2% relative to H2O under matched KV budgets.

Method	Budget	WikiText-2 PPL ↓	NarrativeQA	Qasper	HotpotQA
Full KV	100%	10.28	33.8	39.7	43.5
H2O	20%	22.40	22.1	26.3	28.4
H2O	40%	15.20	26.7	30.4	33.1
SnapKV	20%	16.30	26.5	31.2	34.0
SnapKV	40%	12.70	29.4	35.0	38.1
PyramidKV	20%	13.40	29.8	35.6	38.9
PyramidKV	40%	11.30	32.1	38.0	41.5
Ours	**20%**	**11.90**	**32.5**	**38.4**	**41.9**
Ours	**40%**	**10.90**	**33.4**	**39.1**	**42.8**

Table 1: Main results on WikiText-2 and representative LongBench tasks using LLaMA-3-8B. Lower perplexity is better; higher QA scores are better. Bold indicates the best compression result at each budget.

α	WikiText-2 PPL ↓	LongBench Avg ↑
0.1	11.42	38.6
0.3	11.08	40.1
0.5	**10.91**	**41.8**
0.7	11.16	40.7
0.9	11.58	39.2
1.0	12.03	38.1

Table 2: Ablation over EMA coefficient α on LLaMA-3-8B with a 40% KV budget. Moderate EMA weighting achieves the best balance between adaptability and stability.

3.2 Main Results

Table 1 summarizes perplexity and LongBench QA performance across all methods and budgets. At the aggressive 20% KV budget, EMA-Attention substantially outperforms all baselines: it reduces WikiText-2 perplexity from 13.40 (PyramidKV) to 11.90 and improves NarrativeQA, Qasper, and HotpotQA by 2.7–3.0 points.

Notably, EMA-Attention at 20% budget already approaches Full KV performance (32.5 vs. 33.8 on NarrativeQA, 41.9 vs. 43.5 on HotpotQA). At 40% budget, it further narrows the gap to 10.90 perplexity (vs. 10.28 for Full KV) and outperforms PyramidKV across all datasets.

The larger performance margin at 20% budget supports our hypothesis that static importance estimates accumulate tracking error, making incorrect token evictions more harmful under aggressive compression.

We ablate the EMA coefficient at 40% KV budget.

Performance follows a clear inverted-U trend, with the best results achieved at $\alpha = 0.5$ on both perplexity and LongBench evaluation. When α is too small (e.g., 0.1), historical scores dominate and the method adapts too slowly to attention drift. Conversely, when α approaches 1.0, the scoring function becomes overly sensitive to short-term attention fluctuations, leading to unstable pruning decisions.

Importantly, all values within $\alpha \in [0.3, 0.7]$ consistently outperform both extremes, demonstrating that EMA-Attention is robust and does not require precise hyperparameter tuning. These results directly support our theoretical analysis: effective KV pruning requires balancing temporal stability with dynamic adaptability, and EMA weighting provides a favorable trade-off between the two.

4 Conclusion

A fundamental limitation of prevailing KV cache pruning techniques is their reliance on static importance scores computed during the prefill phase, despite the fact that token importance evolves dynamically at every step of autoregressive decoding. We identify this mismatch as **attention drift** and propose **EMA-Attention**, a simple training-free approach that dynamically tracks token importance using exponentially weighted attention updates.

Both theoretically and empirically, EMA-Attention consistently improves long-context KV cache pruning under aggressive memory budgets. On LLaMA-3-8B, EMA-Attention substantially outperforms H2O, SnapKV, and PyramidKV at matched retention ratios, with the largest gains appearing at the 20% budget where stale scoring errors become most severe. At a 40% KV budget, EMA-Attention achieves performance close to Full KV decoding while retaining less than half of the cache.

Our ablation studies further show that neither pure recency nor pure historical accumulation is sufficient: effective KV pruning requires balancing adaptability with temporal stability, which EMA weighting naturally provides.

Limitations

EMA-Attention introduces an inherent recency bias, which may be less effective for tasks requiring persistent attention to early-context information throughout long reasoning trajectories. Future work includes adaptive scheduling of the EMA coefficient based on attention dynamics, as well as combining EMA-based scoring with orthogonal compression techniques such as KV quantization.

References

Vaswani Ashish. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:I.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, and 1 others. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling, 2024. *URL https://arxiv.org/abs/2406.02069*.

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.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, and 1 others. 2021. A framework for few-shot language model evaluation. *Zenodo*.

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.

Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2023. Lm-infinite: Simple on-the-fly length generalization for large language models.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun S Shao, Kurt Keutzer, and Amir Gholami. 2024. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970.

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

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. 2024. Transformers are multi-state rnns. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18724–18741.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Yuwei Wan, Yixuan Liu, Aswathy Ajith, Clara Grazian, Bram Hoex, Wenjie Zhang, Chunyu Kit, Tong Xie, and Ian Foster. 2024. Sciqag: A framework for auto-generated science question answering dataset with fine-grained evaluation. *Preprint*, arXiv:2405.09939.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pages 21875–21895.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.