EXACT FINITE-QUERY KV CACHE COMPILATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Query-aware key value (KV) cache compression fits a chosen slot count to reference queries, but a successful fit cannot show that fewer slots would fail. We characterize the number of shared KV slots needed to preserve the normalized attention outputs of a finite protected query set exactly. A ratio identity maps each query to a ray of feasible replacement values. With query-specific positive weights, the minimum count is one plus the smallest dimension of an affine subspace meeting all rays. Two queries can already rule out one slot. For grouped query attention (GQA) with g query heads per KV head, protecting t positions requires $\min(k, tg)$ slots for k entries under explicit rank and key-realizability conditions. All 431,256 evaluated KV-head configurations satisfy the conditional law. With full query rank and realizable weights, a related construction represents an entire prefix by Tg standard KV slots per head for a known T-step trace. Across four models, native 32-bit floating-point (FP32) execution preserves every protected token through eight steps, removing 89.31% to 96.95% of same-dtype KV tensor bytes on 524-slot prefixes. The results separate fitting a cache from certifying its capacity within the shared KV representation.

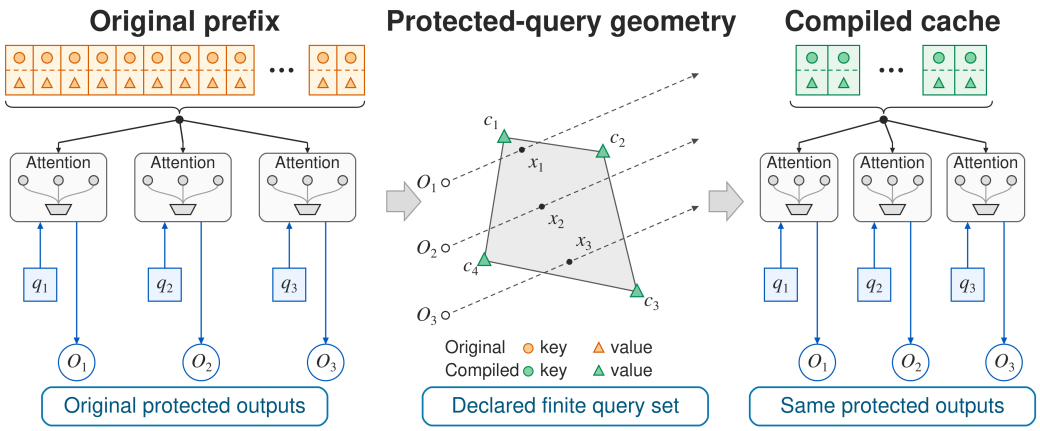


Figure 1: Cache compilation for a finite protected query set. In the cache rows, circles denote keys and triangles denote values. The original (orange) and compiled (green) caches return the same O_r for each q_r. Unchanged tokens outside the replaced prefix are omitted. In the center, positive rays start at distinct outputs O_r (open circles) and have no common point. Each marked x_r lies inside the convex hull of shared values c_j. A physical cache also requires keys that realize the positive weights. Slot counts are illustrative.

1 INTRODUCTION

The KV cache grows with prefix length, layer count, and concurrent sequences, which makes it a central memory cost in long-context inference. Memory managers improve allocation and prefix reuse (Kwon et al., 2023; Zheng et al., 2024), while eviction, quantization, and compaction reduce stored content (Zhang et al., 2023; Xiao et al., 2024; Li et al., 2024; Liu et al., 2024; Kim et al., 2025). Query-aware methods also merge or synthesize states for a current query or a reference set

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.

(Zhang et al., 2024; Tian et al., 2026; Cui & Xu, 2025; Liu et al., 2026; Zweiger et al., 2026). These methods construct or merge caches at a chosen budget. A successful fit shows that the count is feasible, but cannot show that a smaller count is impossible. We ask how many shared KV slots are necessary and sufficient to preserve the attention outputs of a finite query set exactly. Figure 1 shows cache compilation and its underlying value geometry.

Existing exact and theoretical results leave this finite-query question open. KeepKV gives a lossless merge for one current step and bounds later error (Tian et al., 2026), but one query does not determine whether the same cache can serve several queries. Worst-case space bounds and minimax risk analyses characterize approximation limits (Haris & Onak, 2025; Haverbeck et al., 2026). The latter includes compressors that know the query distribution. BalanceKV gives streaming approximation guarantees, and attention coresets give uniform guarantees over bounded-norm queries (Kochetkova et al., 2025; Liberty et al., 2026). Our question concerns pointwise equality on a finite set and the minimum number of shared synthetic KV slots.

A protected query set makes this capacity question well posed. A protected query is a query-head instance whose normalized attention output must remain unchanged after cache replacement; a slot is one cache sequence position. We call direct construction of synthetic KV states from a known prefix and this set *cache compilation*. In the score-relaxed setting, each query may choose its positive unnormalized replacement weights independently. This separates value geometry from key construction. The attention ratio identity then maps every protected query to a ray of feasible values. One query always admits one slot, whereas two disjoint rays form the smallest obstruction. For several slots, the rays must meet the relative interior of one convex hull, meaning its interior within the affine span. The minimum affine dimension of these intersections gives the exact score-relaxed slot count.

The geometry gives a conditional slot law for grouped query attention (GQA) (Ainslie et al., 2023). Let H_q and H_{kv} be the query-head and KV-head counts, and let $g = H_q/H_{kv}$. Protecting t decode positions creates tg joint constraints for each physical KV head. When the stated ray-rank condition holds and shared synthetic keys can realize the required positive weights, a group of k entries has minimum $m^\star_+ = \min(k, tg)$. Thus, for $g \geq 2$, even one decode position excludes a two-to-one merge when the pair's augmented ray matrix has full column rank. The proof gives both the common values and the shared-key solve.

The same geometry extends from a candidate group to a whole prefix. For a known T-step trace, it replaces an L-slot prefix by Tg shared slots when the joint protected-query matrix has full row rank and the required weight columns are key-realizable. This whole-prefix count is a constructive upper bound in the standard KV format. For the same known queries, one can instead store prefix outputs and log-normalizers and combine them with new tokens using the attention-state merge identity (Juravsky et al., 2024; Ye et al., 2025). That smaller, query-indexed representation needs no rank condition. Our construction shows when the same information can be represented by shared keys and values.

Across six public checkpoints and four public data sources, two-query certificates rule out one slot for every evaluated group. All 431,256 sampled KV-head configurations match the conditional slot law, including 60,492 cases whose exact count is smaller than the original group. The whole-prefix construction reconstructs the protected outputs at every layer and head in eight model-prompt cases, each tested with nested one-, two-, four-, and eight-step windows. All protected tokens match under native FP32 teacher forcing and greedy decoding. Even with eight protected steps, the 524-slot prefixes use 89.31% to 96.95% fewer same-dtype KV tensor bytes.

- We characterize the exact score-relaxed slot count through affine intersections of protected-query rays. Two queries give the smallest possible witness of one-slot infeasibility.
- We prove the conditional GQA slot law $m^\star_+ = \min(k, tg)$ and construct shared synthetic keys and values. The law determines a feasible headwise count before fitting.
- We construct a Tg-slot cache for a known T-step trace and validate its protection-window and capacity relation in native execution.

2 Related work

KV cache systems reduce memory through allocation, reuse, and approximate content reduction. PagedAttention manages noncontiguous cache pages, and RadixAttention reuses identical prefixes (Kwon et al., 2023; Zheng et al., 2024). StreamingLLM, H_2O, and SnapKV retain selected tokens (Xiao et al., 2024; Zhang et al., 2023; Li et al., 2024). KIVI quantizes keys and values, SQuat controls key residuals in a query-spanned subspace, and KVzip selects entries through context reconstruction (Liu et al., 2024; Wang et al., 2025; Kim et al., 2025). These methods optimize memory under an approximation or task criterion. We instead ask for the exact slot count required by one declared finite query set.

Cache merging is closest to this representation question. CaM merges evicted states into retained entries, while KeepKV gives a lossless merge for one current step and bounds the error at later steps (Zhang et al., 2024; Tian et al., 2026). AsymKV combines approximate local key merging with cardinality-aware value merging, and KVSlimmer develops objectives for asymmetric KV merging (Cui & Xu, 2025; Liu et al., 2026). Fast KV Compaction fits synthetic keys, values, and slot biases to reference outputs and attention mass (Zweiger et al., 2026). These methods show how to fit or merge a cache at a chosen count. Our analysis determines when one slot is impossible, gives the exact score-relaxed count, and constructs shared keys when they can realize the required positive weights.

Provable attention approximation addresses streaming and bounded-norm query regimes. BalanceKV uses discrepancy theory to obtain a streaming algorithm and space bounds, while attention coresets approach the optimal uniform approximation size for bounded-norm queries (Kochetkova et al., 2025; Liberty et al., 2026). The coreset guarantee holds simultaneously over its query domain. Our finite-query setting asks for exact interpolation of a known set, its minimum per-KV-head slot count, and a shared synthetic cache. The two settings therefore answer different capacity questions.

The guarantee also depends on which queries are available when the cache is formed. Matched-budget experiments show that query visibility can change the ranking of compression methods (Luo et al., 2026). Worst-case space analysis bounds attention approximation (Haris & Onak, 2025); minimax analysis studies distribution-averaged risk in both query-aware and query-agnostic settings (Haverbeck et al., 2026). VeriCache obtains decoded output identity by retaining and consulting the full cache when verification is needed (Yao et al., 2026). In our setting, the protected queries are known, the compressed cache itself satisfies their normalized attention equations, and the construction can be checked without consulting the original cache on the protected set.

Exact attention-state composition provides a second representation for known queries. Hydragen combines blockwise attention outputs using their log-normalizers, and FlashInfer exposes this operation as an attention-state merge (Juravsky et al., 2024; Ye et al., 2025). Fast KV Compaction also uses output and attention-mass matching (Zweiger et al., 2026). These identities motivate a direct query-indexed alternative to shared KV slots, compared in Appendix B.12.

Table 1 separates these approaches by query scope, cache representation, and guarantee. Our object of study is the exact slot count and a constructive representation for one finite protected query set under the per-head equality criterion defined below.

3 Exact representations for finite query sets

3.1 Setting and ratio identity

We study normalized attention within one query head (Vaswani et al., 2017). Let d_k and d_v denote the key and value dimensions. A protected constraint r consists of a query $q_r \in \mathbb{R}^{d_k}$, its visible memory $\mathcal{M}_r = \{(\mathbf{k}_i, v_i, \pi_i)\}_{i \in V(q_r)}$, and its original output $O_r \in \mathbb{R}^{d_v}$. We write $\mathcal{R}$ for the finite set of protected constraints and $n = |\mathcal{R}|$. Position metadata π_i contains the information used by the attention operator. Rotary position embeddings can be absorbed into the executed keys and queries (Su et al., 2021). For a positive unnormalized attention weight ψ, the memory read is

$$O(q; \mathcal{M}) = \frac{\sum_{i \in V(q)} \psi(q, \mathbf{k}_i) v_i}{\sum_{i \in V(q)} \psi(q, \mathbf{k}_i)}, \qquad \psi(q, \mathbf{k}_i) > 0, \tag{1}$$

Table 1: Representative cache guarantees. A protected set is a finite collection of queries available to the cache construction.

Approach	Query scope	Representation	Guarantee
Prefix reuse (Zheng et al., 2024)	identical prefix	original KV states	exact reuse without transformation
KeepKV (Tian et al., 2026)	current step	merged KV states	lossless single step merge
Fast compaction (Zweiger et al., 2026)	reference queries	synthetic KV with biases	approximate matching objective
VeriCache (Yao et al., 2026)	decoded candidates	compressed plus full cache	decoded output verification
This work	finite protected set	synthetic KV states	exact score-relaxed count and conditional physical construction

where $V(q)$ is the visible index set. Equation 1 includes dot-product softmax and softmax with logit soft-capping.

A candidate group S is visibility-consistent if every protected query either sees the group and its replacement or sees neither. We replace S by m slots with shared values $c_1, \dots, c_m$ and positive query-specific weights $y_{r1}, \dots, y_{rm}$. In the score-relaxed class, these weights may be chosen independently for each protected query. A physical cache also requires each weight column to be realized by one shared synthetic key. Exact replacement requires

$$O(q_r; \mathcal{M}'_r) = O(q_r; \mathcal{M}_r), \qquad r = 1, \dots, n. \tag{2}$$

Per-head equality defines the scope of our slot count. It is sufficient for equality of the attention block and is also necessary when the output projection is injective. We minimize the positive slot count for each head. A joint operator may exploit the kernel of a noninjective output projection, but that global problem is outside the headwise result studied here. Appendix B.4 states the exact projection condition.

The exact constraint can be computed from the original cache. For each protected query, define

$$\begin{aligned} a_{ri} &= \psi(q_r, \mathbf{k}_i), & Z_r &= \sum_{i \in V(q_r)} a_{ri}, & p_{ri} &= a_{ri}/Z_r, \\ O_r &= \sum_{i \in V(q_r)} p_{ri} v_i, & \widetilde{D}_r &= \sum_{i \in S} p_{ri}(v_i - O_r). \end{aligned} \tag{3}$$

The signed contribution $\widetilde{D}_r$ records the change that the replacement must reproduce relative to the normalized output.

We separate deletion from positive-slot replacement. Let $m^\star_+$ denote the score-relaxed headwise minimum over $m \geq 1$. If every weight column at this minimum is realized by a shared synthetic key, it is also the physical key-realizable minimum. A zero-slot deletion is exact if and only if $\widetilde{D}_r = 0$ for every protected constraint and the visible complement retains a positive denominator. The positive-slot formula below does not include this deletion case.

Theorem 1 (Exact ratio identity)**.** *The m replacement slots preserve protected constraint r if and only if*

$$\sum_{j=1}^{m} \frac{y_{rj}}{Z_r}(c_j - O_r) = \widetilde{D}_r. \tag{4}$$

Cross multiplication cancels the unchanged complement and yields Equation 4. The identity preserves the normalized output, so it allows numerator and denominator changes that cancel in the ratio. Appendix B.2 gives both directions.

3.2 From one slot obstruction to exact slot count

For one replacement slot, Equation 4 restricts its value to the positive query ray

$$\mathcal{L}_r^+(S) = \{O_r + \tau \widetilde{D}_r : \tau > 0\}. \tag{5}$$

When $\widetilde{D}_r = 0$, the ray is the singleton $\{O_r\}$.

Corollary 2 (Sharp one slot boundary)**.** *One protected query always admits one positive-weight slot. Two protected queries are the smallest set that can exclude one slot.*

For one query, any $\tau > 0$ selects a feasible value and weight. For two queries, disjoint positive rays exclude every common value. Appendix B gives an explicit dot product instance with two disjoint rays.

Corollary 3 (Two-query certificate)**.** *If $\mathcal{L}_a^+(S) \cap \mathcal{L}_b^+(S) = \varnothing$, then no positive-weight one-slot replacement is exact on the protected set.*

The certificate assumes only positive weights. It therefore rules out every key parameterization before a replacement optimizer is chosen.

Multiple slots replace point intersection by convex representation. Let $K = \mathrm{conv}\{c_1, \ldots, c_m\}$. Normalizing the positive coefficients in Equation 4 shows that every ray must meet $\mathrm{relint}(K)$, as illustrated by the shaded hull in Figure 1. This condition is also sufficient in the score-relaxed class, where every $y_{rj} > 0$ can be selected independently.

Theorem 4 (Exact score-relaxed positive-slot count)**.** *For a finite protected set, the score-relaxed headwise minimum over $m \geq 1$ is*

$$m_+^\star = 1 + \min_{\tau_1, \ldots, \tau_n > 0} \mathrm{affdim}\{O_r + \tau_r \widetilde{D}_r : r = 1, \ldots, n\}. \tag{6}$$

In particular, $m_+^\star \leq d_v + 1$.

Any m values span an affine space of dimension at most $m - 1$, which gives the lower bound. Conversely, finitely many selected ray points in a δ dimensional affine space can be placed in the relative interior of a δ simplex. Its vertices give $\delta + 1$ values and strictly positive query weights. Appendix B.6 gives the complete proof.

3.3 Conditional physical GQA law

Physical GQA couples several query heads to one KV head (Ainslie et al., 2023). Let $g = H_q/H_{kv}$. Protecting t decode positions creates $n = tg$ joint query constraints for each physical KV head. All query heads mapped to the same KV head must therefore be solved jointly.

Under the stated rank conditions, Equation 6 has a closed form. Let $k = |S|$, $b = \min(k, n)$, and augment $\bar{O}_r = (O_r, 1)$ and $\bar{D}_r = (\widetilde{D}_r, 0)$. For any b constraints, define

$$R_{\mathrm{ray}} = [\bar{O}_1, \ldots, \bar{O}_b, \bar{D}_1, \ldots, \bar{D}_b]. \tag{7}$$

Let $Q \in \mathbb{R}^{n \times d_k}$ contain the protected queries. A table $Y = (y_{rj})$ is *key realizable* if each column satisfies $\psi(q_r, \mathbf{k}_j) = y_{rj}$ for one shared key $\mathbf{k}_j$; soft capping also requires the inverse target logits to lie in range.

Theorem 5 (Conditional physical GQA slot law)**.** *Suppose R_{ray} has full column rank for the selected b constraints. When $n < k$, also suppose Q has full row rank and the required positive weight table is key realizable. Then the score-relaxed and physical key-realizable minima coincide:*

$$m_+^\star = \min(k, n) = \min(k, tH_q/H_{kv}). \tag{8}$$

The rank of R_{ray} forces any b selected ray points to be affinely independent, which gives $m \geq b$. If $n \geq k$, retaining the original k entries gives equality. If $n < k$, expanding the selected points gives n common values and strictly positive weights. Full row rank of Q realizes each weight column through a linear log-weight solve; soft-capped attention also checks the inverse range. The complete construction, dimension conditions, and dense layer alignment appear in Appendix B.7.

Pairwise merging illustrates the lower bound without a key assumption. At one protected position with $g \geq 2$, a two-entry group needs two slots whenever the augmented matrix for two mapped query heads has full column rank. This follows from $\min(2, g) = 2$; retaining the original pair attains the bound.

4 WHOLE PREFIX COMPILATION

The group construction extends to a complete visible prefix. Fix a reference trace of T decode positions after an L slot prefix. For each physical KV head, the Tg mapped query constraints share the entire prefix group $S = \{1, \ldots, L\}$. We preserve the original weight mass, uniformly expand the selected values, solve the positive weight columns for synthetic keys, and use the same sequence count across all physical heads.

Theorem 6 (Finite reference whole prefix compiler)**.** *Assume $L > Tg$, $Tg \leq d_k$, full row rank of each joint query matrix, and key realizability for every physical head. Then the complete prefix admits a constructive representation with*

$$M = Tg = TH_q/H_{kv} \tag{9}$$

shared sequence slots that preserves every declared normalized head output.

The theorem gives a constructive upper bound on the whole prefix size. Its proof applies Theorem 1 to the full prefix in each head, then aligns the M synthetic slots across heads. Exact replacements of disjoint groups also compose because each preserves the same output ratio and their numerator and denominator changes add. Appendix B.10 gives both arguments, the executable tensor construction, and its cost.

A declared trace is preserved by induction in exact arithmetic. Equal attention outputs preserve the hidden state at the first protected step. The resulting query, logits, and token match the reference, so the same argument applies to the next step and then to every layer and position. This trajectory result requires every query in the declared trace to be included in the construction. The native experiment therefore tests all layers and physical heads together, with position IDs and GQA mappings. Under the theorem's conditions, doubling the protected window doubles the constructed count; the original prefix length does not enter this count.

5 EXPERIMENTS

5.1 EVALUATION SCOPE AND NUMERICAL CRITERIA

The one-slot study asks whether two protected queries can certify that one slot is insufficient for groups drawn from real cache activations. It uses early, middle, and final layers from six public Qwen, Gemma, and Llama checkpoints (Qwen et al., 2024; Yang et al., 2025; Gemma Team et al., 2024; Grattafiori et al., 2024). Protected queries come from the AI2 Reasoning Challenge (ARC), Grade School Math 8K (GSM8K), OpenBookQA, and WikiText-103 (Clark et al., 2018; Cobbe et al., 2021; Mihaylov et al., 2018; Merity et al., 2016). We enumerate all groups of two to four positions on the short question-answering prefixes and all pairs on six 512-token prefixes. The long triples and quadruples use deterministic selections described in Appendix C.

The conditional GQA study asks whether the predicted slot count matches real model activations. It covers four checkpoints, two input types, three layers, and two physical KV heads per layer. We evaluate $k \in \{2, 3, 4, 8, 16, 32\}$ and $t \in \{1, 2, 3, 4, 6, 8, 9\}$. Each configuration checks the augmented-ray and joint-query ranks. When the law predicts $0 < m < k$, we also solve for the synthetic keys and values and replay the protected outputs.

The whole-prefix study asks whether the construction forms a dense native cache. For each of four checkpoints, it uses one fixed ARC prompt and one fixed WikiText-derived long prompt. The final construction covers every layer and physical KV head for $T \in \{1, 2, 4, 8\}$ protected steps. The 32 nested windows share eight fixed model-prompt traces, each continued for 12 steps. Matched-slot controls use $T = 4$. A separate breadth scan covers five layers and every physical KV head for candidate groups that are strict subsets of the prefix, which we call proper groups.

Table 2 separates source examples from the algebraic checks nested within them. This distinction matters because groups, configurations, and layer-head instances test mathematical claims; they are not independent benchmark samples.

We recompute the attention identities from saved float32 operands in extended precision. A construction meets the mathematical criterion when the maximum protected-query relative output error

Table 2: Evaluation scope and reporting units. Long-prefix inputs are derived from the WikiText-103 training split and are not a separate benchmark evaluation.

Stage	Models	Source and split	Prompt and query selection	Scope and reported unit
One-slot certificate	6	ARC-Challenge, GSM8K, OpenBookQA test; WikiText-103 raw train	8 spaced IDs per short source; 6 fixed 512-token prefixes; steps 0, 8, 15	3 layers; candidate groups and layer-group evaluations
Conditional GQA law	4	ARC-Challenge test; WikiText-derived long prefix	4 fixed IDs per input type; first 3 protected, fourth held out	3 layers and 2 physical KV heads per layer; algebraic configurations
Whole-prefix construction	4	ARC-Challenge test; WikiText-derived long prefix	1 fixed prompt per input type; windows of 1, 2, 4, 8 steps	Every layer and physical KV head; 32 nested windows on 8 fixed traces

defined in Appendix C is below 10^{-9}. In native FP32 execution, we use the model's position IDs and GQA mapping, require finite outputs, and report token equality together with relative logit and context errors over each protected window. Token equality is a discrete trace result and does not imply bitwise-equal logits.

6 Results

6.1 From the one slot obstruction to the slot law

Two protected queries rule out one slot throughout the evaluation. Every one of the 928,695 candidate groups is infeasible at early, middle, and final layers, for 2,786,085 layer-group evaluations across six checkpoints and four data sources. The short prefixes enumerate every group of two to four positions, and the long prefixes enumerate every pair. Appendix C gives the deterministic selection policy for larger long-prefix groups.

The conditional GQA law gives the exact tested budget. All 431,256 sampled KV-head configurations satisfy the required rank conditions and match $m_+^\star = \min(k, tg)$. The 60,492 configurations with $tg < k$ admit a smaller cache, and every constructed replacement reconstructs the protected outputs within tolerance. The remaining configurations already use the minimum count, and none admits exact deletion. A separate all-head scan resolves 219,598 of 219,600 proper-group configurations and constructs all 45,620 eligible replacements. The two unresolved configurations share a near-singular rank cell and remain numerically unresolved in Appendix D.

Figure 2 shows the transition implied by the law. Increasing the protected-position count raises the required fraction until it reaches one. A smaller query-to-KV-head ratio g leaves more room for compression at the same protection count. Appendix D reports the observed counts for each model and stratum.

6.2 Whole prefix compilation

The whole-prefix construction succeeds at every tested protection window. Every layer and physical KV head satisfies the joint-query rank condition, admits a shared-key solve for the required weights, and reconstructs the protected outputs within tolerance. All 5,568 layer-head-window checks pass, nested within 32 windows on eight fixed traces. The largest saved-operand float64 relative error is 2.40×10^{-13}.

Eight-step protection retains substantial cache savings across all four models. On the 524-slot prefixes, $M = 8g$ gives $9.36\times$ to $32.75\times$ slot compression, or 89.31% to 96.95% fewer same-dtype KV

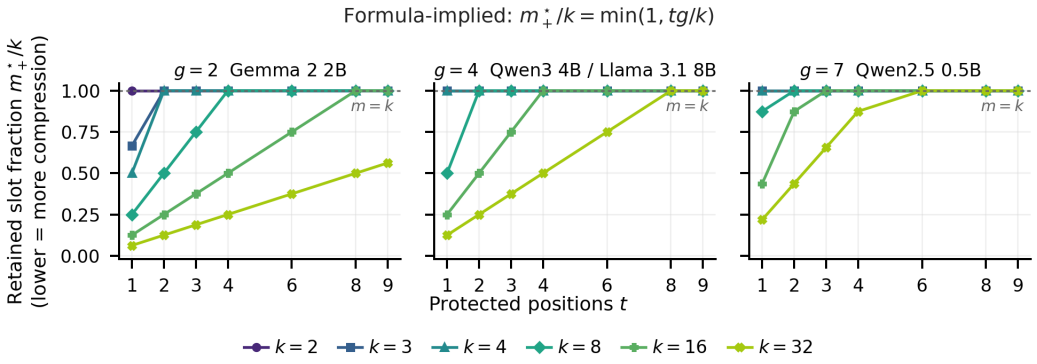


Figure 2: The conditional GQA law predicts how protection consumes cache capacity. The curves plot the formula-derived fraction $\min(k, tg)/k$ for the three GQA widths in the tested models; lower values permit more compression. Across two physical KV heads at each target layer, all 431,256 evaluated configurations satisfy the stated rank and realizability checks.

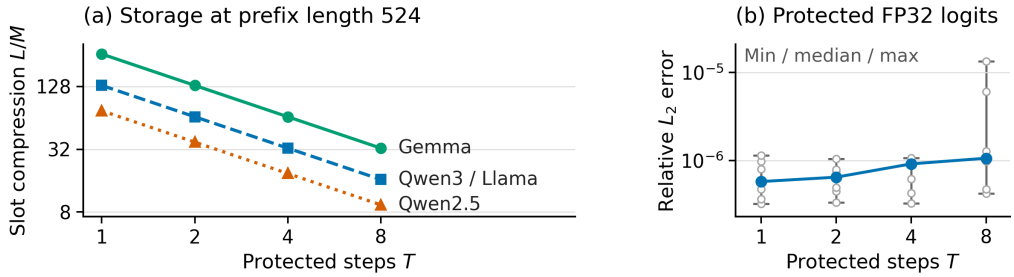


Figure 3: Longer protected windows use more slots while preserving the declared tokens. Left: $L/(Tg)$ for four models at $L = 524$; Qwen3 and Llama have the same GQA ratio. Right: each point is a case's maximum protected-logit relative error in native FP32; the line and bars show the median and range across eight fixed model-prompt cases, not confidence intervals. Every protected token matches at every window size.

tensor bytes. Figure 3 shows how this capacity changes with the protected window. All 32 windows preserve every protected token under teacher forcing and greedy decoding; the largest relative logit error is 1.35×10^{-5}. Appendix F reports every case, including the shorter ARC prefixes, precision, and post-window continuation.

At the four-step matched budget, compilation alone preserves every evaluated protected trace in native execution. The compiled caches have median per-case maximum relative errors of 9.18×10^{-7} for logits and 8.16×10^{-7} for attention context. Figure 4 compares three matched-slot heuristic controls that isolate exact construction at a fixed slot count. Mean pooling preserves no complete trace, uniform sampling preserves two of eight, and attention top M, which retains the M positions with highest protected attention, preserves six. Their median protected logit errors range from 0.249 to 1.172.

7 Discussion

The slot law determines when exact merging can reduce a cache. With a nonempty visible complement, deletion is exact when all contributions $\widetilde{D}_r$ vanish. Disjoint rays rule out one slot. Under the conditional GQA assumptions, $\min(k, tg)$ gives the minimum per-head count and the proof constructs shared keys and values. Feasibility and minimality then follow from the algebraic conditions, independently of an optimizer's success.

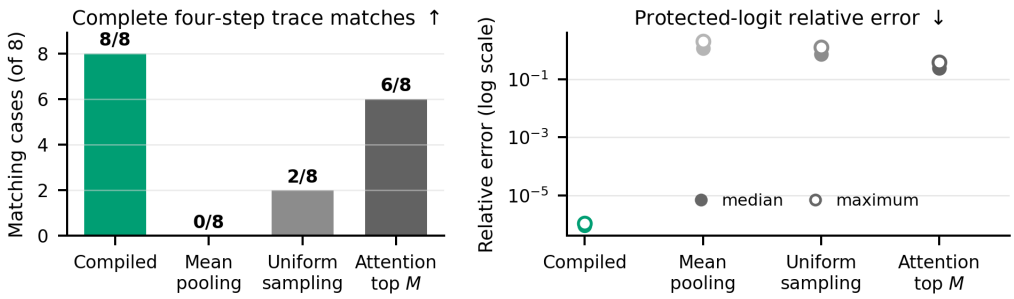


Figure 4: Among the evaluated same-slot representations, compilation alone preserves all eight complete four-step traces in native FP32. The left panel counts matching model-prompt cases. In the right panel, each case contributes its maximum protected-logit relative error over four steps; points report the median and maximum across applicable cases.

Longer protection costs more slots. Each decode position adds g constraints per physical KV head, raising the required fraction $\min(k, tg)/k$. Group size is therefore also a design choice. Under the conditional law, exact compression is possible if and only if $k > tg$. A larger group satisfying these conditions can be compressed jointly to tg slots per physical KV head even when each pair is irreducible: exact pairwise merges are not a prerequisite. Under its conditions, the whole-prefix construction can cover $\lfloor B_{\text{slot}}/g \rfloor$ steps with a slot budget B_{slot}. Appendix F.4 derives the conditional update count when successive reference queries are available; an adaptive refresh policy remains to be developed.

The representation class matters when interpreting this budget. For known queries, cached prefix outputs and log-normalizers use $Tg(d_v + 1)$ scalars per KV head, whereas our Tg-slot construction uses $Tg(d_k + d_v)$. The attention-state merge is exact without a rank condition (Juravsky et al., 2024; Ye et al., 2025). This construction uses more storage to retain the shared-KV format. Our slot law characterizes that constrained representation; it is not a lower bound on all ways to store a known trace.

The protected query set defines the guarantee. Query visibility can change empirical compression rankings (Luo et al., 2026), and existing theory studies worst-case space, distribution-averaged minimax risk, streaming approximation, or uniform bounded-query approximation (Haris & Onak, 2025; Haverbeck et al., 2026; Kochetkova et al., 2025; Liberty et al., 2026). Our whole-prefix result constructs an upper bound from queries obtained in a full-cache reference run. In 31 of the 32 windows, the greedy continuation first diverges after protection ends and by step 12. Online use therefore requires a way to obtain or verify the next protected queries. Appendix E.5 explains why the proper-group lower test cannot establish a whole-prefix minimum here.

Machine execution preserves protected tokens with small numerical errors. The maximum relative error is 2.14×10^{-13} for extended-precision output replay and 1.35×10^{-5} for native FP32 logits. Four-step long-prefix median decode latency stays within 1.3% of the full cache. The measured benefit is cache capacity; faster execution would require specialized kernels and amortized construction. Full-cache verification, as in VeriCache (Yao et al., 2026), could extend decoded identity beyond the protected trace.

8 Conclusion

Finite protected queries determine an exact slot count through their ray geometry. Under the stated rank and key-realizability conditions, this yields the GQA law $\min(k, tg)$ and a standard-KV construction for known traces. The experiments verify both the capacity law and protected execution. Extending the construction to changing protected queries remains an algorithmic question.

AI USE STATEMENT

Generative AI tools supported mathematical formalization, experimental code checking, and manuscript polishing. The authors developed the theoretical framework and experimental design, interpreted the results, and verified the citations, equations, code, figures, insights, claims, and prose. The authors take responsibility for all content.

ETHICS STATEMENT

This work studies public language-model checkpoints and established public benchmarks. It involves no human subjects, personal data collection, or model deployment. The main foreseeable risk is applying a finite query guarantee to queries outside its declared set. We state the protected scope, arithmetic semantics, models, and workload cases in the main paper and appendix.

REPRODUCIBILITY STATEMENT

The complete reproduction code and machine-readable result summaries are provided in the Supplementary Material.

Appendix B gives complete proofs, and Appendix C records model revisions, query and candidate generation, numerical rules, native cache injection, controls, and extended results. The supplement provides a complete one-cell Colab program for the one-slot certificates, conditional slot law, whole-prefix compiler, protection-window evaluation, and CPU appendix calculations. It downloads the fixed models and data, resumes from completed checkpoints, and generates fresh result archives. Machine-readable summaries and the source data for the slot-law and protection-window figures are included.

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 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Wanyun Cui and Mingwei Xu. Homogeneous keys, heterogeneous values: Exploiting local KV cache asymmetry for long-context LLMs. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.05410.

Gemma Team et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Themistoklis Haris and Krzysztof Onak. Compression barriers in autoregressive transformers. In *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pp. 2757–2785. PMLR, 2025. URL https://proceedings.mlr.press/v291/haris25a.html.

Lukas Haverbeck, Carmen Amo Alonso, Andres Felipe Posada-Moreno, Sebastian Trimpe, and Marco Pavone. The risk of KV cache compression. *arXiv preprint arXiv:2607.01520*, 2026. URL https://arxiv.org/abs/2607.01520.

Jordan Juravsky, Bradley Brown, Ryan Ehrlich, Daniel Y. Fu, Christopher Ré, and Azalia Mirhoseini. Hydragen: High-throughput LLM inference with shared prefixes. *arXiv preprint arXiv:2402.05099*, 2024. URL https://arxiv.org/abs/2402.05099.

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. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.23416.

Ekaterina Kochetkova, Kshiteej Jitesh Sheth, Insu Han, Amir Zandieh, and Michael Kapralov. Streaming attention approximation via discrepancy theory. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0705. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1de510838faddd72f79a7a0e9c564f0c-Abstract-Conference.html.

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 *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

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. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2404.14469.

Edo Liberty, Alexandr Andoni, and Eldar Kleiner. Nearly optimal attention coresets. *arXiv preprint arXiv:2605.05602*, 2026. URL https://arxiv.org/abs/2605.05602.

Lianjun Liu, Hongli An, Weiqi Yan, Xin Du, Shengchuan Zhang, Huazhong Liu, and Yunshan Zhong. KVSlimmer: Theoretical insights and practical optimizations for asymmetric KV merging. *arXiv preprint arXiv:2603.00907*, 2026. URL https://arxiv.org/abs/2603.00907.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24bz.html.

Daming Luo, Christy Liang, and Junyu Xuan. How query visibility changes KV-cache compression rankings: A matched-budget audit. *arXiv preprint arXiv:2607.11942*, 2026. URL https://arxiv.org/abs/2607.11942.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. URL https://arxiv.org/abs/1609.07843.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Qwen, An Yang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

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. URL https://arxiv.org/abs/2104.09864.

Yuxuan Tian, Zihan Wang, Yebo Peng, Aomufei Yuan, Zhiming Wang, Bairen Yi, Xin Liu, Yong Cui, and Tong Yang. KeepKV: Achieving periodic lossless KV cache compression for efficient LLM inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33259–33267, 2026. doi: 10.1609/aaai.v40i39.40611. URL https://doi.org/10.1609/aaai.v40i39.40611.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.03762.

Hao Wang, Ligong Han, Kai Xu, and Akash Srivastava. SQuat: Subspace-orthogonal KV cache quantization. *arXiv preprint arXiv:2503.24358*, 2025. URL https://arxiv.org/abs/2503.24358.

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. URL https://arxiv.org/abs/2309.17453.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Jiayi Yao, Samuel Shen, Kuntai Du, Shaoting Feng, Dongjoo Seo, Rui Zhang, Yuyang Huang, Yuhan Liu, Shan Lu, and Junchen Jiang. VeriCache: Turning lossy KV cache into lossless LLM inference. *arXiv preprint arXiv:2605.17613*, 2026. URL https://arxiv.org/abs/2605.17613.

Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, and Luis Ceze. FlashInfer: Efficient and customizable attention engine for LLM inference serving. In *Proceedings of Machine Learning and Systems*, volume 7. MLSys, 2025. URL https://proceedings.mlsys.org/paper_files/paper/2025/hash/dbf02b21d77409a2db30e56866a8ab3a-Abstract-Conference.html.

Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. CaM: Cache merging for memory-efficient LLMs inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58840–58850. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhang24n.html.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.14048.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2312.07104.

Adam Zweiger, Xinghong Fu, Han Guo, and Yoon Kim. Fast KV compaction via attention matching. *arXiv preprint arXiv:2602.16284*, 2026. URL https://arxiv.org/abs/2602.16284.

A Appendix map

The appendix follows the main argument in six parts:

1. Appendix B proves the ratio identity, exact slot count, conditional physical GQA law, dense common slot rule, composition theorem, whole-prefix construction, and protected continuation result. It also records the separate full-query frequency result.
2. Appendix C specifies models, queries, candidates, numerical criteria, and the complete one-slot evidence.

3. Appendix D reports every evaluated slot-law stratum, construction residuals, and held out diagnostics.
4. Appendix E reports whole-prefix layer-head instances, native controls, capacity, continuation scope, precision, and runtime.
5. Appendix F gives the full protection-window results, precision and continuation diagnostics, and conditional refresh count.
6. Appendix G describes the one-cell reproduction program and supplied summaries.

B Complete proofs

B.1 Notation and visibility

The notation separates vector keys from cache sizes and numerical checks. We use $\mathbf{k}_i$ for a key and $k = |S|$ for group cardinality; $n = |\mathcal{R}|$ counts protected query-head constraints, while N_S is a group's unnormalized numerator. The augmented ray matrix is R_{ray}, a slot budget is B_{slot}, and the number of compiled blocks is J. The common dense slot count is M; μ and $\varepsilon_{\mathrm{guard}}$ denote the numerical rejection margin and guard. The memory itself is $\mathcal{M}$.

Fix one attention head and one protected query $q \in \mathcal{R}$. The visible original memory is partitioned into the candidate group S and its complement C. Define

$$z = \sum_{s \in S} a_s, \qquad N_S = \sum_{s \in S} a_s v_s,$$
$$Z_0 = \sum_{j \in C} a_j, \qquad N_0 = \sum_{j \in C} a_j v_j,$$

where $a_i = \psi(q, \mathbf{k}_i) > 0$. The original output is $O = (N_0 + N_S)/(Z_0 + z)$. The replacement has positive weights $y_1, \ldots, y_m$ and values $c_1, \ldots, c_m$. Write $Y = \sum_j y_j$ and $C_y = \sum_j y_j c_j$. Its output is $O' = (N_0 + C_y)/(Z_0 + Y)$. Visibility consistency ensures that both expressions contain the same complement C. A query that sees only a proper subset of S defines a different replacement problem.

B.2 Proof of the exact ratio identity

Proof of Theorem 1. The equality $O' = O$ is equivalent to

$$(N_0 + C_y)(Z_0 + z) = (N_0 + N_S)(Z_0 + Y).$$

Expanding both sides and cancelling $N_0 Z_0$ gives

$$(Z_0 + z)(C_y - YO) = Z_0 N_S - z N_0.$$

Since $(Z_0 + z)O = N_0 + N_S$, division by $Z = Z_0 + z > 0$ gives

$$\sum_j y_j (c_j - O) = \frac{Z_0 N_S - z N_0}{Z} = N_S - zO.$$

Finally,

$$Z\widetilde{D} = Z \sum_{s \in S} \frac{a_s}{Z}(v_s - O) = N_S - zO.$$

Dividing by Z proves Equation 4. Every step is reversible. □

This proof also shows the status of separate mass and local-output matching. If $Y = z$ and $C_y = N_S$, then $O' = O$. These conditions preserve removed mass and numerator contribution separately. Theorem 1 also permits compensating changes that preserve the final normalized ratio.

B.3 PROTECTED CONTINUATION PRESERVATION

Proposition 7 (Protected continuation preservation)**.** *Consider a deterministic decoder, or a decoder coupled to the same external randomness, and a protected continuation beginning from the same boundary hidden state. If every modified head preserves its output at every continuation position and layer on the complete visible memory at that time, then all hidden states, queries, logits, and generated tokens in the continuation are identical.*

Proof. Proceed by induction over continuation position t and, within each position, over layer ℓ. The boundary hidden state is equal by assumption. If the layer input at (t, ℓ) is equal, its normalization and query projection are equal. A modified attention module returns the same head output by hypothesis. An unmodified module receives identical queries, keys, and values and therefore also returns the same output. The output projection, residual update, and feed-forward block are deterministic, so the next layer state is equal. Layer induction gives equal logits at position t. Greedy decoding therefore chooses the same token; sampling does so under coupled randomness. The next position starts from equal states, completing the position induction. □

For offline prefix reuse, the boundary state can be stored together with the compressed cache. If a deployment instead recomputes the compressed prefix, then the recomputed prefix positions also belong to the protected continuation and must satisfy the same condition.

B.4 HEADWISE EQUALITY AND THE OUTPUT PROJECTION

The output projection specifies exactly when compensation across heads can preserve an attention block. Let $o = \mathrm{concat}_h O^h$ and let o' be the concatenated head outputs before a fixed output projection W_O.

Proposition 8 (Output-projection criterion)**.** *The attention-block outputs are equal if and only if* $o' - o \in \ker(W_O)$*. Consequently, per-head equality is always sufficient and is necessary whenever* W_O *is injective.*

Proof. Any fixed projection bias cancels, so equality after projection is equivalent to $W_O(o' - o) = 0$. This is precisely membership in $\ker(W_O)$. If W_O is injective, its kernel contains only zero, which forces $o' = o$ and therefore equality in every head. □

Cross-head compensation requires a joint cache operator whose concatenated error lies in the output-projection kernel. The empirical certificate studies the headwise operator; Proposition 8 states when this criterion also becomes necessary at the block output.

B.5 COMMON RAYS AND SCORE REALIZABILITY

Proof of Corollary 2. Fix any available score $y(q) > 0$ and set $c = O(q) + [Z(q)/y(q)]\widetilde{D}_S(q)$. Then $[y(q)/Z(q)](c - O(q)) = \widetilde{D}_S(q)$, so Theorem 1 gives exact preservation. Therefore no one-query protected set is infeasible in the positive-score one-slot class. The explicit instance below is infeasible for two queries, establishing minimality for the score-relaxed ratio criterion. □

The two-query boundary is attained by a valid dot-product attention instance. Take one-dimensional queries $q_a = 1$ and $q_b = -1$, keys $(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3) = (0, \log 4, 0)$, values $(v_1, v_2, v_3) = (0, 2, 1)$, and group $S = \{1, 2\}$ with complement $C = \{3\}$. The unnormalized scores are $(1, 4, 1)$ for q_a and $(1, 1/4, 1)$ for q_b. Direct substitution gives the following values, where D_r abbreviates $\widetilde{D}_r$ throughout the remaining numerical appendix:

$$O_a = \frac{3}{2}, \quad D_a = \frac{1}{12}, \qquad O_b = \frac{2}{3}, \quad D_b = -\frac{4}{27}.$$

Hence $\mathcal{L}_a^+(S) = (3/2, \infty)$ and $\mathcal{L}_b^+(S) = (-\infty, 2/3)$ are disjoint. One query is always feasible by Corollary 2, whereas this pair is infeasible by Corollary 3.

Proof of Corollary 3. For every exact one slot replacement, Equation 4 implies $c = O_r + \tau_r \widetilde{D}_r$ with $\tau_r = Z_r/y_r > 0$. Thus c belongs to every $\mathcal{L}_r^+(S)$. If any pair has empty intersection, no such value exists. A particular key parameterization restricts the attainable positive weights and therefore preserves the rejection. □

Ray intersection is necessary but does not complete the dot-product problem. If $\widetilde{D}_r = 0$, the ray is the singleton $\{O_r\}$; when $c = O_r$, that query does not determine a unique positive replacement weight. Let $Q \in \mathbb{R}^{n \times d_k}$ have protected queries as rows. If a common value fixes or permits a vector of required positive weights $y \in \mathbb{R}^n_{>0}$, with free positive entries on such zero-direction rows, an unmodified dot-product key exists exactly when

$$\log y \in \operatorname{col}(Q). \tag{10}$$

If a constant score weight $w = e^\beta$ is allowed, the condition becomes $\log y \in \operatorname{col}([Q, \mathbf{1}])$. These statements follow directly from solving $Q\mathbf{k}^* = \log y$ or $Q\mathbf{k}^* + \beta\mathbf{1} = \log y$.

The extra bias is not always available in an unmodified attention layer. A pair $(\mathbf{k}^*, \beta)$ can be represented by one ordinary key $\mathbf{k}_{\text{actual}}$ on the protected set if and only if

$$\beta\mathbf{1} \in \operatorname{col}(Q). \tag{11}$$

Indeed, $\delta\mathbf{k} = \mathbf{k}_{\text{actual}} - \mathbf{k}^*$ must satisfy $Q\delta\mathbf{k} = \beta\mathbf{1}$. For $\beta \neq 0$, this is equivalent to the affine hull of the protected queries excluding the origin. If 0 is an affine combination of the queries, applying $\delta\mathbf{k}^\top q = \beta$ to that combination gives $0 = \beta$. Conversely, if the affine hull excludes the origin, a separating normal is constant and nonzero on that affine hull and can be scaled to β. This distinction is why the main result rejects all positive weights before discussing key realizability.

B.6 Exact slot count with score-relaxed positive weights

The positive-ray geometry extends from one replacement slot to an arbitrary finite slot budget. Replace S by m entries with shared values $c_1, \ldots, c_m \in \mathbb{R}^{d_v}$ and query-specific positive weights $y_{r,a} > 0$, where $m \geq 1$. All replacement entries obey the same visibility rule as S. Repeating the cross-multiplication in Theorem 1 gives the necessary and sufficient condition

$$\sum_{a=1}^{m} y_{r,a}(c_a - O_r) = Z_r \widetilde{D}_r \qquad \text{for every protected query } r. \tag{12}$$

Let $K = \operatorname{conv}\{c_1, \ldots, c_m\}$. With $Y_r = \sum_a y_{r,a}$ and $\alpha_{r,a} = y_{r,a}/Y_r$, Equation 12 says that

$$x_r = \sum_a \alpha_{r,a} c_a = O_r + \frac{Z_r}{Y_r}\widetilde{D}_r \in \operatorname{relint}(K) \cap \mathcal{L}_r^+(S). \tag{13}$$

Strict score positivity is the reason for the relative interior rather than the boundary of K.

For the proof of Theorem 4, define

$$\delta = \min\left\{\dim A : \ A \subseteq \mathbb{R}^{d_v} \text{ is an affine subspace and } A \cap \mathcal{L}_r^+(S) \neq \varnothing \text{ for every } r\right\}. \tag{14}$$

If each replacement score may be chosen as an arbitrary positive number on each protected query, then the minimum exact headwise slot count over $m \geq 1$ is $\delta + 1$. Equivalently, this is Equation 6. In particular, it is at most $d_v + 1$.

Proof of Theorem 4. First, a finite list of points $c_1, \ldots, c_m$ represents a point using strictly positive convex coefficients if and only if that point lies in $\operatorname{relint}(K)$. For the converse direction, let $b = m^{-1}\sum_a c_a$, which lies in $\operatorname{relint}(K)$. If $x \in \operatorname{relint}(K)$, then for sufficiently small $\epsilon > 0$, $z = (x - \epsilon b)/(1 - \epsilon)$ lies in K. Combining any convex representation of z with weight ϵ on b gives a representation of x with every coefficient positive.

Suppose an exact m-slot replacement exists. Equation 13 gives, for every r, a point in both $\operatorname{aff}(K)$ and $\mathcal{L}_r^+(S)$. Therefore $\delta \leq \dim \operatorname{aff}(K) \leq m - 1$, so $m \geq \delta + 1$.

For the reverse inequality, choose a δ-dimensional affine subspace A that attains Equation 14, and choose $x_r \in A \cap \mathcal{L}_r^+(S)$ for each protected query. A finite set of points in A is contained in the

relative interior of some δ-simplex in A. Let its $\delta + 1$ vertices be the replacement values. Write $x_r = \sum_a \alpha_{r,a} c_a$ with every $\alpha_{r,a} > 0$. If $\widetilde{D}_r \neq 0$, let $x_r = O_r + \tau_r \widetilde{D}_r$ and set $y_{r,a} = (Z_r/\tau_r)\alpha_{r,a}$. If $\widetilde{D}_r = 0$, then $x_r = O_r$ and any positive total score, distributed according to the same $\alpha_{r,a}$, satisfies Equation 12. This constructs an exact score-relaxed replacement with $\delta + 1$ slots. Finally, $A = \mathbb{R}^{d_v}$ always intersects every nonempty ray, so $\delta \leq d_v$. □

The separate zero-slot case follows directly from Equation 4. Deleting S is exact for every protected query if and only if $\widetilde{D}_r = 0$ for all r, provided the unchanged visible complement has positive denominator. This case is not part of $m^\star_+$.

For one slot, K is a single point, and Theorem 4 reduces to the common-ray condition used by Corollary 3. In the explicit one-dimensional two-query example above, no point meets both rays, but the line $\mathbb{R}$ does; hence $\delta = 1$ and two score-relaxed slots suffice. This characterization isolates the value geometry. Dot product or capped score realizability adds the key constraints proved next.

B.7 Augmented ray lower bound and the conditional slot law

We first prove the lower bound in Theorem 5. Select $b = \min(k, n)$ protected constraints and suppose the augmented matrix in Equation 7 has full column rank. For arbitrary $\tau_r > 0$, define

$$x_r = O_r + \tau_r \widetilde{D}_r, \qquad \bar{x}_r = \bar{O}_r + \tau_r \bar{D}_r.$$

An affine dependence among $x_1, \ldots, x_b$ gives coefficients λ_r, not all zero, with $\sum_r \lambda_r = 0$ and $\sum_r \lambda_r x_r = 0$. Equivalently,

$$\sum_r \lambda_r \bar{O}_r + \sum_r \lambda_r \tau_r \bar{D}_r = 0.$$

Full column rank of R_{ray} forces every $\lambda_r = 0$, a contradiction. Thus every selection of one point from each of the b rays is affinely independent. Any convex hull that meets these rays in its relative interior has affine dimension at least $b - 1$, so it needs at least b vertices. Therefore $m^\star_+ \geq b$.

The upper bound for $n \geq k$ retains the original k slots. Consider $n < k$. Select any positive τ_r and let $x_r = O_r + \tau_r \widetilde{D}_r$. The lower bound argument shows that the n points are affinely independent. Let $\bar{x} = n^{-1} \sum_r x_r$ and choose a constant $\lambda > 1$. Define the n common values

$$c_j = \bar{x} + \lambda(x_j - \bar{x}).$$

Every x_r has a strictly positive barycentric representation in this expanded simplex:

$$x_r = \sum_{j=1}^{n} \alpha_{rj} c_j, \qquad \alpha_{rr} = \frac{1}{\lambda} + \frac{1 - \lambda^{-1}}{n}, \quad \alpha_{rj} = \frac{1 - \lambda^{-1}}{n} \quad (j \neq r). \tag{15}$$

Set $Y_r = Z_r/\tau_r$ and $y_{rj} = Y_r \alpha_{rj}$. All weights are positive, and substitution into Equation 12 gives exact preservation.

It remains to realize the weight table with shared keys. For dot product attention, let Q contain the n protected queries as rows. Each weight column requires

$$Q\mathbf{k}_j = \log y_{\cdot j}.$$

Full row rank of Q makes this system solvable for every column when $n \leq d_k$. For a monotone soft cap, apply the inverse score function before the same solve; every target logit must lie strictly inside the cap range. This proves Theorem 5. The dimensional requirements are visible: the lower certificate needs $2b \leq d_v + 1$, and the constructive key solve needs $n \leq d_k$.

B.8 Dense common slot count

A dense cache requires every physical KV head to use the same sequence count. For heads satisfying the key-realizability conditions, let $m^\star_h$ be the common score-relaxed and physical positive-slot count of head h. The score-relaxed feasible set is upward closed: duplicate any value and split its positive weight between the copies. Hence the minimum common score-relaxed dense count is

$$M^\star = \max_h m^\star_h. \tag{16}$$

Under dot product attention, the same score split remains key realizable when each joint query matrix has full row rank. For a layer with uniform GQA width whose heads satisfy Theorem 5, Equation 16 becomes $M^\star = \min(k, tg)$. The sampled head study tests individual $m_h^\star$; the whole prefix study tests the common dense construction across all heads.

B.9 Composition of disjoint exact replacements

Exact ratio preserving replacements of disjoint groups compose. Fix a query with original output $O = N/Z$. For a replaced group s, let its changes to the unnormalized numerator and denominator be ΔN_s and ΔZ_s. Equation 4 is equivalent to

$$\Delta N_s = O \Delta Z_s.$$

For pairwise disjoint groups, the changes add. Therefore

$$N' = N + \sum_s \Delta N_s = O\left(Z + \sum_s \Delta Z_s\right) = OZ',$$

which gives $N'/Z' = O$. This argument permits separate synthetic keys and values for each group and requires their visibility states to match the groups they replace. The partition construction in Appendix E tests this algebra inside native cache execution.

B.10 Proof of the whole prefix compiler

Set S to all L visible prefix positions of one physical KV head. The T protected positions and g mapped query heads give $n = Tg$ constraints. Let $z_r = \sum_{i \in S} a_{ri}$ be the original prefix weight mass and set

$$\tau_r = Z_r / z_r, \qquad x_r = O_r + \tau_r \widetilde{D}_r.$$

Use the expanded values and coefficients from Equation 15 and set $y_{rj} = z_r \alpha_{rj}$. The replacement preserves the original prefix weight mass because $\sum_j y_{rj} = z_r$, and

$$\sum_j \frac{y_{rj}}{Z_r}(c_j - O_r) = \frac{z_r}{Z_r}(x_r - O_r) = \widetilde{D}_r.$$

Theorem 1 gives exact normalized output preservation.

Full row rank of each joint Q_h realizes the n positive weight columns as synthetic keys for physical head h. Every physical head uses the same $M = n = Tg$ sequence positions, although its synthetic key and value tensors differ. The shared-key solve incorporates the assigned logical positions, including the executed positional transform. Stacking the head tensors therefore gives a valid dense cache of shape $M \times H_{kv}$ at every layer. Applying the construction independently at each layer proves Theorem 6. Proposition 7 then preserves a declared reference trace by induction.

B.11 Compiler procedure, tensor shapes, and cost

The whole-prefix compiler applies the constructive proof directly. For one layer and physical KV head, let $n = Tg$. Its captured inputs are the joint query matrix $Q \in \mathbb{R}^{n \times d_k}$, original outputs and ratio directions $O, \widetilde{D} \in \mathbb{R}^{n \times d_v}$, prefix weight masses $z \in \mathbb{R}^n_{>0}$, and the logical positions assigned to the synthetic slots. The compiler performs four steps:

1. Verify the joint query rank and score range, then form the selected ray points $X \in \mathbb{R}^{n \times d_v}$ with $x_r = O_r + (Z_r/z_r)\widetilde{D}_r$.
2. Uniformly expand the n ray points about their centroid to form $C \in \mathbb{R}^{n \times d_v}$ and its strictly positive convex-coefficient table $A \in \mathbb{R}^{n \times n}$. Set $Y_{rj} = z_r A_{rj}$.
3. Apply the inverse attention-score function to Y, then solve the joint system $QK^{*\top} = G$ for $K^* \in \mathbb{R}^{n \times d_k}$. The executed positional transform is inverted for the assigned logical positions.
4. Repeat across physical heads and layers, then stack the results into dense key and value tensors with sequence dimension n.

The coefficient formula remains valid when the ray points are affinely dependent. Whole-prefix compilation requires query rank for the key solve, not affine independence of the selected values.

Given captured attention statistics, forming the values and weight table costs $O(n^2 d_v)$. A dense thin factorization and the n key right-hand sides cost $O(n^2 d_k + n^3)$ per physical head. Across N_ℓ layers and H_{kv} physical heads, the construction cost is therefore

$$O\big(N_\ell H_{kv}\left[n^2(d_k + d_v) + n^3\right]\big).$$

Capturing the protected statistics still reads the original prefix; with standard dense attention it adds $O(N_\ell H_{kv} nL(d_k + d_v))$ arithmetic. The compiled sequence dimension is independent of L, but the current offline compiler does not remove this one-time source pass.

B.12 Query-indexed attention states and shared KV representations

A known query can store its prefix contribution without storing prefix keys. For protected instance r, define its prefix-only output and log mass by

$$u_r = \frac{\sum_{i\in S} a_{ri} v_i}{z_r}, \qquad z_r = \sum_{i\in S} a_{ri}, \qquad \ell_r = \log z_r.$$

Let (u_r^C, ℓ_r^C) be the corresponding state for the disjoint visible complement. The standard attention-state composition identity gives

$$h_r = \operatorname{logaddexp}(\ell_r, \ell_r^C), \qquad O_r = e^{\ell_r - h_r} u_r + e^{\ell_r^C - h_r} u_r^C. \tag{17}$$

For an empty complement, its contribution is zero and its log mass is $-\infty$. Multiplying each local output by its mass recovers the local numerator; adding the numerators and masses proves the identity. Hydragen uses this blockwise decomposition, and FlashInfer implements attention-state composition (Juravsky et al., 2024; Ye et al., 2025). The identity holds for both dot-product and soft-capped scores because it needs only positive unnormalized weights. It is an established identity, not a new result here.

For a declared T-step trace, the states can be indexed by layer, step, and query head. One physical KV head serves Tg query-head instances, so the table stores $Tg(d_v + 1)$ scalars. It requires no rank or key-solve condition. The same induction as Proposition 7 preserves the declared trace in exact arithmetic: the stored state is used at its matching query instance, and the appended-token contribution is computed normally. Both representations require the reference queries in advance. Neither establishes exactness for a changed query. The cited shared-prefix systems implement the merge identity; their documented kernels still receive prefix KV tensors, rather than implementing this persistent table of future-query states.

Our whole-prefix construction realizes exactly these states in shared KV form. By Equation 3,

$$\widetilde{D}_r = \frac{z_r}{Z_r}(u_r - O_r), \qquad x_r = O_r + \frac{Z_r}{z_r}\widetilde{D}_r = u_r.$$

The construction chooses common values and weights with $\sum_j y_{rj} = z_r$ and $\sum_j y_{rj} c_j = z_r u_r$, then realizes the weight columns using shared keys. The rank and score-range conditions are needed for this last representation step, not for the existence of the attention state. The more general ratio identity also allows numerator and denominator changes that cancel against a fixed complement; the whole-prefix construction uses its mass-preserving case.

The storage comparison makes this constraint explicit. Per layer and physical KV head, the original prefix, query-indexed states, and our $M = Tg$ compiled cache store, respectively,

$$L(d_k + d_v), \qquad Tg(d_v + 1), \qquad Tg(d_k + d_v)$$

scalars, excluding the identical appended-token cache. At equal element width, the state table is smaller than this compiled cache whenever $d_k > 1$, with size ratio $(d_v + 1)/(d_k + d_v)$. If precisions differ, the comparison must instead count the bytes of values, keys, and log masses separately. The state table retains the source statistics directly and avoids the additional shared-key solve. Its lookup and merge cost is $O(d_v)$ per query in addition to complement attention, whereas the compiled prefix evaluates its synthetic entries. We report no measured runtime comparison between these representations. The slot theorems concern shared KV representations, not a minimum over arbitrary encodings of a known trace.

B.13 FULL-QUERY COEFFICIENT CHARACTERIZATION

The finite query results have a separate full query consequence for dot product attention. We retain it here because it describes the limiting scope without interrupting the main finite query argument.

Theorem 9 (Full query coefficient characterization)**.** *For a fixed visibility consistent one slot replacement, exactness for every $q \in \mathbb{R}^d$ holds if and only if the summed vector coefficient of every distinct induced exponential frequency is zero. If all frequencies in the uncollected expansion are pairwise distinct, these equations force $v_i = c$ for every original entry i.*

For dot-product attention, let $a_i(q) = e^{q^\top \mathbf{k}_i}$ and let the replacement score be $y(q) = we^{q^\top \mathbf{k}^*}$, $w > 0$. Before dividing by the original denominator, the exact ratio identity is equivalent to

$$F(q) = \sum_{j\in C}\sum_{s\in S} e^{q^\top(\mathbf{k}_j+\mathbf{k}_s)}(v_j - v_s) + w\sum_{s\in S} e^{q^\top(\mathbf{k}^*+\mathbf{k}_s)}(c - v_s) + w\sum_{j\in C} e^{q^\top(\mathbf{k}^*+\mathbf{k}_j)}(c - v_j) = 0. \tag{18}$$

For each induced frequency λ, define the vector coefficient

$$C_\lambda = \sum_{\substack{j\in C, s\in S\\ \mathbf{k}_j+\mathbf{k}_s=\lambda}} (v_j - v_s) + w\sum_{\substack{s\in S\\ \mathbf{k}^*+\mathbf{k}_s=\lambda}} (c - v_s) + w\sum_{\substack{j\in C\\ \mathbf{k}^*+\mathbf{k}_j=\lambda}} (c - v_j). \tag{19}$$

Proof of Theorem 9. The derivation in Appendix B.2, before division by Z, gives

$$y\big(c(Z_0 + z) - (N_0 + N_S)\big) - (Z_0 N_S - zN_0) = 0.$$

Substituting the exponential scores and grouping terms gives equation 18. After collecting equal frequencies, $F(q) = \sum_\lambda e^{q^\top \lambda} C_\lambda$. Exponentials with distinct vector frequencies are linearly independent on $\mathbb{R}^d$, coordinate by coordinate. To see this, choose a direction r for which the finitely many $r^\top \lambda$ are distinct and restrict to $q = tr$; one-dimensional exponentials are linearly independent. Hence $F(q) = 0$ for every q if and only if every $C_\lambda = 0$.

If all individual frequency vectors in the uncollected expansion equation 18 are pairwise distinct, the frequency $\mathbf{k}^* + \mathbf{k}_i$ has coefficient $w(c - v_i)$ for each $i \in S \cup C$. Since $w > 0$, every coefficient equation gives $v_i = c$. The cross-group frequencies then vanish as well. Conversely, if all values equal c, both the original and replaced outputs equal c for every query. □

B.14 AN EXPLICIT COLLISION EXCEPTION

Let the replaced group contain $(\mathbf{k}_0, v)$ and $(\mathbf{k}_1, u)$, and let its complement contain $(\mathbf{k}_0, 2u - v)$ and $(\mathbf{k}_1, u)$, where $\mathbf{k}_0 \neq \mathbf{k}_1$ and $v \neq u$. Write $a = e^{q^\top \mathbf{k}_0}$ and $b = e^{q^\top \mathbf{k}_1}$. The original denominator is $2(a + b)$, and its numerator is

$$av + bu + a(2u - v) + bu = 2(a + b)u.$$

Thus the original output is u. After replacing the group by $(\mathbf{k}_0, v)$, the denominator is $2a + b$ and the numerator is

$$av + a(2u - v) + bu = (2a + b)u.$$

The replaced output is also u for every q. The values are nonuniform when $v \neq u$. Exact key collisions cause several terms in equation 19 to share a frequency, so their coefficients can cancel. This example establishes existence, not a complete classification of collision patterns.

C COMPLETE EXPERIMENTAL DETAILS

Table 2 in the main paper summarizes the three empirical stages. This appendix records the exact public revisions, query selection, candidate sets, numerical criteria, and complete stratified results.

C.1 MODELS AND FIXED REVISIONS

Table 3 records the exact public checkpoint revisions and sampled target layers. All six checkpoints enter the one-slot study. The conditional-law and whole-prefix studies use Qwen2.5-0.5B, Qwen3-4B-2507, Gemma-2-2B, and Llama-3.1-8B; the whole-prefix study covers every layer rather than only the listed targets. Layer indices are zero-based. TensorFloat-32 (TF32) is disabled, and model activations are captured in float32. The corresponding public repository IDs are `Qwen/Qwen2.5-0.5B-Instruct`, `Qwen/Qwen2.5-7B-Instruct`, `Qwen/Qwen3-4B-Instruct-2507`, `google/gemma-2-2b-it`, `google/gemma-2-9b-it`, and `meta-llama/Llama-3.1-8B-Instruct`, in table order.

Table 3: Public model revisions and sampled target layers for the one-slot study. The conditional-law layer selection is specified below.

Model	Public revision	Target layers	H_q	H_{kv}	Short length
Qwen2.5-0.5B	`7ae557604adf67be5041 7f59c2c2f167def9a775`	2,12,23	14	2	11
Qwen2.5-7B	`a09a35458c702b33eeac c393d103063234e8bc28`	2,14,27	28	4	11
Qwen3-4B-2507	`cdbee75f17c01a7cc42f 958dc650907174af0554`	2,18,35	32	8	11
Gemma-2-2B	`299a8560bedf22ed1c72 a8a11e7dce4a7f9f51f8`	2,13,25	8	4	12
Gemma-2-9B	`11c9b309abf73637e4b6 f9a3fa1e92e615547819`	2,21,41	16	8	12
Llama-3.1-8B	`0e9e39f249a16976918f 6564b8830bc894c89659`	2,16,31	32	8	12

C.2 PROTECTED-QUERY SELECTION

ARC-Challenge contains 1,172 rows, GSM8K contains 1,319, and OpenBookQA contains 500. We choose eight evenly spaced sample identifiers in each dataset and record decode steps 0, 8, and 15 from a 16-token continuation. For a given model, every short prompt begins with the same fixed system prefix; concatenating the three datasets gives 72 protected queries per query head on one prefix cache. The long-prefix case constructs one fixed model-tokenized 512-token WikiText-103 prefix and appends 384 document-grounded suffix prompts. The same eight-identifier, three-step rule gives 24 protected queries per query head on that fixed cache. Length bucketing is used only for equivalent generation; query statistics are united across all buckets and short benchmarks before scanning.

C.3 REPORTED ERROR METRICS

The mathematical replay reports relative output error for each protected constraint. If O_r and $\widehat{O}_r$ are the original and reconstructed head outputs, the reported value is

$$e_{\text{out}} = \max_r \frac{\|\widehat{O}_r - O_r\|_2}{\max(\|O_r\|_2, \epsilon_{\text{tiny}})}, \tag{20}$$

where ϵ_{tiny} is the smallest positive normal number in the replay data type. The 10^{-9} criterion applies to this maximum over the protected constraints.

Native trace errors use the same relative L_2 form. For a field x, such as logits or attention context, the error at step s is

$$e_x(s) = \frac{\|\widehat{x}_s - x_s\|_2}{\max(\|x_s\|_2, \epsilon_{\text{tiny}})}. \tag{21}$$

Native errors are evaluated in binary64, with its smallest positive normal number as ϵ_{tiny}. Each model-prompt case contributes its maximum over its protected steps. Figures and tables report the median and maximum of these case-level values across the applicable cases. Displayed diagnostics use about three significant digits; the machine-readable supplement retains the recorded precision.

C.4 QUERY-COUNT, SOURCE, AND DECODE-STEP SENSITIVITY

The saved statistics support additional query-set sensitivity tests without a new model forward pass. We apply the unchanged ray test and float64 guard to query pairs specified by fixed index rules. The query-count analysis uses the first element of the schedule fixed before candidate scanning. The source-by-step analysis covers every recorded cell using its first and last saved prompt identifiers, so no pair is selected by its outcome.

Table 4: Infeasibility coverage as the protected-query budget grows. One query cannot reject by Corollary 2; the first scheduled pair certifies all tested layer cases. Additional pairs add no rejection.

Protected-query rule	Certified / tested	New certificates
One query	0 / 2,786,085 by theorem	0
First scheduled pair	2,786,085 / 2,786,085	2,786,085
Fixed eight-pair prefilter	2,786,085 / 2,786,085	0
Complete pair schedules	2,786,085 / 2,786,085	0

The first scheduled pair attains complete coverage at the smallest query count that can certify infeasibility. It is the recorded witness for 2,784,669 layer cases. Evaluating the same pair across the saved query heads rejects the remaining 1,416 cases, and long-double replay confirms every decision. Only one case has a float64 margin at most 10^{-4}; its replayed margin remains positive at 6.32×10^{-5}. At the earliest target layer, query head 0 and the same pair directly reject all 928,695 groups.

Table 5: Two-query short-prefix sensitivity within one benchmark and one decode step. Prompt IDs are the first and last entries in each saved eight-prompt order. Every cell reports all six models, three group sizes, and three target layers.

Benchmark	Prompt IDs	Step 0	Step 8	Step 15
ARC-Challenge	501, 1003	11,979 / 11,979	11,979 / 11,979	11,979 / 11,979
GSM8K	376, 941	11,979 / 11,979	11,979 / 11,979	11,979 / 11,979
OpenBookQA	356, 0	11,979 / 11,979	11,979 / 11,979	11,979 / 11,979

Each cell in Table 5 uses only two queries from one benchmark and decode step. Every pair rejects its complete 11,979-case short-prefix cell, giving 107,811 rejections across the nine cells. Long-double replay confirms all 1,080 decisions with margin at most 10^{-4}; the smallest margin remains positive and changes from 5.08×10^{-10} to 5.50×10^{-10}.

The same-step result also holds on the long prefix. For each step, we use the first and last saved suffix identifiers, 0 and 383, at that step. Each fixed pair independently rejects all 2,774,106 long layer-candidate cases, including exhaustive pairs and the fixed triple and quadruple selections. Across the three steps, all 38,831 cases with margin at most 10^{-4} remain rejected in long-double arithmetic; the minimum replayed margin is 1.203×10^{-11}. Together, these tests establish complete coverage for every recorded long-prefix decode step while keeping the model, candidate, and numerical choices fixed.

C.5 CANDIDATE GENERATION

Table 6 gives the per-model long candidate design. The short design enumerates all combinations of the actual 11- or 12-token system prefix. Long selected triples and quadruples contain every in-range arithmetic progression whose stride is a power of two, unioned with 2,048 uniform groups from each of four fixed seeds. Duplicates are removed and the final groups are sorted lexicographically before scanning. The reproduction program regenerates the candidate arrays deterministically.

C.6 MODEL-BY-REGIME COVERAGE

Every model and candidate regime has zero unresolved layer cases. The six model specific long pair cells each contain 130,816 exhaustive candidates. The selected long triple and quadruple cells

Table 6: Long-prefix candidate design per model. Population is the complete combination count for 512 positions.

Group size	Population	Selected	Policy
2	130,816	130,816	all combinations
3	22,238,720	11,778	power-of-two progressions and fixed-seed groups
4	2,829,877,120	11,523	power-of-two progressions and fixed-seed groups

retain their fixed candidate policy. Table 7 gives the exact accounting by model, and Table 8 gives the corresponding mechanism and margin breakdowns.

Table 7: Candidate and layer-case accounting by model. Every candidate is tested at three target layers and rejected at all three.

Model	Short	Long	Candidates	Layer cases	Unresolved
Qwen2.5-0.5B	550	154,117	154,667	464,001	0
Qwen2.5-7B	550	154,117	154,667	464,001	0
Qwen3-4B-2507	550	154,117	154,667	464,001	0
Gemma-2-2B	781	154,117	154,898	464,694	0
Gemma-2-9B	781	154,117	154,898	464,694	0
Llama-3.1-8B	781	154,117	154,898	464,694	0
Total	3,993	924,702	928,695	2,786,085	0

C.7 Rejection mechanism and margin by candidate regime

The dominant rejection mechanism is consistent across the long candidate regimes. Table 8 uses each recorded first witness and its guarded float64 margin. Supporting-line separation accounts for at least 98.85% of every long regime. The selected triple and quadruple sets show higher median and first-percentile guarded margins than the exhaustive pair regime.

Table 8: Rejection cause and guarded margin by long group size. Long pairs are exhaustive; triples and quadruples are fixed selections.

k	Set	Layer cases	Geometric	Sign
2	exhaustive	2,354,688	2,342,099	12,589
3	fixed	212,004	210,120	1,884
4	fixed	207,414	205,023	2,391

k	Set	Median margin	1st percentile
2	exhaustive	0.329	1.14×10^{-4}
3	fixed	0.482	9.32×10^{-4}
4	fixed	0.539	2.83×10^{-3}

The two long selection policies give the same qualitative result. Table 9 partitions each fixed triple and quadruple set into its complete multiscale component and its fixed-seed component. Every cell has complete rejection, and geometric separation accounts for at least 98.74% within each cell.

Complete rejection and geometric dominance hold separately for the structured and uniform candidate policies. Their geometric shares differ by at most 0.15 percentage points within each group size, and the fixed-seed groups have slightly higher median guarded margins. The two rejection causes capture complementary properties of the ray system. A geometric rejection means that the output displacement has a guarded component outside the span of the two ray directions, so even their supporting lines do not intersect. A sign rejection has a supporting-line intersection but requires a nonpositive coefficient, placing the intersection outside at least one positive ray.

Table 9: Long selected groups by construction policy. The multiscale component contains all power-of-two arithmetic progressions; the seeded component contains the remaining groups added by four fixed seeds.

k	Set	Layer cases	Geometric	Sign
3	multiscale	64,548	63,946 (99.07%)	602
3	seeded	147,456	146,174 (99.13%)	1,282
4	multiscale	59,958	59,205 (98.74%)	753
4	seeded	147,456	145,818 (98.89%)	1,638

k	Set	Median margin	1st percentile
3	multiscale	0.462	1.09×10^{-3}
3	seeded	0.492	8.51×10^{-4}
4	multiscale	0.516	3.84×10^{-3}
4	seeded	0.549	2.43×10^{-3}

All 11,979 short layer cases are supporting-line rejections: 1,089, 3,465, and 7,425 for group sizes two, three, and four. Across all regimes, supporting-line separation gives 2,769,221 of 2,786,085 cases. The positive-ray sign condition supplies the remaining 16,864.

C.8 Ray classification and rounding guard

Query pairs are evaluated in a fixed sequence. Within one query head, they are sorted by decreasing witness-index separation and then lexicographically. The first eight form the initial schedule. Cross-query-head pairs use the same ordering after concatenating the query heads mapped to one KV head. Candidates without an early witness are checked against the complete within-head and cross-query-head schedules in long-double arithmetic.

For a query pair, normalize nonzero directions as $u_a = D_a/\|D_a\|$ and $u_b = D_b/\|D_b\|$, set $s = u_a^\top u_b$, let $s_O = \max\{1, \|O_a\|_2, \|O_b\|_2\}$, and solve

$$xu_a - yu_b = O_b - O_a. \tag{22}$$

The normal matrix $A^\top A$ has eigenvalues $1 \pm |s|$, giving condition number $\kappa = (1 + |s|)/(1 - |s|)$. Writing $b = O_b - O_a$, $b_0 = u_a^\top b$, and $b_1 = -u_b^\top b$, the implemented solution is $x = (b_0 + sb_1)/(1 - s^2)$ and $y = (sb_0 + b_1)/(1 - s^2)$. We then compute

$$\rho = \frac{\|xu_a - yu_b - b\|_2}{s_O}, \qquad \eta = \frac{\max(-x, -y)}{\max\{1, |x|, |y|\}}, \qquad \mu = \max\{\rho - \varepsilon_{\text{guard}}, \eta - \varepsilon_{\text{guard}}\}. \tag{23}$$

The fixed float64 guard uses $\gamma_m = m\epsilon/(1 - m\epsilon)$ with $m = 128d + 32k + 2048$. The $128d$ term covers two norms, inner products, and residual formation; $32k$ covers the group contribution accumulation; the constant covers scalar elimination and fixed overhead. The classification bound is $\varepsilon_{\text{guard}} = \gamma_m(1 + \kappa)$. A pair remains unresolved if $\min(\|D_a\|_2, \|D_b\|_2) \le \gamma_m s_O$ or $\kappa > 10^{12}$; otherwise it is rejected only when $\mu > 0$. Thus, the stored margin is already net of the guard. These integers and the condition limit are fixed before scanning. Float64 supplies the main classification; extended precision is used for saved low-margin replay. The guard is not a formal error theorem for the complete data-dependent construction of O, P, D, and normalized directions. Its role is to prevent rejections at the scale of the counted float64 operations. No empirical quantile or tuned threshold enters the scan.

C.9 Controls and score semantics

Table 10 reports the synthetic controls. The feasible rays are not rejected. Both infeasible mechanisms are rejected with large margin, and the constructed mask aggregation has the expected output.

Gemma requires a separate execution-semantic check because its reference score is a scaled dot product followed by logit soft-capping. Table 11 compares context reconstruction under this score with an ordinary dot-product counterfactual. The reference score is consistently about four orders of magnitude closer to the executed context.

Table 10: Synthetic controls for the numerical classifier and aggregation.

Control	Margin	Outcome
Feasible intersecting rays	-7.76×10^{-16}	not rejected
Disjoint supporting lines	9.44×10^{-1}	rejected
Negative required coefficient	1.00	rejected by sign

Table 11: Gemma context reconstruction root-mean-square discrepancy.

Model	Query unit	Reference soft-cap	Dot counterfactual
Gemma-2-2B	ARC	4.42×10^{-7}	4.63×10^{-3}
	GSM8K	4.56×10^{-7}	4.66×10^{-3}
	OpenBookQA	4.80×10^{-7}	3.94×10^{-3}
	Long prefix	4.88×10^{-7}	6.43×10^{-3}
Gemma-2-9B	ARC	6.03×10^{-7}	8.45×10^{-3}
	GSM8K	5.85×10^{-7}	7.52×10^{-3}
	OpenBookQA	5.60×10^{-7}	7.52×10^{-3}
	Long prefix	6.31×10^{-7}	1.11×10^{-2}

C.10 Count hierarchy and low-margin replay

The six model-level candidate totals are 154,667 for each Qwen model and 154,898 for each Gemma or Llama model. Each candidate is evaluated at three target layers, so the 2,786,085 layer-group decisions are three nested checks of the 928,695 groups. The recorded causes are 2,769,221 geometric residuals and 16,864 negative coefficients. A query-head witness in the fixed eight-pair prefilter resolves each decision; the broader searches add no rejection.

Low-margin replay covers the 23,394 decisions whose stored margin is at most 10^{-4}. All are geometric-residual witnesses. Reconstruction from the saved O, P, and V arrays followed by extended-precision replay preserves their rejection and mechanism. The minimum replayed margin is 8.442×10^{-12}, the minimum raw residual is 8.444×10^{-12}, and the largest normal-matrix condition number is 6.697×10^{3}.

D Conditional slot law results

D.1 Physical GQA evaluation design

The physical GQA evaluation treats the union of mapped query heads as one constraint system. It uses four public checkpoints, two input types, three layers, and two physical KV heads per layer. The layers follow Table 3, except that Gemma-2-2B uses layers 3, 15, and 25. Group sizes are $k \in \{2, 3, 4, 8, 16, 32\}$; protected position counts are $t \in \{1, 2, 3, 4, 6, 8, 9\}$. Candidate policies include the complete short groups and three deterministic long-prefix policies. Each configuration records $n = tg$, $b = \min(k, n)$, the augmented ray rank, the joint query rank, and the predicted count $\min(k, n)$ before construction. Constraint rows follow the fixed step-major protected-position order and ascending mapped query-head order. The lower certificate uses the first b rows in this deterministic order.

All 431,256 configurations satisfy the predicted exact count. Every one of the 60,492 compressible configurations passes the homothetic value construction, positive weight solve, direct ratio replay, and finite checks. No configuration has $m = 0$, and none is unresolved. The model counts differ because the GQA width changes $n = tg$ and the candidate populations differ between workloads.

Table 12: Physical GQA slot-law results by model. A compressible configuration has $0 < m_+^\star < k$; a retained configuration has $m_+^\star = k$.

Model	g	Configurations	Compressible	Retained
Qwen2.5 0.5B	7	84,252	6,012	78,240
Qwen3 4B	4	84,252	7,164	77,088
Gemma 2 2B	2	131,376	36,192	95,184
Llama 3.1 8B	4	131,376	11,124	120,252
Total		431,256	60,492	370,764

D.2 Exact counts by group size and protection budget

The compressible region follows the formula rather than a fitted threshold. Table 13 reports the number of evaluated nontrivial positive constructions. Groups of size two have no room because every physical head has $g \geq 2$. Larger groups remain compressible until tg reaches k.

Table 13: Nontrivial exact constructions by original group size and protected position count. Each count is a subset of the same 60,492 constructions.

Stratum	1	2	3	4	6	8	9
By t	32,484	11,124	10,548	2,880	2,304	576	576

k	2	3	4	8	16	32
Count	0	3,216	6,516	31,176	7,488	12,096

The ARC exhaustive policies contribute 334,488 KV-head configurations and 36,300 constructions. Each of the three long prefix policies contributes 32,256 configurations and 8,064 constructions. The counts show that the same law holds across all three selected long-prefix policies.

D.3 Rank, score, and reconstruction diagnostics

The rank certificate uses a fixed float64 guard rather than a fitted threshold. Let $R \in \mathbb{R}^{r\times c}$ be the augmented matrix after dividing each nonzero column by its norm. With

$$p = 8rc + 32\min(r, c), \qquad \gamma_p = \frac{p\epsilon}{1 - p\epsilon},$$

the arithmetic term is $\gamma_p \|R\|_F$. The total guard adds the measured singular value decomposition (SVD) reconstruction residual and the left and right orthogonality defects:

$$G = \gamma_p \|R\|_F + \|R - U\Sigma V^\top\|_F + \sigma_{\max}\left(\|U^\top U - I\|_2 + \|VV^\top - I\|_2\right).$$

Full column rank is certified only when $\sigma_{\min} > G$. A zero column, too few rows, or a singular value inside the guard remains unresolved. Column equilibration preserves rank and prevents the arbitrary scale of a ray direction from controlling the decision. This is a certificate for the supplied float64 matrix, not an interval proof for upstream model activations.

Among the 60,492 constructed representations, the guarded rank margin has minimum 1.40×10^{-7} and median 1.14×10^{-2}, while the joint-query condition number has median 3.64 and maximum 100. The float64 output reconstruction error has median 2.01×10^{-16} and maximum 7.08×10^{-14}; the shared-key solve residual has median 4.44×10^{-15} and maximum 5.01×10^{-13}. FP32 replay is reported separately, with median 1.26×10^{-8} and maximum 4.62×10^{-6}. For the capped Gemma score, the minimum target-range margin is 36.2, and no range failure occurs.

Together, the guarded rank margins and solve residuals place the evaluated constructions inside the certificate conditions. The conditional theorem itself remains an exact-arithmetic statement. Rank failure leaves a decision unresolved, while a score-range or key-solve failure prevents the constructive upper bound.

D.4 Protected and held out query behavior

Protected exactness does not imply improvement on queries omitted from the construction. Among the 60,492 constructed representations, 28,672 have lower held-out relative error than same-slot mean pooling, a rate of 47.4%. The median held-out errors are 0.210 for the exact protected construction and 0.091 for mean pooling. This result matches the theorem's scope: the constructed weight table satisfies the declared finite equations and makes no distributional objective for additional queries.

E Whole prefix and native execution results

E.1 Breadth and whole-prefix coverage

The breadth evaluation extends proper-group cells to five layers and every physical KV head. It resolves 219,598 of 219,600 evaluated configurations. Both unresolved cases belong to one near-singular Qwen2 long-prefix rank cell. All 45,620 eligible configurations admit the constructed replacement. These cases test the conditional minimum law for proper groups.

The four-step whole-prefix study uses the eight fixed model-prompt cases in Table 2. Table 14 reports the layer-head checks nested within those cases. The construction and numerical criteria are satisfied for all 1,392 instances.

Table 14: Whole-prefix layer-head checks and native four-step trace matches, aggregated across the ARC and long-prefix cases.

Model	Construction and numerical checks	Full native traces matched
Qwen2.5 0.5B	96 / 96	2 / 2
Qwen3 4B	576 / 576	2 / 2
Gemma 2 2B	208 / 208	2 / 2
Llama 3.1 8B	512 / 512	2 / 2
Total	1,392 / 1,392	8 / 8

The maximum extended-precision relative output error is 6.31×10^{-14}, the maximum float64 output error is 8.22×10^{-14}, and the maximum absolute score residual is 9.09×10^{-13}. Each remains below 10^{-12}.

E.2 Capacity across models and workloads

Table 15 gives the sequence dimensions for the four-step comparison. The constructed count is $M = Tg$. The ARC cases therefore achieve $2.68\times$ to $9.38\times$ slot compression, and the longer cases achieve $18.71\times$ to $65.50\times$.

Table 15: Sequence capacity for four protected decode positions. Compression is L/M. With the same data type and unchanged layer and head dimensions, the KV tensor-byte reduction is the equivalent quantity $1 - M/L$; it is not measured allocator or peak-device memory.

	ARC		Long prefix	
Model	$L \to M$	L/M	$L \to M$	L/M
Qwen2.5-0.5B	$75 \to 28$	$2.68\times$	$524 \to 28$	$18.71\times$
Qwen3-4B-2507	$75 \to 16$	$4.69\times$	$524 \to 16$	$32.75\times$
Gemma-2-2B	$75 \to 8$	$9.38\times$	$524 \to 8$	$65.50\times$
Llama-3.1-8B	$76 \to 16$	$4.75\times$	$524 \to 16$	$32.75\times$

E.3 SAME BUDGET NATIVE CONTROLS

The native study uses three heuristic controls to isolate exact construction at the same sequence count. Mean pooling averages keys and values within consecutive bins, uniform sampling keeps evenly spaced positions, and attention top M retains the positions with the largest mean protected attention across steps and mapped query heads. Table 16 reports teacher-forced errors; greedy decoding gives the same full-trace match counts.

Table 16: Native protected traces. Whole-prefix compilation and the three heuristic controls use M slots; the separate four-partition composition test uses $4M$ slots and has seven applicable cases. A trace matches only when all four protected tokens agree. Errors follow Equation 21; each case contributes its four-step maximum, followed by the median and maximum across applicable cases.

Representation	Trace matches	Logits median / max	Context median / max
Exact whole prefix	8 / 8	9.18×10^{-7} / 1.08×10^{-6}	8.16×10^{-7} / 1.24×10^{-6}
Mean pooling	0 / 8	1.172 / 2.103	2.395 / 3.891
Uniform sampling	2 / 8	0.740 / 1.271	0.851 / 0.946
Attention top M	6 / 8	0.249 / 0.400	0.437 / 0.599
Partition ($4M$ slots)	7 / 7	9.59×10^{-7} / 1.77×10^{-6}	8.18×10^{-7} / 2.33×10^{-6}

The partition construction replaces four disjoint cache groups and validates the composition proof in Appendix B.9. It is applicable in seven model-prompt cases, and all 5,376 layer-head constructions meet the numerical criteria. Qwen2 ARC has no valid partition under the deterministic group policy.

E.4 SCOPE AFTER THE PROTECTED TRACE

Every exact whole-prefix case preserves steps 1 through 4 under teacher-forced and greedy execution. The matching prefixes over the full eight step continuations have lengths $4, 5, 5, 5, 4, 6, 4, 6$. All eight cases have diverged by the final step. This result provides a direct scope diagnostic: the construction preserves the finite queries used to form it, and later queries may leave that set.

The proper-group held-out comparison gives the same conclusion. Exact constructions outperform same slot mean on 22,342 of 45,620 held out comparisons, or 49.0%. Neither diagnostic supports an unknown-query generalization claim.

E.5 WHOLE PREFIX LOWER BOUND

The whole-prefix setting has a structural rank constraint that is absent from general proper groups. At decode position t, its complement contains only the t newly appended values. These values are shared by the g query heads mapped to a physical KV head. The following identity explains why the full-column-rank certificate in Theorem 5 cannot establish the whole-prefix minimum for the tested GQA models.

Proposition 10 (Whole-prefix augmented rank)**.** *Fix a layer and physical KV head, and protect T positions with $n = Tg$ query-head constraints. Let the group contain the complete original prefix, and suppose its visible complement consists of at most T shared appended values. Then the augmented matrix in Equation 7, formed from all n constraints, satisfies*

$$\operatorname{rank}[U, \bar{D}] \le \min(d_v + 1, n + T). \tag{24}$$

Here $U = [\bar{O}_1, \dots, \bar{O}_n]$ and $\bar{D} = [\bar{D}_1, \dots, \bar{D}_n]$. For $g > 1$, this rank is strictly less than $2n$.

Proof. Index a constraint by $i = (t, h)$. Let μ_i be its normalized attention mass on the original prefix, let v_s^{new} be the value appended at position s, and let p_{is}^{out} be its attention probability. Set

$p_{is}^{\text{out}} = 0$ for $s > t$. Normalization gives

$$1 - \mu_i = \sum_{s=1}^{T} p_{is}^{\text{out}},$$

$$\widetilde{D}_i = (1 - \mu_i) O_i - \sum_{s=1}^{T} p_{is}^{\text{out}} v_s^{\text{new}}. \tag{25}$$

Define $W = [(v_1^{\text{new}}, 1), \ldots, (v_T^{\text{new}}, 1)]$ and $P_{si} = p_{is}^{\text{out}}$. The zero last coordinate of each $\bar{D}_i$ then yields

$$\bar{D} = U \operatorname{diag}(1 - \mu) - WP, \qquad \operatorname{col}[U, \bar{D}] \subseteq \operatorname{col}[U, W]. \tag{26}$$

The latter matrix has $n+T$ columns and d_v+1 rows. If $g > 1$, then $n = Tg > T$, so $n+T < 2n$. □

This rank bound limits the lower certificate, not the construction. All four evaluated models have $g > 1$, so the full $2n$-column condition is unavailable for every whole-prefix window. A proper group can have many more complement values and need not satisfy this bound. The mass-preserving whole-prefix construction uses query rank and key realizability, independently of the augmented lower matrix. Its $M = Tg$ representation therefore remains valid, while the actual whole-prefix minimum remains open.

Floating-point rank diagnostics must retain this structural distinction. The four-step panel has no positive lower diagnostic. In the extended window panel, two one-step layer-head matrices appear full rank after their separately rounded columns are normalized. At $T = 1$, a diagnostic least-squares projection onto the shared-complement identity changes the direction entries by at most 1.21×10^{-15}; the normalized smallest singular values then fall from approximately 10^{-11} and 10^{-12} to 10^{-16}. These two rounded-matrix diagnostics are not physical minimum certificates. This projection does not recover the original appended value, which is not stored in the diagnostic arrays. The original observations are retained, and no whole-prefix minimality claim depends on them.

E.6 Arithmetic precision and runtime

Native FP32 logits and hidden states are finite and close, but they are not bitwise equal. The protected token criterion is exact at the discrete output level. Bfloat16 (BF16) replay is a diagnostic outside that criterion. Its maximum whole-prefix operator relative error is 0.585, while the separate partition replay reaches 0.612. A BF16 exactness claim is therefore unsupported.

The same-dtype byte comparison in Table 15 is not the only capacity reference point. Comparing the compiled FP32 cache with a shape-matched two-byte original cache still gives 25.33% to 78.67% reduction on ARC and 89.31% to 96.95% on the long prefixes. These values follow directly from the saved tensor shapes and element widths; they are capacity calculations rather than new native precision measurements.

The construction reduces cache capacity while leaving median decode latency close to the full-cache baseline. Long-prefix median speed ratios are 0.990, 0.987, 1.003, and 0.997 for Qwen2.5, Qwen3, Gemma, and Llama, respectively, where a ratio above one indicates faster execution. Each ratio uses 20 timed forward passes after five warmup passes, with alternating execution order and cache cloning excluded. The recorded setup interval spans 1.14 to 4.72 seconds on the long prefixes. It includes the whole-prefix construction followed by four separate partition constructions, with their numerical checks. It is therefore a combined experiment setup cost, not the cost of compiling one whole prefix. Appendix F times the whole-prefix call alone and reports 0.28 to 1.33 seconds across the eight four-step cases. Both timers exclude reference-query acquisition and native decoding. The combined timer does not support a whole-prefix break-even estimate. The measured capacity reduction is established; an end-to-end acceleration claim requires a separately timed deployment implementation.

Table 17: Capacity across all four protected horizons. Each row combines the two fixed inputs for one model and horizon. The constructed slot count is $M = Tg$, compression is L/M, and saved KV tensor bytes are $100(1 - M/L)\%$ at the same data type. Every listed case preserves all T protected tokens in both execution modes.

			ARC			Long prefix, $L = 524$	
Model	T	M	L	L/M	Saved (%)	L/M	Saved (%)
Qwen2.5-0.5B	1	7	75	10.71	90.67	74.86	98.66
	2	14	75	5.36	81.33	37.43	97.33
	4	28	75	2.68	62.67	18.71	94.66
	8	56	75	1.34	25.33	9.36	89.31
Qwen3-4B	1	4	75	18.75	94.67	131.00	99.24
	2	8	75	9.38	89.33	65.50	98.47
	4	16	75	4.69	78.67	32.75	96.95
	8	32	75	2.34	57.33	16.38	93.89
Gemma-2-2B	1	2	75	37.50	97.33	262.00	99.62
	2	4	75	18.75	94.67	131.00	99.24
	4	8	75	9.38	89.33	65.50	98.47
	8	16	75	4.69	78.67	32.75	96.95
Llama-3.1-8B	1	4	76	19.00	94.74	131.00	99.24
	2	8	76	9.50	89.47	65.50	98.47
	4	16	76	4.75	78.95	32.75	96.95
	8	32	76	2.38	57.89	16.38	93.89

F Protected horizon, capacity, and numerical precision

F.1 Nested reference queries

The horizon study measures how long a compiled prefix can preserve a specified reference continuation. We use the same four model checkpoints and two fixed inputs per model as in Appendix E. One input comes from ARC-Challenge and the other from WikiText-103. The latter has 524 prefix tokens. The ARC prefix has 75 tokens for three models and 76 for Llama. These inputs define eight model-input cases.

Each case supplies one twelve-step reference continuation and four nested protected horizons, $T \in \{1, 2, 4, 8\}$. For each horizon, we construct a new whole-prefix representation using the first T reference queries at every layer and mapped query head. Holding the model, prefix, and reference continuation fixed isolates the effect of the protected horizon. The 32 constructions are nested evaluations of eight cases, not 32 independent samples. The two inputs serve as native execution cases rather than full benchmark evaluations.

All layers and physical KV heads meet the construction criteria at every horizon. There are 1,392 layer-head constructions per horizon and 5,568 in total. Every window preserves its complete protected trace under both teacher-forced and greedy execution, giving 120 matching token comparisons per mode across the nested horizons. At $T = 8$, incorrect cache positions, incorrect physical-head mappings, and corrupted compiled values are detected in every case by the three negative controls.

F.2 Capacity as the horizon grows

The constructed slot count grows linearly with the protected horizon. For g query heads sharing a physical KV head, the whole-prefix construction uses $M = Tg$ slots under its query-rank and score-range conditions. Table 17 reports every model, input, and horizon. Doubling T doubles the slot count and halves the compression ratio. At eight protected steps, the length-524 prefixes retain a slot compression factor of 9.36 to 32.75, corresponding to 89.31% to 96.95% fewer KV tensor bytes at the same data type.

The byte reduction follows from tensor shape and element width. With unchanged layer, head, and channel dimensions, the retained fraction is M/L. This counts the compiled prefix tensors. Reference-query acquisition, model weights, temporary construction tensors, and any retained ref-

Table 18: Arithmetic precision and construction cost on ARC. Error columns are the per-case maxima defined in the text. Time is the single recorded construction duration in seconds. The final column gives the first differing greedy token after the protected horizon; > 12 denotes right censoring at the twelve-step observation limit.

Model	T	ε_{64}	$\varepsilon_{\mathrm{ext}}$	ε_{32}	$\varepsilon_{\mathrm{BF16}}$	$\varepsilon_{\mathrm{logits}}$	Time (s)	First diff.
Qwen2.5-0.5B	1	3.21×10^{-14}	6.44×10^{-14}	1.24×10^{-5}	0.4875	8.00×10^{-7}	1.229	2
	2	5.34×10^{-14}	6.26×10^{-14}	1.10×10^{-5}	0.4983	7.53×10^{-7}	0.259	4
	4	6.70×10^{-14}	6.31×10^{-14}	2.85×10^{-5}	0.5849	9.17×10^{-7}	0.725	5
	8	2.40×10^{-13}	2.14×10^{-13}	5.37×10^{-5}	0.5548	6.05×10^{-6}	0.484	9
Qwen3-4B	1	3.31×10^{-14}	3.28×10^{-14}	6.29×10^{-6}	0.2427	5.62×10^{-7}	0.417	2
	2	2.58×10^{-14}	1.84×10^{-14}	6.62×10^{-6}	0.3981	6.81×10^{-7}	0.690	4
	4	5.21×10^{-14}	2.60×10^{-14}	1.40×10^{-5}	0.5235	1.08×10^{-6}	1.283	6
	8	4.52×10^{-14}	2.94×10^{-14}	2.49×10^{-5}	0.4897	1.29×10^{-6}	4.242	12
Gemma-2-2B	1	5.43×10^{-15}	4.22×10^{-15}	1.46×10^{-6}	0.0318	3.22×10^{-7}	0.116	3
	2	4.36×10^{-15}	6.33×10^{-15}	1.12×10^{-6}	0.0400	3.33×10^{-7}	0.175	3
	4	4.98×10^{-15}	3.72×10^{-15}	1.31×10^{-6}	0.0616	3.29×10^{-7}	0.282	5
	8	5.39×10^{-15}	3.58×10^{-15}	1.29×10^{-6}	0.0429	4.24×10^{-7}	0.494	> 12
Llama-3.1-8B	1	4.20×10^{-15}	3.90×10^{-15}	2.23×10^{-6}	0.0564	9.74×10^{-7}	0.343	2
	2	5.91×10^{-15}	4.86×10^{-15}	1.28×10^{-6}	0.0595	7.92×10^{-7}	0.638	4
	4	1.65×10^{-14}	1.44×10^{-14}	3.39×10^{-6}	0.0862	9.68×10^{-7}	1.148	5
	8	2.30×10^{-14}	1.81×10^{-14}	6.09×10^{-6}	0.1380	1.12×10^{-6}	3.831	9

erence cache contribute separate memory costs. The four-step cases recover the capacities and protected token sequences in Appendix E.

F.3 Arithmetic precision and construction cost

The precision measurements distinguish construction arithmetic from native model execution. We denote the maximum normalized construction-output relative error in binary64 arithmetic by ε_{64}, using Equation 20, and the corresponding extended-precision replay error by $\varepsilon_{\mathrm{ext}}$. For each case, these maxima range over all protected queries, layers, and physical KV heads. The diagnostic operator replays give ε_{32} in FP32 and $\varepsilon_{\mathrm{BF16}}$ in BF16. Finally, $\varepsilon_{\mathrm{logits}}$ is the largest native FP32 logit relative L_2 error over the protected greedy steps, using Equation 21. Tables 18 and 19 report these distinct quantities.

Native FP32 execution preserves protected tokens across the full horizon range. The largest binary64 construction error is 2.40×10^{-13}, and the largest extended-precision replay error is 2.14×10^{-13}. Native FP32 protected-logit error is at most 1.35×10^{-5}, while the separate FP32 operator diagnostic reaches 5.37×10^{-5}. These small nonzero errors distinguish finite-precision agreement from the real-arithmetic identity. The BF16 operator error reaches 0.918, so the numerical evidence supports the reported FP32 execution.

Construction time measures one whole-prefix call, including its numerical checks, after the reference queries are available. The single recorded durations range from 0.114 to 4.293 seconds across the 32 windows. These measurements exclude reference-query acquisition and end-to-end decoding. They provide the observed construction costs for each horizon, without repeated-run timing uncertainty or an inferred online speedup. At $T = 4$, they range from 0.28 to 1.33 seconds. The 1.14 to 4.72 second interval in Appendix E also includes four partition constructions and measures a different workload.

F.4 Behavior after protection and a conditional refresh schedule

The twelve-step continuations show where the finite-query guarantee ends. All protected tokens match, while 31 of the 32 windows first differ after their protected horizon and within the observation window. For $T = 8$, seven of eight cases first differ at steps 9 through 12. Gemma on ARC continues to match through step 12. This last observation is right censored: the first difference, if

Table 19: Arithmetic precision and construction cost on the long prefix. Error columns are the per-case maxima defined in the text. Time is the single recorded construction duration in seconds. The final column gives the first differing greedy token after the protected horizon; > 12 denotes right censoring at the twelve-step observation limit.

Model	T	ε_{64}	$\varepsilon_{\mathrm{ext}}$	ε_{32}	$\varepsilon_{\mathrm{BF16}}$	$\varepsilon_{\mathrm{logits}}$	Time (s)	First diff.
Qwen2.5-0.5B	1	5.71×10^{-14}	5.16×10^{-14}	1.46×10^{-5}	0.3761	5.94×10^{-7}	0.122	2
	2	5.16×10^{-14}	7.88×10^{-14}	1.93×10^{-5}	0.3812	6.16×10^{-7}	0.240	4
	4	8.22×10^{-14}	4.30×10^{-14}	2.97×10^{-5}	0.3594	9.19×10^{-7}	0.660	6
	8	9.22×10^{-14}	7.56×10^{-14}	3.62×10^{-5}	0.9182	1.35×10^{-5}	0.464	9
Qwen3-4B	1	2.85×10^{-14}	1.81×10^{-14}	6.85×10^{-6}	0.1915	3.66×10^{-7}	0.463	2
	2	6.71×10^{-14}	5.37×10^{-14}	1.29×10^{-5}	0.6890	4.49×10^{-7}	0.697	6
	4	6.09×10^{-14}	4.92×10^{-14}	2.24×10^{-5}	0.5367	6.19×10^{-7}	1.326	6
	8	6.53×10^{-14}	2.03×10^{-14}	2.19×10^{-5}	0.4351	9.75×10^{-7}	4.293	9
Gemma-2-2B	1	4.96×10^{-15}	4.65×10^{-15}	1.29×10^{-6}	0.0326	4.76×10^{-7}	0.114	2
	2	4.99×10^{-15}	2.12×10^{-15}	8.39×10^{-7}	0.0488	4.97×10^{-7}	0.177	3
	4	4.95×10^{-15}	3.39×10^{-15}	9.05×10^{-7}	0.0416	4.30×10^{-7}	0.294	7
	8	7.12×10^{-15}	3.10×10^{-15}	1.15×10^{-6}	0.0494	4.71×10^{-7}	0.500	11
Llama-3.1-8B	1	7.12×10^{-15}	5.00×10^{-15}	1.57×10^{-6}	0.0671	1.15×10^{-6}	0.344	2
	2	5.04×10^{-15}	5.41×10^{-15}	1.61×10^{-6}	0.0361	1.04×10^{-6}	0.642	3
	4	1.12×10^{-14}	8.56×10^{-15}	2.40×10^{-6}	0.0399	1.04×10^{-6}	1.161	7
	8	7.31×10^{-15}	5.72×10^{-15}	5.44×10^{-6}	0.0499	1.01×10^{-6}	3.785	12

any, lies beyond the measured window. The final columns of Tables 18 and 19 preserve the case-level observations.

A fixed-block refresh schedule follows conditionally from the construction. Suppose that the reference queries for each successive block are available and satisfy the construction conditions. Protecting a total of N steps in blocks of at most T then requires

$$J = \left\lceil \frac{N}{T} \right\rceil$$

compiled representations, including the first, and $J - 1$ subsequent refreshes. Each full block uses Tg slots per physical KV head. The asymptotic refresh rate is one refresh per T steps, and a shorter final block can use its actual number of protected queries. This calculation links the storage budget to a potential refresh interval when the next reference queries can be supplied.

The experiments evaluate each protected block separately. Acquiring the next reference queries, selecting a refresh time from observed execution, and measuring repeated refresh costs remain algorithmic questions for an online method. The current result supplies the exact construction for each specified block and its measured capacity and precision.

G Reproducibility

The supplement provides a complete one-cell Colab program for the one-slot scan, conditional GQA evaluation, whole-prefix native study, protection-window evaluation, and CPU appendix calculations. It downloads the listed model and data revisions, adapts work chunks to accelerator memory, resumes from checkpoints, and regenerates the reported summaries from fresh outputs. Included comma-separated value (CSV) files contain aggregate slot-law counts and the curves in Figures 2 and 3; raw arrays are regenerated rather than bundled.