NoHairCache explained: one power-law exponent that unifies H2O and StreamingLLM KV-cache eviction

25 November 2026 · 8 min read · blogs.iictp.org/nohaircache-kv-cache-eviction-power-law-exponent-explained

Short answer: NoHairCache is a KV-cache eviction method from an arXiv preprint by Shuyang Xiang, submitted on 21 November 2026, that scores each cached token as its cumulative attention multiplied by an age penalty with exponent alpha, and shows that H2O and StreamingLLM are the two endpoints of that single family. The author reports that alpha can be calibrated once on one corpus and reused across datasets, with the lowest perplexity on WikiText, PG19 and arXiv among the methods compared. This is an arXiv preprint. It has not been peer reviewed, and its numbers are the authors' own claims.

TL;DR The short version

- NoHairCache, from an arXiv preprint submitted on 21 November 2026, scores each cached token by its cumulative attention times an age penalty with exponent alpha.
- It shows H2O and StreamingLLM are the two ends of one family, and reports the lowest perplexity on WikiText, PG19 and arXiv with alpha calibrated once.
- Tests used models of 410M to 1.5B parameters and gains are small, so measure full-cache quality on your own prompts before switching eviction policy.

Read online, listen to it, or change the text size: https://blogs.iictp.org/nohaircache-kv-cache-eviction-power-law-exponent-explained

What is KV-cache eviction?

During generation, a transformer stores keys and values for earlier tokens (the KV cache), which grows with context length. Eviction drops some entries to fit a memory budget. H2O keeps tokens with high accumulated attention. StreamingLLM keeps recent tokens plus an "attention sink" at the start.

How it works

- **One scoring rule.** Each token gets s = A × (t − i)^(−alpha), where A is cumulative attention and (t − i) is the token's age.
- **Endpoints.** The paper says H2O corresponds to alpha = 0 (no age penalty) and StreamingLLM to alpha going to infinity (recency only), with StreamingLLM additionally hard-coding an attention-sink patch.
- **Calibrate once.** The author claims alpha is a property of the model, calibrated once on a single corpus and transferable across domains. Calibrating on WikiText gives a range of 0.556 to 0.822 for Qwen2.5-1.5B-Instruct.

What is new compared with earlier approaches

Previous work treats the two methods as unrelated. The paper unifies them and replaces per-dataset tuning with a single calibrated number. The author also ties the admissible range of alpha to a “no-hair” mechanism from earlier work.

Results reported by the source

- Ranges of alpha are consistent across five models from 410M to 1.5B parameters and match optimal regions found by perplexity sweeps.
- NoHairCache gets the lowest perplexity across WikiText, PG19 and arXiv without per-dataset tuning, beating H2O and StreamingLLM ($p < 0.05$ on all three), with gains of 0.06 to 0.10 perplexity over StreamingLLM at a 512-token budget.
- On four-way multiple-choice QA it matches or beats StreamingLLM in all nine task-and-budget cells and retains 98.2% of full-cache accuracy on NarrativeQA at 25% retention against 91.1% for StreamingLLM. Per-cell significance tests are not significant, so the author treats QA as corroborating only.

Limitations and open questions

- Models are small (410M to 1.5B parameters); behaviour at 70B scale is untested here.
- Perplexity gains of 0.06 to 0.10 are small and QA differences are not statistically significant.
- The author says the calibration script and scorer will be released upon publication, so they cannot be run yet. Sources were read on 25 November 2026. Products, rules and standards change, so check the linked originals before you rely on any detail.

How to test it in your own serving stack

1. Measure full-cache quality on your own prompts first, as the reference.
2. Implement the score as attention accumulation times age to the power minus alpha.
3. Sweep alpha between 0 and a large value on a small calibration set and compare with H2O-style and recency-only baselines.
4. Check long-context tasks you care about, such as retrieval and summarisation, not only perplexity.

What it means for engineers running LLMs locally

If memory limits your context length, eviction policy is a lever. A single tunable exponent is simpler to maintain than separate policies, if the transfer claim holds for your model. See MoE expert streaming on 32 GB for another memory-saving technique.

Frequently asked questions

- **What is NoHairCache?** A KV-cache eviction scorer that combines cumulative attention with an age penalty controlled by one exponent.
- **How does it relate to H2O and StreamingLLM?** The paper says they are the alpha = 0 and alpha-to-infinity endpoints of the same family.

- **Does it need tuning per dataset?** The author reports calibrating once on WikiText and reusing the exponent.
- **Is it peer reviewed?** No, it is a preprint.

Related reading

- MoE expert streaming: running a 180B model in 32 GB
- llama.cpp v1.0.0
- Local LLM Practitioner certification

> Related credential: Local LLM Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Xiang S. NoHairCache: Calibrate-Once KV Cache Eviction with One Power-Law Exponent. arXiv 2611.29006, 21 Nov 2026 (preprint). arxiv.org/abs/2611.29006

Published by IICTP, an independent technical credential body, on 25 November 2026. The current version of this post is at https://blogs.iictp.org/nohaircache-kv-cache-eviction-power-law-exponent-explained. More guides: blogs.iictp.org.