CAPE: COMPRESSED ATTENTION WITH PRESERVED ENTROPY

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning models answer hard problems by generating long chains of thought (CoT), and the KV cache grows with the reasoning length until it becomes the binding memory constraint on inference. KV-cache compression aims to maintain the same model performance while keeping only a bounded number of cached entries. Eviction-based methods have made this practical: scoring cached keys by the attention they receive from recent decode queries retains much of the uncompressed accuracy at a small fraction of the cache. We show that under a reduced cache budget, Softmax attention over the remaining keys is sharper than the distribution the model is calibrated for at its logical length, and the mismatch widens as the trace grows. Another issue is the loss of the objective: because keys are scored by the attention of recent queries, the model becomes less certain of the question as generation length grows under this eviction scheme. We propose CAPE (Compressed Attention with Preserved Entropy), which restores the entropy of compressed attention to its uncompressed level, and reserves a bounded share of the budget for objective anchors selected by context reconstruction. Across three reasoning models on MATH-500 and AIME 24/25 benchmarks, CAPE recovers most of the accuracy lost under compression while saving almost 11 $\times$ memory.

1 INTRODUCTION

Reasoning models generate a long chain-of-thought (Wei et al., 2022; Kojima et al., 2022) before answering (OpenAI, 2024; DeepSeek-AI et al., 2025; Yang et al., 2025). On hard mathematical and scientific problems the trace can run to tens of thousands of tokens: the model plans, backtracks, and verifies rather than emitting an answer in a few steps (Snell et al., 2025; Muennighoff et al., 2025). Because Transformer decoding stores a key–value (KV) pair for every generated token, this style of inference turns accuracy on difficult tasks into a memory problem. The KV cache grows linearly with the reasoning length and quickly dominates GPU memory (Kwon et al., 2023).

A substantial line of work shows that not every past token needs to remain in the cache for generation to continue. StreamingLLM finds that decode-time attention concentrates on a small set of initial *attention sinks* together with a sliding window of recent tokens, and that keeping only those positions is enough to stream far beyond the trained context (Xiao et al., 2023). Eviction methods such as H2O and SnapKV retain a sparse subset of heavy-hitter or recently attended keys, scored from accumulated or local-window attention (Zhang et al., 2023; Li et al., 2024). R-KV targets reasoning traces specifically, arguing that CoT is highly redundant and that a redundancy-aware selection of keys can preserve accuracy at a fraction of the full-cache size (Cai, 2025). Together these results establish that a fixed budget $B \ll L$ can support generation without sacrificing much performance.

Existing compressors primarily optimize which tokens to retain by scoring keys with attention from a local window of recent queries. We find that this design induces two dominant failure modes that are poorly reflected by aggregate accuracy. Under compression, models frequently *lose the objective* (omitting or restating the question) or exhibit *commitment collapse* (verification loops, hedging, or generation that never produces a final answer). Local arithmetic errors occur, but they are rarely the primary failure in our traces and are attributable mainly to base-model capability rather than to the compression mechanism studied here.

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.

We leverage these findings and propose CAPE. CAPE reserves a small, bounded share of the budget for *objective anchors*: prompt tokens selected once at prefill by reconstructing the question, and exempt from eviction thereafter. The rest of the budget is still managed by the KV cache selection scorer. This allows the scorer to safely evict key entries without risking lost of objective. For commitment collapse, CAPE restores the entropy of compressed attention to its uncompressed level by dividing attention logits by a calibrated temperature $T(L, B)$. We show on mathmatical reasoning tasks, CAPE maintains accuracy close to the uncompressed KV cache (FullKV) level.

In summary, our contributions are: **(i)** A diagnosis of long-CoT failure under compression. Rather than reading a single aggregate accuracy drop, we identify and characterize two failure modes: models lose the objective as generation continues, and models exhibit commitment collapse, becoming answer-ready but failing to commit to a final answer. **(ii)** An in depth theoretical analysis of why compression causes commitment collapse. We show that compressing a trace of logical length L into a physical cache of size B leaves Softmax attention more concentrated than the distribution the model is calibrated for. **(iii)** We propose CAPE, a simple method combining Softmax entropy correction with a bounded budget of reconstruction-selected objective anchors, which achieves state-of-the-art results among KV-cache compressors on mathematical reasoning benchmarks.

2 RELATED WORK

KV-cache compression. A large body of work reduces the KV cache by retaining a sparse subset of keys and values. StreamingLLM (Xiao et al., 2023) keeps attention sinks and a recent window, consistent with models' bias toward nearby context (Liu et al., 2024b); H2O (Zhang et al., 2023) and related methods (Liu et al., 2023; Oren et al., 2024) evict by accumulated or current attention; SnapKV (Li et al., 2024) scores keys from a local observation window of recent queries, and follow-ups allocate the budget adaptively across layers and heads (Ge et al., 2024; Cai et al., 2025; Feng et al., 2025). Reconstruction-based methods such as KVzip (Kim et al., 2025), and later KVzap (Jégou & Jeblick, 2026) and FastKVzip (Kim et al., 2026) score keys by their contribution to reconstructing the context, which is query-agnostic, and achieve state-of-the-art performance on retrieval benchmarks (Devoto et al., 2025). Another complementary line of work reduces the cache along the precision axis, quantizing keys and values to low bit-width (Liu et al., 2024c; Hooper et al., 2024), pruning channels (Xu et al., 2025), merging layers (Liu et al., 2024a), learned compression (Nawrot et al., 2024), or shared and latent KV heads (Shazeer, 2019; Ainslie et al., 2023; DeepSeek-AI et al., 2024).

Most KV cache compression methods focus on reducing memory at the prefilling stage; R-KV (Cai, 2025) was the first to target the decoding stage, with repeated eviction events during generation. TriAttention (Mao et al., 2026) follows the same regime but uses pre-RoPE latent geometry for scoring keys.

Attention entropy and length scaling. Our work is inspired by the length extrapolation literature which argues that attention should preserve entropy or scale statistics as the context length changes (Li et al., 2025; Anson et al., 2025; Gao et al., 2025). When a Transformer is evaluated on sequences longer than those used for calibration, Softmax entropy grows approximately as $\log n$ and attention becomes more diffuse, a phenomenon variously described as score dilution or attention fading. Proposed remedies sharpen logits with a length-dependent scale: $\log n$ scaling motivated by Hahn's $O(1/n)$ influence bound (Hahn, 2020; Chiang & Cholak, 2022); entropy-invariant temperatures (Li et al., 2025); YaRN attention temperature (Peng et al., 2023); LogN scaling in Qwen (Bai et al., 2023); and Scalable-Softmax (Nakanishi, 2025). Chen et al. (2025a) identify a critical scaling that interpolates between diffuse (rank-collapse) attention and identity-like attention. Conversely, pathologically low attention entropy is a known training failure mode (Zhai et al., 2023).

We apply the same invariance principle in the opposite setting: the logical trace grows to length L, but Softmax is normalized over a physical key set of size $\approx B$, so a model calibrated to support that grows with L produces attention more concentrated than the corresponding FullKV distribution (§4.2). Closest to our use of entropy scaling in a KV-cache pipeline, AhaKV (Gu et al., 2025b) adjusts Softmax temperature to de-bias accumulated eviction scores, sharpening them toward a sparse top-k distribution. AhaKV does not modify decoding attention; CAPE instead matches the decoding distribution to the uncompressed entropy at logical length L.

3 FAILURE MODES OF LONG COT UNDER FIXED-BUDGET COMPRESSION

Throughout, B denotes the KV-cache budget, Δ the number of decode steps between compression events, L the logical sequence length at a given decode step, and $n_{\text{phys}} \approx B$ the number of physically stored keys. The retained fraction B/L falls as generation proceeds; we refer to this regime as *densification*, since a fixed number of cache slots must represent an ever longer trace. §4.1 specifies the compression loop in full.

We group observed CoT failures under compression into:

- **Lost objective / question drift:** the model forgets or misstates what was asked (wrong quantity, dropped constraints, solving a related but different problem).
- **Commitment collapse:** the model becomes answer-ready but does not commit: it continues verification, hedges, or truncates without emitting a valid \boxed{} answer.
- **Local arithmetic mistakes:** uncommon as the primary break in our traces, and attributable mainly to base-model capability rather than the compression mechanism we target.

Lost objective. During long CoT decoding, the immediate next-token distribution is often determined by the most recent derivation steps or by intermediate conclusions already established in the trace. The original problem statement remains the task specification, but it need not receive high attention at every intermediate step. This separation between local computational demand and global task relevance is problematic for attention-based eviction. Prior work on streaming language models observes that decode-time attention is concentrated on a recent window, together with a small set of attention sinks (Xiao et al., 2023; Gu et al., 2025a). Evictors such as SnapKV and H2O use this recent-window attention as their selection signal, so they preferentially retain tokens with high immediate utility for the current continuation. Question and constraint tokens, although globally relevant, can have low instantaneous utility once the trace has moved into intermediate derivations. As compression is applied every Δ steps, a question token that falls below the top-B set at any event is removed permanently. The resulting cache can therefore lose the task specification while preserving a locally coherent reasoning suffix; subsequent verification then operates on the intermediate trace alone, making objective drift likely.

We verify this with a probe on matched FullKV traces (Figure 1). At selected depths L we record the question keep-rate r_Q, the fraction of original question tokens still in the physical cache, then append "Wait the original problem is:" and measure the perplexity of restating the gold question, which asks whether those tokens remain usable rather than merely stored. The measurements match the account above: once the cache first reaches B, H2O and SnapKV drive r_Q toward zero, and recall perplexity rises with it.

Commitment collapse. Table 8 in Appendix C shows a representative paired trace. Related to overthinking in uncompressed models (Chen et al., 2025b; Wang et al., 2025), we define commitment collapse by two times in the generated trace. Let t_{ans} be the first position at which the benchmark answer normalizer derives a gold answer anywhere in the reasoning text, and let t_{box} be the first position at which a valid boxed answer is emitted. A trace is *answer-ready* if t_{ans} exists. For answer-ready traces, the commitment delay is $t_{\text{box}} - t_{\text{ans}}$.

We find that compression-related commitment collapse is concentrated in the tail rather than appearing as a uniform delay. Most answer-ready traces commit on a similar timescale, but compression increases the fraction that continue through extended verification or never emit a boxed answer. Figure 2 makes this concrete on MATH-500. Panel (A) shows that compression does not delay finding the answer: the answer-ready survival curves coincide out to ≈3k tokens. However, it leaves more traces never answer-ready. Panel (B) shows the same tail-only signature for committing: the median commitment delay is similar under compression, and the methods separate in the tail of the delay distribution, where SnapKV and TriAttention leave three to four times as much mass that never emits a boxed-answer as FullKV. Following the theory of entropy-invariant attention scaling (Li et al., 2025; Gu et al., 2025b), we hypothesize that softmax entropy mismatch in attention operation due to KV compression is one cause of these failures, and thus run a causal intervention: FullKV with no keys dropped, but Softmax temperature T$<$1 chosen so the entropy deficit matches the cache densification L/B (FullKV+deficit in Figure 2). We observe a similar tail behavior with a matched entropy deficit to FullKV, which motivates our entropy temperature calibration.

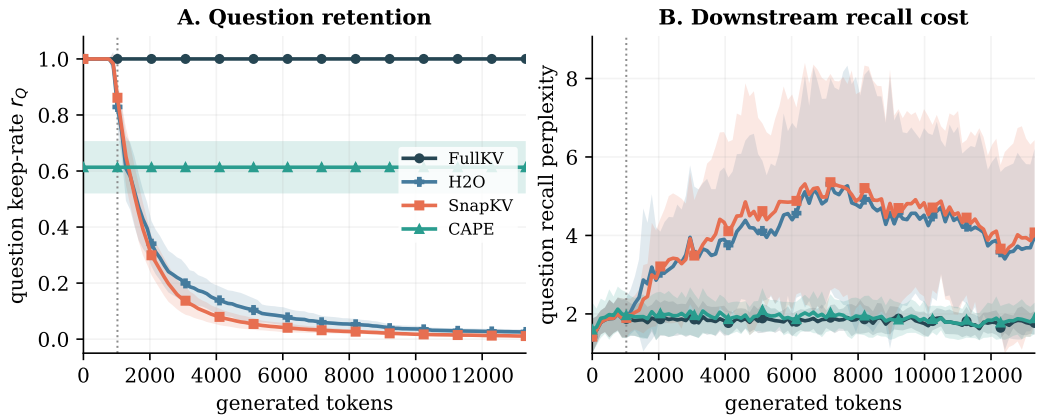

Figure 1: Question loss under decode-time compression on AIME 24 (Qwen3-8B, B=1024, Δ=128, mean±std over n=5 traces with CoT length $\geq$ 10k; the dotted line marks the step at which the cache first reaches B). **A**: keep-rate r_Q of original question tokens remaining in the physical cache. **B**: perplexity of restating the gold question after the probe "Wait the original problem is:". H2O and SnapKV drive r_Q toward zero as L grows, and recall perplexity rises with it; FullKV and CAPE are shown for comparison.

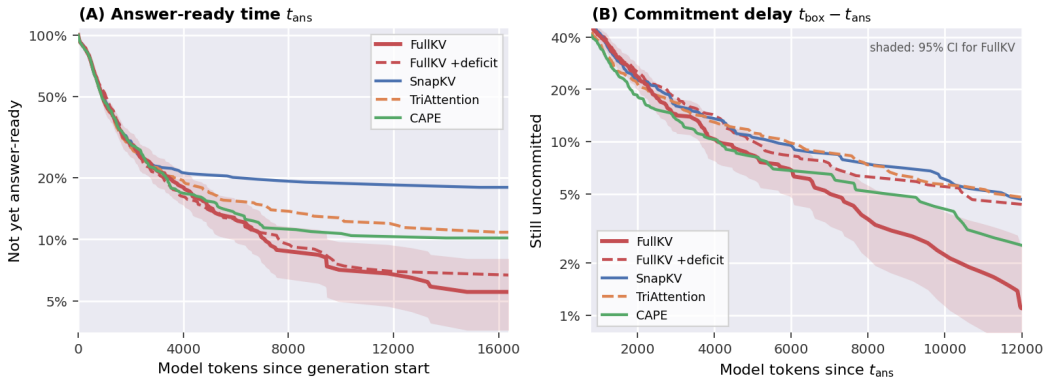

Figure 2: Commitment analysis on MATH-500 (R1-Llama-8B, B=512). **(A)** Kaplan–Meier (Kaplan & Meier, 1958) survival curve of the not yet answer-ready fraction against model tokens elapsed since generation start. **(B)** Kaplan–Meier survival of the uncommitted fraction against model tokens elapsed since t_{ans}.

4 CAPE: COMPRESSED ATTENTION WITH PRESERVED ENTROPY

Based on the analysis in §3, we propose CAPE, a decode-time KV compression method that preserves both the task objective and the attention entropy scale. We first summarize the fixed-budget decode-time setting and local-window importance estimator (§4.1). CAPE then adds two components: a logical–physical temperature for normalizing attention over the retained keys (§4.2), and objective anchors selected at prefill by reconstruction scoring and held outside the moving scorer (§4.3).

4.1 DECODE-TIME COMPRESSION AND LOCAL-WINDOW SCORING

We follow the fixed-budget decode-time inference setting used by prior work (Cai, 2025). A request consists of a prefill of P tokens followed by autoregressive decoding; every Δ decode steps, if the cache exceeds the budget B, the method scores cached entries and gathers the top-B keys and values. We state the notation for grouped-query attention (GQA) (Ainslie et al., 2023), with multi-

head attention as the special case $G = 1$. Let H be the number of key/value head groups and G the number of query heads per group. For head group h, the cache contains keys and values $K^h, V^h \in \mathbb{R}^{n\times d}$ shared by query heads $g \in [0, G)$, where n is the current number of cached entries and d is the head dimension. The physical cache size n_{phys} equals B after a gather and grows to at most $B + \Delta - 1$ before the next one, while the logical sequence length L continues to increase.

Attention-based compressors (Li et al., 2024; Cai, 2025) estimate I^h from the attention weights assigned by recent decode queries to cached keys, using recent-query demand as a proxy for future utility. CAPE inherits this estimator unchanged for the part of the budget it allocates to the reasoning trajectory. Let W be the observation-window size and let $Q^{h,g} \in \mathbb{R}^{W\times d}$ collect the last W RoPE-rotated (Su et al., 2021) decode queries for query head g in group h. The window attention over cached keys is

$$A^{h,g} = \text{softmax}\Big(Q^{h,g}\,(K^h)^\top/\sqrt{d} + M\Big) \in \mathbb{R}^{W\times n}, \tag{1}$$

with M the causal mask over the window positions. Following the stabilization step used in prior attention-based scoring (Li et al., 2024; Cai, 2025), we average over the observation queries, apply a sliding max-pool over key positions, and then average across the G query heads that share the same KV cache, giving the importance score

$$a_i^{h,g} = \frac{1}{W}\sum_{j=0}^{W-1} A_{j,i}^{h,g}, \qquad \tilde{a}_i^{h,g} = \max_{|\delta|\le r} a_{i+\delta}^{h,g}, \qquad I_i^h = \frac{1}{G}\sum_{g=0}^{G-1} \tilde{a}_i^{h,g}. \tag{2}$$

Finally the W observation-window positions are assigned $\max_{i'} I_{i'}^h + 1$ so that the window itself is never evicted.

4.2 LOGICAL–PHYSICAL TEMPERATURE

Local-window scoring changes which keys are stored; it does not change how the stored keys are normalized. Compressing to B entries while the trace grows to logical length L leaves Softmax normalized over a support much smaller than the one the model operates on uncompressed. We treat this as a calibration problem: matching the entropy of compressed attention to its uncompressed operating point yields a temperature on the attention logits that depends only on L and the physical cache size, which we derive below.

Entropy of scaled attention. The scaled dot-product of Vaswani et al. (2017) already introduces a temperature $1/\sqrt{d}$ in order to keep Softmax away from saturation as the head dimension grows. A second scale is the number of keys n. Let $p = \text{softmax}(z)$ denote attention of a query against n keys, with $p_j = e^{z_j}/Z$ and $Z = \sum_{i=1}^n e^{z_i}$. The Shannon entropy (Cover & Thomas, 2006) is $H(p) = -\sum_j p_j \log p_j$, so that $H(p) = 0$ exactly when p is a point mass and $H(p) = \log n$ exactly when p is uniform; in particular $H(p) \le \log n$, a bound we return to below. Substituting $\log p_j = z_j - \log Z$ yields the identity

$$H(p) = \log \sum_{i=1}^n e^{z_i} - \sum_{j=1}^n p_j z_j, \tag{3}$$

which holds for arbitrary logits. The first term increases with n; the second is the p-weighted mean logit.

To obtain the dependence of H on n, assume that $z_1, \dots, z_n$ are i.i.d. copies of a random variable z with finite $\mathbb{E}[e^z]$ and $\mathbb{E}[ze^z]$, following the analysis of Li et al. (2025) and Gu et al. (2025b). Applying the law of large numbers to both terms of equation 3 (Appendix A.1) gives

$$H \;\longrightarrow\; \log n + \underbrace{\log \mathbb{E}[e^z] - \frac{\mathbb{E}[ze^z]}{\mathbb{E}[e^z]}}_{-C}, \qquad \text{i.e.} \qquad H \;\approx\; \log n - C. \tag{4}$$

The equation 4 is the basis of entropy-invariant attention scaling (Li et al., 2025; Gu et al., 2025b): holding H fixed as n varies requires a temperature that depends on n. InfoScale applies this principle to length extrapolation, where the key set is larger than at training, entropy is inflated, and logits are sharpened. AhaKV applies it to eviction scores, again sharpening toward a sparse top-k distribution. We apply the same dependence to decoding Softmax, in the opposite direction.

Fixed-budget compression as the converse of length extrapolation. Length-extrapolation methods operate in the regime $n_{\text{test}} > n_{\text{train}}$. equation 4 then predicts an entropy surplus of approximately $\log(n_{\text{test}}/n_{\text{train}})$, which is corrected by multiplying logits by a factor of order $\log n$ (§2). Fixed-budget compression is the converse. At logical length L, the operating entropy of the uncompressed model is $H^\star \approx \log L - C$. Compressed Softmax, however, is taken over $n_{\text{phys}} \approx B$ keys and yields $H_{\text{comp}} \approx \log B - C$. The difference

$$\Delta H \;\approx\; \log \tfrac{L}{B} \tag{5}$$

is an entropy *deficit*: the compressed distribution is more concentrated than the uncompressed distribution at the same decode step. Appendix A.1 decomposes this gap exactly for eviction without logit changes.

Entropy-matching temperature. To bridge the entropy gap, one simple solution is by dividing logits by $T > 1$. Assume logits are Gaussian distribution $z \sim \mathcal{N}(\mu, \sigma^2)$, for which equation 4 evaluates to $C = \sigma^2/2$; rescaling $z \mapsto z/T$ sends $\sigma \mapsto \sigma/T$ and hence $C \mapsto C/T^2$. Equating $H_{\text{comp}}(T) = \log B - C/T^2$ with the uncompressed operating point $H^\star = \log L - C$ yields

$$\log B - \frac{C}{T^2} = \log L - C \qquad \Rightarrow \qquad T^\star \;=\; \Big(1 - \tfrac{\log(L/B)}{C}\Big)^{-1/2}, \tag{6}$$

which is defined if and only if $C > \log(L/B)$. Then to ensure entropy of compressed attention remains stable, we follow Bai et al. (2023); Su (2021) and fit C at the compressed budget B, so $C = \kappa \log B$, where κ is a constant parameter. This gives, together with its first-order form used for implementation ($\beta = \frac{1}{2\kappa}$),

$$T^\star = \Big(1 - \frac{1}{\kappa}\Big(\frac{\log L}{\log B} - 1\Big)\Big)^{-1/2} \;\approx\; 1 + \beta\Big(\frac{\log L}{\log B} - 1\Big).$$

With $n_{\text{phys}} \approx B$, the temperature CAPE applies at each decode step is therefore

$$T = 1 + \beta\Big(\tfrac{\log L}{\log n_{\text{phys}}} - 1\Big). \tag{7}$$

Therefore $T = 1$ when $L = n_{phys}$, and grows monotonically with L.

4.3 OBJECTIVE ANCHORS FROM CONTEXT RECONSTRUCTION

We now address the objective drift described in §3. An intuitive remedy is to exclude the problem statement from eviction altogether. This guarantees $r_Q = 1$, but it spends the budget indiscriminately: chat template, boilerplate, and redundant phrasing occupy slots on the same terms as the constraints that define the task. The cost is not marginal at practical budgets. With the common setting $B = 512$, a MATH-500 prompt together with its chat template can claim a substantial fraction of the physical cache, so preserving the whole question increases the compression pressure on rest of the CoT decoding. CAPE therefore reserves a bounded share of the budget for the question, filled by context reconstruction. It partitions the budget into k anchor slots, selected once at the end of prefill and held fixed, and $B - k$ slots managed by the estimator of §4.1.

Query-agnostic scoring. Reconstruction-based KV compressors score keys by their contribution to repeat the prompt. Kim et al. (2025), Jégou & Jeblick (2026) and Kim et al. (2026) attain high compression ratios with near-lossless quality, and outperform eviction-based methods by a large margin. Their scoring procedures, however, require extra forward passes over the context, which is expensive when the context is long. In the long-CoT settings, the problem statement is typically short while the reasoning trace is long, so the cost of reconstruction at prefill is negligible relative to decoding. We therefore ask which keys are required to reproduce the problem statement, rather than which keys the current query attends to. Following KVzip (Kim et al., 2025), this is measured by a reconstruction pass: after prefill we run one additional forward pass whose input is a fixed probe ("Repeat the question exactly.") followed by the question tokens themselves, and read off the cross-attention that the replayed tokens direct onto the cached prefill keys. Because both the probe and the replayed content are determined by the prompt, the resulting score is a function of the prompt only, and is identical at every subsequent decode step.

Let $\tilde{Q}^{\ell,h,g} \in \mathbb{R}^{R\times d}$ be the queries of the reconstruction pass at layer ℓ, where R is the combined length of probe and replay, and let $\tilde{A}^{\ell,h,g} \in \mathbb{R}^{R\times P}$ denote the corresponding attention weights restricted to the P prefilled key columns. The anchor score of key i is

$$s_i^{\ell,h} = \max_{g\in[0,G)} \max_{j\in[0,R)} \tilde{A}_{j,i}^{\ell,h,g}, \tag{8}$$

To accommodate different budgets, we design an adaptive anchor budget:

$$k(P,B) = \begin{cases} P, & P < P_0, \\ \min\big(P,\ \mathrm{clip}(\lfloor \rho P \rfloor,\ k_{\min},\ \lfloor \gamma B \rfloor)\big), & \text{otherwise}, \end{cases} \tag{9}$$

so that short prompts are retained whole, longer prompts are compressed to a fraction ρ of their length, and the cap γ prevents the anchors from starving the decoding trajectory. We use $P_0 = k_{\min} = 64$ and $\rho = \gamma = \frac{1}{2}$ throughout. The first $n_{\text{sink}} = 4$ positions are forced into the anchor set as attention sinks (Xiao et al., 2023). The anchor set of layer ℓ and head group h is $\mathcal{A}^{\ell,h} = \text{top-}k\,(s^{\ell,h})$, and the prefill cache is gathered onto it, so decoding starts from a physical cache of length k rather than P. The count k is shared by all layers and head groups, preserving the dense layout of §4.1.

The gathered anchors are then held through decoding by a pin mask $\pi^h \in \{0,1\}^n$, set to 1 on the reconstruction-selected slots and to 0 on every entry appended later. At each compression event the local-window score is overridden on pinned entries,

$$\hat{I}_i^h = \pi_i^h \Big(\max_j I_j^h + 1 \Big) + \big(1 - \pi_i^h\big)\, I_i^h, \tag{10}$$

which guarantees the anchors survive each top-B selection, while the remaining $B - k$ slots are ranked by the unmodified local-window score.

5 Experiments

5.1 Setup

We evaluate three reasoning models—Qwen3-8B, DeepSeek-R1-Distill-Llama-8B, and DeepSeek-R1-Distill-Qwen-14B on MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023) and AIME 24/25 under decode-time compression with budget B and compression interval Δ. We compare to FullKV and SnapKV (Li et al., 2024), R-KV (Cai, 2025) and TriAttention (Mao et al., 2026). All runs share a matched prompt and generation-length limit, default to B=512, Δ=128 on MATH-500; B=2048 on AIME unless otherwise stated.

5.2 Main results

In Table 1, CAPE matches or exceeds SnapKV, R-KV, and TriAttention on all three datasets, with the single exception of Qwen3-8B AIME25, where it ties the next-best baseline TriAttention. On R1-Qwen-14B, for instance, CAPE leads the next best TriAttention on all three suites, with SnapKV and R-KV trailing both. Figure 3 sweeps the cache budget on Qwen3-8B: the gain of CAPE is largest at small B and shrinks as the budget approaches the FullKV operating point, consistent with both mechanisms of §3 intensifying with the ratio L/B. On AIME, where traces are longer, CAPE matches the FullKV baseline on AIME24 and exceeds it on AIME25 by B=4k.

5.3 Commitment collapse

With t_{ans} and t_{box} as defined in §3, we measure three diagnostics on answer-ready traces: mean observed commitment delay $t_{\text{box}} - t_{\text{ans}}$, the conditional no-box rate, and the number of *reopen episodes*—self-interruption markers such as "Wait" (Muennighoff et al., 2025) or "Let me check" (full list in Table 7, Appendix C) between t_{ans} and the first boxed answer or trace end. Table 2 reports these on R1-Llama-8B MATH-500 at B=512.

Compressed methods become answer-ready less often than FullKV and among answer-ready traces, SnapKV and TriAttention have a higher mean observed delay and no-box rate. Longer delays are

Table 1: Accuracy on MATH-500 and AIME24/25 across three models. MATH-500 accuracies are averaged over 5 runs and AIME24/25 accuracies are averaged over 8 runs. Best results are in **bold**.

Model	Method	MATH-500 Acc.	AIME24 Acc.	AIME25 Acc.
Qwen3-8B	FullKV	89.2	66.7	63.3
	SnapKV	60.8	43.3	16.7
	R-KV	46.4	25.4	17.5
	TriAttention	67.6	43.3	**50.0**
	CAPE	**77.0**	**56.7**	**50.0**
R1-Llama-8B	FullKV	88.6	60.0	33.3
	SnapKV	68.6	30.0	26.7
	R-KV	76.9	25.8	11.3
	TriAttention	80.6	46.7	26.7
	CAPE	**83.0**	**52.1**	**33.3**
R1-Qwen-14B	FullKV	92.2	76.7	53.3
	SnapKV	72.4	50.0	40.0
	R-KV	82.2	52.1	32.1
	TriAttention	85.4	70.8	41.3
	CAPE	**87.8**	**73.3**	**50.0**

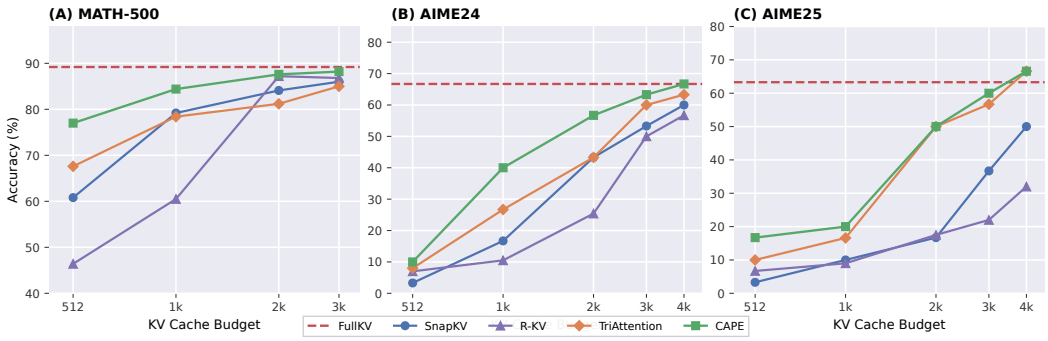

Figure 3: Accuracy vs. KV-cache budget on Qwen3-8B.

associated with more reopening, e.g. SnapKV averages 15.6 reopen episodes per answer-ready trace against FullKV's 11.4. Median episode counts are nearly identical (3 vs. 3), so the gap is a tail of traces trapped in verification loops rather than a uniform reluctance to commit; the survival curves of Figure 2B show the same shape.

To isolate Softmax densification from token eviction, we apply a matched entropy deficit to FullKV by empirically calibrating a Softmax temperature $T<1$ that matches the compressed entropy level (FullKV+deficit in Table 2). Mean observed delay rises to 1896 tokens, no-box to 4.1%, and reopens to 14.8. In this setting, an entropy deficit without eviction is sufficient to increase these commitment-failure metrics. CAPE has 437 answer-ready traces, with mean observed delay of 1521 tokens, a 3.2% no-box rate, and 11.6 reopen episodes per trace, close to the FullKV values; this shows that restoring Softmax entropy helps recover the commitment-failure cases. TriAttention improves the no-box rate (3.9%) but not the delay and reopen counts, which remains close to SnapKV's (1921 vs. 1956; 16.3 vs. 15.6).

5.4 EFFICIENCY

Accuracy is only half of the deployment story: a compression method must bound KV memory, sustain throughput under concurrency, and not inflate generation length by much. We profile all five methods at the same setting summarized in Table 3. For memory, at a matched budget every compressed method stores the same B keys and values, so KV memory is identical across them by

Table 2: Commitment diagnostics on answer-ready MATH-500 traces (R1-Llama-8B, B=512). Mean observed delay is in model tokens; no-box traces contribute their observed follow-up from t_{ans} to trace end, capped at the generation limit. This mean is descriptive and is not adjusted for censoring. FullKV+deficit applies an entropy-matched Softmax temperature with no eviction.

Method	n_{ready}	Delay (tok.)	No-box (%)	Reopens
FullKV	459	1564	1.1	11.4
FullKV+deficit	461	1896	4.1	14.8
SnapKV	397	1956	4.3	15.6
TriAttention	434	1921	3.9	16.3
CAPE	437	1521	3.2	11.6

Table 3: Efficiency at a matched budget (Qwen3-8B, B=512, Δ=128). Throughput follows the max-batch protocol of Cai (2025): batch scaled to the largest that fits one H100-80GB.

	FullKV	SnapKV	R-KV	TriAttention	CAPE
Throughput (max batch, 8k fixed generation)					
Tokens/s	893	6549	6259	6833	5971
Speedup vs. FullKV	1.00×	7.34×	7.01×	**7.65**×	6.69×
Generation (MATH-500)					
Tokens generated (M)	2.230	3.181	4.299	4.058	2.996
vs. FullKV	1.00×	1.43×	1.93×	1.82×	**1.34**×
Single-stream wall time (whole dataset)					
Hours	17.48	25.74	36.18	33.21	26.95
vs. FullKV	1.00×	1.54×	2.07×	1.90×	**1.47**×
MATH-500 Acc. (%)	89.2	60.8	46.4	67.6	**77.0**

construction; on Qwen3-8B at B=512 this lets them fit roughly 11× more sequences on one H100 than FullKV (Appendix A.2).

Throughput and generation length. Under the max-batch protocol every compressed method is much faster than FullKV, and CAPE is last among them due to extra compute from context reconstruction. Compression makes the model reason longer. This inflation is a general symptom of a shorter memory budget: with only a bounded window of history reliably in cache, the model cannot trust the compressed context to remember what it already established, so it repeatedly brings that content back into the recency window (restating the problem, re-deriving intermediate steps, etc). R-KV needs 1.93× as many tokens as FullKV to complete MATH-500, TriAttention 1.82×, SnapKV 1.43×, and CAPE only 1.34×. Therefore, although CAPE has the slowest measured throughput, its single-stream wall time on MATH-500 is the best among the compressed methods.

5.5 ABLATIONS

We ablate the question anchors and the entropy temperature in a 2×2 design on Qwen3-8B at B=512 (Table 4; β sensitivity in Table 6; Appendix B). On MATH-500 / AIME25, anchors alone lift the SnapKV baseline from 60.8 / 16.7 to 72.6 / 33.3, the temperature alone to 68.8 / 26.7, and both together to 77.0 / 43.3. Pinning a random set of k prefill slots already recovers part of the gap, while reconstruction scoring (77.0) matches pinning the full question (75.6) with half the slots (Table 5, Appendix B).

6 CONCLUSION

In this work, we found KV compression induces two principal issues in long CoT decoding: Softmax entropy mismatch between physical cache size B and logical length L; and lost objectives, driven by local-window scoring bias. By matching compressed attention entropy to the uncompressed operating point and fixing an anchored budget for the question, CAPE recovers most of the FullKV accuracy in the mathematical reasoning benchmarks.

AI USE STATEMENT

In this work, we used generative AI tools for drafting and polishing writing, assisting in the writing of proofs, providing feedback on research experiments. We have not used generative AI tools for helping develop theoretical models, and generating synthetic datasets, cleaning and reformating dataset are not applicable to this work. Additionally, we used generative AI tools for modifying scientific figures and editing code. We have reviewed all AI-assisted work. LLM-generated code was verified and tested, experimental results are checked and discussed with both authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies efficiency methods for language-model inference on public math benchmarks. We do not involve human subjects or release new sensitive datasets.

REPRODUCIBILITY STATEMENT

We will release configs, scripts, and evaluation code used for MATH-500 / AIME experiments and document hyperparameters in the appendix.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Ben Anson, Xi Wang, and Laurence Aitchison. Scale-invariant attention. *arXiv preprint arXiv:2505.17083*, 2025.

Jinze Bai, Shuai Bai, Yunfei Chu, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. LogN attention scaling for context extension.

others Cai. R-KV: Redundancy-aware KV cache compression for reasoning models. *arXiv preprint arXiv:2505.24133*, 2025. Update authors from camera PDF.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *Conference on Language Modeling (COLM)*, 2025.

Shi Chen, Zhengyu Lin, Yury Polyanskiy, and Philippe Rigollet. Critical attention scaling in long-context transformers. *arXiv preprint arXiv:2510.05554*, 2025a.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? on the overthinking of long reasoning models. In *International Conference on Machine Learning (ICML)*, pp. 9487–9499, 2025b. arXiv:2412.21187.

David Chiang and Peter Cholak. Overcoming a theoretical limitation of self-attention. In *ACL*, 2022.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, et al. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Alessio Devoto, Maximilian Jeblick, and Simon Jégou. Expected attention: KV cache compression by estimating attention from future queries distribution. *arXiv preprint arXiv:2510.00636*, 2025. URL https://arxiv.org/abs/2510.00636.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Bo Gao, Michael W. Spratling, and Letizia Gionfrida. Softplus attention with re-weighting boosts length extrapolation in large language models. *arXiv preprint arXiv:2501.13428*, 2025.

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 (ICLR)*, 2024.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *International Conference on Learning Representations (ICLR)*, 2025a.

Yifeng Gu, Zicong Jiang, Jianxiu Jin, Kailing Guo, Ziyang Zhang, and Xiangmin Xu. AhaKV: Adaptive holistic attention-driven KV cache eviction for efficient inference of large language models. *arXiv preprint arXiv:2506.03762*, 2025b.

Michael Hahn. Theoretical limitations of self-attention in neural sequence models. *TACL*, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Simon Jégou and Maximilian Jeblick. KVzap: Fast, adaptive, and faithful KV cache pruning. *arXiv preprint arXiv:2601.07891*, 2026.

Edward L. Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481, 1958.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W. Lee, Sangdoo Yun, and Hyun Oh Song. KVzip: Query-agnostic KV cache compression with context reconstruction. *arXiv preprint arXiv:2505.23416*, 2025.

Jang-Hyun Kim, Dongyoon Han, and Sangdoo Yun. Fast KVzip: Efficient and accurate LLM inference with gated KV eviction. *arXiv preprint arXiv:2601.17668*, 2026.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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. In *ACM Symposium on Operating Systems Principles (SOSP)*, 2023.

Kewei Li, Yanwen Kong, Yiping Xu, Jianlin Su, Lan Huang, Ruochi Zhang, and Fengfeng Zhou. Information entropy invariance: Enhancing length extrapolation in attention mechanisms. *arXiv preprint arXiv:2501.08570*, 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. *arXiv preprint arXiv:2404.14469*, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. MATH-500 subset.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. MiniCache: KV cache compression in depth dimension for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics (TACL)*, 12:157–173, 2024b.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 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. In *International Conference on Machine Learning (ICML)*, 2024c.

Weian Mao, Xi Lin, Wei Huang, Yuxin Xie, Tianfu Fu, Bohan Zhuang, Song Han, and Yukang Chen. TriAttention: Efficient long reasoning with trigonometric KV compression. *arXiv preprint arXiv:2604.04921*, 2026.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 20275–20321, 2025.

Ken M. Nakanishi. Scalable-softmax is superior for attention. *arXiv preprint arXiv:2501.19399*, 2025.

Piotr Nawrot, Adrian Łańcucki, Marcin Chochowski, David Tarjan, and Edoardo M. Ponti. Dynamic memory compression: Retrofitting LLMs for accelerated inference. In *International Conference on Machine Learning (ICML)*, pp. 37396–37412, 2024.

OpenAI. OpenAI o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. Transformers are multi-state RNNs. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 18724–18741, 2024.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2408.03314.

Jianlin Su. Understanding the scaling operation in attention from the perspective of entropy invariance. `https://kexue.fm/archives/8823`, Dec 2021.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Thoughts are all over the place: On the underthinking of long reasoning models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2501.18585.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Yuhui Xu, Zhanming Jie, Hanze Dong, Lei Wang, Xudong Lu, Aojun Zhou, Amrita Saha, Caiming Xiong, and Doyen Sahoo. ThinK: Thinner key cache by query-driven pruning. In *International Conference on Learning Representations (ICLR)*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M. Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *ICML*, 2023.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. *arXiv preprint arXiv:2306.14048*, 2023.

A Additional Details

A.1 Entropy scaling with the number of keys

This subsection gives the steps behind equation 4 and the exact entropy gap under eviction referenced in §4.2. Under the i.i.d. assumption of §4.2, the law of large numbers gives

$$\log \sum_{i=1}^{n} e^{z_i} = \log n + \log\left(\tfrac{1}{n}\sum_{i} e^{z_i}\right) \longrightarrow \log n + \log \mathbb{E}[e^z],$$

and

$$\sum_j p_j z_j = \frac{\frac{1}{n}\sum_j z_j e^{z_j}}{\frac{1}{n}\sum_i e^{z_i}} \longrightarrow \frac{\mathbb{E}[z e^z]}{\mathbb{E}[e^z]}.$$

Substituting both limits into equation 3 yields equation 4.

Exact gap under eviction. Let p be the FullKV attention distribution and S the retained set, with $|S| = B$ and retained mass $p_S = \sum_{j\in S} p_j$. If keys are dropped without altering the remaining logits, compressed attention is the renormalization of p onto S. The grouping identity for Shannon entropy (Cover & Thomas, 2006) then gives

$$H_{\text{full}} - H_{\text{comp}} = h(p_S) + (1 - p_S)\big(H_{S^c} - H_S\big), \tag{11}$$

where h denotes binary entropy and H_S, H_{S^c} are the entropies of p restricted to S and its complement. The discarded support is large (H_{S^c} is at most $\log(L - B)$) while $H_S \le \log B$, so the gap increases as p_S decreases. The temperature correction therefore targets H_S, fixing how concentrated attention is on the keys that survive.

A.2 Memory accounting

KV memory is set by the physical cache length, so at a matched budget every compressed method stores the same keys and values by construction. For Qwen3-8B (36 layers, 8 KV heads, d=128, bfloat16) a cached token costs 144 KiB, so all compressed methods hold 0.075 GB at B=512 against 1.24 GB for FullKV at the median completed trace ($L \approx 8.4$k). The decode-time scorer buffers Δ steps of hidden states, adding 0.038 GB per sequence for every method built on it (FullKV runs no scorer), so a sequence costs 0.113 GB under CAPE against the same 1.24 GB under FullKV once that state is counted. With this saving the compressed methods fit roughly $11\times$ more sequences on a single H100 GPU than FullKV.

B Extended Results

Table 4: 2×2 ablation of the two CAPE components (Qwen3-8B, $B = 512$). Pin = reconstruction anchors of §4.3; Temp = logical–physical temperature of §4.2.

Pin	Temp	MATH-500 Acc.	AIME25 Acc.
–	–	60.8	16.7
✓	–	72.6	33.3
–	✓	68.8	26.7
✓	✓	**77.0**	**43.3**

Anchor selector controls. A random-anchor pin isolates the anchor selector of §4.3 from the act of pinning itself: it pins a uniformly random set of k prefill slots, at the k given by equation 9. Table 5 shows that pinning anything recovers part of the gap: even a content-blind random set improves over pinning nothing, so part of the benefit is the act of exempting prefill slots from the moving scorer rather than the choice of which ones. Reconstruction scoring reaches 77.0 using k anchors, against 75.6 when every prompt token is pinned. So it matches full-question pinning at half the slots, leaving the remaining budget to the long reasoning trajectory. Recall perplexity behaves accordingly: pinning the whole question is the most faithful (1.9) and reconstruction is close behind (2.4).

Table 5: Anchor selector controls (Qwen3-8B, $B = 512$), all with the temperature enabled and the anchor count k from equation 9. Recall PPL is the perplexity of restating the question (§3).

Anchor selection	MATH-500 Acc.	Recall PPL
None	68.8	*7.2*
Random k	*71.4*	*4.7*
Full question	*75.6*	*1.9*
Reconstruction (CAPE)	**77.0**	*2.4*

Table 6: Sensitivity to the temperature strength β of equation 7 on MATH-500 (Qwen3-8B, B=512, Δ=128). The bold row is the default setting.

β	MATH-500 Acc.
0	72.6
0.25	74.0
0.5	**77.0**
1.0	74.0
2.0	61.6

C FAILURE TAXONOMY DETAILS

Reopen markers. Table 7 lists some example self-interruption markers counted as reopen episodes in §5.3. Markers are matched case-sensitively as whole words in the text between t_{ans} and the first valid \boxed{} answer (or trace end if none is emitted), and every non-overlapping occurrence counts as one episode.

Table 7: Self-interruption markers counted as reopen episodes (case-sensitive, whole-word match).

Category	Markers
Interjection	"Wait", "But wait", "Hold on", "Hmm", ...
Explicit check	"Let me check", "Let me double-check", "Let me verify", ...

SnapKV	CAPE
[Question and Instruction – 20 words] Evaluate $\lceil (3.6)^2 \rceil - (\lceil 3.6 \rceil)^2$. Please reason step by step and put the final answer in \boxed{}. **[Think – 741 words]** … "Ceiling of 12.96 is 13. Then ceiling of 3.6 is 4, squared is 16. $13 - 16$ is -3. So, answer is -3?" **[Post-answer verification loop – 5639 words, 12 "Wait" tokens] "Wait, but maybe I need to check if the ceiling function is defined differently?"** … **"ceiling(3.1) is 4? Wait, no! Wait, ceiling(3.1) is 4? Wait, no!"** … **[Conclusion]** Truncated at the generation limit without a boxed answer.	**[Question and Instruction – 20 words]** Same problem and output instruction. **[Think – 517 words]** … "Ceiling of 12.96 is 13. Ceiling of 3.6 is 4. 4 squared is 16. $13 - 16$ is -3." **[Verification – 516 words, 2 "Wait" tokens]** … "Yep, same result. So that's correct. Therefore, I think my calculations are right. The answer is -3." **[Conclusion]** "**Final Answer** \boxed{-3}."

Table 8: Matched MATH-500 trace pair for problem `test/algebra/2232.json`. SnapKV becomes answer-ready but repeatedly reopens verification instead of committing, eventually truncating without a boxed answer; CAPE reaches the same value and commits after a few verification steps.