COMPACT-BENCH: ONE BUDGET AXIS FOR MEMORY COMPACTION IN LLMS AND AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Methods that reduce the memory a language model retains about its context are developed in largely separate literatures, and each reports its savings in a different unit: KV-cache compression in memory saved at fixed perplexity, prompt compression in tokens removed at fixed accuracy, sub-quadratic architectures in recall at a fixed state size, and agent-memory systems in end-task scores without the size of the store. These methods consume the same resource, yet their results cannot be compared. We introduce COMPACT-Bench, which expresses every method in a single unit, the bytes of retained state per token of history (BPT), and evaluates four properties at matched budgets: the accuracy a method retains, the value of exact over lossy retention, whether the system can report what it discarded, and whether its stated confidence reflects what its memory still supports. Each self-report is validated against uncompacted controls in which the relevant fact is present or absent. We evaluate KV eviction, KV quantization, prompt compression, and model-written summaries on 21 settings of 16 open models (0.5B–24B parameters) and report four findings. First, once every stored byte is counted, mechanisms differ substantially at equal cost, and their ordering depends on the model: on Qwen3-1.7B a 2-bit KV cache retains 0.98 accuracy at 22% of the full cache, whereas no other mechanism exceeds 0.56 below 30%, while on Qwen2.5-1.5B the same quantizer fails below 8 bits. Second, tolerance to eviction tracks release date far more closely than scale. In five same-family pairs of matched size, the model released from November 2024 onwards reaches half its accuracy at a budget 17–33 points smaller, with every 95% interval excluding zero, and across all settings the later models collapse at 33–69% of their budget against 76–86% for the earlier ones. Third, attention-based scorers fall below random eviction because the standard protocol compresses before the question arrives; when the question is visible at compression time, SnapKV and TOVA gain a median of 0.52 in accuracy. Fourth, eviction that removes a fact in fragments misleads the model's self-report. On the six models whose self-report passes both controls, the model claims to retain the fact in 54% of SnapKV failures against 3% for StreamingLLM, which removes contiguous spans, and the misreported failures are those in which most of the fact's key tokens survived. We release the benchmark and the per-generation records behind every reported number at `https://anonymous.4open.science/w/compact-bench-6A71/index.html`.

1 INTRODUCTION

The memory that holds a language model's context has become a dominant cost of inference. The KV cache grows linearly with the number of tokens processed and occupies most of the accelerator's memory at long context (Kwon et al., 2023); the interaction history of an agent quickly exceeds any context window, and a persistent assistant accumulates state across sessions. A large body of work reduces this memory at every level of the stack, by evicting or quantizing cache entries (Zhang et al., 2023; Li et al., 2024; Liu et al., 2024), by pruning or compressing prompts (Jiang et al., 2023; Mu et al., 2024), by bounding the state architecturally (Gu & Dao, 2024; Bulatov et al., 2022), and by summarizing or consolidating agent memory (Packer et al., 2024; Chhikara et al., 2025).

Results from these literatures are difficult to compare. Each holds a different quantity fixed and reports a different ratio, so a practitioner choosing among a 4-bit cache, an evictor that retains one

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

token in eight, and a rolling summarizer has no common measure, although all three consume the same resource, bytes of retained state. Standard evaluation is also narrow. It measures whether single-turn accuracy survives compression, but not whether the system can identify what it has lost, whether its stated confidence reflects the reduced memory, or how accuracy evolves under repeated compaction, as in long-running agents.

COMPACT-Bench addresses both gaps. Its central quantity is *bytes per token of history* (BPT), the number of bytes of state a method retains divided by the number of context tokens it has processed. Because retained state is ultimately read through a KV cache, eviction, quantization, low-rank projection, prompt compression, and summarization can all be expressed on this axis (Section 3). The benchmark evaluates four task families with fixed metrics: the *frontier* of accuracy against budget; *reversibility*, the difference between exact and lossy retention at equal budget; *attribution*, whether the system can report what it discarded; and the *calibration* of its stated confidence after compaction.

Our contributions are as follows.

- **A common budget axis and benchmark.** An accounting of retained state that is independent of model family and includes quantization metadata, a four-task protocol with uncompacted controls for every self-report, and an open reference implementation (Section 3).
- **A rate–distortion formulation of compaction.** We cast compaction as a constrained rate–distortion problem and derive a Fano-type lower bound on error (Section 2). The formulation defines the quantities the benchmark varies, namely the rate, query conditioning, and reversibility, and supplies the information ceiling used in the reversibility analysis. At the budgets studied the bound does not bind, so the measured failures reflect which information a method retains and not how much.
- **A cross-literature comparison** (Section 4.1). At matched bytes, eviction, KV quantization, their combination, prompt compression, and summarization differ by up to 0.83 in accuracy, and the best mechanism changes between two model generations of the same family.
- **Tolerance to eviction tracks release date more than scale** (Sections 4.2 and 4.3). Across five size-matched pairs and 21 settings, models released from November 2024 onwards tolerate substantially more eviction at every size we tested, and the deficit of attention-based scorers against random eviction is attributable to the question-agnostic protocol.
- **Self-report under compaction** (Section 4.5). By recording which tokens of a fact survive eviction, we show that fragmentary retention causes the model to report a lost fact as present and to state high confidence in wrong answers, whereas eviction of contiguous spans is reported correctly.

2 Problem formulation

We model a compaction method as the composition of two operators. A *compaction operator* $\mathcal{C}_\theta : H \mapsto Z$ maps the history H, the information on which the model could otherwise condition, to a representation Z with $\mathrm{rate}(Z) \le B$, and a *usage operator* $\mathcal{U}(Z, Q) \to \hat{Y}$ answers a query from that representation alone. The rate is measured in cache bytes for a KV evictor, in tokens for a prompt compressor, and in state dimensions for an architectural memory; Section 3 converts all three to a common unit. Selecting $\mathcal{C}_\theta$ is then a rate–distortion problem,

$$\min_{\theta} \ \mathbb{E}_{(H,Q,Y)}\big[\ell(\mathcal{U}(\mathcal{C}_\theta(H), Q),\, Y)\big] \quad \text{s.t.} \quad \mathrm{rate}(Z) \le B. \tag{1}$$

In information-bottleneck form, Eq. 1 asks $\mathcal{C}_\theta$ to maximize $\mathrm{I}(Z; Y \mid Q)$ subject to $\mathrm{I}(Z; H) \le B$ (Tishby et al., 2000; Wang et al., 2025), so that the representation retains the parts of H that bear on the answer and nothing else. Because $\hat{Y}$ depends on H only through Z, the error of any method is bounded below once the budget is small relative to the information the answer requires (Proposition 1).

Proposition 1 (Error floor)**.** *Let $I^\star(Q) = \mathrm{I}(Y; H \mid Q)$ and let $\mathcal{Y}$ be finite. For any scheme with $\hat{Y} = \mathcal{U}(Z, Q)$,*

$$P_e \ \ge \ \frac{H(Y \mid Q) - B - 1}{\log |\mathcal{Y}|}, \qquad \textit{whenever } B < I^\star(Q). \tag{2}$$

Appendix A derives Eq. 2 from the data-processing and Fano inequalities. As a converse, the bound holds for every θ, so once $B < I^\star(Q)$ no choice of scorer, architecture, or retrieval scheme avoids it. It implies that the budget required for a given accuracy grows with $I^\star(Q)$, so queries that require exact values or several facts are the hardest to compress. An operator that observes the query needs to retain only the information that query requires, whereas one that does not must cover every admissible query, and the difference between the two rates is at most $H(Q)$. The fixed-state limit of attention-free architectures (Wen et al., 2024) and the $\Theta(nd)$ cost of exact attention (Haris & Onak, 2025) are special cases of the same inequality. The importance scores used in practice, including attention mass (Zhang et al., 2023; Li et al., 2024), self-information (Jiang et al., 2023), explicit mutual-information estimates (Wang et al., 2025), output-error bounds (Feng et al., 2025), and an LLM's own judgment of salience (Chhikara et al., 2025; Park et al., 2023), are estimates of the distortion $\mathcal{D}(\theta)$, which cannot be computed directly because it depends on the unknown answer Y.

Two properties of $\mathcal{C}_\theta$ determine where a method operates relative to this floor. **(P-rev) Reversibility** is whether the information needed for Q can be recovered exactly from the retained state. A quantized or merged entry remains addressable in Z, but its original value cannot be reconstructed, so it is irreversible at a finer granularity than an evicted entry. **(P-q) Query conditioning** is whether $\mathcal{C}_\theta$ observes Q before deciding what to discard, or commits to a compaction in advance.

Together with Eq. 2, these properties yield four testable predictions, one for each layer of the stack. (1) KV eviction: needle recall collapses once the per-token budget falls below the needle's share of attention mass, at a point governed by the entropy of the answer's location. (2) Prompt compression: a query-aware method outperforms a query-agnostic one by at most the mutual-information gap between them. (3) Fixed-state architectures: an s-bit state cannot answer a query that requires more than s bits, so state-space and linear-attention models exhibit a sharp recall cliff where standard attention degrades gradually. (4) Agent memory: under repeated irreversible summarization, the error on a fact accumulates with the number of compaction events it has been exposed to, whereas a reversible, retrieval-backed memory does not degrade with the number of events. Section 4 measures the recall curves addressed by (1), a KV-cache analogue of the query-conditioning gap in (2), and the equal-budget mechanism underlying (4); prediction (3) and the multi-round form of (4) are left to future work. The attribution and calibration tasks test whether a system's report of what it discarded agrees with what it retained.

The bound is far from binding at practical budgets. When the 1.5B reference model has lost half its accuracy, it still retains about 48 MB of cache for a 2,000-token context (3.9×10^8 bits), whereas the answer, a six-digit number, carries at most 20 bits. The failures reported below therefore lie many orders of magnitude above Eq. 2 and reflect which information a method retains, not how much. The formulation serves two purposes in what follows. It determines which comparisons isolate a single property of $\mathcal{C}_\theta$, and its counting argument supplies the information ceiling against which Section 4.4 evaluates summaries. Because $I^\star(Q)$ cannot be measured directly, we do not evaluate Eq. 2 numerically.

3 COMPACT-BENCH

3.1 THE BPT AXIS

Each literature reports its savings in the unit its own methods produce: a KV evictor reports the fraction of tokens retained, a quantizer the bits per entry, a prompt compressor a token ratio, and an agent-memory system the size of its store. These units cannot be converted into one another without the model's dimensions. COMPACT-Bench replaces them with a single unit, the bytes of retained state per token of original history (BPT), computed from the model's architecture and not from the ratio a paper reports.

Given the key-value dimensions, the conversion is direct. A transformer with L layers, H key-value heads, and head dimension d stores $2LHd$ scalars of KV cache per token; multiplied by the bytes per scalar, this is the full-cache BPT, of which every method is expressed as a fraction (Table 1). All stored state is priced, so a quantized cache is charged for its per-group scales and zero points and for any entries retained at full precision. A 4-bit cache with groups of 32 and 1% fp16 outliers therefore costs 34% of the full cache, not 25%, which is the cost of a 66% eviction (Appendix B.3).

Table 1: BPT accounting: every compaction mechanism reduces to bytes retained per token of original history. b_{meta} is the per-scalar cost of a quantizer's scales, zero points, and full-precision outliers, 1.48 bits for ours.

Mechanism	BPT cost
Full cache (uncompacted)	$2LHd\cdot$ bytes/scalar
Token eviction, keep fraction f	$f\times$ full-cache BPT
Quantization to b bits	$\big(b + b_{\text{meta}}\big)/16\times$ full-cache BPT
Prompt or summary compression, m of n tokens kept	$(m/n)\times$ full-cache BPT
Soft/gist compression, g vectors of dimension d_g	$(g\, d_g \cdot \text{bytes/scalar})/n$

BPT measures memory and deliberately excludes compute; the latency of a scoring pass or a summarization call is not part of the budget.

3.2 Task families and metrics

A budget axis measures how much memory a method uses, not how well it uses it. Single-turn accuracy, the metric of almost every long-context benchmark, conflates several distinct properties, which COMPACT-Bench separates into four task families.

Frontier. A single fact is inserted at a controlled depth in natural filler text, and the model is asked a question that can be answered only from that fact. BPT is swept from the full-cache value through the operating range of every method under test, and accuracy is reported as a curve over budget, so that no operating point is selected after the results are known. The uncompacted model answers every question, so every error on this task is attributable to compaction.

Reversibility. Two operators compact only on overflow and are limited to the same token budget. The first retains raw chunks of history and evicts the oldest chunk when full, so everything it keeps is exact; the second summarizes its buffer to the budget at each overflow, covering more of the history but none of it verbatim. Because both use the same memory, any difference in accuracy between them is attributable to reversibility.

Attribution. From one compacted cache, the system answers a yes/no probe asking whether it still holds the fact and then the question itself, so that both read the same retained set. The overclaim rate is the fraction of wrong answers for which the probe answered yes. For eviction methods we also record, for each token of the fact's key and value, the fraction of layer and head slots that retain it. *Calibration* uses the confidence the system states with its answer in the same generation.

A self-report is informative only if it can report a missing fact when nothing has been compacted. Both tasks therefore include two uncompacted controls: a *present* control with the full context, and an *absent* control with the same filler and the fact removed, which no system can answer. An overclaim rate is interpreted only for models whose probe passes both controls, answering yes when the fact is present and no when it is absent. Confidence after compaction is compared with confidence on the absent control, which separates the effect of compaction from the error of the instrument.

Two variants test the design of the frontier task. In the *question-aware* variant the question is appended to the context before compression, so that only the information available to query-scored methods changes; in the *multikey* variant three distractor facts with the same template and different keys are inserted, so that a method must preserve the binding between key and value. The reversibility task has a *dense* variant with four times as many facts and one filler sentence per chunk.

The tasks rank methods differently. On Qwen3-4B, SnapKV answers more questions than StreamingLLM at half the budget (0.48 against 0.38) but claims to hold the fact on 74% of its failures, whereas StreamingLLM does so on none (Section 4.5). Unless stated otherwise, KV methods are evaluated in the question-agnostic setting that KVPress uses by default, in which the context is compressed once before the question arrives, as when a cache is reused across queries (Appendix B.4).

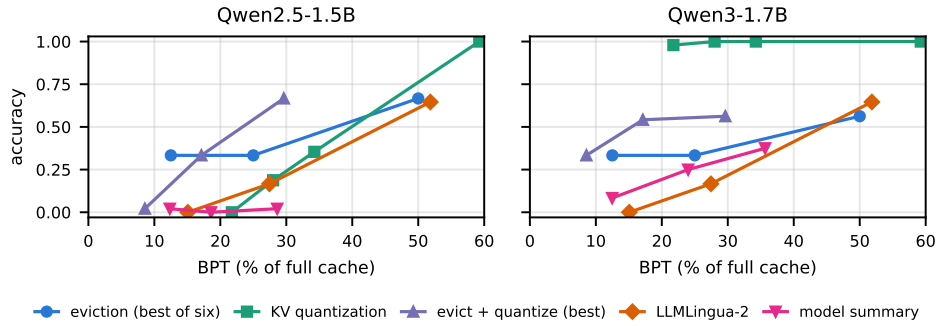

Figure 1: Needle accuracy against bytes of retained state per token of history, as a percentage of the full cache, for five compaction mechanisms answering the same 48 questions. Eviction and evict-then-quantize show the best of their methods at each budget. Quantization is priced with its metadata and outliers. On Qwen3-1.7B a 2-bit cache keeps 0.98 accuracy at 22% of the bytes, and on Qwen2.5-1.5B the same quantizer fails below 8 bits.

4 Experiments

The reference model is `Qwen2.5-1.5B-Instruct` on a single 8 GB consumer GPU. Section 4.1 compares five mechanisms on the BPT axis for the reference model and its successor, Qwen3-1.7B. Sections 4.2 and 4.3 study KV eviction on the reference model and on 21 settings of 16 models, and Sections 4.4 and 4.5 address reversibility and self-report. Unless stated otherwise, each cell contains 48 generations, and intervals are 95% intervals from a parametric bootstrap over cells (Appendix B.2); Appendix B lists every setting and prompt.

4.1 Five mechanisms on one axis

Figure 1 places eviction, KV quantization, eviction followed by quantization, LLMLingua-2 prompt compression (Pan et al., 2024), and a summary written by the model itself on a common axis. Every point is evaluated on the same 48 needle questions, and quantized caches are charged for their scales, zero points, and fp16 outliers (Appendix B.3). Mechanisms that their literatures report in incompatible units differ widely at equal cost. On Qwen3-1.7B, quantization retains 1.00 accuracy at 3 bits (28% of the full cache) and 0.98 at 2 bits (22%); at comparable cost, the best eviction method retains 0.33, the best combination of eviction and quantization 0.56, the model's summary 0.25, and LLMLingua-2 0.17.

The ordering of the two KV mechanisms reverses on the reference model. Its first layer holds keys with magnitudes near 300, and even with 1% of entries kept at full precision, accuracy falls to 0.35 at 4 bits and 0.00 at 2 bits. Its best configuration near 30% of the full cache combines the two literatures, retaining half the tokens with StreamingLLM and storing them in 8 bits (0.67 at 29.6%). On both models, 8-bit storage changes the accuracy of each of the four eviction methods with which we combined it by at most 0.04, so it reduces an evictor's memory by a further 41% without measurable loss. Summaries of this filler rarely preserve the needle; on the reference model they answer at most 2% of questions at any budget. None of these comparisons can be derived from published results, because "keep 25% of tokens" and "4 bits" denote the same budget only when quantization metadata is ignored, and on Qwen3-1.7B the best eviction method at a quarter of the tokens answers 0.33 where the 4-bit cache answers 1.00.

4.2 The eviction frontier and the effect of query conditioning

Figure 3 (Appendix B) places six KV methods (SnapKV, StreamingLLM, TOVA, Knorm, ExpectedAttn, and a random-eviction control) on the reference model's BPT axis over 1,395 generations, with the needle at five depths in contexts of 2,000 to 8,000 tokens. The uncompacted model answers every question. Averaged over the five non-random scorers, accuracy is 0.67 at 90% of the full-cache budget of 28,672 bytes per token, 0.43 at 75%, 0.20 at 50%, and zero at 10%. Three of

the four content-based scorers fall below random eviction at high budgets; at 90% of the budget, SnapKV, Knorm, and ExpectedAttn reach 0.53, 0.58, and 0.29, against 0.93 for random eviction. The positional StreamingLLM performs best at every budget and answers correctly exactly when the needle lies within its recent window, that is, at three of five depths at half the budget (0.60) and at one at a quarter (0.20). The frontier is not an artifact of short contexts. At 16,000 tokens (three depths, four trials, 12 generations per cell) the uncompacted reference model still answers every question, StreamingLLM again answers exactly the depths inside its window (0.67 at half the budget), and the collapse point of the scorer average moves from 84% to 64% of the budget, so a longer context tolerates proportionally more eviction.

The question-aware variant identifies the source of this deficit. SnapKV and TOVA score tokens by the attention they receive from the end of the context, which in the question-agnostic setting consists of filler. With the question appended before compression, SnapKV on the replication subset of the reference model (Appendix B) rises from 0.08 to 0.85 at half the budget and TOVA from 0.17 to 1.00; at a quarter of the budget they rise from 0.00 to 0.29 and from 0.04 to 0.69. Across the six models evaluated with this variant (Qwen2.5 at 0.5B, 1.5B, and 3B, and Qwen3 at 0.6B, 1.7B, and 4B), SnapKV and TOVA gain a median of 0.52 over the four largest budgets and improve in 90% of model–budget cells. Knorm, ExpectedAttn, and random eviction, which do not score against the question, change by a median of 0.02 and serve as a control; the single exception is Knorm on Qwen3-4B, whose accuracy is erratic even without the question (0.04 at 90% of the budget and 0.81 at 75%). For the attention-based scorers, the deficit in the question-agnostic setting therefore reflects the queries against which they score, not the scoring rule, and the resulting penalty of up to 0.98 in accuracy at equal budget is a direct measurement of the query-conditioning gap of Section 2.

4.3 Tolerance to eviction tracks release date more than scale

We evaluated the frontier on 21 settings of 16 models from five families (Appendix B, Table 5). The earlier group, released in August and September 2024, comprises Qwen2.5 at 0.5B, 1.5B, 3B, 7B, and 14B and Phi-3.5-mini; the later group, released between November 2024 and April 2025, comprises Qwen3 at 0.6B, 1.7B, 4B, 8B, and 14B, OLMo-2 at 1B and 13B, SmolLM2-1.7B, Phi-4, and Mistral-Small-24B. Nine settings use 4-bit NF4 weights, either because the model does not fit on one GPU at 16 bits or as a precision control; paired runs at both precisions move the collapse point by at most 4 points. Full-cache sizes range from 12,288 to 819,200 bytes per token.

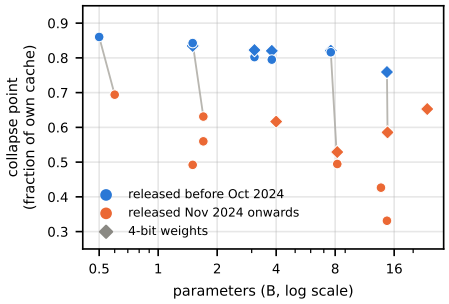

Figure 2: Budget fraction at which scorer-averaged accuracy halves, for 21 settings of 16 models. Grey lines join five same-family pairs. Every model released from November 2024 onwards (orange) tolerates more compaction than every earlier one (blue), at every size.

Every later model tolerates more compaction than every earlier model, at every size (Figure 2). The earlier settings reach half their baseline accuracy at 76–86% of their own budget and the later settings at 33–69%, and the two ranges remain disjoint in 98% of bootstrap samples. The five same-family pairs of matched size differ by 17 to 33 points (Qwen2.5-0.5B and Qwen3-0.6B, 86% and 69%; 1.5B and 1.7B, 84% and 63%; 3B and 4B, 82% and 62%; 7B and 8B, 82% and 49%; 14B, 76% and 59%), and the 95% interval of every paired difference excludes zero, the smallest lower bound being 12 points. A regression of the collapse point on release group and log parameter count, with one setting per model, attributes 25 points to release group and 2.4 points to each doubling of parameters, and a permutation test on the group labels gives $p \leq 10^{-4}$. Scale has a smaller effect within the earlier group, where Qwen2.5-14B tolerates 4 to 10 points more compaction than the smaller Qwen2.5 models, and no consistent effect within the later group, in which Phi-4 tolerates the most compaction (33%) and Qwen3-0.6B the least (69%). The separation persists with distractors: with three same-format needles for other keys, the three Qwen2.5 models collapse at 87% or later and their Qwen3 counterparts at 68–77%. Release date serves here as a proxy, and the data do not identify which change in training produces the effect (Section 6).

Table 2: Reversibility at equal token budget, compact-on-overflow (accuracy, summary / raw buffer). Standard documents hold 12 facts in about 1,600 tokens (96 questions per budget) and dense documents 48 facts in about 1,300 tokens (192 questions per budget).

Budget (% of history)	10%	25%	50%	75%	100%
Qwen2.5-1.5B, standard	0.30 / 0.07	0.47 / 0.25	0.32 / 0.45	0.40 / 0.69	1.00 / 0.94
Qwen3-1.7B, standard	0.78 / 0.07	0.98 / 0.25	1.00 / 0.50	1.00 / 0.75	1.00 / 1.00
Qwen2.5-1.5B, dense	0.11 / 0.08	0.12 / 0.22	0.27 / 0.47	0.49 / 0.71	0.90 / 0.96
Qwen3-1.7B, dense	0.19 / 0.08	0.40 / 0.22	0.70 / 0.47	0.90 / 0.72	0.93 / 0.97

The collapse point is a property of the model and not of the absolute size of its cache. Among the nine earlier settings up to 7.6B parameters, whose full caches span a 32× range, scorer-averaged accuracies at each budget fraction lie within 0.12 of one another, or within 0.16 when StreamingLLM, whose accuracy is determined by the needle's position, is excluded. In absolute bytes the same settings differ widely. At a quarter of its budget, Phi-3.5-mini retains 98,304 bytes per token, more than the full cache of any Qwen2.5 model up to 7B, and answers 17% of questions. OLMo-2-13B and Phi-3.5-mini collapse at nearly the same retained state (349,365 and 312,607 bytes per token) but at 43% and 80% of their budgets.

Method rankings also differ between the groups. StreamingLLM is the best method at half the budget on every Qwen2.5 model, and at least three of the four content-based scorers fall below random eviction at 90% of the budget on every earlier setting. In the later group, a content-based scorer is the best method at half the budget on nine of the ten models, by up to 0.65 over StreamingLLM (SnapKV answers 96% of questions on Phi-4 at a quarter of the budget, where random eviction answers 2%); the exception is SmolLM2, on which StreamingLLM leads narrowly (0.67 against 0.65). The best scorer varies across models, and Knorm retains full accuracy on Qwen3-8B down to half the budget but answers no question on OLMo-2-13B at any budget. We had hypothesized in advance that QK-norm, introduced in Qwen3, would account for the shift. The hypothesis is not supported, since OLMo-2 uses QK-norm and defeats Knorm, whereas Phi-4 uses none and shares its attention implementation with Phi-3.5-mini. The difference therefore points to training and not to this architectural change, and a method's ranking must be measured on the model with which it will be deployed.

4.4 Reversibility at equal budget

At a fixed token budget, we compare a rolling summary, which covers the whole history but cannot be inverted, with a bounded raw buffer, which covers only the most recent chunks but can be read back exactly. The history is a synthetic document of 24 chunks; at answer time, the raw buffer returns the retained chunks that mention the queried entity. Table 2 reports accuracy at five matched budgets.

The accuracy of the raw buffer tracks the fraction of the history it retains on every model, since it keeps each fact either exactly or not at all, so which arm leads is determined by the summary. On the reference model the two cross at 41% of the budget; below that point the summary is more accurate because it is the only representation that has observed the entire history. On standard documents, the summary leads at every budget below the full history on every 16-bit model of 3B or more that we evaluated and on Qwen3-1.7B, Qwen3-4B, and Qwen2.5-14B, answering 0.49–1.00 at a quarter of the budget against 0.25 for the raw buffer. These documents contain twelve facts in about 1,600 tokens of repeated filler, so a capable summarizer retains most of them even at a tenth of the budget. Dense documents remove this slack. With 48 facts in 1,307 tokens, the most compact encoding of every fact as an entity, attribute, and value occupies 42% of the history, so a summary at a quarter of the budget can hold at most about 59% of the facts and at a tenth about 24%. Qwen3-1.7B's summary reaches 0.40 and 0.19 at these budgets, and its advantage over the raw buffer at a quarter of the budget falls from 0.73 to 0.18. On the reference model the raw buffer now leads from a quarter of the budget upwards, and the crossover moves from 41% to 13%. Whether exact retention outperforms a summary at a given budget therefore depends on how much of the history the questions require, the quantity $I^\star(Q)$ of Section 2, as well as on the quality of the summarizer. The dense documents also reproduce the release-date effect: in the three same-size pairs evaluated

Table 3: Self-report after compaction on the six models whose probe passes both uncompacted controls, saying yes to the fact when it is present and never when it is absent (48 needles per cell). Overclaim is the share of wrong answers at compression ratio 0.5 on which the probe still claimed the fact. Confidence is the mean stated confidence on the absent control and on the wrong answers of each method. Qwen2.5-14B's run capped generation at 50 tokens and scored a completion without a number as 0.5, so its confidence is omitted.

	Probe yes		Overclaim (r=0.5)			Stated confidence		
Model	present	absent	StrLLM	SnapKV	Random	absent	StrLLM	SnapKV
Qwen2.5-1.5B	0.58	0.00	0.00	0.44	0.47	0.98	0.96	0.91
Qwen2.5-3B*	0.58	0.00	0.11	0.36	0.61	0.23	0.10	0.43
Qwen2.5-7B*	0.90	0.00	0.00	0.74	0.20	0.08	0.49	0.80
Qwen2.5-14B*	0.96	0.00	0.00	0.42	0.24	–	–	–
Qwen3-0.6B	1.00	0.00	0.06	0.61	0.00	0.50	0.90	0.96
Qwen3-4B*	1.00	0.00	0.00	0.74	0.53	0.51	0.78	0.85

on them (0.5B and 0.6B, 1.5B and 1.7B, 3B and 4B), the Qwen3 summary retains more facts than the Qwen2.5 summary at every budget below the full history.

4.5 What a compacted system reports about its losses

Before interpreting a self-report, we verify that it can report a missing fact without compaction. Of the eleven models audited, six pass both controls. The remaining five either claim the fact when it was never present (Qwen2.5-0.5B, Qwen3-1.7B, OLMo-2-1B, and SmolLM2, on 50–100% of absent trials) or seldom claim it when it is present (Phi-3.5-mini, 17%), so their overclaim rates are uninformative about compaction.

On the six models with a valid probe, whether the system can report a loss depends on how eviction removes the fact (Table 3). StreamingLLM removes a contiguous span, so the needle is either retained or removed in its entirety, and pooled over these models the probe claims the fact on 3% of its failures (95% interval 1–8%). SnapKV and random eviction distribute their losses across the context, and the probe claims the fact on 54% (48–60%) and 34% (28–40%) of their failures. The recorded survival of the needle's tokens accounts for the difference. Pooled over these models and both compression ratios, the SnapKV failures on which the probe answered yes retained 70% of the key's layer and head slots and 49% of the value's, against 24% and 16% for the failures it reported correctly; random eviction shows the same pattern (40% against 26%), and StreamingLLM's 408 correctly reported failures retained none of the key. The relation is graded. Over all 1,631 failures of the three scattering methods on these models, a logistic regression of the probe's answer on the share of the key's slots retained has slope 3.8 (standard error 0.25), so the probe claims the fact on 4% of the failures in which none of the key survived and on 66% of those in which all of it did. A partially retained fact is reported as present, and a scorer that retains the key while discarding digits of the value produces precisely this state. Because random eviction, which applies no scoring rule, also overclaims, the cause is the fragmentation of the fact and not the scoring rule.

Stated confidence shows the same dependence wherever the instrument is informative. On Qwen2.5-1.5B, confidence is 0.98 even when the fact was never present, so its confidence after compaction carries no information about compaction. Five audited models do state low confidence on the absent control (0.08–0.60), namely Qwen2.5-3B and 7B, Qwen3-0.6B and 4B, and Phi-3.5-mini. On each of them, confidence after a SnapKV failure is higher by 0.16–0.72 (Table 3 lists four; Phi-3.5-mini rises from 0.60 to 0.76), and confidence after a StreamingLLM failure is lower than after a SnapKV failure. Compaction therefore produces confident errors where it leaves a fragment of the fact, and a model that recognizes an absent fact does not recognize a damaged one.

5 Related work

Long-context and memory benchmarks. Long-context evaluation began with the depth-by-length heatmap of Needle-in-a-Haystack (Kamradt, 2023) and has since expanded to multi-hop synthetic

suites (Hsieh et al., 2024) and to broader, more realistic ones (Bai et al., 2024; Zhang et al., 2024; Yen et al., 2025). NoLiMa removes the lexical overlap between needle and query (Modarressi et al., 2025); our needles retain it for comparability with NIAH, and because the uncompacted models answer them without error, every measured error is attributable to compaction. LOCOMO and LongMemEval evaluate multi-session dialogue and memory at the agent level (Maharana et al., 2024; Wu et al., 2025). Two suites are closest to this work. SCBench studies KV-centric long-context methods across generation, compression, retrieval, and loading (Li et al., 2025), and KVPress provides a common harness for eviction and quantization (NVIDIA, 2024). Neither reports results on a common budget axis or tests whether a system can report what it discarded. COMPACT-Bench builds on both, reusing their infrastructure and adding the BPT axis, the reversibility, attribution, and calibration task families, and the uncompacted controls.

Compaction methods. KV-cache methods evict scored tokens (Zhang et al., 2023; Li et al., 2024; Xiao et al., 2024; Oren et al., 2024) or lower the bit-width of every entry (Liu et al., 2024; Hooper et al., 2025). Prompt compression deletes low-value tokens (Jiang et al., 2023; Pan et al., 2024) or maps the context to trained latent vectors (Mu et al., 2024). Architectural methods build a fixed-size memory into the model (Gu & Dao, 2024; Bulatov et al., 2022), and agent-memory systems page content in and out of an external store (Packer et al., 2024) or extract and update facts (Chhikara et al., 2025).

Calibration. Prior work on stated confidence concerns uncompressed models, which evaluate their own answers well when prompted in a suitable format (Kadavath et al., 2022) and often state verbal confidence that is better calibrated than token probabilities (Tian et al., 2023; Xiong et al., 2023). The effect of context compaction has not been examined. Our absent control shows that on several small models verbal confidence cannot signal a missing fact even without compaction, so calibration under compaction must be interpreted relative to that control.

6 Limitations

Our largest model has 24B parameters, and we evaluate only open-weight models, since the benchmark requires direct access to the KV cache. KVPress cannot compress a model sharded across GPUs, so the largest models use 4-bit weights, which in paired runs moves the collapse point by at most 4 points. The release-date effect rests on ten later and six earlier models from five families, and all five size-matched pairs come from one family; it holds at every size tested but does not identify the responsible training change, and our QK-norm hypothesis was not supported. The cross-mechanism comparison covers two models, the question-aware and multikey variants six settings, and the self-report findings the six models whose probe passes both controls. The tasks use synthetic English text with lexical needles and contexts of 2,000 to 8,000 tokens, extended to 16,000 for the reference model; they isolate compaction errors cleanly but do not measure multi-hop or naturalistic retrieval. The bootstrap resamples cells independently and ignores the correlation induced by sharing needles across budgets. Architectural memory (prediction 3) and multi-round agent memory (the full form of prediction 4) are not evaluated; testing (3) requires state-space models, which KVPress does not support.

7 Conclusion

COMPACT-Bench expresses memory-compaction methods from several literatures in a single unit, bytes of retained state per token of history, and measures properties that single-turn accuracy does not capture. Measured in bytes, mechanisms differ far more than their native units suggest, and the best mechanism depends on the model. Tolerance to eviction tracks release date far more closely than scale, and the deficit of attention-based scorers relative to random eviction is a consequence of the question-agnostic protocol. Eviction that fragments a fact leads the model to report the fact as present and to answer with unwarranted confidence, whereas eviction of contiguous spans is reported correctly. We release COMPACT-Bench so that compaction methods can report what a system has lost alongside its accuracy at each budget.

AI USE STATEMENT

We used generative AI tools to help polish the text and to help implement, run, and debug the benchmark harness and the experiments reported here. All AI-assisted work was reviewed by the authors. Every number in the paper was recomputed from the per-generation records released with the benchmark, every change to the harness was tested against the reference implementation it modifies, and the hypothesis stated in advance in Section 4.3 was committed to version control before the runs that tested it. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix B lists the hardware, model, and weight precision of every setting (Tables 4 and 5), the BPT accounting (Appendix B.3), and every prompt, sweep, and metric (Appendix B.4), including the correction to the reversibility harness and the handling of completions without a stated confidence. Appendix A states the assumptions of Proposition 1 and gives its full derivation. The supplementary material contains an anonymized copy of the benchmark code and the per-generation JSON records behind every reported number.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding, 2024. URL https://arxiv.org/abs/2308.14508.

Aydar Bulatov, Yuri Kuratov, and Mikhail S. Burtsev. Recurrent memory transformer, 2022. URL https://arxiv.org/abs/2207.06881.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory, 2025. URL https://arxiv.org/abs/2504.19413.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient llm inference, 2025. URL https://arxiv.org/abs/2407.11550.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Themistoklis Haris and Krzysztof Onak. Compression barriers for autoregressive transformers, 2025. URL https://arxiv.org/abs/2502.15955.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization, 2025. URL https://arxiv.org/abs/2401.18079.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models, 2023. URL https://arxiv.org/abs/2310.05736.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom

Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022. URL https://arxiv.org/abs/2207.05221.

Greg Kamradt. Needle in a haystack – pressure testing LLMs. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023. Software.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

Yucheng Li, Huiqiang Jiang, Qianhui Wu, Xufang Luo, Surin Ahn, Chengruidong Zhang, Amir H. Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and Lili Qiu. Scbench: A kv cache-centric analysis of long-context methods, 2025. URL https://arxiv.org/abs/2412.10319.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation, 2024. URL https://arxiv.org/abs/2404.14469.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache, 2024. URL https://arxiv.org/abs/2402.02750.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents, 2024. URL https://arxiv.org/abs/2402.17753.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. Nolima: Long-context evaluation beyond literal matching, 2025. URL https://arxiv.org/abs/2502.05167.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens, 2024. URL https://arxiv.org/abs/2304.08467.

NVIDIA. KVPress: LLM KV cache compression made easy. https://github.com/NVIDIA/kvpress, 2024. Software library.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. Transformers are multi-state rnns, 2024. URL https://arxiv.org/abs/2401.06104.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. URL https://arxiv.org/abs/2310.08560.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression, 2024. URL https://arxiv.org/abs/2403.12968.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023. URL https://arxiv.org/abs/2304.03442.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback, 2023. URL https://arxiv.org/abs/2305.14975.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000. URL https://arxiv.org/abs/physics/0004057.

Yihang Wang, Xu Huang, Bowen Tian, Yueyang Su, Lei Yu, Huaming Liao, Yixing Fan, Jiafeng Guo, and Xueqi Cheng. Quito-x: A new perspective on context compression from the information bottleneck theory, 2025. URL https://arxiv.org/abs/2408.10497.

Kaiyue Wen, Xingyu Dang, and Kaifeng Lyu. Rnns are not transformers (yet): The key bottleneck on in-context retrieval, 2024. URL https://arxiv.org/abs/2402.18510.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory, 2025. URL https://arxiv.org/abs/2410.10813.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024. URL https://arxiv.org/abs/2309.17453.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms, 2023. URL https://arxiv.org/abs/2306.13063.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. Helmet: How to evaluate long-context language models effectively and thoroughly, 2025. URL https://arxiv.org/abs/2410.02694.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞bench: Extending long context evaluation beyond 100k tokens, 2024. URL https://arxiv.org/abs/2402.13718.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2o: Heavy-hitter oracle for efficient generative inference of large language models, 2023. URL https://arxiv.org/abs/2306.14048.

A Derivation of the error floor

We restate Proposition 1 with its assumptions made explicit and give the short derivation. Fix a finite answer set $\mathcal{Y}$, use base-two logarithms, and let the budget bound the description length of the representation, $H(Z) \le B$ bits. Any physical memory budget gives such a bound. Assume the usage operator reads the history only through the compact representation and the query, so $\hat{Y} = \mathcal{U}(Z, Q)$, and Y and $\hat{Y}$ are conditionally independent given (Z, Q). All quantities below are implicitly conditioned on Q and averaged over its distribution, and $P_e = \Pr[\hat{Y} \neq Y]$.

Conditioned on Q, the representation $Z = \mathcal{C}(H, Q)$ is a function of H alone, whether or not $\mathcal{C}$ also depends on Q, so $Y - H - Z$ forms a Markov chain given Q. The same holds for $Y - Z - \hat{Y}$, since $\hat{Y}$ is a function of (Z, Q). Applying the data-processing inequality along each chain gives two separate bounds on how much $\hat{Y}$ can tell us about Y. The first comes from the chain through H:

$$\mathrm{I}(Y; \hat{Y} \mid Q) \;\le\; \mathrm{I}(Y; Z \mid Q) \;\le\; \mathrm{I}(Y; H \mid Q) \;=\; I^\star(Q). \tag{3}$$

That is, $\hat{Y}$ carries no more information about Y than the representation from which it is computed, and the representation carries no more than the history. The second bound follows from the budget, since $\hat{Y}$ is a deterministic function of (Z, Q):

$$\mathrm{I}(Y; \hat{Y} \mid Q) \;\le\; H(\hat{Y} \mid Q) \;\le\; H(Z \mid Q) \;\le\; H(Z) \;\le\; B. \tag{4}$$

Together these give $\mathrm{I}(Y; \hat{Y} \mid Q) \le \min(I^\star(Q), B)$, so no scheme transmits more information about the answer than the smaller of what the history contains and what the budget can hold. Fano's inequality converts this information bound into an error bound. Using the standard relaxations $H_b(P_e) \le 1$ and $\log(|\mathcal{Y}| - 1) \le \log|\mathcal{Y}|$, Fano gives $H(Y \mid \hat{Y}, Q) \le 1 + P_e \log|\mathcal{Y}|$. Since $H(Y \mid \hat{Y}, Q) = H(Y \mid Q) - \mathrm{I}(Y; \hat{Y} \mid Q)$ by the definition of conditional entropy, rearranging and substituting the information bound above gives

$$P_e \;\ge\; \frac{H(Y \mid Q) - \mathrm{I}(Y; \hat{Y} \mid Q) - 1}{\log|\mathcal{Y}|} \;\ge\; \frac{H(Y \mid Q) - \min(I^\star(Q), B) - 1}{\log|\mathcal{Y}|}. \tag{5}$$

When the budget is the limiting factor, $B < I^\star(Q)$, the minimum equals B, which gives Eq. 2.

Three remarks qualify the interpretation of the bound. First, it is a converse and not an achievability result: it bounds from below the error of every scheme that satisfies the assumptions above but does not indicate whether a particular scheme approaches it, so a well-designed method may operate well above the floor. Second, it is loose, because the relaxations in the Fano step are coarse, and it is informative only when $H(Y \mid Q)$ exceeds the budget by a clear margin. Third, query conditioning does not move the $I^\star(Q)$ ceiling, since that quantity is defined without reference to any operator and the derivation does not assume that $\mathcal{C}$ ignores the query. Query conditioning instead determines how efficiently a method spends its budget in approaching that ceiling. The penalty for ignoring the query, discussed in Section 2, is the gap between the conditional and unconditional rates a representation would need, $\mathrm{I}(Z; H \mid Q) - \mathrm{I}(Z; H)$, which is at most $H(Q)$ and usually much smaller in practice.

B Benchmark details

B.1 Hardware and models

The reference-model frontier (Figure 3) uses `Qwen2.5-1.5B-Instruct` on a single consumer GPU with 8 GB of VRAM (RTX 4060), with peak usage of 3.6 GiB in fp16 on the 8,000-token contexts. The standard-document rows of Table 2 for Qwen2.5-1.5B come from the corrected-harness run on the server described below. We chose a small reference scale so that the protocol can be reproduced on commodity hardware. The harness accepts any Hugging Face causal LM via `--model`, and the accounting in Appendix B.3 does not depend on the model. The first cross-scale ladder repeats a subset of the protocol on the same GPU at the five further settings listed in Table 4. Every setting fits within 6.4 GiB because weights above 1.5B are loaded in 4-bit NF4 (`--quant nf4`). This gives up some weight precision in exchange for testing three to four times as many parameters on the same card.

Table 4: Cross-scale hardware settings, measured on the RTX 4060.

Model	Precision	Full-cache BPT	Peak VRAM
Qwen2.5-0.5B-Instruct	fp16	12,288 B/tok	1.12 GiB
Qwen2.5-1.5B-Instruct	fp16	28,672 B/tok	3.25 GiB
Qwen2.5-1.5B-Instruct	nf4	28,672 B/tok	2.69 GiB
Qwen2.5-3B-Instruct	nf4	36,864 B/tok	5.19 GiB
Phi-3.5-mini-instruct	nf4	393,216 B/tok	6.39 GiB
Qwen2.5-7B-Instruct	nf4	57,344 B/tok	5.94 GiB

The second ladder ran on a server with two 32 GB RTX 5000 Ada GPUs, one job per GPU. It adds 16-bit runs of Qwen2.5-3B, Qwen2.5-7B, and Phi-3.5-mini and the five newer models, with peak VRAM of 16.2 GiB for Qwen3-8B, 28.9 GiB for Phi-4, and 29.2 GiB for OLMo-2-13B in bf16. KVPress cannot compress a model that `accelerate` shards across GPUs, because the attention mask is built for the compressed length while layers on the other device keep the full cache. Qwen3-14B and Mistral-Small-24B, which do not fit on one card at 16 bits, therefore run in NF4. KVPress also rebuilds queries for scoring without OLMo-2's QK-norm, and multiplies OLMo-2's float32 rotary tables against bf16 queries. Our harness corrects both inside the scoring hook, which leaves the model's own forward pass and the cache it compresses unchanged.

A third batch adds the models that separate release date from scale. Qwen3 at 0.6B, 1.7B, and 4B (the last in NF4), SmolLM2-1.7B, and OLMo-2-1B ran on the RTX 4060, as did every mechanism, question-aware, multikey, and dense reversibility run and every audit except Qwen2.5-14B's. Qwen2.5-14B ran on the server from a published bitsandbytes NF4 checkpoint (`unsloth/Qwen2.5-14B-Instruct-bnb-4bit`), the same quantization our loader applies to Qwen3-14B and Mistral-Small-24B, because the server's scratch disk could not hold its 16-bit weights.

B.2 Protocol rules

Every comparison in COMPACT-Bench holds everything fixed except the method under test and its budget. The same random seed generates the same needle documents and fact sets at every point on

Table 5: Cross-scale results for each setting (replication subset: lengths $\{2000, 4000\}$, depths $\{0.1, 0.5, 0.9\}$, 8 trials, 48 generations per cell). Collapse is the interpolated budget fraction at which accuracy averaged over the five non-random scorers first falls below half the model's own baseline, on the grid $\{0.9, 0.75, 0.5, 0.25, 0.1\}$; the content-only column leaves out StreamingLLM, and >90% means accuracy is already below half at the largest budget. Multikey is the collapse point with three distractor needles (– where not run). Crossover is where the raw buffer overtakes the summary in the standard reversibility task (– means the summary is at least as accurate at every budget below the full history, † run with the earlier summary cap of Appendix B.4, n/r not run with the corrected cap). SmolLM2's summaries copy the repeated filler verbatim and keep no facts, which accounts for its early crossover. The rule separates models released before October 2024 (top) from later ones (bottom).

Model (precision)	Released	Full BPT	Collapse	Content only	Multikey	Crossover
Qwen2.5-0.5B (fp16)	2024-09	12,288	86%	>90%	>90%	48%
Qwen2.5-1.5B (fp16)	2024-09	28,672	84%	89%	88%	41%
Qwen2.5-1.5B (nf4)	2024-09	28,672	83%	88%	–	n/r
Qwen2.5-3B (fp16)	2024-09	36,864	80%	87%	–	–
Qwen2.5-3B (nf4)	2024-09	36,864	82%	88%	87%	n/r
Qwen2.5-7B (fp16)	2024-09	57,344	82%	90%	–	–
Qwen2.5-7B (nf4)	2024-09	57,344	82%	>90%	–	n/r
Qwen2.5-14B (nf4)	2024-09	196,608	76%	78%	–	–
Phi-3.5-mini (bf16)	2024-08	393,216	80%	87%	–	–
Phi-3.5-mini (nf4)	2024-08	393,216	82%	90%	–	n/r
Qwen3-0.6B (bf16)	2025-04	114,688	69%	71%	77%	61%
OLMo-2-1B (bf16)	2025-04	131,072	49%	53%	–	40%
Qwen3-1.7B (bf16)	2025-04	114,688	63%	63%	69%	–
SmolLM2-1.7B (bf16)	2024-11	196,608	56%	61%	–	14%
Qwen3-4B (nf4)	2025-04	147,456	62%	62%	68%	–
Qwen3-8B (bf16)	2025-04	147,456	49%	46%	–	–†
Qwen3-8B (nf4)	2025-04	147,456	53%	50%	–	n/r
OLMo-2-13B (bf16)	2024-11	819,200	43%	44%	–	n/r
Phi-4 (bf16)	2024-12	204,800	33%	31%	–	–†
Qwen3-14B (nf4)	2025-04	163,840	59%	58%	–	n/r
Mistral-Small-24B (nf4)	2025-01	163,840	65%	71%	–	n/r

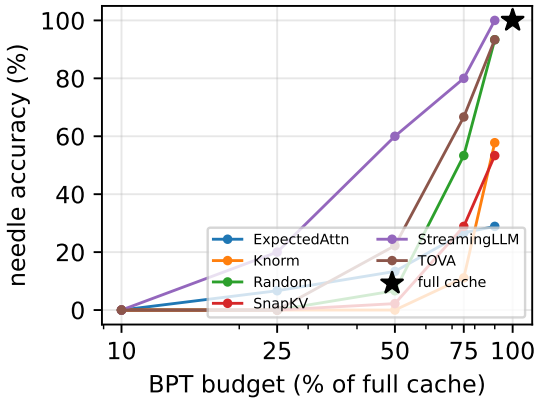

Figure 3: Needle accuracy against BPT for six KV methods on the reference model Qwen2.5-1.5B (1,395 generations), with the cache compressed before the question arrives (Section 4.2).

a sweep, so two methods compared at the same budget answer the same questions and not different questions drawn from a shared distribution. This matters because a method's difficulty depends heavily on where the answer sits in the document, and letting that position vary between conditions would confound the comparison. We report the number of generations behind every summary statis-

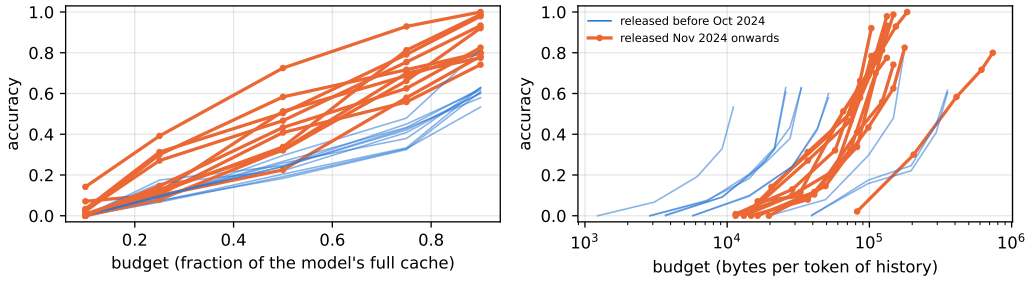

Figure 4: The sweep of Table 5 plotted against budget fraction (left) and against absolute bytes per token (right, log scale), for all 21 settings. In bytes the curves spread over two orders of magnitude, while as a fraction they separate into models released before October 2024 and those released from November 2024 onwards.

tic, since our cells are small (48 generations in most tasks). Intervals on collapse points and their paired differences come from a parametric bootstrap with 2,000 replicates, in which the accuracy of every (method, budget) cell and of the baseline is resampled binomially at its observed rate and sample size, and the collapse point is recomputed by the same interpolation; intervals on rates are Wilson score intervals. No method chooses the budget at which it is evaluated: every method in a sweep runs at the same fixed set of budgets, so none can be reported only at its most favourable operating point, as often happens when method papers report a single ratio. Appendix B.4 describes the question-agnostic setting in which all KV methods are run.

B.3 BPT ACCOUNTING, WORKED EXAMPLE

For a model with L layers, H KV heads, and head dimension d, one full-precision token of KV cache costs $2LHd \cdot$ (bytes/scalar) bytes, where the factor of 2 accounts for keys and values. For Qwen2.5-1.5B-Instruct (L=28, H=2, d=128, fp16), this is $2{\cdot}28{\cdot}2{\cdot}128{\cdot}2 = 28{,}672$ bytes/token, the full-cache value in Figure 3. An evictor that keeps a fraction f of entries costs $f \times 28{,}672$ bytes/token. A b-bit quantizer stores b bits per scalar, and it also stores metadata that the "b bits" of its literature leaves out. Our quantizer (Section 4.1) groups keys per channel and values per token in groups of 32 with one fp16 scale and zero point per group, which adds 1 bit per scalar, and keeps the largest 1% of entries in each layer at fp16 with a 32-bit index, as KVQuant does (Hooper et al., 2025), which adds 0.48 bits. It therefore costs $(b + 1.48)/16$ of the full cache, 34.2% at 4 bits and 21.8% at 2 bits. Without the outlier set, 4-bit keys break Qwen2.5 entirely, because its first layer holds key channels with magnitudes near 300. Reported in their own units, 75% eviction and 4-bit quantization look equivalent ("25% of tokens kept" and "4 bits of 16"), but once every stored byte is counted the 4-bit cache costs as much as a 66% eviction. Eviction followed by quantization multiplies the two costs, so keeping half the tokens at 8 bits costs $0.5 \times 9.48/16 = 29.6\%$. Prompt compression from n history tokens to m kept tokens costs $(m/n) \times 28{,}672$ bytes/token, and soft or gist compression to g vectors of dimension d_g costs $g \cdot d_g \cdot$ (bytes/scalar)$/n$ bytes/token, amortized in the same way. The same short accounting module prices all four mechanisms (`compactbench/bpt.py`, with the quantizer in `compactbench/presses.py`). Physical bytes are an upper bound on retained information but not an estimate of its entropy, since a byte can be spent on redundant content.

B.4 TASK PROTOCOLS AND PROMPTS

All four tasks use the same needle-in-filler construction. A key-value fact ("The special magic number for {key} is {value}.") is inserted at a controlled relative depth into a pool of natural filler text truncated to a target token length. Random seeds are fixed across every budget, so that only the compaction method and its ratio vary between cells.

- **Frontier** (§4.2, Fig. 3): the question "What is the special magic number for {key}? Answer with just the number." is asked after compaction, and an answer counts as correct if it contains the value. The reference sweep covers lengths $\{2000, 4000, 8000\}$ tokens $\times$ depths $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ $\times$ 3 trials (45 needles per cell) $\times$ six

methods (SnapKV, StreamingLLM, TOVA, Knorm, ExpectedAttn, Random) $\times$ ratios $\{0.1, 0.25, 0.5, 0.75, 0.9\}$, plus a full-cache baseline, giving the 1,395 generations behind Figure 3.

- **Reversibility** (§4.4, Table 2): this task does not use the needle. Each document has 24 chunks of repeated synthetic filler (about 1,600 tokens in total), with twelve facts of the form "the {attribute} of {entity} is {value}" placed in randomly chosen chunks. Both arms read the document chunk by chunk and compact on overflow. The summary arm asks the model to summarize its buffer within the budget while keeping every name and number. The raw arm evicts the oldest chunks and, at answer time, passes on the kept chunks that mention the entity asked about (all kept chunks if none do). Each fact is asked as "What is the {attribute} of {entity}? Answer with just the value." Budgets are $\{0.1, 0.25, 0.5, 0.75, 1.0\}$ of the document length, with 8 documents (96 questions per budget). The summary may use the whole budget. Generation is capped at the budget plus 40 tokens and the output is truncated to the budget. An earlier version of the harness capped generation at 380 tokens, which cut off any summary longer than that, possible at budgets above 24% of the document. Every reversibility number in the main text uses the corrected harness except those for Qwen3-8B and Phi-4. Rerunning the other 16-bit settings with the correction changed summary accuracy by at most 0.19 and never changed which arm leads at any budget, and under the earlier harness the summary already leads Qwen3-8B's and Phi-4's raw buffer by at least 0.21 at every budget below the full history.

- **Frontier variants** (§4.2): the question-aware variant appends "Question: {question}" to the context before compression and asks the question again afterwards. The multikey variant inserts three further needles of the same template for other keys at random depths at least 0.08 apart. Both use the replication subset of Table 5.

- **Mechanisms** (§4.1): the 48 needles of the replication subset (lengths $\{2000, 4000\}$, depths $\{0.1, 0.5, 0.9\}$, 8 trials) are answered after six eviction presses at keep fractions $\{0.5, 0.25, 0.125\}$, KV quantization at $\{8, 4, 3, 2\}$ bits, SnapKV, StreamingLLM, Knorm, and Random eviction followed by quantization at (8 bits, keep 0.5), (4 bits, keep 0.5), and (4 bits, keep 0.25), LLMLingua-2 (`xlm-roberta-large-meetingbank`) at rates $\{0.5, 0.25, 0.125\}$, and a summary written by the model itself, prompted "Summarize the following text in at most $\lfloor 0.75m \rfloor$ words, preserving every specific name and number" and truncated to m tokens. For prompt compression and summaries the BPT is the measured ratio of kept to original tokens.

- **Attribution and calibration** (§4.5): from one compressed cache, the probe "Do your notes contain the special magic number for {key}? Answer with just yes or no." and the answer question followed by "Then, on a new line, state how confident you are that your answer is correct as a number from 0 to 100." are asked in turn, each starting from the same compressed cache, with up to 100 new tokens. Confidence is the last number in $[0, 100]$ in the completion, with the answer's own digits removed first; a completion without one is recorded as missing and left out of the confidence statistics, not scored as 50. Each of the 48 needles (lengths $\{2000, 4000\}$, depths $\{0.2, 0.5, 0.8\}$, 8 trials) is run under the present and absent controls and under SnapKV, StreamingLLM, Knorm, and Random at ratios $\{0.5, 0.9\}$. Survival is the share of (layer, KV head) slots whose retained set contains each needle token, averaged over the tokens of the key or of the value, computed from the indices the press keeps. Because both questions read one cache, the random press takes part here, which it could not in a two-call design.

- **Dense reversibility** (§4.4): 24 chunks, each with one filler sentence and two facts, 48 facts in about 1,300 tokens, 4 documents (192 questions per budget).

Question-agnostic compression. KVPress compresses the context during prefill and appends the question afterwards, so every method commits to a retained set before it sees the question, as when a compressed cache is reused across queries. SnapKV's observation window therefore covers the last 64 tokens of the context, which in the question-agnostic setting is filler, and the question-aware variant of Section 4.2 measures what that costs.

B.5 RUNTIME

A full four-task sweep at the reference scale (the 1.5B model with all the lengths, positions, methods, and ratios above) takes under two hours on a single RTX 4060. Most of the time goes to the attribution and calibration tasks, which need two generations per cell (probe and answer, or answer and confidence). The replication subset used in Section 4.3 is a smaller sweep over fewer lengths, positions, and ratios. It took 23 minutes at 0.5B, 46 at 1.5B fp16, 58 at 1.5B NF4, 73 at 3B, 103 at Phi-3.5-mini, and 102 at 7B, for a total of 6h45m on the same GPU. The quantized settings ran more slowly than their parameter counts alone would suggest, because of NF4 dequantization overhead and longer generations. The third batch ran 44 jobs in 19.1 hours on the same GPU, four of them audits repeated after the confidence parser was fixed: 1.8 hours of frontiers, 4.2 of question-aware and multikey frontiers, 3.3 of audits, 7.5 of standard and dense reversibility, and 2.3 of mechanism sweeps, most of the last two spent writing summaries. Qwen2.5-14B took 42, 23, and 44 minutes for its frontier, audit, and reversibility runs on one RTX 5000 Ada. Per-run logs and raw JSON records (one row per generation, from which the summary statistics in Section 4 are computed as simple aggregates) are released with the benchmark.