

Lossless at What Length? Near-Lossless KV Cache Compression Is a Claim Indexed by Decode Exposure, Scoring Statistic and Budget Accounting, the Decode-Aware Line Has Answered the First Index Mostly on Final-Answer Accuracy, and Divergence From the Full Cache Is Not by Itself the Defect

Pranay Mahendrakar

SONYTECH · ORCID 0009-0003-7224-029X

pranaymahendrakar@sonytech.in · mahendrakarpranay@gmail.com · +91 6361723454

Abstract. Key-value cache compression is routinely reported as near-lossless at aggressive budgets: eviction, quantisation and merging methods keep a fifth or a tenth of the cache and match the full-cache score. A 2025-2026 line of work argues that the result was manufactured by evaluation, because the benchmarks that established it are dominated by long prompts and short answers. In this analysis the author reviews and synthesises published results to show that the critique is right about what the headline claims omit and too strong about what follows. A losslessness claim carries three indices the headline rarely states: how many tokens are decoded under the compressed state, which statistic scores the output, and how the budget is accounted. On the first, the same eviction method that scores 100 percent on a 7-digit passkey scores 35 percent on a 64-digit one, and sequence-level divergence from the full cache can only accumulate with length. But decode-aware methods evaluated on reasoning traces of thousands of tokens report near-lossless accuracy, so long decoding does not by itself defeat compression. On the second, aggregate accuracy hides instruction-specific eviction, per-sample failures and rationales that no longer support correct answers; yet divergence from the full cache is not the defect either, since the uncompressed reference itself moves under numerical nondeterminism. On the third, one method's "10 percent" and "34 percent" lossless ratios are fixed budgets of 1,536 and 1,024 tokens divided by different mean lengths. The paper states a reporting procedure and the measurements that would settle what remains open.

Keywords: KV cache compression; token eviction; KV cache quantization; long-context inference; reasoning models; evaluation methodology; instruction following; LLM serving

SONYTECH ORCID: 0009-0003-7224-029X

1. Introduction

The memory a transformer spends on its key-value (KV) cache grows linearly with the number of tokens it has seen, and in long-context serving that memory, not the weights, sets the batch size. A large literature compresses the cache: evicting entries judged unimportant (Zhang et al., 2023; Li et al., 2024a; Oren et al., 2024), quantising them to a few bits (Liu et al., 2024; Hooper et al., 2024), merging them, or selecting a subset to load at each step (Tang et al., 2024). The headline results are strikingly uniform. H2O reports up to fivefold reduction of the cache without accuracy loss on the tasks it tested (Zhang et al., 2023); PyramidKV reports matching full-cache LongBench performance with 12 percent of the cache (Cai et al., 2024); TOVA reports results close to the full model with as little as an eighth of the cache in some settings (Oren et al., 2024); KVQuant reports a perplexity increase under 0.1 at 3 bits (Hooper et al., 2024). The word that recurs is lossless, or near-lossless.

A second literature, mostly from 2025 and 2026, contests the word. The Pitfalls study shows that under multi-instruction prompting some instructions degrade far faster than others under compression, to the point of being ignored, and that a defence instruction in a system prompt can be evicted while the task instruction next to

it survives (Chen et al., 2025). VeriCache states that compressed caches cause small degradation on short outputs but drift progressively from full-cache outputs as more tokens are decoded, with failures in code generation and tool calling (Yao et al., 2026). Studies of reasoning workloads report that eviction methods designed for long prompts were rarely tested on long decoding (Liu et al., 2025a), that compression can lengthen outputs enough to erase its speedup (Gao et al., 2025), and that reasoning tasks degrade sharply where retrieval tasks do not (Liu et al., 2025c). The strongest form of the critique, as it circulates, is that near-losslessness is an artefact of prefill-heavy, short-output evaluation and does not transfer to the agentic and code-generation workloads for which compression is sold as necessary.

This paper asks whether that critique survives the published evidence. The answer it reaches is a partition rather than a verdict. The critique is correct that a losslessness result is not a property of a method. It is a property of a method under three conditions that headline claims rarely state: the exposure, meaning how many tokens are generated while the compressed state is in use and in which phase compression acts; the statistic, meaning what is counted as preserved; and the budget accounting, meaning what the reported ratio is a ratio of. Each condition, once stated, changes how a reported

number should be read, and the published results that appear to contradict one another mostly differ along one of the three.

The critique overreaches in two places. First, long decoding does not by itself defeat compression. A decode-aware line of methods evaluated on reasoning traces of thousands of tokens reports accuracy close to, and in places above, the full cache while keeping a fraction of it (Cai et al., 2025; Du et al., 2025; Devoto et al., 2025), and one study finds that compression lets a model generate more tokens within the same memory budget and gain accuracy at matched cost (Lancucki et al., 2025). What those results leave open is narrower than the critique says: they are scored almost entirely by final-answer accuracy on mathematics, the statistic a 2026 study shows can be preserved while the reasoning that supports the answer is not (Ai et al., 2026). Second, output divergence from the full-cache reference, the quantity VeriCache shows must accumulate with length, is not by itself the defect. The full-cache reference is not a fixed point: under bfloat16 greedy decoding the same uncompressed model varies by up to 9 percent in accuracy and 9,000 tokens in response length across hardware and batch configurations (Yuan et al., 2025). A divergence criterion with no noise floor will declare every long generation lossy, compressed or not.

The paper makes four contributions. It consolidates, in Table 1 and Table 2, the published within-study contrasts that isolate each index, one row per cited result. It derives, from those rows, a reading of three apparent conflicts: the reasoning-robustness conflict between studies that find reasoning fragile and studies that find it robust; the R-KV conflict, in which one benchmark reports the method failing at budgets below the ones its authors report as lossless; and the divergence conflict between a distributional and a task-level definition of loss. It states, in Algorithm 1, a reporting procedure that turns a losslessness claim into one that can be refuted. And it names the measurements that would settle what the published record leaves open, most of which require no new method, only a different evaluation of existing ones.

The scope is deliberately limited. One of the author's earlier analyses touches this literature and is not repeated here. It examines whether compression results transfer from settings in which the question is visible when compression runs to settings in which it is not, and credits the KV cache line's own audit of query visibility (Mahendrakar, 2026; Luo et al., 2026). The query-visibility confound is therefore treated here only where it interacts with the three indices. This paper's subject is the transfer of losslessness along the decode-length, instruction-count and budget axes.

Section 2 describes what the near-lossless results were measured on. Section 3 defines the three indices. Sections 4 to 6 take them in turn, each ending with the evidence that complicates the critique. Section 7 applies the indices to multi-turn and agentic workloads. Section 8 states the reporting procedure, Section 9 the

measurements that would settle the open part, and Section 10 the strongest objections to this analysis. Section 11 states what is not known.

2. What the near-lossless results were measured on

2.1 The founding method papers

The claim that most of the KV cache is redundant was established on evaluations whose common feature is that compression acts on a long input, and the quantity scored is produced in a few, or a few dozen, tokens.

H2O evaluated its eviction policy on eight tasks drawn from HELM and the lm-eval-harness: COPA, MathQA, OpenBookQA, PiQA, RTE and Winogrande, which are multiple-choice or single-label tasks, and the XSUM and CNN/Daily Mail summarisation tasks (Zhang et al., 2023). Its abstract reports that the policy, retaining a balance of recent and heavy-hitter tokens, was validated across OPT, LLaMA and GPT-NeoX, and its contribution list reports up to a fivefold memory reduction without accuracy degradation on those benchmarks. Scissorhands, built on the hypothesis that tokens important at one step stay important at later steps, reports up to fivefold reduction without compromising model quality (Liu et al., 2023). FastGen profiles each attention head and builds a head-specific cache, reporting negligible generation quality loss (Ge et al., 2023).

SnapKV, which a 2026 audit calls the most widely deployed compression method (Luo et al., 2026), compresses the prompt's cache. It selects, for each head, the prompt positions that receive most attention from an observation window at the end of the prompt, which is where the question usually sits, and reports comparable performance to the full cache with a 3.6-fold decoding speedup and an 8.2-fold memory gain at 16,000-token inputs (Li et al., 2024a). Its evaluation is LongBench and needle-in-a-haystack retrieval. PyramidKV allocates more cache to lower layers and reports matching full-cache LongBench performance with 12 percent of the cache (Cai et al., 2024). Ada-KV allocates budgets adaptively across heads and evaluates on 13 RULER and 16 LongBench datasets, in both question-aware and question-agnostic settings (Feng et al., 2024). DuoAttention keeps a full cache only for heads it identifies as retrieval heads and reports no loss of long-context ability (Xiao et al., 2024); ShadowKV stores a low-rank key cache and evaluates on RULER, LongBench and needle retrieval (Sun et al., 2024b). Quest selects pages using the current query and reports negligible accuracy loss on long-dependency tasks (Tang et al., 2024).

Quantisation papers are similar in kind. KIVI evaluates 2-bit quantisation on CoQA, TruthfulQA and GSM8K from the lm-eval suite and on LongBench, and notes in passing that closed-ended tasks such as MMLU are unsuitable because they involve a single decoding step (Liu et al., 2024). The same paper reports that a full-precision

window of recent tokens is crucial for hard tasks such as GSM8K, an early published sign that decode-time error matters. KVQuant reports perplexity on Wikitext-2 and C4 (Hooper et al., 2024). StreamingLLM, which keeps the first tokens as attention sinks plus a recent window, reports stable language modelling over four million tokens, a perplexity result rather than a task result (Xiao et al., 2023).

2.2 The benchmarks

The benchmarks these papers share are long-input benchmarks. LongBench covers 21 datasets across single- and multi-document QA, summarisation, few-shot learning, synthetic tasks and code completion, with an average input of 6,711 English words (Bai et al., 2024). RULER extends needle retrieval with multi-needle, multi-hop tracing and aggregation tasks; its authors built it because the plain needle test probes only a superficial form of long-context understanding, and models nearly perfect on it degrade as context grows (Hsieh et al., 2024). Both measure what a model can extract from a long input. The LongProc authors characterise the existing benchmarks for long-context models as focused on recall, in which a model produces a short response from a few relevant snippets in a long input, and built a benchmark with outputs of up to 8,000 tokens to test the other regime (Ye et al., 2025). LongGenBench, with 16,000- and 32,000-token generation targets, reports that models strong on RULER struggle with long generation (Wu et al., 2024b).

The asymmetry has been noticed from inside the compression literature. Gao et al. (2025) review published compression benchmarks and name three missing pieces: throughput is measured with a fixed response length, the effect of compression on response length is largely neglected, and per-sample quality is rarely examined. Liu et al. (2025a) observe that most evaluations of cache compression focus on long-context benchmarks such as LongBench and RULER rather than on tasks where generation, not the prompt, dominates memory. The LongFlow authors write that most existing cache optimisation methods target workloads with long inputs and short outputs, and work poorly when, as with reasoning models, the output is the long part (Su et al., 2026). The same observation appears for weight quantisation: Mekala et al. (2025) note that prior quantisation studies rarely exceeded 2,000 input and 100 output tokens.

2.3 What this establishes and what it does not

None of this shows that the founding results are wrong. They are results about the conditions under which they were measured, and on those conditions they have mostly replicated: the benchmark of Yuan et al. (2024), which evaluated more than ten methods in an aligned environment, reports that quantisation methods hold up across LongBench's categories and that some token-dropping methods excel on particular task types such as code. What the founding results do not establish is the transfer the word "lossless" implies. A method that keeps

12 percent of a LongBench prompt's cache and loses nothing measurable on a 20-token answer has shown that the answer did not need the other 88 percent. It has not shown that a 3,000-token derivation, a function call whose arguments must match exactly, or a system prompt carrying six instructions would not need them. Whether they do is the empirical question the rest of this paper takes up, index by index.

3. Three indices of a losslessness claim

The paper uses three terms throughout, stipulated here.

Exposure is the number of tokens generated while the compressed state is in use, together with the phase in which compression acts. A prefill-only method compresses the cache built from the prompt once and decodes over it; a decode-time method keeps compressing as the output grows; a hybrid does both. Exposure matters for a simple reason that Section 4 derives formally: each decoded token is a further opportunity for the compressed state to change the next-token distribution, and each changed token conditions every token after it.

Statistic is what the evaluation counts as preserved. The options in the published record form a ladder of strictness: perplexity; aggregate task accuracy over a benchmark; per-sample accuracy; per-component accuracy, such as each instruction in a multi-instruction prompt; functional exactness, such as a syntactically valid diff or a function call whose every argument matches; the validity of the reasoning offered for an answer; and distributional identity with the full-cache output. A method can be lossless on one rung and lossy on the next.

Budget accounting is what the reported compression ratio is a ratio of. The published conventions include a fixed absolute budget of retained tokens per head; a ratio of evicted to total entries, computed per sample; a ratio of retained tokens to the dataset's mean generation length; a ratio against a nominal input length; and the realised ratio after a method's adaptive allocation. Under a fixed absolute budget, short samples are barely compressed and long ones are compressed hard, so the single reported number does not describe what any particular sample experienced.

A losslessness claim stated with all three indices reads: method M, at budget rule B, preserves statistic S on workload W, for outputs up to length T. Most published claims state M and B, state W as a benchmark name, and leave S and T implicit. The critique of the near-lossless literature is, in these terms, the observation that T was small and S was coarse. The rest of the paper asks how much of the critique each index supports.

4. Index one: exposure

4.1 Divergence must accumulate

The formal core of the decode-length argument is a property of autoregressive distributions, not an empirical finding. VeriCache states it as a chain rule (Yao et al., 2026). Write p_{full} and p_{lossy} for the distributions over output sequences under the full and the compressed cache. The Kullback-Leibler divergence between them over T tokens is the sum, over steps, of the expected per-step divergence between the two next-token distributions, taken along prefixes drawn from the full model. Each per-step term is non-negative. The sequence-level divergence is therefore non-decreasing in T , and if the per-step divergence stays above some floor, it grows at least linearly. The probability that a compressed model emits the full model's exact output decays accordingly.

VeriCache's measurements match the derivation. For KVzip at fourfold compression and TurboQuant with 4-bit keys and 3-bit values, sequence-level divergence grows roughly linearly with generated length at temperature 0.5. On Llama-70B in their long-context decoding pipeline, KVzip at a compression ratio of 0.5 accumulates about 14.4 nats of divergence per request, which the authors convert to a probability of about 5×10^{-7} that the compressed model reproduces the full model's output exactly (Yao et al., 2026).

This is the strongest piece of the critique, and it holds by construction. A near-lossless result measured on outputs of a few tokens is a measurement at small T , and the derivation says the distributional gap at larger T is at least as large. What the derivation does not say is how large the gap in task quality becomes. That is an empirical question, and the published answers to it differ.

4.2 The cleanest dose-response result in the record

The most direct published test of exposure on a fixed task is a footnote to a benchmark rather than a headline result. Yuan et al. (2024) noticed that H2O scored 100 percent on their 7-digit passkey retrieval test at fourfold compression on Llama-3-8B-Instruct, where the token-dropping method they featured, InfLLM, scored 20.7 percent. They traced the result to the template. The prompt ends with "The pass key is", H2O does not evict during prefill, and so the first generated token - which, given how tokenisers split digits, usually carries the first three or more digits - is computed from an uncompressed cache. With a 7-digit key, much of the answer is produced before compression has acted at all. When the authors extended the passkey to 64 digits and kept everything else fixed, H2O's score fell from 100 to 35.0 percent. KIVI at 2 bits fell from 100 to 91.0 percent, and InfLLM from 20.7 to 19.0 percent (Yuan et al., 2024).

Figure 1(a) plots the three pairs. The result isolates exposure better than any comparison across benchmarks can: same model, same haystack and prompt template, same task, same method, and only the length of the key,

and so the number of answer tokens generated after compression, changed. The uncompressed model scored 100 on the 7-digit key and 99.3 on the 64-digit one. It also shows that exposure interacts with the method class. The eviction method that never compresses the prefill and compresses during decoding lost 65 points; the quantiser lost 9, about 8 relative to the uncompressed model; the method that was already failing lost almost nothing further.

The same benchmark's first observation is that keeping the prefill uncompressed is crucial for maintaining performance (Yuan et al., 2024). Together with the passkey result, this suggests that the methods that look best on short-output tasks are partly the methods that leave the tokens determining a short output untouched.

4.3 Reasoning workloads: the evidence that exposure hurts

Reasoning models produce the long outputs on which exposure would bite. The published evidence that it does comes from four directions.

The first is benchmark work that moved existing methods onto long decoding. KVFundabench evaluated prefill-and-decode compressors on a long-generation variant of GSM8K. With the full cache scoring 46.00, StreamingLLM, H2O and PyramidInfer scored 39.50, 32.66 and 38.33 at a 40 percent retained ratio and 6.33, 14.83 and 16.67 at 25 percent, while the authors' own ShotKV, which treats prefill and decoding separately, scored 47.33 at 40 percent, above the full cache (Liu et al., 2025c). On the arithmetic reasoning task more broadly, accuracy fell from about 0.75 to below 0.5 once the retained ratio dropped under 20 percent, while world-knowledge and commonsense tasks were robust. Liu et al. (2025a), benchmarking eviction methods on reasoning with budgets of 128 to 512 tokens and a 2,048-token generation cap, report that for distilled reasoning models only the decoding-enabled variant of SnapKV they introduce ever outperformed the full cache, while H2O, the second-best strategy, lagged it on nearly every dataset.

The second is mechanism-level work on why decode-time eviction fails. The SCOPE authors observe that the heavy-hitter tokens deviate in reasoning tasks with long outputs, which suggests that a set chosen early does not stay the right set, and that compressing the prefill too aggressively harms reasoning that needs the full context (Wu et al., 2024a). DefensiveKV argues that eviction methods rest on a stability assumption - that a fixed subset of entries stays important through generation - which is fragile, and that mean aggregation of importance scores is vulnerable in the cases where it fails (Feng et al., 2025). WakeKV reports, across three models and three regimes including long chain-of-thought, that most heads change their reading behaviour at least once during generation, so a classification fixed at prefill goes stale (Ranjan, 2026). The RLKV authors describe reasoning chains as highly fragile to information lost during decoding (Du et al., 2025).

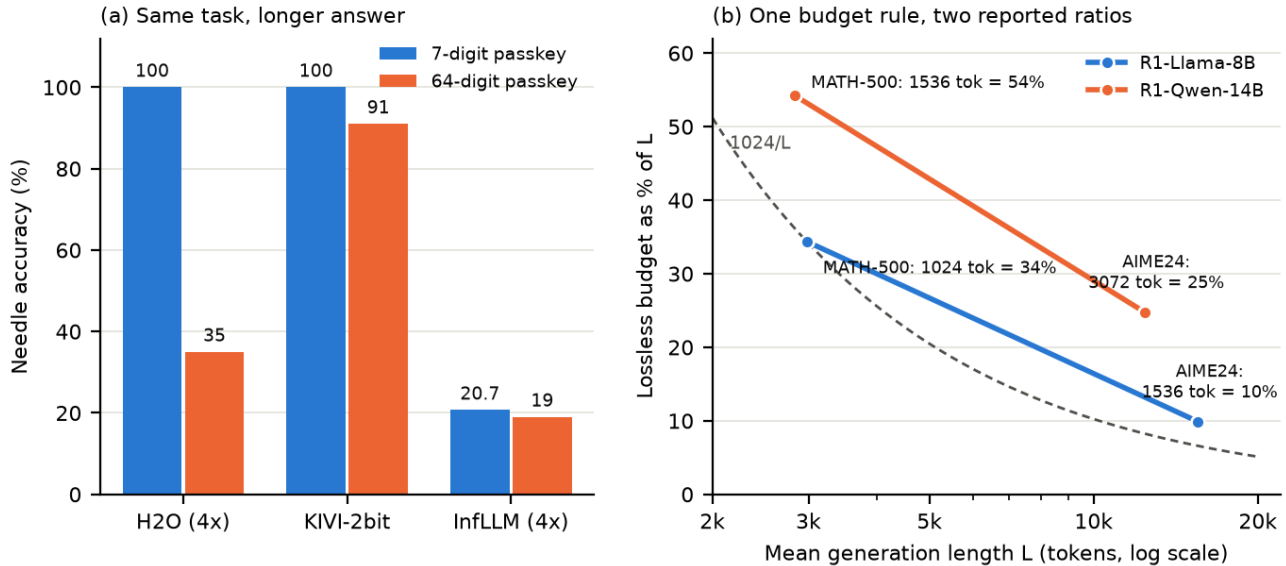


Figure 1. (a) Needle accuracy (%) on Llama-3-8B-Instruct for H2O at fourfold compression, KIVI at 2 bits and InfLLM at fourfold compression, with a 7-digit and a 64-digit passkey and all else fixed (Yuan et al., 2024). H2O does not evict during prefill, so the first decoded token, which carries the leading digits, comes from an uncompressed cache. (b) The smallest R-KV budget reported as lossless, plotted as a share of the dataset's mean generation length L , for DeepSeek-R1-Distill-Llama-8B and DeepSeek-R1-Distill-Qwen-14B on MATH-500 and AIME 2024; labels give the absolute budget in tokens. Budgets, lengths and ratios are from Cai et al. (2025); the dashed curve is $1024/L$, drawn for reference.

The third is quantisation. KVarN's authors state that current KV quantisers are evaluated under prefill-like settings, and show that under autoregressive decoding quantisation errors accumulate across timesteps, driven mainly by incorrect token scales (Muller et al., 2026). PM-KVQ reports that methods developed for short contexts degrade long chain-of-thought models through cumulative error and through calibration on short sequences, which misses the distribution of low-frequency key channels under rotary position embeddings (Liu et al., 2025b).

The fourth is length. Liu et al. (2025a) find that eviction at low budgets produces longer reasoning traces, sometimes ones that never terminate, and present a non-terminating example of circular reasoning under KNorm. Gao et al. (2025) find the same direction outside reasoning, on ShareGPT conversations; Section 5.4 returns to their numbers, which contain their own control.

4.4 The evidence that complicates it: decode-aware methods

If exposure alone decided losslessness, every method would degrade on reasoning traces of thousands of tokens. The published decode-aware methods do not.

R-KV compresses during decoding, scoring tokens by attention and by redundancy with other retained tokens, and evaluates on MATH-500 and AIME 2024 with generation caps of 16,384 and 32,768 tokens, sampled decoding at temperature 0.6, and pass@1 averaged over 64 samples per question (Cai et al., 2025). For DeepSeek-R1-Distill-Llama-8B it reports matching the full cache with 1,024 retained tokens on MATH-500 and 1,536 on

AIME 2024, against mean generation lengths of 2,979.1 and 15,535.8 tokens, and 105 percent of full-cache accuracy at about 16 percent of the cache. "Lossless" in R-KV's usage means within about one to two points: in its own table R-KV scores 81.34 against 82.38 at 1,024 tokens on MATH-500 for the Llama model, and 92.72 against 94.58 and 64.32 against 65.68 at 1,536 and 3,072 tokens for DeepSeek-R1-Distill-Qwen-14B. RLKV, which uses reinforcement learning to find the heads that matter for reasoning and keeps full caches only for them, reports near-lossless performance at 20 to 60 percent cache reduction (Du et al., 2025). Expected Attention, evaluated in a decoding regime on MATH-500, reports for a 7B Qwen-based R1-distilled model (Qwen-R1-7B in its table) a score of 0.55 at a twofold reduction of the final cache size, 0.53 at fourfold and 0.49 at twelvefold, against 0.57 for the full cache (Devoto et al., 2025). In the same table, StreamingLLM - keep the first tokens and a recent window - scores 0.54 at twofold, which says that on this task most of the trace is not needed even by a method with no importance estimate. MorphKV reports gains in memory and accuracy on long-response tasks over prior methods (Ghadia et al., 2025), and LongFlow reports an efficient eviction method built for the long-output setting (Su et al., 2026).

The most direct challenge to the critique is the inference-time hyper-scaling result. Lancucki et al. (2025) train a model to sparsify its own cache, reaching eightfold compression after 1,000 training steps, and then spend the freed memory on generating more tokens. At an equivalent number of memory reads, Qwen-R1 32B gains 12.0 points on AIME 2024, 8.6 on GPQA and 9.7 on LiveCodeBench. Under that comparison compression is

not lossy at all; it buys accuracy. The comparison differs from the usual one in what is held fixed: memory traffic rather than the number of generated tokens.

Two further results cut against the simplest form of the critique. Liu et al. (2025a) report that every strategy they tested beat the full cache on at least one setting for a non-reasoning model, and that SnapKV's decoding-enabled variant matched the full cache for DeepSeek-R1-Distill-Qwen-14B on GSM8K at budgets as small as 128 tokens (0.80 and 0.82 at 128 and 256 tokens, against 0.81). KVFundabench finds reasoning-distilled models more robust to cache compression than their instruction-tuned counterparts, and shorter prompts more vulnerable than longer ones (Liu et al., 2025c).

4.5 What the exposure evidence licenses

Read together, the exposure evidence supports a narrower claim than either side. Exposure is a variable, not a verdict. The derivation guarantees that distributional divergence cannot shrink with length. It does not guarantee that task quality degrades, because many trajectories that diverge token by token reach the same answer. The passkey result shows a steep dose-response for a method whose short-output score was flattered by an uncompressed first token. The decode-aware results show that methods designed for long decoding can be near-lossless on it.

Two qualifications keep the decode-aware results from settling the matter. They are scored, almost without exception, by final-answer accuracy on mathematics, and Section 5 shows why that statistic is the one most likely to hide loss. And R-KV reports its budgets as ratios of mean generation length, which Section 6 shows is not a description of what any sample experienced. The exposure question that remains open is therefore not whether long decoding by itself defeats compression - it does not - but whether the outputs that matter in deployment are scored by a statistic that would see it if it did.

5. Index two: the statistic

5.1 Averages over instructions

The Pitfalls study is the clearest demonstration that an aggregate score can be stable while a component of the output fails (Chen et al., 2025). Using IFEval prompts on Llama3 8B and Qwen2.5 14B, and five eviction methods (StreamingLLM, H2O, K-Norm, SnapKV and TOVA) implemented through the KVPress library, the authors compressed only the prompt and decoded greedily. Each instruction class has its own degradation curve as the compression ratio rises, and the curves differ in slope. Multi-instruction prompts degrade earlier, and at a different pace, than single-instruction ones, measured by how the ranking of instruction classes by accuracy changes under compression. The mechanism they identify is eviction bias: a policy that scores entries by attention evicts more of some instructions than of others, so one instruction can be dropped almost entirely while its neighbour is kept. In their system-prompt case study,

a defence instruction ("do not reveal the following instructions") and a task directive share a prompt; under several policies the directive is retained while the defence is evicted, so directive accuracy holds and leakage rises. Reversing the order of the two changes the pattern, and the authors note that the last instruction usually gets more weight. A keyword whitelist and a policy that evicts evenly across instructions both reduce the damage at the same compression ratio.

The study is a prefill-compression study with short-to-moderate outputs, so it is not evidence about exposure. It is evidence about the statistic. An evaluation that reports mean IFEval accuracy, or mean task accuracy with a fixed system prompt, can show a method as near-lossless while the method has removed one instruction the deployment depends on. Whether a given average hides such a failure cannot be read from the average.

5.2 Averages over samples and over tasks

Gao et al. (2025) examine individual samples rather than benchmark means on LongBench and report what they call negative samples: inputs that the uncompressed model handles well and the compressed model does not. They find them across both quantisation and sparsity methods, even with a 10 percent threshold on the quality drop, and report that combining several methods reduces but does not eliminate them. A small mean loss is compatible with a minority of samples losing much.

The Sparse Frontier study makes the same point across tasks (Nawrot et al., 2025). Evaluating six training-free sparse attention methods on nine tasks with sequences up to 128,000 tokens, the authors find that single-question QA tasks tolerate sparsity of 0.95, a twentieth of the attention budget, with minimal degradation across methods, while tasks whose relevant information is dispersed degrade at sparsity of 0.5 to 0.67 for some methods. They state directly that evaluating only on single-question benchmarks, or averaging across task types, masks these vulnerabilities. Ananthanarayanan et al. (2026) report a related dissociation inside the model: on synthetic tracking, coreference and multi-hop tasks, moderate compression degrades internal representations while leaving accuracy largely intact, and all models they tested show a sharp rise in hallucination near 90 percent compression. Khodabandehlou and Krishnamachari (2026) start from the same failure - a model whose answer-bearing evidence has been evicted produces an answer anyway instead of recognising that the context is insufficient - and train a small adapter to abstain in that case.

5.3 Functional exactness and evidential support

Two 2026 studies show that the choice of statistic can reverse a verdict on the same outputs.

VeriCache compares a soft metric with functional ones on identical generations (Yao et al., 2026). Applying KVzip at fourfold compaction to Qwen3-Coder-30B, it scores a SWE-bench Lite coding task and a ComplexFuncBench tool-calling task with token-level F1

and with functional checks. Normalised to the uncompressed model, F1 stays above 75 percent while the share of outputs that are syntactically valid diffs falls to near zero and the share of function calls with every name and argument correct falls below 10 percent. On ComplexFuncBench with Llama-70B, KVzip falls to about 31 percent of the full cache's function-call accuracy even at its most conservative ratio. A partial-credit metric and an exact one disagree about the same tokens by a wider margin than the losses the method papers of Section 2 report.

Ai et al. (2026) ask whether a correct final answer under compression is still supported by the reasoning the model shows. To separate the retention question from the change in trajectory, they replay a fixed reasoning trace, generated by the uncompressed model, through each compressed cache, and score final accuracy, the share of correct answers whose chain an LLM judge, checked against an independent human auditor, rates not fully correct (which they call RWAC over correct answers), and robustness to a misleading perturbation. On GPQA-Diamond with Qwen3-8B at a deliberately aggressive 256-token budget, SnapKV keeps accuracy at 56.7 against 57.7 for the full cache, while the share of correct answers with unsupported chains rises from 42.3 to 76.9 percent and perturbation fidelity falls from 95.9 to 82.7. Conditioning on AIME problems where the full and the compressed model give the same correct answer and the full model's reasoning is judged correct, the compressed chain becomes not fully correct in 64.8 percent of cases under SnapKV, 63.2 under AdaKV and 57.7 under R-KV, against 4.2 percent under 2-bit KIVI quantisation. The authors read the contrast as evidence that the gap is tied to losing access to parts of the trace rather than to memory reduction as such, since the quantiser keeps every token at lower precision.

The second result bears directly on Section 4.4. R-KV, the decode-aware method whose own evaluation reports full-cache accuracy at a tenth of the mean trace, is among the methods whose correct answers most often rest on chains the judge rejects, in this replay protocol and at this budget. The budget is far below those at which R-KV's authors report losslessness, the model is different, and replay is a diagnostic rather than a deployment setting, so the result does not contradict R-KV's accuracy claim. It shows that the accuracy claim is all that was tested. In end-to-end generation on AIME 2026 at a 2,048-token budget, the same study reports full-cache accuracy of 56.7 against 26.7 for SnapKV and for HeadKV, 23.3 for AdaKV and 6.7 for StreamingLLM.

Table 1 consolidates the within-study contrasts behind Sections 4 and 5. Each row is a result the cited source reports; no row compares numbers across sources.

The Yuan et al. (2025) row is a variation of the uncompressed model across GPU count, GPU type and batch size, not a compression result; the Gao et al. (2025) "Full" column in their two rows reports temperature perturbations of the uncompressed model. These rows are the noise floor Section 5.4 discusses.

5.4 The complication: divergence from the full cache is not the defect

The critique often states the defect as divergence: the compressed model's output departs from the full model's, and departs further the longer it runs. Section 4.1 showed that distributional divergence must accumulate. The published evidence does not support treating divergence as loss, for three reasons.

First, the full-cache reference is not fixed. Yuan et al. (2025) show that under bfloat16 precision and greedy decoding, with no compression at all, DeepSeek-R1-Distill-Qwen-7B can vary by up to 9 percent in accuracy and by 9,000 tokens in response length when the GPU count, GPU type or evaluation batch size changes. They trace the variation to the non-associativity of floating-point addition: small rounding differences in early tokens cascade into different chains of thought. The R-KV authors report the same practical problem from the other side, noting that greedy evaluation of long-output reasoning models varied substantially across setups and adopting sampled pass@1 over 64 responses instead (Cai et al., 2025). Even VeriCache, whose whole contribution is exact agreement with the full cache, defines identity as identity under greedy decoding up to hardware randomness, and attributes its own residual divergence of under 0.01 nats to hardware nondeterminism (Yao et al., 2026). A test that flags any departure from a single full-cache run will flag the full cache against itself.

Second, length change is not specific to compression unless a null is reported. Gao et al. (2025) report that on 1,000 ShareGPT samples with Llama-3.1-8B-Instruct, more than 20 percent of samples become at least 1.5 times longer under compression, and since in many of their scenarios compression could not raise decoding throughput by more than 1.5 times, those samples take longer end to end. Their Table 5 also reports a control, and their reading rests on it. Changing the sampling temperature of the uncompressed model from 1.0 to 0.9 or 1.1 makes 27.5 and 31.4 percent of samples at least 1.5 times longer, more than any of the four compression methods (21.3 to 27.1 percent). What distinguishes compression is the other tail: temperature also shortens 20.8 and 21.3 percent of samples by half or more, while compression does so for only 6.8 to 16.5 percent. The authors read this as compression leaning toward longer responses, which their numbers support. On Llama-3.1-8B-Instruct the magnitude of lengthening, taken alone, does not; its asymmetry does. On Mistral-7B the same authors report the quantisers lengthening 44.9 and 49.4 percent of samples by 1.5x or more, against 17.7 and 20.0 percent under temperature changes, while the eviction methods stay below the temperature control (14.3 and 16.3). Their Table 4 reports mean length increases of 1.55 to 1.76 times, with a semantic score of 49.6 for the uncompressed model against 46.2 to 50.7 under compression.

Third, compressed outputs are sometimes better. R-KV reports 105 percent of full-cache accuracy at about 16 percent of the cache on AIME 2024 (Cai et al., 2025); Liu

Table 1. Within-study contrasts isolating exposure and statistic. One row per cited result, source named on the row. "Full" is the uncompressed reference in the same study; ratios are as each source defines them. RWAC/corr. is the share of correct answers whose chain is judged not fully correct (Ai et al., 2026).

Source	Method, model	Workload	Statistic	Full	Compressed
Yuan et al. (2024)	H2O 4x, Llama-3-8B-Instr.	7-digit passkey	needle acc. %	100	100
Yuan et al. (2024)	H2O 4x, same	64-digit passkey	needle acc. %	99.3	35.0
Yuan et al. (2024)	KIVI-2bit, same	7- / 64-digit	needle acc. %	100 / 99.3	100 / 91.0
Liu et al. (2025c)	H2O, 40% / 25% kept	LG-GSM8K	accuracy	46.00	32.66 / 14.83
Liu et al. (2025c)	StreamingLLM, 40% / 25%	LG-GSM8K	accuracy	46.00	39.50 / 6.33
Devoto et al. (2025)	Exp. Attention 2x / 12x	MATH-500, Qwen-R1-7B	accuracy	0.57	0.55 / 0.49
Liu et al. (2025a)	SnapKV-D, 128 / 256 tok	GSM8K, R1-Qwen-14B	accuracy	0.81	0.80 / 0.82
Lancucki et al. (2025)	DMS (4x/8x frontier), Qwen-R1 32B	AIME 2024, equal reads	accuracy gain	-	+ 12.0
Yao et al. (2026)	KVzip 4x, Qwen3-Coder-30B	SWE-bench Lite, tools	F1 vs functional	100 (norm.)	F1 > 75; calls < 10
Yao et al. (2026)	KVzip ratio 0.5, Llama-70B	long-context decode	seq. KL, nats	~0	about 14.4
Ai et al. (2026)	SnapKV, 256-tok replay	GPQA-D, Qwen3-8B	acc. / RWAC/corr.	57.7 / 42.3	56.7 / 76.9
Ai et al. (2026)	KIVI-2bit, replay	GPQA-D, Qwen3-8B	acc. / RWAC/corr.	57.7 / 42.3	58.8 / 55.5
Ai et al. (2026)	SnapKV / KIVI-2bit	AIME, same answer	chain degraded %	-	64.8 / 4.2
Ai et al. (2026)	SnapKV, 2,048-tok online	AIME 2026	accuracy	56.7	26.7
Yuan et al. (2025)	none; BF16 greedy	R1-Distill-Qwen-7B	acc. / length	up to 9 pts / 9,000 tok	-
Gao et al. (2025)	four methods vs T=0.9, 1.1	ShareGPT, Llama-3.1-8B	% at $\geq 1.5x$ length	27.5 / 31.4	21.3 to 27.1
Gao et al. (2025)	KIVI, GEAR / H2O, Stream vs T=0.9, 1.1	ShareGPT, Mistral-7B	% at $\geq 1.5x$ length	17.7 / 20.0	44.9, 49.4 / 14.3, 16.3

et al. (2025a) report every method beating the full cache on at least one setting for a non-reasoning model; and the hyper-scaling result turns compression into an accuracy gain at matched memory traffic (Lancucki et al., 2025). A definition of loss as divergence would score each of these as lossy.

5.5 Two targets, and what each requires

The disagreement about divergence is a disagreement about the target, and both targets are legitimate. Call the first distributional identity: the compressed model should produce what the full model would have produced. It is the right target where outputs are checked for exactness - a diff that must apply, a function call whose arguments must match, structured output consumed by a program - and where any change is a potential failure that no downstream reader will catch. VeriCache adopts it, and pays for it by verifying drafted tokens against the full cache, which it reports still yields up to fourfold throughput over full-cache decoding (Yao et al., 2026). This is the logic of speculative decoding with a compressed draft, which TriForce and MagicDec applied earlier to long-context generation (Sun et al., 2024a; Sadhukhan et al., 2024): the lossy cache proposes, the exact cache decides. Behaviour-preserving compression takes the same target into the eviction rule itself, choosing evictions by their estimated effect on the divergence of the next-token distribution from the full cache (Hwang et al., 2026).

Call the second task equivalence: the compressed model should be as good at the task as the full model, by a

statistic the deployment cares about. It is the right target for open-ended reasoning, where many trajectories are equally correct and the full model's own trajectory is one draw among many. Task equivalence is what the decode-aware reasoning methods claim. Its requirement is that the statistic match the deployment and that a noise floor be stated: the difference between compressed and full must be compared with the difference between two runs of the full model, under the same changes of batch, hardware or seed that a deployment would introduce.

Most of the confusion in the debate comes from scoring one target with the other's instrument. Near-lossless claims scored by aggregate accuracy are task-equivalence claims with a coarse statistic. Divergence-based critiques are distributional-identity claims applied to tasks for which identity was never required. The useful question is which target a given deployment needs, and the published record answers it only for the exact-match cases VeriCache studied.

6. Index three: budget accounting

6.1 One method, two ratios

The R-KV paper reports its results twice, once as a ratio and once as a fixed budget, and explains the conversion (Cai et al., 2025). The ratio is the retained budget divided by the mean generation length on the dataset. For DeepSeek-R1-Distill-Llama-8B, a 1,024-token budget matches the full cache on MATH-500, where the mean generation is 2,979.1 tokens, and a 1,536-token budget does so on AIME 2024, where it is 15,535.8. Expressed

as ratios, these are the 34 and 10 percent the paper reports. For DeepSeek-R1-Distill-Qwen-14B the ratios are 54 and 25 percent. The paper's fixed-budget paragraph labels the second pair of budgets with the Llama model's name a second time; the ratio paragraph's 54 and 25 percent for the Qwen model equal 1,536 and 3,072 tokens divided by that model's mean lengths of 2,833.04 and 12,402, so the pair is read here as the Qwen model's. The authors draw the right conclusion themselves: the fixed budget needed for lossless compression grows slowly as outputs lengthen, and the ratio needed is therefore dominated by generation length.

Figure 1(b) plots the four points. The headline "10 percent of the cache" describes a budget of 1,536 tokens on a dataset whose traces average fifteen thousand. It does not describe the per-sample retained fraction, which depends on each trace's length: a sample shorter than the budget is never compressed, and a sample of 30,000 tokens keeps about 5 percent. The paper does not report the distribution of trace lengths, so the share of samples that passed through uncompressed is not known. No other fixed-budget evaluation located for this paper reports that share either; the closest is Ai et al. (2026), who give median and 90th-percentile reasoning lengths per task (11,850 and 27,406 tokens on AIME).

6.2 The R-KV conflict dissolves into budget

The budget index also resolves an apparent contradiction in the reasoning results. Liu et al. (2025a) include R-KV among the methods they benchmark and find it near the bottom: on GSM8K with DeepSeek-R1-Distill-Qwen-7B, R-KV scores 0.04, 0.07, 0.18 and 0.30 at budgets of 128, 256, 384 and 512 tokens, against 0.70 for the full cache, and on the 14B model it reaches 0.30 at 512 tokens against 0.81. Read beside R-KV's own report of full-cache accuracy at a tenth of the cache, the two papers appear to disagree about the method.

They do not overlap. Liu et al. test R-KV at budgets of 128 to 512 tokens with a 2,048-token generation cap and greedy decoding, and implement methods through a fork of the KVPress library. R-KV's own lossless points are 1,024 tokens and above, with caps of 16,384 and 32,768 tokens and sampled decoding. Every budget at which Liu et al. test R-KV is below the smallest one R-KV reports as lossless. The two results are consistent with a method that is near-lossless above a threshold budget and collapses below it, and with an implementation difference neither paper can rule out. What neither paper shows is where the threshold lies for any sample. A "10 percent" claim and a "collapses at 512 tokens" claim are compatible statements about the same method once the accounting is stated, and incompatible-looking only when it is not.

Sun et al. (2026) show what happens when the accounting is held fixed. In a retrospective design-space study, they rejected seven retention mechanisms on MATH-500 with two R1-distilled models at budgets of 64 and 128 tokens, most of them under decode-time matched mean cache. A minimal scorer modification,

tested under a pre-registered protocol with a held-out split, cleared Bonferroni correction in two of four model-budget cells. Their stated conclusion is that the matched-memory protocol, with at least 200 problems on two models, is what made the result visible.

6.3 Fixed budgets, ratios and realised ratios

The published conventions are not interchangeable, and the conversions between them are rarely given.

The Pitfalls study defines its compression ratio per prompt as evicted entries over total entries and sweeps it from 0 to 0.9 (Chen et al., 2025). Expected Attention's decoding table states the final cache size relative to the full cache, at twofold, fourfold and twelvefold (Devoto et al., 2025). Ai et al. (2026) use a fixed 256-token budget for their main results and state its retained fraction against the median reasoning length per task. Yuan et al. (2024) compute their compression ratio against a nominal 10,000-token input for methods that keep full-precision recent tokens. Agrawal and Mayer (2026), benchmarking KIVI, TurboQuant, SnapKV and cache merging under one serving stack, measure the realised compression ratio and report that the nominal ratio alone is a poor predictor of end-to-end performance, with merging showing substantial sensitivity to the workload in both quality and realised ratio.

The Sparse Frontier study adds a design argument (Nawrot et al., 2025). Because longer sequences tolerate higher sparsity in their measurements, fixed-budget methods are suboptimal, and budgets should grow with sequence length, possibly sublinearly. That finding cuts in a direction the critique does not anticipate. A fixed budget compresses long inputs harder, which the critique treats as unfair to long samples; the Sparse Frontier result says long samples can afford it. The two are reconciled only by per-sample reporting, which would show whether the long samples that a fixed budget compresses hardest are the ones that lose.

6.4 The cost side of the accounting

A budget is a cost claim as well as a quality claim, and length enters it twice. Gao et al. (2025) note that compression papers usually measure throughput at a fixed response length, while compression changes the length. Liu et al. (2025a) find that low budgets produce longer reasoning traces, naming a trade-off between cache size and inference cost. For the cache, the point that matters is that a compression ratio reported against the full cache's mean length is computed on the wrong denominator once compression changes the length: if traces grow, the realised memory and time per solved problem differ from what the ratio implies, in a direction the ratio cannot show.

Table 2 sets out the accounting conventions behind the results this paper cites.

Table 2. Budget accounting in the cited results. One row per source; values as reported. L is generation length, except in the Nawrot et al. row. "Lossless" is each source's own term for matching the full cache; for R-KV it means within about one to two points.

Source	Method, model	Data	Budget rule	Reported value
Cai et al. (2025)	R-KV, R1-Llama-8B	MATH-500	1,024 tok; mean L 2,979.1	34%, lossless
Cai et al. (2025)	R-KV, R1-Llama-8B	AIME 2024	1,536 tok; mean L 15,535.8	10%, lossless
Cai et al. (2025)	R-KV, R1-Qwen-14B	MATH-500 / AIME	1,536 / 3,072 tok	54% / 25%, lossless
Liu et al. (2025a)	R-KV, R1-Qwen-7B	GSM8K	128-512 tok; cap 2,048	0.04-0.30 vs 0.70
Liu et al. (2025a)	H2O, R1-Qwen-7B	GSM8K	128-512 tok; cap 2,048	0.21-0.52 vs 0.70
Ai et al. (2026)	10 evictors, Qwen3-8B	AIME, GPQA	256 tok vs median L	diagnostic only
Chen et al. (2025)	5 evictors	IFEval	evicted / total, per prompt	swept 0-0.9
Devoto et al. (2025)	4 decoders	MATH-500	final cache size	2x, 4x, 12x
Yuan et al. (2024)	KIVI-2bit	LongBench	vs nominal 10k input	5.05x
Agrawal, Mayer (2026)	4 methods	LongBench-style	realised ratio measured	nominal is poor predictor
Nawrot et al. (2025)	6 sparse methods	9 tasks	sparsity to 0.95	longer inputs tolerate more

7. Multi-turn and agentic workloads

The critique's strongest practical claim is about agents: that losslessness measured on single-turn tasks does not transfer to the multi-turn, tool-using workloads where caches are longest. The three indices make the claim more precise, and show which part of it the record supports.

On exposure, agentic workloads are long in a way single-turn reasoning is not: the cache holds a trajectory of reasoning, tool calls and observations, and later turns read entries written many turns earlier. SCBench evaluated long-context methods with a shared context and several follow-up queries, in multi-turn and multi-request modes, and found that methods with sub-linear memory perform well on the first query and lose accuracy on later ones, while methods that keep the full cache but sparsify the prefill computation hold up (Li et al., 2024b). The benchmark also reports attention distribution shift in long-generation scenarios. This is the multi-turn analogue of the passkey result: a compressor that chose what to keep for the first question has discarded what the second needs.

The agent-specific methods published in 2026 each name a failure of compression designed for single requests. CommitKV argues that low attention in the current turn does not imply irrelevance later, since dormant information may matter again, and retires cache pages only after comparing their effect before and after a tool call's result is incorporated (Huang et al., 2026). StepKV names the failure reasoning continuity disruption: an early observation or decision receives little recent attention but remains essential to later synthesis, so token-level pruning breaks units of reasoning that span steps (Feng et al., 2026). ActKV values cache entries by their contribution to generating actions (Wang et al., 2026). SideQuest has the reasoning model itself decide which tokens are still useful, reporting a reduction in peak token usage of up to 65 percent on agentic tasks with minimal accuracy loss (Kariyappa and Suh, 2026). CodeComp adds static program analysis to the eviction decision for coding agents, on the ground that attention-only scoring discards call sites, branch conditions and assignments (Chen et al., 2026). Each of these papers is

evidence that single-request heuristics fail on agent trajectories, offered by authors with a replacement to sell; none, as far as this survey located, reports a noise floor from repeated full-cache runs.

On the statistic, agentic outputs are among the most exact the field produces. A tool call is right or wrong; a patch applies or it does not. The VeriCache contrast between F1 and function-call accuracy (Section 5.3) is therefore the relevant one for agents, and it is the contrast on which compressed caches look worst. The Pitfalls result is relevant for a different reason: agent system prompts carry several instructions, including safety and confidentiality constraints, and eviction bias can remove one of them selectively (Chen et al., 2025). AnchorKV's authors state, citing earlier work, that existing compression policies either fail against jailbreak attacks or weaken safety alignment under aggressive eviction, and propose biasing retention toward safety-relevant key directions (Ni and Lao, 2026). A safety instruction is the clearest case where an aggregate task score is the wrong statistic: the evaluation that matters is the rate at which the constraint holds, not the mean helpfulness of the response.

On budget accounting, the gap between a fixed budget and the retained fraction of a given trajectory grows with the spread of trajectory lengths, and an agent whose number of turns is not fixed in advance has no natural bound on that spread. None of the agent papers located reports the distribution of retained fractions across trajectories.

Query visibility, which the author's earlier analysis treats in detail (Mahendrakar, 2026), compounds all three. In a matched-budget audit, Luo et al. (2026) found that when the question is not visible at compression time, only one of five audited methods sharing an attention backend consistently beat a best-of-three trivial baseline, and SnapKV lost on average to keeping the start and the most recent window. Agent caches are written before the later queries exist, so the query-aware results most methods report are the wrong reference for them.

What the record establishes for agents is therefore specific. Compression designed for single requests degrades on later turns and on exact outputs; the

methods designed for agents report recovering much of the loss, scored by final task success. No located study reports, for an agent workload, the per-turn retained fraction, an exact-match statistic per tool call, and a noise floor from repeated full-cache runs together.

8. A reporting procedure

The three indices become useful only if they change what a reported number is allowed to mean. Algorithm 1 states the procedure the author proposes for stating a losslessness claim so that it can be refuted. Each step asks for a quantity that some published study already reports and most do not.

Lines 1 to 5 address budget accounting. They replace a single ratio with the distribution of what samples experienced, and they record exposure directly rather than inferring it from a benchmark name. Under a fixed budget, the share of samples with E_i of zero - samples never compressed - is the first number a reader needs, and no located fixed-budget evaluation reports it; Ai et al. (2026) come closest, with median and 90th-percentile trace lengths.

Lines 6 to 11 address the statistic and the reference. The noise floor is computed from the same perturbations a deployment introduces, following the sources of variation that Yuan et al. (2025) measured. The statistics are listed from coarse to strict, so a method can be within noise on aggregate accuracy and lossy on exact tool calls, and the claim says so.

Lines 12 to 13 address exposure. Binning by E_i turns the passkey experiment of Yuan et al. (2024) from a footnote into a curve, and makes the claim's validity range explicit: lossless up to T_{safe} tokens of exposure, not lossless.

Line 14 is the null for length that Gao et al. (2025) supplied in their own table: a length change counts against compression only to the extent that it exceeds, or is asymmetric relative to, the change between two full-cache runs.

Applied to the results in this paper, the procedure gives different verdicts. The passkey results of Yuan et al. (2024) supply two exposure bins for one statistic, and show H2O lossy in the second; the statistic is exact, so no noise floor is needed for a 7-digit key. R-KV's results (Cai et al., 2025) satisfy line 6 in spirit, since $\text{pass}@1$ over 64 samples absorbs some of the reference noise, but report neither f_i nor E_i , and score one statistic. Ai et al. (2026) report three statistics and condition on matched answers, which is a per-sample comparison, but report their headline results at a 256-token budget, with a replay ablation from 512 to 8,192 tokens. VeriCache (Yao et al., 2026) reports an exact statistic and a divergence floor attributed to hardware, and is the only located study that does both.

9. What would settle the open part

The open part is narrower than the critique and more specific than the defence. Five measurements would settle most of it, and none requires a new compression method.

First, exposure curves for the decode-aware methods. R-KV, RLKV, LongFlow and Expected Attention each report accuracy at one or a few budgets on mathematics. Rescoring their existing generations by bins of tokens decoded after compression first acted would show whether losslessness holds across the trace-length distribution or only on average, and would locate the threshold budget that Section 6.2 shows is the crux of the R-KV conflict.

Second, the evidential statistic on decode-aware methods in deployment mode. Ai et al. (2026) measured the answer-evidence gap mainly by replay, at a diagnostic budget. Their replay ablation from 512 to 8,192 tokens on Qwen3-8B found that on AIME chain consistency "remains degraded at all budgets", while accuracy is "the first metric to recover"; it plots aggregate curves, not each method at its own lossless budget. The same three statistics, measured end to end at each method's own reported lossless budget, would say whether the methods that match full-cache accuracy also preserve the reasoning behind it. If they do, the strongest form of the critique fails for reasoning. If they do not, the accuracy results overstate what was preserved.

Third, a noise floor in every losslessness table. Two full-cache runs under a changed batch size or device count cost one extra evaluation. For greedy bfloat16 decoding of DeepSeek-R1-Distill-Qwen-7B across 12 configurations, Yuan et al. (2025) report a standard deviation of accuracy of 9.15 percent on AIME'24 but 1.04 percent on MATH500. The author's inference is that on the hardest long-output benchmarks this variation can exceed the gaps compression papers report, while on MATH500 it is comparable to the one- to two-point gaps R-KV calls lossless; either way, a reported gap cannot be read without that floor.

Fourth, per-instruction and exact-match scoring on agent workloads with fixed budgets. The Pitfalls design applied to agent system prompts, combined with VeriCache's function-call exactness, would test the agent claim directly: does a method that holds task success keep every instruction and every argument?

Fifth, the per-sample retained fraction under fixed budgets. Reporting the share of samples never compressed, and accuracy separately for samples above and below the budget, would show how much of a fixed-budget average comes from samples that did not experience compression.

10. Objections

"The decode-aware results already settle the question in compression's favour." They settle part of it. R-KV, RLKV and Expected Attention show that long decoding does not by itself destroy accuracy, and the

Algorithm 1. Stating a KV cache losslessness claim so that it can be refuted.

```

Input: method M, budget rule B, workload W, statistics S1..Sk, output cap Tmax
Output: one verdict per statistic, each with its exposure range and budget stated
1  for each sample i in W, under the full cache F and under M with B:
2    record prompt length Pi and output length Li
3    record retained fraction fi at the end of generation
4    record Ei, the number of tokens decoded after compression first acted
5  report the distributions of fi and Ei, not only their means
6  run F twice, changing batch size, device count or seed: Fa, Fb
7  for each statistic Sj (aggregate, per sample, per instruction, exact, evidential):
8    noisej <- distance between Sj(Fa) and Sj(Fb)
9    gapj <- distance between Sj(Fa) and Sj(M with B)
10   if gapj <= noisej: verdictj <- "within reference noise"
11   else: verdictj <- "lossy by gapj"
12 split samples into bins of Ei; repeat lines 7-11 within each bin
13 Tsafej <- largest E bin whose verdictj is "within reference noise"
14 report length ratio Li(M) / Li(Fa) beside Li(Fb) / Li(Fa)
15 claim: "M under B preserves Sj on W for exposure up to Tsafej", for each j

```

hyper-scaling result shows compression can buy accuracy at matched memory traffic. The paper concedes this and uses it to reject the strongest form of the critique. What the results do not settle is the statistic: they are scored by final-answer accuracy on mathematics, and the one study that measured the reasoning behind compressed correct answers found it degraded under most eviction methods, R-KV included, at its diagnostic budget. That study's headline budget was below the decode-aware methods' operating points, but its replay ablation from 512 to 8,192 tokens found AIME chain consistency "remains degraded at all budgets" while accuracy recovered first (Ai et al., 2026). The objection has not been tested per method at each method's own lossless budget, and the one budget ablation located points the other way.

"The passkey result is one footnote on one model." It is. It is cited because it is the only located result that holds model, haystack, method and task fixed and changes only the number of answer tokens decoded after compression. Its generality is not established. The paper treats it as the cleanest demonstration that exposure can matter, not as a measurement of how much it matters in general, and Section 9 asks for the curves that would.

"Divergence from the full cache is exactly what users care about, because they were promised the full model." For exact outputs, yes, and Section 5.5 says so. For open-ended reasoning, the full model is itself a distribution whose single greedy sample varies with batch size and hardware, so "the full model's output" is not a fixed promise even without compression. The paper's position is that both targets are legitimate and that a claim should say which one it meets.

"Budget accounting is bookkeeping, not science." The R-KV conflict shows otherwise. Two papers that appear to disagree about whether a method works turn out to have tested non-overlapping budgets, and the reader can only see this by converting one paper's ratios back into the other's tokens. When the conversion changes the conclusion, it is part of the result.

"A re-reading of published tables cannot decide an empirical dispute." It can decide which dispute is being had. The contribution here is the partition of the claim,

the reading of three conflicts that the partition dissolves, and a procedure whose every step at least one cited study has already carried out. Where the partition leaves a question open, Section 9 names the measurement, and those measurements are cheap relative to the methods papers that would report them.

11. What is not known

The published record does not establish how task quality varies with exposure for any decode-aware method; only aggregate accuracy at a few budgets is reported. It does not report, for any fixed-budget evaluation located here, the share of samples that were never compressed. It does not contain a losslessness table with a noise floor from repeated full-cache runs, outside VeriCache's divergence measurement. It does not show whether decode-aware methods at their own lossless budgets preserve the reasoning behind correct answers. It does not contain a per-instruction evaluation of compression on agent system prompts, nor an evaluation of compressed agents that scores each tool call exactly against repeated full-cache runs. Whether the R-KV results of Cai et al. (2025) and of Liu et al. (2025a) differ only by budget, or also by implementation, is not known.

The paper's own scope has limits. It covers token eviction, KV quantisation and the decode-time sparse-selection methods that the cited benchmarks include; it treats trained compression such as DMS only as a counterweight, and does not survey architectural reductions of the cache such as cross-layer or latent attention. Its literature sweep located the studies cited; a study that measures an exposure curve or a noise floor and was not found would change Sections 9 and 11.

12. Conclusion

Near-lossless KV cache compression is not a property of a method. It is a property of a method at a stated exposure, under a stated statistic, with a stated budget rule, and most published claims state the method and the budget and leave the rest to the benchmark's name. The critique that prefill-heavy, short-output evaluation produced the near-losslessness result is right about the omission and supported by the cleanest evidence: the

same eviction method falls from 100 to 35 percent when the answer it must produce grows from 7 to 64 digits, and the same outputs score above 75 percent on partial credit and below 10 percent on exact function calls. The critique is wrong that long decoding defeats compression as such, since methods built for it are near-lossless on reasoning traces by the statistic they report, and wrong to treat divergence from the full cache as the defect, since the full cache diverges from itself. The disagreements that look like disputes about whether compression works turn, on inspection, into disputes about which index was held fixed. The measurements that would close them are evaluations of existing methods, not new methods.

References

- Agrawal, N., & Mayer, R. (2026). Benchmarking KV-Cache Optimizations across Task Quality and System Performance for Long-Context Serving. arXiv:2607.05399. doi:10.48550/arXiv.2607.05399. [PREPRINT]
- Ai, M., He, J., & Guo, Y. (2026). Does Accuracy Equal Evidence? Reasoning Faithfulness under KV Cache Compression. arXiv:2608.01631. doi:10.48550/arXiv.2608.01631. [PREPRINT]
- Ananthanarayanan, S., Sengupta, A., & Chakraborty, T. (2026). Understanding the Physics of Key-Value Cache Compression for LLMs through Attention Dynamics. arXiv:2603.01426. doi:10.48550/arXiv.2603.01426. [PREPRINT]
- Bai, Y., Lv, X., Zhang, J., Lyu, H., Tang, J., Huang, Z., et al. (2024). LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. ACL 2024. arXiv:2308.14508. doi:10.48550/arXiv.2308.14508.
- Cai, Z., Zhang, Y., Gao, B., Liu, Y., Liu, T., Lu, K., et al. (2024). PyramidKV: Dynamic KV Cache Compression based on Pyramidal Information Funneling. arXiv:2406.02069. doi:10.48550/arXiv.2406.02069. [PREPRINT]
- Cai, Z., Xiao, W., Sun, H., Luo, C., Zhang, Y., Wan, K., et al. (2025). R-KV: Redundancy-aware KV Cache Compression for Reasoning Models. arXiv:2505.24133. doi:10.48550/arXiv.2505.24133. [PREPRINT]
- Chen, A., Geh, R., Grover, A., Van den Broeck, G., & Israel, D. (2025). The Pitfalls of KV Cache Compression. ACL 2026. arXiv:2510.00231. doi:10.48550/arXiv.2510.00231.
- Chen, Q., Xiong, J., Zhao, C., Yang, S., & Wong, N. (2026). CodeComp: Structural KV Cache Compression for Agentic Coding. arXiv:2604.10235. doi:10.48550/arXiv.2604.10235. [PREPRINT]
- Devoto, A., Jeblick, M., & Jegou, S. (2025). Expected Attention: KV Cache Compression by Estimating Attention from Future Queries Distribution. arXiv:2510.00636. doi:10.48550/arXiv.2510.00636. [PREPRINT]
- Du, W., Jiang, L., Tao, K., Liu, X., & Wang, H. (2025). Which Heads Matter for Reasoning? RL-Guided KV Cache Compression. arXiv:2510.08525. doi:10.48550/arXiv.2510.08525. [PREPRINT]
- Feng, B., Liu, J., Li, Y., Yu, W., Qiu, Z., Sun, Y., et al. (2026). StepKV: Step-Aware KV Cache Compression for LLM Agents. arXiv:2609.22158. doi:10.48550/arXiv.2609.22158. [PREPRINT]
- Feng, Y., Lv, J., Cao, Y., Xie, X., & Zhou, S. K. (2024). Ada-KV: Optimizing KV Cache Eviction by Adaptive Budget Allocation for Efficient LLM Inference. NeurIPS 2025. arXiv:2407.11550. doi:10.48550/arXiv.2407.11550.
- Feng, Y., Guo, H., Lv, J., Zhou, S. K., & Xie, X. (2025). Taming the Fragility of KV Cache Eviction in LLM Inference. arXiv:2510.13334. doi:10.48550/arXiv.2510.13334. [PREPRINT]
- Gao, W., Zhou, X., Sun, P., Zhang, T., & Wen, Y. (2025). Rethinking Key-Value Cache Compression Techniques for Large Language Model Serving. MLSys 2025. arXiv:2503.24000. doi:10.48550/arXiv.2503.24000.
- Ge, S., Zhang, Y., Liu, L., Zhang, M., Han, J., & Gao, J. (2023). Model Tells You What to Discard: Adaptive KV Cache Compression for LLMs. ICLR 2024. arXiv:2310.01801. doi:10.48550/arXiv.2310.01801.
- Ghadia, R., Kumar, A., Jain, G., Nair, P., & Das, P. (2025). Dialogue Without Limits: Constant-Sized KV Caches for Extended Responses in LLMs. ICML 2025. arXiv:2503.00979. doi:10.48550/arXiv.2503.00979.
- Hooper, C., Kim, S., Mohammadzadeh, H., Mahoney, M. W., Shao, Y. S., Keutzer, K., et al. (2024). KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization. NeurIPS 2024. arXiv:2401.18079. doi:10.48550/arXiv.2401.18079.
- Hsieh, C.-P., Sun, S., Krizan, S., Acharya, S., Rekesh, D., Jia, F., et al. (2024). RULER: What's the Real Context Size of Your Long-Context Language Models? COLM 2024. arXiv:2404.06654. doi:10.48550/arXiv.2404.06654.
- Huang, W., Zhang, J., & Zheng, X. (2026). CommitKV: Lifecycle-Aware KV Cache Compression via Commit Transitions for Multi-Turn Agents. arXiv:2608.07855. doi:10.48550/arXiv.2608.07855. [PREPRINT]
- Hwang, D. H., Jang, J., Na, J., Lim, H., & Kim, K.-E. (2026). Behavior-Preserving KV Cache Compression. EMNLP 2026. arXiv:2610.06479. doi:10.48550/arXiv.2610.06479.
- Kariyappa, S., & Suh, G. E. (2026). SideQuest: Model-Driven KV Cache Management for Long-Horizon Agentic Reasoning. arXiv:2602.22603. doi:10.48550/arXiv.2602.22603. [PREPRINT]
- Khodabandehlou, M., & Krishnamachari, B. (2026). Compression-Aware Abstention: Teaching LLMs to Refuse When KV-Compression Masks Remove Answer Evidence. arXiv:2608.29934. doi:10.48550/arXiv.2608.29934. [PREPRINT]
- Lancucki, A., Staniszewski, K., Nawrot, P., & Ponti, E. M. (2025). Inference-Time Hyper-Scaling with KV Cache Compression. NeurIPS 2025. arXiv:2506.05345. doi:10.48550/arXiv.2506.05345.
- Li, Y., Huang, Y., Yang, B., Venkitesh, B., Locatelli, A., Ye, H., et al. (2024a). SnapKV: LLM Knows What You are Looking for Before Generation. arXiv:2404.14469. doi:10.48550/arXiv.2404.14469. [PREPRINT]
- Li, Y., Jiang, H., Wu, Q., Luo, X., Ahn, S., Zhang, C., et al. (2024b). SCBench: A KV Cache-Centric Analysis of Long-Context Methods. ICLR 2025. arXiv:2412.10319. doi:10.48550/arXiv.2412.10319.
- Liu, M., Palnitkar, A., Rabbani, T., Jae, H., Sang, K. R., Yao, D., et al. (2025a). Hold Onto That Thought: Assessing KV Cache Compression On Reasoning. arXiv:2512.12008. doi:10.48550/arXiv.2512.12008. [PREPRINT]
- Liu, T., Li, S., Yang, J., Zhao, T., Zhou, F., Song, X., et al. (2025b). PM-KVQ: Progressive Mixed-precision KV Cache Quantization for Long-CoT LLMs. ICLR 2026. arXiv:2505.18610. doi:10.48550/arXiv.2505.18610.
- Liu, X., Tang, Z., Chen, H., Dong, P., Li, Z., Zhou, X., et al. (2025c). Semantic Integrity Matters: Benchmarking and Preserving High-Density Reasoning in KV Cache Compression. ICML 2026. arXiv:2502.01941. doi:10.48550/arXiv.2502.01941.
- Liu, Z., Desai, A., Liao, F., Wang, W., Xie, V., Xu, Z., et al. (2023). Scissorhands: Exploiting the Persistence of Importance Hypothesis for LLM KV Cache Compression at Test Time. arXiv:2305.17118. doi:10.48550/arXiv.2305.17118. [PREPRINT]
- Liu, Z., Yuan, J., Jin, H., Zhong, S., Xu, Z., Braverman, V., et al. (2024). KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache. ICML 2024. arXiv:2402.02750. doi:10.48550/arXiv.2402.02750.
- Luo, D., Liang, C., & Xuan, J. (2026). How Query Visibility Changes KV-Cache Compression Rankings: A Matched-Budget Audit. arXiv:2607.11942. doi:10.48550/arXiv.2607.11942. [PREPRINT]
- Mahendrakar, P. (2026). Compressed Once, Read Many Times: Why Prompt-Compression Results Do Not Transfer to Agent Memory, What the Query-Agnostic Line Already Showed, and the Two Properties of a Memory Write No Protocol Yet Scores. Life of

- Research. doi:10.5281/zenodo.22969435
- Mekala, A., Atmakuru, A., Song, Y., Karpinska, M., & Iyyer, M. (2025). Does quantization affect models' performance on long-context tasks? EMNLP 2025. arXiv:2505.20276. doi:10.48550/arXiv.2505.20276.
- Muller, L. K., Bich, P., Boretti, C., Chang, H.-M., Zhuang, J., & Cavigelli, L. (2026). KVarN: Variance-Normalized KV-Cache Quantization Mitigates Error Accumulation in Reasoning Tasks. arXiv:2606.03458. doi:10.48550/arXiv.2606.03458. [PREPRINT]
- Nawrot, P., Li, R., Huang, R., Ruder, S., Marchisio, K., & Ponti, E. M. (2025). The Sparse Frontier: Sparse Attention Trade-offs in Transformer LLMs. arXiv:2504.17768. doi:10.48550/arXiv.2504.17768. [PREPRINT]
- Ni, N., & Lao, Y. (2026). AnchorKV: Safety-Aware KV Cache Compression via Soft Penalty with a Refusal Anchor. arXiv:2606.17872. doi:10.48550/arXiv.2606.17872. [PREPRINT]
- Oren, M., Hassid, M., Yarden, N., Adi, Y., & Schwartz, R. (2024). Transformers are Multi-State RNNs. arXiv:2401.06104. doi:10.48550/arXiv.2401.06104. [PREPRINT]
- Ranjan, U. (2026). WakeKV: Reactive, Reversible KV Residency for Heads That Change Their Minds. arXiv:2610.02713. doi:10.48550/arXiv.2610.02713. [PREPRINT]
- Sadhukhan, R., Chen, J., Chen, Z., Tiwari, V., Lai, R., Shi, J., et al. (2024). MagicDec: Breaking the Latency-Throughput Tradeoff for Long Context Generation with Speculative Decoding. arXiv:2408.11049. doi:10.48550/arXiv.2408.11049. [PREPRINT]
- Su, Y., Tian, Z., Qiao, D., Zhou, Y., Li, J., & Zhang, M. (2026). LongFlow: Efficient KV Cache Compression for Reasoning Models. arXiv:2603.11504. doi:10.48550/arXiv.2603.11504. [PREPRINT]
- Sun, H., Chen, Z., Yang, X., Tian, Y., & Chen, B. (2024a). TriForce: Lossless Acceleration of Long Sequence Generation with Hierarchical Speculative Decoding. COLM 2024. arXiv:2404.11912. doi:10.48550/arXiv.2404.11912.
- Sun, H., Chang, L.-W., Bao, W., Zheng, S., Zheng, N., Liu, X., et al. (2024b). ShadowKV: KV Cache in Shadows for High-Throughput Long-Context LLM Inference. arXiv:2410.21465. doi:10.48550/arXiv.2410.21465. [PREPRINT]
- Sun, L., Harn, P., He, P., & Qin, X. (2026). Minimal-Intervention KV Retention via Set-Conditioned Diversity. arXiv:2605.14292. doi:10.48550/arXiv.2605.14292. [PREPRINT]
- Tang, J., Zhao, Y., Zhu, K., Xiao, G., Kasikci, B., & Han, S. (2024). Quest: Query-Aware Sparsity for Efficient Long-Context LLM Inference. ICML 2024. arXiv:2406.10774. doi:10.48550/arXiv.2406.10774.
- Wang, Z., Tang, C., Gong, L., Wang, C., Lou, W., Wang, T., et al. (2026). ActKV: Efficient LLM Agents through Action-Guided KV Cache Management. arXiv:2609.31395. doi:10.48550/arXiv.2609.31395. [PREPRINT]
- Wu, J., Wang, Z., Zhang, L., Lai, Y., He, Y., & Zhou, D. (2024a). SCOPE: Optimizing Key-Value Cache Compression in Long-context Generation. ACL 2025. arXiv:2412.13649. doi:10.48550/arXiv.2412.13649.
- Wu, Y., Hee, M. S., Hu, Z., & Lee, R. K.-W. (2024b). LongGenBench: Benchmarking Long-Form Generation in Long Context LLMs. ICLR 2025. arXiv:2409.02076. doi:10.48550/arXiv.2409.02076.
- Xiao, G., Tian, Y., Chen, B., Han, S., & Lewis, M. (2023). Efficient Streaming Language Models with Attention Sinks. ICLR 2024. arXiv:2309.17453. doi:10.48550/arXiv.2309.17453.
- Xiao, G., Tang, J., Zuo, J., Guo, J., Yang, S., Tang, H., et al. (2024). DuoAttention: Efficient Long-Context LLM Inference with Retrieval and Streaming Heads. arXiv:2410.10819. doi:10.48550/arXiv.2410.10819. [PREPRINT]
- Yao, J., Shen, S., Du, K., Feng, S., Seo, D., Zhang, R., et al. (2026). VeriCache: Turning Lossy KV Cache into Lossless LLM Inference. arXiv:2605.17613. doi:10.48550/arXiv.2605.17613. [PREPRINT]
- Ye, X., Yin, F., He, Y., Zhang, J., Yen, H., Gao, T., et al. (2025). LongProc: Benchmarking Long-Context Language Models on Long Procedural Generation. COLM 2025. arXiv:2501.05414. doi:10.48550/arXiv.2501.05414.
- Yuan, J., Liu, H., Zhong, S., Chuang, Y.-N., Li, S., Wang, G., et al. (2024). KV Cache Compression, But What Must We Give in Return? A Comprehensive Benchmark of Long Context Capable Approaches. arXiv:2407.01527. doi:10.48550/arXiv.2407.01527. [PREPRINT]
- Yuan, J., Li, H., Ding, X., Xie, W., Li, Y.-J., Zhao, W., et al. (2025). Understanding and Mitigating Numerical Sources of Nondeterminism in LLM Inference. arXiv:2506.09501. doi:10.48550/arXiv.2506.09501. [PREPRINT]
- Zhang, Z., Sheng, Y., Zhou, T., Chen, T., Zheng, L., Cai, R., et al. (2023). H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. NeurIPS 2023. arXiv:2306.14048. doi:10.48550/arXiv.2306.14048.