NEVER EVICT ALWAYS TAPER: A STATIC LOGARITHMIC BIT-WIDTH SCHEDULE FOR KV CACHE QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-context inference relies on an ever-growing key-value (KV) cache, making precision allocation a key challenge for memory-efficient generation. We study how to distribute precision across cached tokens while preserving access to older information, without requiring online attention tracking. Under an exponential distortion surrogate and a prescribed precision floor, reverse water-filling yields an optimal allocation in which precision depends logarithmically on token importance; with power-law age weighting, this naturally produces a precision schedule that decreases with token age. A complementary perturbation analysis establishes a sufficient precision floor for preserving the highest-scoring attention key. Motivated by these results, we propose TaperKV, a token-preserving KV cache with a static, age-indexed integer-precision schedule that is compatible with a broad range of quantizers. Experiments on retrieval, language modeling, and mathematical reasoning show that TaperKV substantially reduces cache storage while maintaining strong task performance.

1 INTRODUCTION

KV-cache compression must reconcile two competing goals: reducing memory consumption while preserving access to information from earlier in the context. Unlike model weights, the KV cache grows with sequence length and is repeatedly accessed throughout autoregressive generation (Pope et al., 2023; Kwon et al., 2023a). Its memory footprint, therefore, becomes particularly consequential in long-context applications and in models that produce extended reasoning traces (Guo et al., 2025; Qwen Team, 2025). The central challenge is not merely to store fewer bits but to do so without discarding information that may become important for a future query.

Existing approaches largely follow two strategies. *Eviction* retains a subset of cache entries and removes the rest, typically based on recency, attention sinks, or estimated token importance (Xiao et al., 2024; Zhang et al., 2023; Li et al., 2024). *Quantization*, in contrast, retains cache entries but represents them with reduced precision (Liu et al., 2024; Hooper et al., 2024; Zandieh et al., 2025). Uniform quantization is simple but allocates the same number of bits to tokens whose expected importance may differ substantially. Recent mixed-precision and progressive methods mitigate this mismatch using layer-wise sensitivity, calibration data, or reasoning-phase information (Li et al., 2025; Liu et al., 2025; Ramachandran et al., 2025). We instead ask whether a simple static rule can allocate precision effectively without requiring online attention statistics, calibration, or phase detection.

> *How should a static, age-based precision schedule compress the KV cache while preserving access to older information?*

This question naturally splits into two problems: how to distribute precision when token importance is nonuniform, and how much precision is needed to prevent an older entry from becoming too distorted to retrieve. The first problem is governed by average reconstruction distortion; the second depends on the margin between competing attention scores. These objectives are distinct: a token may have low expected importance while still being critical for a particular query.

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.

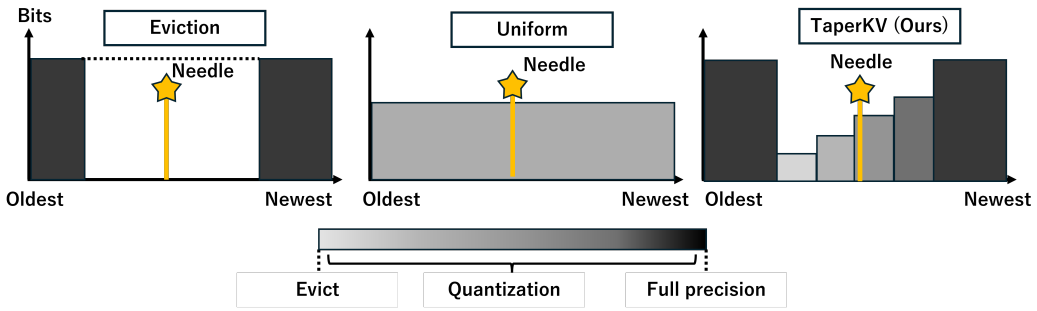


Figure 1: Horizontal axis is the bit width at each cached position. The star marks a needle at an old position. TaperKV instead reduces precision across equal-width age bands while protecting sink and recent positions. Darker bands denote higher configured precision, not necessarily full precision. The illustration is schematic and does not represent an equal-memory comparison. Retaining an entry prevents its information from being deleted; preserving its original status as the highest-scoring key additionally requires the margin condition in Theorem 3.5.

We address both aspects through an allocation analysis and a corresponding cache design. Under a stylized exponential distortion–rate model, reverse water-filling implies that optimal precision varies logarithmically with token importance. When importance decays with age according to a power law, the resulting ideal precision profile decreases logarithmically with age. Separately, a perturbation analysis yields a sufficient lower bound on key precision for preserving the maximizer of the attention score. TaperKV translates these principles into a static integer-precision schedule with equal-width age bands, a protected sink, recent positions, and a strictly positive precision floor (Figure **??**). The deployed schedule is motivated by the continuous optimum rather than claimed to reproduce it exactly. TurboTaper further combines this age-based schedule with a rotated quantizer.

Contributions. First, we characterize optimal continuous and discrete bit allocations under a floored distortion surrogate, and identify conditions under which graded precision improves over a zero-or-high-precision allocation. Second, we derive a conditional guarantee for preserving key selection and analyze the monotonicity, accumulated quantization error, and storage cost of the resulting static ladder. Third, we evaluate the design on retrieval, language modeling, and mathematical reasoning tasks.we can substantially reduce encoded cache storage Across these settings, the results show that substantial reductions in encoded cache storage can be achieved while retaining strong retrieval and reasoning performance, and that an overly low precision floor is a major source of degradation. Together, these findings motivate a simple design principle: *reduce the precision of older information before removing its representation*.

2 Problem Formulation

Attention and storage. Let n, d be positive integers. Consider a single attention head with query $q \in \mathbb{R}^d$, keys $k_i \in \mathbb{R}^d$, and values $v_i \in \mathbb{R}^d$ for $i \in \{1, \dots, n\}$. All norms and inner products are Euclidean. The attention scores, weights, and output are

$$s_i = \langle q, k_i \rangle / \sqrt{d}, \qquad a_i = \mathrm{softmax}(s)_i = \frac{e^{s_i}}{\sum_{j=1}^{n} e^{s_j}}, \qquad o = \sum_{i=1}^{n} a_i v_i. \tag{1}$$

For a layer with h *KV heads*, storing keys and values at a common width b requires $2nhdb$ code bits, excluding metadata. Under grouped-query attention, multiple query heads may share a KV head; hence, h denotes the number of stored KV heads rather than the number of query heads (Ainslie et al., 2023). We analyze keys and values separately and later allow role-specific precision schedules.

Distortion surrogate. For a random vector x satisfying $0 < \mathbb{E}\|x\|^2 < \infty$, let $\hat{x}$ denote its reconstruction at bit width b. We measure distortion by the relative mean squared error

$$\mathrm{relMSE}(b) = \frac{\mathbb{E}\|\hat{x} - x\|^2}{\mathbb{E}\|x\|^2}, \tag{2}$$

where the expectation is over x and any randomness in the codec. Classical high-resolution quantization theory, as well as recent analyses of rotated quantizers, motivates an exponential dependence

of distortion on bit width (Panter & Dite, 1951; Gray & Neuhoff, 1998; Zandieh et al., 2025). We adopt this scaling as a tractable model for allocation, rather than as an exact description of low-bit quantization for every cached vector.

Assumption 2.1 (Allocation distortion model)**.** For a fixed model constant $\kappa \geq 1$, we associate bit width $b \geq 0$ with normalized distortion

$$D(b) = \min\{1, \kappa 4^{-b}\}. \tag{3}$$

Here, $D(0) = 1$ represents an uninformative reconstruction under the surrogate. For $b \geq \log_4 \kappa$, the distortion $D(b) = \kappa 4^{-b}$ is strictly decreasing and strictly convex.

The condition $\kappa \geq 1$ is a modeling assumption, not a universal information-theoretic lower bound for arbitrary KV distributions. Similarly, a zero-rate reconstruction is not equivalent to removing a token from the softmax computation. We use $b_i = 0$ only to represent zero storage allocated to an entry; the effects of actual token eviction on attention are treated separately.

Allocation objective. Let $w_i > 0$ be fixed importance weights satisfying $\sum_i w_i = 1$. These weights may capture factors such as expected attention mass and vector energy, but we do not assume that they exactly represent downstream task sensitivity. Given token-wise widths $b = (b_1, \ldots, b_n)$, a total budget $0 \leq B \leq nb_{\text{hi}}$, and a maximum width $b_{\text{hi}} > 0$, we consider

$$J(b) = \sum_{i=1}^{n} w_i D(b_i), \qquad \sum_{i=1}^{n} b_i \leq B, \qquad 0 \leq b_i \leq b_{\text{hi}}. \tag{4}$$

Here, B is the bit budget summed across tokens per coordinate, corresponding to dB code bits for one KV head and one role. Let $J_0^\star$ denote the minimum of equation 4, and let $J_{\min}^\star$ denote the minimum when an additional feasible floor $0 \leq b_{\min} \leq b_{\text{hi}}$ is imposed. Uniform allocation sets $b_i = B/n$ when feasible, whereas a zero-or-high-precision allocation restricts $b_i \in \{0, b_{\text{hi}}\}$. Metadata is omitted from this optimization and accounted for separately in our experiments.

Age-based importance model. Empirical recency and attention-sink effects suggest that different cache positions need not receive equal precision (Xiao et al., 2024; Zhang et al., 2023). For non-sink tokens, we model importance as

$$w(t) \propto (t + t_0)^{-\gamma}, \tag{5}$$

over a finite set of nonnegative token ages, where $t_0 > 0$ and $\gamma > 0$. This power-law profile is an analytical model rather than an assumption that must hold for every attention head. Importantly, our deployed schedule does not estimate or require access to these weights.

3 Theoretical Analysis: Graded Precision Allocation and Key-Selection Stability

Our answer to the research question has three components: a floored allocation principle, a condition for preserving key selection, and a static schedule that requires only token positions. We first derive the principle, then state the guarantee and implement the schedule. All proofs are provided in Appendix A.

3.1 Allocate precision according to importance

Theorem 3.1 (reverse water-filling with a prescribed floor)**.** *Assume* $\log_4 \kappa \leq b_{\min} < b_{\text{hi}}$ *and* $nb_{\min} \leq B \leq nb_{\text{hi}}$*. The floored version of equation 4, with real-valued widths, has the unique minimizer*

$$b_i^\star = \left[\nu + \tfrac{1}{2}\log_2 w_i\right]_{b_{\min}}^{b_{\text{hi}}}, \qquad [x]_l^u = \min\{u, \max\{l, x\}\}, \tag{6}$$

where a water level ν *is chosen so that* $\sum_i b_i^\star = B$*. Such a level exists; it is unique when at least one coordinate is strictly between the bounds. If no coordinate is clipped, then*

$$J_{\min}^\star = n\kappa \,\mathrm{GM}(w) 2^{-2B/n}, \qquad \mathrm{GM}(w) = \left(\prod_{i=1}^{n} w_i\right)^{1/n}. \tag{7}$$

For two interior coordinates, $b_i^\star - b_j^\star = \frac{1}{2}\log_2(w_i/w_j)$: a twofold importance ratio corresponds to a half-bit precision difference. Changing one weight at fixed total budget also changes the water level. The floor is prescribed, not proved universally necessary; without it, the saturated objective need not be convex (Appendix A.1).

Corollary 3.2 (power-law weights imply a logarithmic profile)**.** *Under Theorem 3.1, weights $w(t) \propto (t + t_0)^{-\gamma}$ yield*

$$b^\star(t) = \left[c - \tfrac{\gamma}{2}\log_2(t + t_0)\right]_{b_{\min}}^{b_{\mathrm{hi}}}, \tag{8}$$

where c enforces the budget. Away from clipping boundaries, precision decreases by $\gamma/2$ bits when $t + t_0$ doubles. On a continuous age axis, unit-step rounding with a fixed offset gives band-width ratio $4^{1/\gamma}$ for complete interior bands; clipping and discrete token boundaries alter endpoint widths.

Proposition 3.3 (integer allocation and rounding)**.** *Let integers $\log_4 \kappa \le b_{\min} < b_{\mathrm{hi}}$ satisfy $nb_{\min} \le B \le nb_{\mathrm{hi}}$. Starting at $b_i = b_{\min}$, perform $\lfloor B \rfloor - nb_{\min}$ unit increments, each maximizing $w_i[D(b_i) - D(b_i + 1)]$ over $b_i < b_{\mathrm{hi}}$, with arbitrary tie-breaking. This attains the optimum among all feasible integer allocations. A* feasible *allocation within half a bit per coordinate of equation 6 has distortion at most $2J_{\min}^\star$; rounding down is feasible and gives at most $4J_{\min}^\star$.*

These are guarantees for the specified optimization problem. They do not certify the equal-width implementation below: geometrically widening ideal bands and equal-width deployed bands are different schedules.

3.2 When can graded precision improve on eviction?

A zero-or-high-precision allocation spends b_{hi} bits on each retained token and none on the rest. Comparing it with graded precision clarifies the potential cost of concentrating the budget, without claiming that the surrogate determines end-to-end accuracy.

Theorem 3.4 (comparison within the allocation surrogate)**.** *Let $0 \le B \le nb_{\mathrm{hi}}$, $b_{\mathrm{hi}} > 0$, and $m = \lfloor B/b_{\mathrm{hi}} \rfloor$. The best zero-or-high-precision allocation has distortion*

$$J_{\mathrm{evict}} = \min_{|S|=m} \left\{ \sum_{i \notin S} w_i + D(b_{\mathrm{hi}}) \sum_{i \in S} w_i \right\}, \tag{9}$$

and $J_0^\star \le J_{\mathrm{evict}}$. If an optimal set S contains k but not e, and some $0 < \delta \le b_{\mathrm{hi}}$ satisfies

$$w_e[1 - D(\delta)] > w_k[D(b_{\mathrm{hi}} - \delta) - D(b_{\mathrm{hi}})], \tag{10}$$

then $J_0^\star < J_{\mathrm{evict}}$.

The strict inequality follows by transferring δ bits from a retained token to a discarded one. The gain must exceed the donor's loss; it need not do so for arbitrary importance weights. Appendix A.4 gives the proof and an analytical example. This comparison concerns $J_0^\star$, not the more restrictive $J_{\min}^\star$, and it is not a theorem that every eviction method loses on every task.

3.3 Preserve key selection with a sufficient floor

Small average distortion does not guarantee that a particular query selects the same key. We therefore analyze score perturbations separately. The following vectorwise error bound is an *additional assumption*; it does not follow from an expected-MSE model.

Theorem 3.5 (key-selection stability and information loss)**.** *(a) Fix q and $n \ge 2$ keys. Let $s_{(1)} > s_{(2)}$ be the two largest scores, with a margin $\Delta = s_{(1)} - s_{(2)} > 0$. Suppose the actual reconstructed keys satisfy $\left\|\hat{k}_i - k_i\right\| \le \sqrt{\kappa}\, 2^{-b_i} \|k_i\|$ for every i. If $b_i \ge b_{\min}$ and*

$$b_{\min} > \log_2 \frac{2\sqrt{\kappa}\, \|q\| \max_i \|k_i\|}{\sqrt{d}\, \Delta}, \tag{11}$$

then $\arg\max_i \left\langle q,\, \hat{k}_i \right\rangle = \arg\max_i \langle q,\, k_i \rangle$.

(b) Let a payload W be uniform on a set of size $M \ge 2$. After deletion, let Z include the entire available state, including the query, metadata, and decoder randomness. If W remains uniform and independent of Z, every decoder f has $\Pr[f(Z) = W] \le 1/M$. For data-dependent deletion, these assumptions are conditional on its occurrence.

Part (a) preserves a highest-scoring *key*, not the value, attention output, or generated answer. A feasible certified floor requires the right-hand side of equation 11 to be below the configured ceiling. The condition is sufficient, not a necessary threshold. Part (b) requires that no payload information remains elsewhere; deletion alone does not imply this independence.

4 TaperKV: Graded Precision without Eviction

The previous section showed that, the optimal bit width grows logarithmically with the token importance. In other words, under the power-law age weighting, bit width decays logarithmically with token age. Two properties of this solution make it easy to implement. First, the bit width depends only on age, so a token index is enough to determine it, and no attention statics, calibration data, or phase classification are needed. Second, the optimal widths span a narrow range. Doubling a token imporance buys only half a bit, so handful of integer levels covers importance ratios of several orders of magnitude, and Proposition 3.3 bounds the cost of rounding to those levels. Guided by these observations, we propose TaperKV, a novel token-preserving KV cache with a static, age-indexed integer-precision schedule.

4.1 TaperKV Schedule

Definition 4.1 (TaperKV schedule)**.** Fix sink size S and recent-window size R as nonnegative integers, widths $1 \le b_{\min} \le b_{\mathrm{hi}}$ as integers, and update stride $r \ge 1$ as an integer. Define $n_L = b_{\mathrm{hi}} - b_{\min} + 1$ and $L[j] = b_{\mathrm{hi}} - j$ for $j \in \{0, \dots, n_L - 1\}$. At integer sequence length $T \ge 1$, a token at absolute position $p \in \{0, \dots, T-1\}$ has age $t = T - 1 - p$ and target nominal width

$$b(t,T) = \begin{cases} b_{\mathrm{hi}}, & p < S \text{ or } t < R, \\ L\left[\left\lfloor n_L \dfrac{t-R}{T-S-R} \right\rfloor\right], & \text{otherwise.} \end{cases} \tag{12}$$

The second case implies $T > S + R$ and a fraction in $[0, 1)$. Apply the rule separately to keys and values. Initialize prompt entries at their targets and append new entries at b_{hi}. After attention at lengths divisible by r, requantize entries whose stored width exceeds their target.

Sink positions and recent tokens use the *highest configured width*, which need not be full precision. The intervening region is divided into equal-width bands, up to integer boundary effects. "Static" means that the parameters and positional rule are fixed: the rule depends on sequence length but never estimates attention or detects reasoning phases. Between updates, an entry may remain at a higher width than its current target.

Proposition 4.2 (monotone precision)**.** *For each fixed token, the target nominal width in Definition 4.1 is non-increasing during decoding. Sink tokens remain at b_{hi}. Consequently, a token undergoes at most $b_{\mathrm{hi}} - b_{\min}$ precision reductions after initial encoding, including when updates are delayed to stride boundaries.*

Proposition 4.3 (accumulated requantization error)**.** *For a fixed original vector x, suppose each encoding or requantization at a visited level $j \le b_{\mathrm{hi}}$ adds an error of norm at most $C2^{-j}\,\|x\|$, for a common $C \ge 0$ relative to the* original *norm. Let $\hat{x}_{\mathrm{prog}}$ be the final reconstruction along strictly decreasing integer levels ending at $b \ge 0$. Then*

$$\|\hat{x}_{\mathrm{prog}} - x\| \le C\,\|x\| \sum_{j=b}^{b_{\mathrm{hi}}} 2^{-j} < 2C2^{-b}\,\|x\| \quad (C\,\|x\| > 0). \tag{13}$$

For $C\,\|x\| = 0$, the accumulated error is zero.

This compares error *upper bounds*, not realized errors. Appendix A.7 treats root-mean-square and current-input-relative bounds and connects the accumulated deterministic error to the key-selection certificate. None follows automatically for an arbitrary codec.

Remark 4.4 (predictable encoded storage). For fixed S, R, r and ladders, each role's mean *nominal* width is at least $(b_{\mathrm{hi}} + b_{\min})/2$ and converges to it as $T \to \infty$. Thus the limiting nominal-code fraction is $(\bar{b}_K + \bar{b}_V)/32$, where $\bar{b}_K, \bar{b}_V$ are the respective midpoints. Extra protected-channel bits and metadata are additional; Appendix A.6 gives the finite-length identity.

TaperKV uses per-token affine integer quantization within each band. TurboTaper uses a rotated Lloyd–Max-based codec inspired by TurboQuant (Zandieh et al., 2025). Both use the same scheduling rule; their actual role widths and codec options are experimental settings, specified in Section6.1. A different rotation, channel-protection rule, or progressive encoding path does not automatically inherit every guarantee of a one-shot codec.

4.2 Allow different precision for keys and values

Lemma 4.5 (first-order attention sensitivity)**.** *Hold q, n, d and the original keys and values fixed. Let δo be the output change when k_i, v_i are replaced by $k_i + \delta k_i, v_i + \delta v_i$. Stack the perturbations into δ, with $\|\delta\|^2 = \sum_i (\|\delta k_i\|^2 + \|\delta v_i\|^2)$. As $\delta \to 0$,*

$$\delta o = \underbrace{\sum_i a_i \delta v_i}_{\textit{value perturbations}} + \underbrace{\sum_i a_i (v_i - o) \frac{\langle q,\, \delta k_i \rangle}{\sqrt{d}}}_{\textit{key perturbations}} + O(\|\delta\|^2). \tag{14}$$

Value errors enter through a convex average. Key errors also change the weights, with sensitivity depending on the query norm and value spread. This motivates role-specific widths, but does not prove that keys always need more bits: when every $v_i = o$, the first-order key term vanishes.

5 Related work

What to retain. Eviction policies exploit sinks and recency (Xiao et al., 2024), persistent importance (Zhang et al., 2023; Liu et al., 2023; Oren et al., 2024), or prompt-dependent selections (Li et al., 2024; Cai et al., 2024; Ge et al., 2024; Adnan et al., 2024). Query-aware sparsity instead reduces which stored entries are read (Tang et al., 2024; Sun et al., 2024). Our focus is different: preserve an encoded representation at every position and vary its precision. The information-loss result in Theorem 3.5(b) states a conditional consequence of deletion, not a universal characterization of these methods.

How to encode and allocate. KV quantizers address channel distributions, outliers, and recent-window sensitivity (Liu et al., 2024; Hooper et al., 2024; He et al., 2024; Dong et al., 2024; Duanmu et al., 2024); rotated codecs improve per-vector distortion (Zandieh et al., 2024; 2025; Han et al., 2025). Allocation is already nonuniform in importance-aware quantization, layer-wise search, progressive precision, and thought-adaptive compression (Yang et al., 2024; Li et al., 2025; Liu et al., 2025; Ramachandran et al., 2025). In particular, PM-KVQ is a quantization method, whereas ThinKV combines quantization with eviction. TaperKV contributes a fixed age-only scheduling rule and an explicit separation between allocation, key-selection conditions, and implementation guarantees; it does not introduce mixed precision itself.

The analytical connection. Reverse water-filling and exponential distortion scaling are classical (Shannon, 1959; Cover & Thomas, 2006; Panter & Dite, 1951; Gersho, 1979; Gray & Neuhoff, 1998). We apply this allocation perspective to a cache surrogate and connect it to a static age schedule. The distinction between the ideal solution and the deployed equal-width ladder is part of the contribution's scope.

6 Experiments

The evaluation addresses the research question in two stages: task quality at the reported storage points (Section6.2); sensitivity to floors, role allocation, and storage (Section6.3).

6.1 Experimental Setup

Baselines. We compare our method with representative quantization, eviction, and hybrid compression approaches, using an uncompressed 16bit KV cache (FullKV) as the reference. Baselines cover KIVI (Liu et al., 2024) for quantization, StreamingLLM (Xiao et al., 2024) for eviction, and ThinKV (Ramachandran et al., 2025) for the hybrid setting. We prepare two precision configurations

for TaperKV and TurboTaper, targeting budgets of 4 and 3 average bits per element. TaperKV-4bit steps each key vector bit width from 6 down to 4 bits and its value bit width from 4 down to 2 bits as the token ages. TaperKV-3bit steps keys from 5 down to 3 and values from 3 down to 1. Both keep $S = 4$ sink tokens and the $R = 128$ most recent tokens at the top step, and re-grade the cache every $r = 16$ steps. TurboTaper-4/3bit follows the same ladder schedule but replaces the affine format with a randomized Hadamard rotation and a Lloyd–Max quantizer, grouping both keys and values per token with a single FP16 RMS scale per 32 coordinates. Each compressed method is evaluated at budgets of 4 and 3 average bits per element, with group size $G = 32$ throughout.

Tasks. For mathematical reasoning, we evaluate the sampled zero-shot accuracy on AIME 2024, AIME 2025, and the first 100 problems of MATH-500. We report pass@1, estimated from $k = 8$ samples per question, with answers extracted and checked by Math-Verify 0.9.0. We also evaluate retrieval accuracy with Needle-in-a-Haystack (NIAH)(Kamradt, 2023), placing a synthetic key at five depths across context lengths of 4k, 8k, and 16k tokens and reporting the exact recovery of the corresponding value.

Models. We evaluate various KV cache compression methods on GPT-OSS-20B (OpenAI et al., 2025), DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025). All models perform chain-of-thought reasoning, so their contexts grow much longer than in standard generation. The GPT-OSS-20B model uses the Harmony response format with the reasoning effort set to medium, scoring only the content of the final channel. We cap generation at 32,768 new tokens for all models, following the DeepSeek-R1-Distill published model card. Weights and activations are 16-bit, except that GPT-OSS-20B keeps its native MXFP4 expert weights. GPT-OSS-20B interleaves 128-token sliding-window layers with full attention layers. The sliding-window layers hold far fewer tokens, so we compress only the full-attention layers and report Mem as the ratio to the FullKV cache.

Reading storage and uncertainty. We report memory efficiency (Mem), the encoded KV cache size including metadata, expressed as a fraction of a 16-bit full-precision cache and computed over MATH-500 generation. We use Python 3.12.12, PyTorch 2.11.0 with CUDA 13.0, and Transformers 5.6.2. All experiments are conducted on a single NVIDIA B200 GPU.

6.2 Experimental results

6.2.1 Retrieval

On GPT-OSS-20B, the full cache scores 1.000, and TaperKV retains it at 4 bits and loses one placement at 3 bits for 0.994. KIVI matches the full cache at both budgets, so retrieval on this model does not separate the age schedule from a uniform allocation. On R1-Qwen-7B, the full cache reaches only 0.567, and TaperKV-4bit records the highest observed value at 0.600. TurboTaper scores 0.956 and 0.772 on GPT-OSS-20B and 0.344 and 0.072 on the distill. It shares the ladder of TaperKV and replaces only the quantizer, so the schedule accepts any codec that takes an integer width.

6.2.2 Mathematical reasoning

On GPT-OSS-20B, TaperKV-4bit records the highest observed AIME24 accuracy at 0.788 against 0.758 for the full cache, and matches it on MATH500 at 0.967. On R1-Qwen-7B, TaperKV-4bit stays within one point of the full cache on both AIME years, at 0.521 and 0.417 against 0.533 and 0.421. TurboTaper approaches TaperKV on GPT-OSS-20B but falls to 0.154 and 0.142 on the AIME years for the distill. Since eviction and hybrid configurations repeat until the 32,768-token cap, leaving no answer to extract, their scores are at or near zero on both AIME and NIAH for both models.

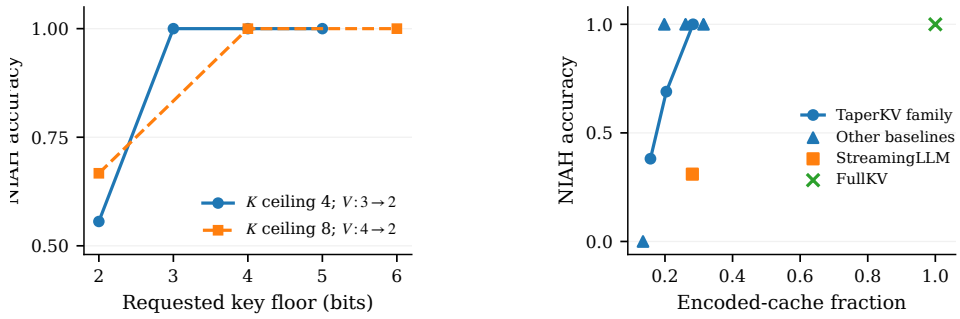
6.3 Sensitivity to floors, role allocation, and storage

6.3.1 Floor sensitivity: purpose and settings.

We test whether retaining tokens is sufficient at aggressive precision and whether increasing the key floor restores quality. On Llama3-8B-Instruct, we vary the requested key floor in two otherwise fixed ladder families. Figure 2a shows the measured retrieval points. These are empirical floor sweeps, not measurements of the worst-case bound in equation 11. Both families lose retrieval accuracy at the lowest tested key floor and recover at higher tested floors. The narrower ladder recovers after a one-bit floor increase; the wider ladder was sampled at two-bit intervals and has no measurement at

Table 1: Task quality at reported storage points. NIAH: retrieval accuracy; AIME24/AIME25/MATH500: pass@1; Mem: encoded-cache fraction including metadata. Bold marks the best compressed value within each bit-width group, not significance; columns with a single measured entry are left unbolded.

GPT-OSS-20B							
Method	Quant	Evict	NIAH↑	AIME24↑	AIME25↑	MATH500↑	Mem↓
FullKV	-	-	1.000	0.758	0.692	0.967	1.000
KIVI-INT4	✓	-	**1.000**	0.754	0.733	0.950	0.382
StreamingLLM-4bit	-	✓	0.200	0.004	0.000	0.225	0.257
ThinKV-4bit	✓	✓	0.000	0.000	0.000	0.321	**0.191**
TaperKV-4bit	✓	-	**1.000**	**0.788**	0.683	**0.967**	0.346
TurboTaper-4bit	✓	-	0.956	0.733	0.683	0.962	0.314
KIVI-INT3	✓	-	**1.000**	**0.750**	**0.688**	0.950	0.316
StreamingLLM-3bit	-	✓	0.200	0.000	0.000	0.217	0.195
ThinKV-3bit	✓	✓	0.000	0.000	0.000	0.354	**0.188**
TaperKV-3bit	✓	-	0.994	0.733	0.679	**0.962**	0.280
TurboTaper-3bit	✓	-	0.772	0.592	0.504	0.912	0.242
DeepSeek-R1-Distill-Qwen-7B							
Method	Quant	Evict	NIAH↑	AIME24↑	AIME25↑	MATH500↑	Mem↓
FullKV	-	-	0.567	0.533	0.421	0.888	1.000
KIVI-INT4	✓	-	0.556	0.512	0.408	**0.938**	0.321
StreamingLLM-4bit	-	✓	0.167	0.000	0.000	0.021	0.250
ThinKV-4bit	✓	✓	0.017	0.000	0.000	0.008	**0.145**
TaperKV-4bit	✓	-	**0.600**	**0.521**	**0.417**	0.908	0.313
TurboTaper-4bit	✓	-	0.344	0.154	0.142	0.733	0.282
KIVI-INT3	✓	-	**0.550**	**0.546**	0.383	0.883	0.258
StreamingLLM-3bit	-	✓	0.144	0.000	0.000	0.021	0.188
ThinKV-3bit	✓	✓	0.011	0.000	0.000	0.000	**0.143**
TaperKV-3bit	✓	-	0.511	0.488	**0.388**	**0.888**	0.251
TurboTaper-3bit	✓	-	0.072	0.013	0.008	0.392	0.219



(a) Key-floor sweep; NIAH $n = 45$.

(b) Separate operating-point run; $n = 42$.

Figure 2: Precision and retention both matter. (a) Raising the key floor improves retrieval in both evaluated ladder families. Points are measured configurations; connecting segments are visual guides, not an inferred threshold. The $K{:}8 \rightarrow b_{\min}$ family has no measurement at floor 3. (b) The reported storage–retrieval points show the cost of aggressive tapering and the low retrieval of the evaluated eviction configuration. This is a separate run from Table 1, not a pooled or exhaustively optimized Pareto frontier.

floor 3. This supports the importance of the floor but does not identify a universal threshold or certify equation 11 from task accuracy alone.

6.3.2 Storage trade-offs: purpose and settings.

Figure 2b uses a separate run to examine progressively more aggressive configurations, alongside the evaluated eviction configuration and the full cache. The plotted points are not a pooled or exhaustively optimized Pareto frontier. Aggressive tapers reduce storage but compromise retrieval quality. The

evaluated eviction configuration remains far below the full cache at a comparable encoded-cache fraction. These are the sampled points of a single run and do not establish an ordering at unsampled fractions.

7 Discussion and limitations

The static schedule answers our research question at the evaluated operating points, not through an end-to-end guarantee. The equal-width ladder is not the logarithmic optimum, the surrogate is not task loss, and fixed-query key selection does not ensure answer recovery. Nonmonotone importance patterns may favor other schedules.

The evaluation uses small retrieval subsets and simulated quantization. Encoded storage does not establish packed-cache residency, throughput, or untested context-length scaling. The position-only rule is compatible with paged attention (Kwon et al., 2023b) and could be integrated into vLLM, which we leave to future work. Baseline adaptations and missing run records limit reproducibility (Appendix B). AIME point estimates remain exploratory; pending protocols are not completed evidence.

8 Conclusion

KV-cache compression need not treat every older token as either fully precise or absent. A floored allocation model favors graded precision, and an independent margin analysis clarifies when key selection is preserved. TaperKV implements the resulting design principle with a fixed age-based ladder, retaining every position without online attention bookkeeping. The reported experiments show that appropriately configured tapers can retain useful retrieval and reasoning performance with smaller encoded caches. The central lesson is to separate *whether information remains represented* from *how precisely it is stored* when designing a compressed cache.

Ethics statement

This work uses existing models and public benchmarks and introduces no new collection of personal data. Cache compression can change model behavior, so deployment requires evaluating capability and safety under the exact configuration served rather than assuming equivalence to an uncompressed model. Lower inference cost can support both beneficial and harmful uses; this paper makes no claim that compression itself improves model safety.

Reproducibility statement

Proofs are in Appendix A; recorded settings and unresolved provenance are in Appendix B. The package contains aggregate results, not the executable harness, per-problem outcomes, or row-to-run manifests. These artifacts and the configuration reconciliations are required for independent experimental reproduction.

LLM usage statement

Large language models assisted with writing, mathematical checks, and document preparation. Reported experimental values were preserved, not independently reproduced. The authors remain responsible for the final proofs, provenance, and claims.

References

Muhammad Adnan, Akhil Arunkumar, Gaurav Jain, Prashant J Nair, Ilya Soloveychik, and Purushotham Kamath. Keyformer: KV cache reduction through key tokens selection for efficient generative inference. In *Proceedings of Machine Learning and Systems*, volume 6, 2024.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, et al. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.

Shichen Dong, Wen Cheng, Jiayu Qin, and Wei Wang. QAQ: Quality adaptive quantization for LLM KV cache. *arXiv preprint arXiv:2403.04643*, 2024.

Haojie Duanmu, Zhihang Yuan, Xiuhong Li, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. SKVQ: Sliding-window key and value cache quantization for large language models. *arXiv preprint arXiv:2405.06219*, 2024.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *International Conference on Learning Representations*, 2024.

Allen Gersho. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, 25(4):373–380, 1979.

Robert M Gray and David L Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44 (6):2325–2383, 1998.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Insu Han, Praneeth Kacham, Amin Karbasi, Vahab Mirrokni, and Amir Zandieh. PolarQuant: Quantizing KV caches with polar transformation. *arXiv preprint arXiv:2502.02617*, 2025.

Yefei He, Luoming Zhang, Weijia Wu, Jing Liu, Hong Zhou, and Bohan Zhuang. ZipCache: Accurate and efficient KV cache quantization with salient token identification. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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. *Advances in Neural Information Processing Systems*, 37, 2024.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023a.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626, New York, NY, USA, 2023b. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Xing Li, Zeyu Xing, Yiming Li, Linping Qu, Hui-Ling Zhen, Wulong Liu, Yiwu Yao, Sinno Jialin Pan, and Mingxuan Yuan. KVTuner: Sensitivity-aware layer-wise mixed precision KV cache quantization for efficient and nearly lossless LLM inference. In *International Conference on Machine Learning*, 2025.

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. *Advances in Neural Information Processing Systems*, 37, 2024.

Tengxuan Liu et al. PM-KVQ: Progressive mixed-precision KV cache quantization for long-CoT LLMs. *arXiv preprint arXiv:2505.18610*, 2025.

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

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. *arXiv preprint arXiv:2402.02750*, 2024.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, et al. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

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

P. F. Panter and W. Dite. Quantization distortion in pulse-count modulation with nonuniform spacing of levels. *Proceedings of the IRE*, 39(1):44–48, 1951.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*, 5, 2023.

Qwen Team. QwQ-32B: Embracing the power of reinforcement learning, 2025. https://qwenlm.github.io/blog/qwq-32b/.

Akshat Ramachandran, Marina Neseem, Charbel Sakr, Rangharajan Venkatesan, Brucek Khailany, and Tushar Krishna. ThinKV: Thought-adaptive KV cache compression for efficient reasoning models. *arXiv preprint arXiv:2510.01290*, 2025.

Claude E Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE National Convention Record*, 4(142–163):1, 1959.

Hanshi Sun, Li-Wen Chang, Wenlei Bao, Size Zheng, Ningxin Zheng, Xin Liu, Harry Dong, Yuejie Chi, and Beidi Chen. ShadowKV: KV cache in shadows for high-throughput long-context LLM inference. *arXiv preprint arXiv:2410.21465*, 2024.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context LLM inference. In *International Conference on Machine Learning*, 2024.

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

June Yong Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, Se Jung Kwon, and Dongsoo Lee. No token left behind: Reliable KV cache compression via importance-aware mixed precision quantization. *arXiv preprint arXiv:2402.18096*, 2024.

Amir Zandieh, Majid Daliri, and Insu Han. QJL: 1-bit quantized JL transform for KV cache quantization with zero overhead. *arXiv preprint arXiv:2406.03482*, 2024.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. TurboQuant: Online vector quantization with near-optimal distortion rate. *arXiv preprint arXiv:2504.19874*, 2025.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2O: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36, 2023.

A PROOFS AND ANALYTICAL SCOPE

This appendix follows the order of the results in Section3. Throughout, $D(b) = \min\{1, \kappa 4^{-b}\}$, $\kappa \geq 1$, and $b_\kappa = \log_4 \kappa$; $\ln$ denotes the natural logarithm. Widths are real in the continuous problem and integer in the discrete problem.

A.1 REVERSE WATER-FILLING WITH A PRESCRIBED FLOOR

Proof of Theorem 3.1. The feasible set is nonempty and compact. On $[b_{\min}, b_{\mathrm{hi}}]$, the objective is $\sum_i w_i \kappa 4^{-b_i}$, which is strictly convex and strictly decreasing in each coordinate. Its minimizer is therefore unique and spends the full budget: any slack would allow an increase of a coordinate below its ceiling. Endpoint budgets fix all coordinates at the corresponding bound.

Define $N(\nu) = \sum_i [\nu + \frac{1}{2}\log_2 w_i]_{b_{\min}}^{b_{\mathrm{hi}}}$. This function is continuous and nondecreasing, equals $n b_{\min}$ for sufficiently small ν, and equals $n b_{\mathrm{hi}}$ for sufficiently large ν. A finite solution to $N(\nu) = B$ exists even at the endpoints. An interior coordinate makes N strictly increasing locally, excluding a second solution; otherwise a plateau can represent the same unique allocation.

For a solution ν, set $\mu = \kappa(\ln 4) 4^{-\nu} > 0$. Each $b_i^\star$ in equation 6 minimizes $w_i \kappa 4^{-b} + \mu b$ on $[b_{\min}, b_{\mathrm{hi}}]$, since its unconstrained stationary point is $\nu + \frac{1}{2}\log_2 w_i$. For every feasible b,

$$J(b) \geq J(b) + \mu\Big(\sum_i b_i - B\Big) \geq J(b^\star) + \mu\Big(\sum_i b_i^\star - B\Big) = J(b^\star). \tag{15}$$

This proves optimality without invoking a constraint qualification at the endpoint budgets. When no coordinate is clipped, $w_i D(b_i^\star) = \kappa 4^{-\nu}$ and $\nu = B/n - \frac{1}{2}\log_2 \mathrm{GM}(w)$, which give equation 7. □

Scope. For $\kappa > 1$, D is flat below b_κ and its derivative jumps downward there. It is not convex on an interval spanning that boundary. Consequently the floored theorem does not characterize every unfloored optimum; any global claim for that problem needs a separate certificate, as in the example in Appendix A.4.

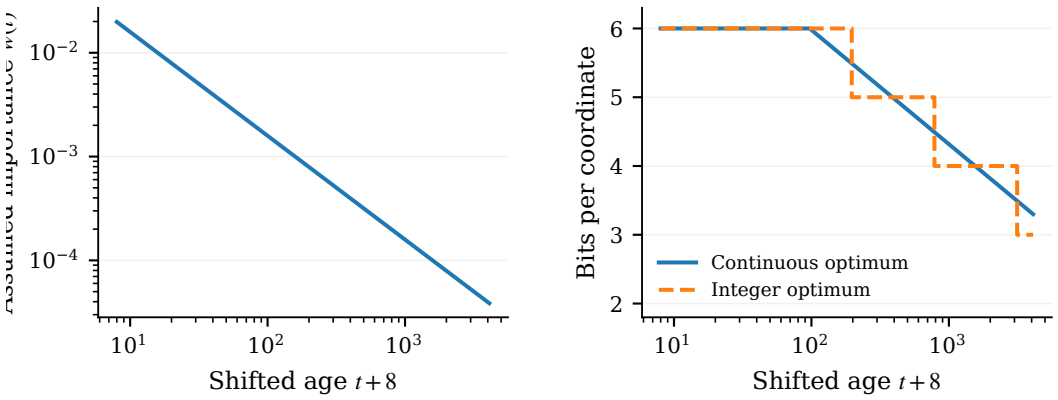
A.2 POWER-LAW WEIGHTS AND INTEGER-BAND GEOMETRY

Proof of Corollary 3.2. For the finite age set, let $Z = \sum_t (t + t_0)^{-\gamma}$. Substituting $w(t) = (t + t_0)^{-\gamma}/Z$ into equation 6 gives equation 8 with $c = \nu - \frac{1}{2}\log_2 Z$. On the unclipped branch, doubling $t + t_0$ subtracts $\gamma/2$ bits.

To specify rounding, fix $0 \leq \theta < 1$ and round x to $\lfloor x + \theta \rfloor$. On a continuous age axis, the transition from integer level b to $b - 1$ occurs at $t_b = 4^{(c+\theta-b)/\gamma} - t_0$, up to tie-breaking at that point. Consecutive transition spacings satisfy

$$t_{b-1} - t_b = 4^{(c+\theta-b)/\gamma}(4^{1/\gamma} - 1). \tag{16}$$

Their ratio is $4^{1/\gamma}$. This concerns complete interior bands before clipping or restricting ages to integers; it is not an exact identity for discrete endpoint bands. □



(a) Assumed importance weights.

(b) Floored optimum and integer reference.

Figure 3: Illustrative allocation, not measured attention. Here $n = 4096$, $t_0 = 8$, $\gamma = 1$, $\kappa = \sqrt{3}\pi/2$, $B/n = 4$, and widths lie in $[2, 6]$. The water level is solved with the bounds in place; both profiles meet the budget. The integer reference uses greedy marginal allocation. Neither profile is the equal-width deployed schedule of Definition 4.1.

A.3 Integer allocation and rounding

Proof of Proposition 3.3. At integer width $b \geq b_{\min}$, the marginal gain $m_i(b) = w_i[D(b) - D(b+1)] = \frac{3}{4} w_i \kappa 4^{-b}$ is positive and decreasing. Every optimum therefore uses exactly $K = \lfloor B \rfloor - n b_{\min}$ increments. Let G be the greedy allocation and O an optimum. If they differ, equal totals give indices i, j with $G_i > O_i$ and $G_j < O_j$. When greedy gave its last increment to i, token j had width $\ell_j \leq G_j \leq O_j - 1$. Hence

$$m_i(G_i - 1) \geq m_j(\ell_j) \geq m_j(O_j - 1). \tag{17}$$

Moving one increment from j to i in O is feasible and changes its objective by $-m_i(O_i) + m_j(O_j - 1) \leq -m_i(G_i - 1) + m_j(O_j - 1) \leq 0$. It reduces $\sum_i |G_i - O_i|$, so finitely many exchanges yield G without increasing the objective. Ties do not affect the argument.

For feasible widths on the exponential branch, $D(\tilde{b}_i)/D(b_i^\star) = 4^{b_i^\star - \tilde{b}_i}$. Losing at most half a bit multiplies distortion by at most 2. Flooring loses less than one bit, preserves the integer lower bound, and cannot increase the budget, giving the factor 4. Nearest rounding need not respect the budget. If the floor equals the ceiling, the single feasible allocation is trivially optimal. □

A.4 Zero-or-high-precision allocations

Proof of Theorem 3.4. Because $D(b_{\text{hi}}) \leq 1$, an optimal zero-or-high allocation can use all $m = \lfloor B/b_{\text{hi}} \rfloor$ available high-width entries. Such allocations are feasible for equation 4, proving $J_0^\star \leq J_{\text{evict}}$. If $D(b_{\text{hi}}) < 1$, an optimal retained set consists of the m largest weights; otherwise all choices tie.

For the stated k, e, δ, replace $(b_k, b_e) = (b_{\text{hi}}, 0)$ with $(b_{\text{hi}} - \delta, \delta)$. The total budget is unchanged and the objective change is

$$w_k[D(b_{\text{hi}} - \delta) - D(b_{\text{hi}})] - w_e[1 - D(\delta)] < 0. \tag{18}$$

This is a feasible strict improvement, as required. □

Analytical example. Let n be divisible by 16, $w_i = 1/n$, $B = 3n$, $b_{\text{hi}} = 16$, and $\kappa = \sqrt{3}\pi/2$. Then

$$J_{\text{evict}} = \frac{13}{16} + \frac{3}{16}\kappa 4^{-16} \approx 0.8125, \qquad J_0^\star = \kappa 4^{-3} \approx 0.04251. \tag{19}$$

For the *unfloored* equality, set $\mu = (\kappa/n)(\ln 4)4^{-3}$. Each scalar Lagrangian is minimized at zero on the flat branch and at 3 on the exponential branch. The latter value, $(\kappa/64n)(1 + 3\ln 4)$, is below $1/n$, the former. Since the budget is tight, Appendix A.1 certifies global optimality. The ratio of about 19.1 concerns this surrogate, not measured accuracy.

A.5 KEY-SELECTION STABILITY AND INFORMATION LOSS

Proof of Theorem 3.5. (a) Let $e_i = \hat{k}_i - k_i$ and $\hat{s}_i = \left\langle q, \hat{k}_i \right\rangle / \sqrt{d}$. Cauchy–Schwarz and the deterministic error assumption give

$$|\hat{s}_i - s_i| \le \frac{\|q\| \, \|e_i\|}{\sqrt{d}} \le \frac{\sqrt{\kappa}\, 2^{-b_{\min}} \, \|q\| \max_j \|k_j\|}{\sqrt{d}} =: \epsilon. \tag{20}$$

Condition equation 11 is $2\epsilon < \Delta$. For the original maximizer i_1 and every $i \neq i_1$, $\hat{s}_{i_1} - \hat{s}_i \ge s_{i_1} - s_i - 2\epsilon \ge \Delta - 2\epsilon > 0$. Hence the maximizer remains unique. This is a fixed-query statement, not a guarantee along a perturbed autoregressive trajectory.

(b) Under the stated post-deletion distribution, independence gives

$$\Pr[f(Z) = W] = \mathbb{E}\big[\Pr(W = f(Z) \mid Z)\big] = 1/M \tag{21}$$

for outputs in the payload alphabet. Outputs outside it can only decrease success. Decoder randomness is already part of Z; no determinism assumption is needed. □

A.6 MONOTONICITY AND FINITE-LENGTH ENCODED STORAGE

Proof of Proposition 4.2. Fix absolute position p. A sink ($p < S$) stays at b_{hi}. Otherwise it is recent until $t = R$, when it enters band zero at the same width. In the interior case, $p \ge S$ and $t \ge R$ imply $T = p + t + 1 > S + R$. Write its normalized age as $\phi_p(T) = 1 - (p + 1 - S)/(T - S - R)$. Then

$$\phi_p(T+1) - \phi_p(T) = \frac{p + 1 - S}{(T - S - R)(T + 1 - S - R)} > 0. \tag{22}$$

The band index is nondecreasing and L is non-increasing. Stored widths, initialized at or above the target, are lowered only to a later target. Delaying updates cannot introduce an increase. There are at most $b_{\text{hi}} - b_{\min}$ strict integer reductions per entry, excluding initial encoding. □

Finite-length identity and proof of Remark 4.4. Set $P = S + R$ and $\bar{b} = (b_{\text{hi}} + b_{\min})/2$. If $T \le P$, every position uses b_{hi}. For $T > P$, let $m = T - P$ and $g = \gcd(m, n_L)$, the greatest common divisor. The *target* widths satisfy the exact identity

$$\frac{1}{T} \sum_{t=0}^{T-1} b(t, T) = \bar{b} + \frac{P(b_{\text{hi}} - b_{\min}) + n_L - g}{2T}. \tag{23}$$

Indeed, the P protected positions use b_{hi} and interior band indices are $\lfloor n_L a/m \rfloor$ for $a = 0, \ldots, m-1$. Pairing a with $m - a$ gives

$$\sum_{a=0}^{m-1} \left\lfloor \frac{n_L a}{m} \right\rfloor = \frac{(n_L - 1)(m - 1) + g - 1}{2} : \tag{24}$$

for $1 \le a < m$, each pair sums to $n_L - 1$, except the $g - 1$ indices for which m divides $n_L a$, where it sums to n_L. Subtracting these indices from $m b_{\text{hi}}$ and adding $P b_{\text{hi}}$ gives equation 23.

Since $1 \le g \le n_L$, the correction is nonnegative and vanishes as $T \to \infty$. Stored nominal widths are at least the current targets. Between successive sweeps, at most r entries are appended. Their bounded widths change the average by $O(r/T)$ for fixed ladder parameters. The same limit therefore holds for stored widths. Dividing the sum of the key and value midpoints by 32 gives the nominal-code fraction of a 16-bit KV cache. Protected-channel extra bits and metadata are outside this calculation. □

A.7 ACCUMULATED ERROR AND THE KEY-SELECTION CERTIFICATE

Proof of Proposition 4.3. Let e_j be the reconstruction increment at visited level j, including initial encoding. Telescoping and the triangle inequality give

$$\|\hat{x}_{\text{prog}} - x\| \le \sum_{j \text{ visited}} \|e_j\| \le C\,\|x\| \sum_{j=b}^{b_{\text{hi}}} 2^{-j} = C2^{-b}\,\|x\|\left(2 - 2^{-(b_{\text{hi}}-b)}\right). \tag{25}$$

The last factor is strictly below 2. If $C\,\|x\| = 0$, every increment vanishes; skipped levels only remove terms. For fixed x, the same proof in the vector-valued L^2 norm replaces each $\|e_j\|$ with $(\mathbb{E}\,\|e_j\|^2)^{1/2}$ and bounds the final root mean squared error. No independence of increments is required. □

Current-input-relative errors. If an encoding at level j changes its current input y by at most $\alpha_j\,\|y\|$, where $\alpha_j = C2^{-j}$, sequentially bounding the input norms gives

$$\|\hat{x}_{\text{prog}} - x\| \le \|x\|\left[\prod_{j \text{ visited}} (1+\alpha_j) - 1\right] \le \|x\|\left(e^{2C2^{-b}} - 1\right). \tag{26}$$

The product bound follows by summing each increment times the product of all preceding norm-growth factors. It is not generally the original-norm bound $2C2^{-b}\,\|x\|$.

Using accumulated error in the floor condition. If Proposition 4.3's *deterministic* increment bound holds for every key with a common $C > 0$, all final widths are at least $b_{\min}$, and the query is fixed, the sufficient condition becomes

$$b_{\min} > \log_2 \frac{4C\,\|q\|\max_i \|k_i\|}{\sqrt{d}\,\Delta}. \tag{27}$$

This follows by substituting the accumulated bound $2C2^{-b_{\min}}\,\|k_i\|$ into the score-error argument. It does not follow from root mean squared error alone. Thus a one-shot codec bound must not be reused unchanged after progressive encoding.

A.8 KEY AND VALUE SENSITIVITY

Proof of Lemma 4.5. The softmax derivative is $\partial a_i/\partial s_j = a_i(\mathbf{1}[i=j] - a_j)$, where $\mathbf{1}[i=j]$ is one when $i = j$ and zero otherwise. Thus

$$\frac{\partial o}{\partial s_j} = a_j(v_j - o), \qquad \delta s_j = \frac{\langle q,\, \delta k_j\rangle}{\sqrt{d}}, \qquad \frac{\partial o}{\partial v_j} = a_j I_d, \tag{28}$$

where I_d is the identity matrix. These derivatives give equation 14. With the base vectors, n, d, q fixed, the attention map has bounded second derivatives on a sufficiently small neighborhood; the remainder's Euclidean norm is therefore bounded by a constant times $\|\delta\|^2$.

The value term has norm at most $\max_i \|\delta v_i\|$. Each key term is bounded by the product of its attention weight, value spread, query norm, and key-error norm, divided by $\sqrt{d}$. Since the value spread can vanish, no universal key-heavy allocation follows. □

B IMPLEMENTATION AND EVALUATION DETAILS

B.1 SCHEDULE AND QUANTIZERS

Schedule. Common ladders and S, R, r appear in Section6.1; ablations override named settings. Prefill encodes at the target width. Algorithm 1 decodes with role-specific targets.

Algorithm 1 TaperKV/ TurboTaper decode step (one layer)

Require: New key, value, and query; both role caches with codes, metadata, and widths; S, R, r and role-specific ladders SectionTATE Append the key and value at their respective ceilings; update length T. SectionTATE Compute attention with the query and reconstructed key and value caches.

1: **if** $T \bmod r = 0$ **then**
2: **for** each role and cached position p **do**
SectionTATE Compute its role-specific target from equation 12, with $t = T - 1 - p$.
3: **if** target width $<$ stored width **then**
SectionTATE Re-encode the current reconstruction; replace codes, metadata, and stored width.
4: **end if**
5: **end for**
6: **end if**

Ensure: All positions retained; no stored nominal width increased.

Affine codec. Each vector uses per-token, per-role affine integer quantization: db code bits for integer b, one fp16 scale, and one fp16 zero-point. Constant-vector handling, rounding, and packing conventions are not specified in executable form.

Rotated codec. The recorded TurboTaper setup uses a randomized Hadamard transform, a shared sign vector per layer, Gaussian Lloyd–Max-based codebooks, and per-token norm scales. Floor protection assigns an extra bit to 32 selected highest-energy channels. The selection domain (per-vector or shared; original or rotated coordinates), refresh rule, and transform padding are unspecified. Extra channel refreshes are not covered by the decreasing-level error bound, nor are these options automatic instances of a one-shot expected-distortion guarantee (Zandieh et al., 2025).

Cost model. A sweep evaluates T targets every r steps: $O(T/r)$ amortized indexing work for fixed ladders, plus dimension- and codec-dependent re-encoding. Proposition 4.2 bounds width reductions, not runtime. Simulated quantization establishes neither packed-cache residency nor kernel speed.

B.2 ENCODED-STORAGE ACCOUNTING

For a layer with h KV heads of dimension d, let $b_{K,i}, b_{V,i}$ be stored *nominal* widths for entry i (absolute position $i-1$), averaged over heads; an omitted entry has width zero. Let $e_i \geq 0$ count its extra protected-channel code bits and $m_i \geq 0$ its metadata bits, each summed over all heads and both roles. Let $m_{\text{shared}} \geq 0$ contain shared overhead. Then

$$\text{mem}(T) = \frac{hd \sum_{i=1}^{T} (b_{K,i} + b_{V,i}) + \sum_{i=1}^{T} (e_i + m_i) + m_{\text{shared}}}{32Thd}. \tag{29}$$

Scales, zero-points, codebooks, rotation state, channel indices, and alignment must be counted. Sum layer numerators and denominators before taking a whole-model ratio. This fixed-width accounting excludes unreported entropy coding or cross-entry deduplication.

Mem is the reported end-of-prompt fraction for fixed-prompt tasks. CR denotes the reported average of instantaneous reciprocals, $1/\text{mem}(T)$, over decode steps; it is not generally the reciprocal of average Mem. Example weighting and sampled decode steps need a run manifest.

Under the stated equal-width schedules, nominal-code lower bounds are $9/32 = 0.28125$ for default TaperKV, $6/32 = 0.1875$ for TaperKV-3bit, and $6.5/32 = 0.203125$ for the recorded TurboTaper-3bit ladder. Equation 23 shows that finite length cannot lower these values; stale higher widths, extra channel bits, and metadata only increase them.