WHEN CAN DICTIONARY REUSE COMPRESS A COMPOSABLE ATTENTION CACHE?

Anonymous authors
Paper under double-blind review

ABSTRACT

A compressed attention prefix must save storage while remaining correct when new tokens are appended. Reusing key and value components offers one route, but it also requires preserving which component pairs occur and how much attention mass they carry. We study additive dictionaries with an occupancy matrix and give an exact contraction that avoids expanding all pairs. Under distinct combined keys, the minimum number of fixed positive product components needed for exact composition is the nonnegative rank of occupancy. A full product of r^2 unit-norm tokens has an exact state with $2r$ dictionary rows, whereas every proper raw-value subset has uniform error at least $1/4$ on a specified query ball. Conversely, four-cycle occupancy requires four positive components even on sixteen finite query/tail tests at nonzero tolerance. A joint perturbation bound separates occupancy and dictionary errors. Fixed-dictionary query continuation verifies the predicted covariance derivative. On 540 saved-cache cases, increasing the positive component count reduces error but quickly exhausts the storage saving: exact pair indices cost less and retain exact occupancy. Ordinary clustering still achieves lower output error. The results characterize when dictionary reuse supports composable compression and which information must survive the reduction in storage.

1 INTRODUCTION

Compressing an attention cache creates an engineering obligation that lasts beyond the query used to build the summary. An older prefix may be replaced by a compact state while recent tokens remain exact. Each subsequent query must then combine that state with a growing tail, without retrieving the discarded prefix. A useful representation must therefore save storage, support query-time evaluation and preserve the information needed for this merge. A small error on the prefix's normalized output alone does not establish the last requirement.

One possible source of savings is reuse within keys and values. A cache may contain many distinct tokens assembled from fewer shared components; for example, an object component can be combined with a relation component. This is a motivating structure, not an assumption that arbitrary model caches have exact such decompositions. Storing each complete key/value pair duplicates shared vectors. Keeping representative tokens instead makes a different choice: it retains selected combinations and asks the decoder to approximate the rest. We ask when reusing components can save space while retaining combinations that an original-token subset cannot represent.

Two dictionaries generate keys $k_{ji} = \mu_j + x_i$ and values $v_{ji} = \nu_j + y_i$. An occupancy matrix W records which pairs occur and with what multiplicities. Full independent occupancy permits the attention partition to factor into two dictionary partitions. Dependent occupancy changes the problem: multiplying assignment marginals invents combinations that may never have occurred. Even exact individual dictionaries do not remove this occupancy error. The storage opportunity thus depends on both vector reuse and the cost of preserving their pairings.

Composition exposes why the unnormalized partition is also necessary. A prefix with two zero-key, unit-value tokens and one synthetic token with the same key and value have identical standalone outputs at every query. After a zero-key, zero-value token is appended, their outputs are $2/3$ and $1/2$. Exact online normalization and weighted merging already retain this mass information (Milakov & Gimelshein, 2018; Dao, 2024; Tian et al., 2025). Our question is which dictionary representations can preserve it together with the numerator at lower cost (Figure 1).

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

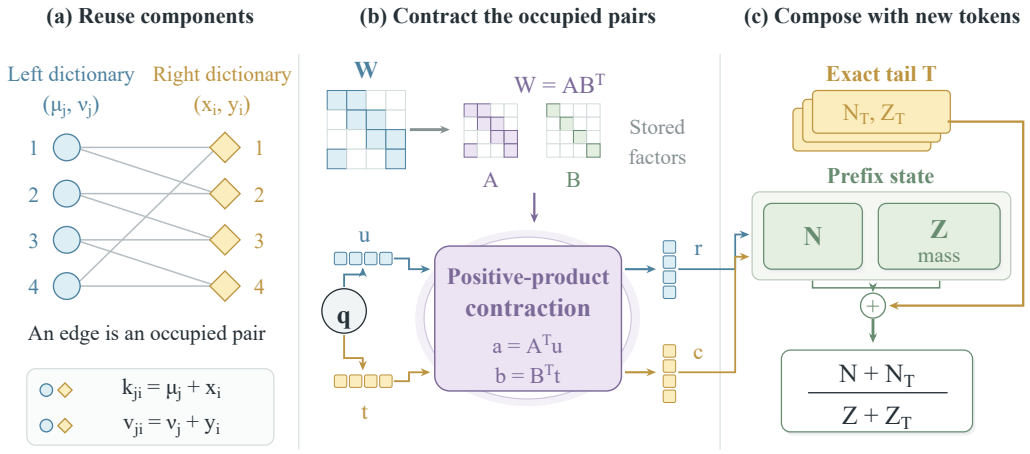

Figure 1: A composable dictionary state preserves pair occupancy and unnormalized mass. (a) Eight edges specify the occupied pairs of two additive key/value dictionaries. (b) The four-cycle uses $A = W_\circ$, $B = I_4$; query-dependent factors yield marginal weights without expanding all pair values. (c) Prefix and exact-tail numerators and partitions are added before normalization. This example illustrates exact composition, not a storage saving: its four positive components cost more than dense occupancy.

Product keys support efficient memory lookup (Lample et al., 2019), and sum-product models provide tractable partition functions (Poon & Domingos, 2011). Building on this algebra, we connect the cache's required merge operation to the complexity of a specified positive representation. With fixed dictionaries and distinct combined keys, exact composition after every tail forces exact occupancy, so its nonnegative rank determines the minimum positive component count. This is a characterization of that interface, rather than a lower bound for every possible compressed encoding.

Two constructions show why the distinction is useful. An $r \times r$ full product has an exact state with $2r$ dictionary rows, whereas every fixed proper subset of original values loses at least $1/4$ on a stated query ball, even with arbitrary query-dependent convex mixing. Conversely, a four-cycle has ordinary rank three but nonnegative rank four. Every three-positive-component state fails at least one of sixteen finite query/tail tests by more than $1/502$. Its standalone prefix output is always zero: the tail makes the hidden mass error observable.

Contributions and evidence. The exact contraction and component characterization establish when the interface is lossless (Theorems 1 and 2); the subset separation and finite obstruction expose its advantages and limits (Theorems 3 and 4). A joint perturbation bound connects imperfect dictionaries and occupancy to output error under explicit construction and prediction access (Theorem 5). Mechanism checks and 540 fixed saved-cache cases then distinguish occupancy error from reconstruction error. Restoring occupancy usually helps the product state, but ordinary clustering retains lower aggregate error. This result locates the practical limitation in the representation and its fitted dictionaries, rather than in a different exact factorization of occupancy.

2 Cache State and Representation Class

For a nonempty weighted prefix A, define

$$Z_A(q) = \sum_{a \in A} w_a e^{q^\top k_a}, \qquad N_A(q) = \sum_{a \in A} w_a e^{q^\top k_a} v_a, \qquad o_A = N_A / Z_A. \tag{1}$$

Keys lie in $\mathbb{R}^d$, values in $\mathbb{R}^p$, and weights are nonnegative with positive sum. Integer weights encode multiplicities of ordinary tokens; arbitrary real weights are supplied explicitly. The usual attention

scale is absorbed into keys. A disjoint exact tail T contributes (N_T, Z_T), with both zero when empty. The combined output is

$$o_{A\cup T} = \frac{N_A + N_T}{Z_A + Z_T} = \lambda o_A + (1-\lambda) o_T, \qquad \lambda = \frac{Z_A}{Z_A + Z_T}. \tag{2}$$

The second expression applies only to a nonempty tail. We use Euclidean output error and exact real arithmetic in theorems.

Construction and prediction access. Construction may inspect the full prefix, fit dictionaries and assignments, and verify a proposed occupancy factorization. Prediction receives only the stored representation, the current query and the exact tail. Discarded tokens and their assignments are unavailable at prediction time. Queries and caches are fixed for the operation under study. Positional rotations must already be included in the keys.

Fix dictionaries $(\mu_j, \nu_j)_{j=1}^m$ and $(x_i, y_i)_{i=1}^s$. The represented cache is

$$k_{ji} = \mu_j + x_i, \qquad v_{ji} = \nu_j + y_i, \qquad W_{ji} \geq 0. \tag{3}$$

An entry $W_{ji} = 0$ means that pair is absent. A *positive product representation* uses R fixed nonnegative component columns,

$$\widehat{W} = AB^\top, \qquad A \in \mathbb{R}_+^{m\times R}, \quad B \in \mathbb{R}_+^{s\times R}, \tag{4}$$

with the same dictionaries. Each column contributes a product of two weighted dictionary states. Factors may be supplied or constructed; no efficient algorithm for finding the minimum possible R is assumed. The lower bounds apply to this class. Signed factors, different dictionaries and unrestricted compressed encodings define other classes.

Why the normalizer is identifiable. A mergeable prefix must supply its relative mass as well as its output. Two one-token tails make this requirement precise: choose a zero key and tail values zero and $z \neq 0$. The difference between their combined outputs is $z/(Z_A + 1)$, which identifies Z_A; either probe then identifies N_A. Thus exact composition determines the unnormalized state even when standalone outputs are identical. Appendix A.1 gives the full argument.

3 Exact contraction and its storage cost

Let $u_j = \exp(q^\top \mu_j)$ and $t_i = \exp(q^\top x_i)$. Store the four dictionaries and factors A, B. The query operation forms

$$\begin{aligned} a &= A^\top u, & b &= B^\top t, \\ r &= u \odot (Ab), & c &= t \odot (Ba), \\ Z &= \mathbf{1}^\top r = \mathbf{1}^\top c = a^\top b, & N &= \sum_j r_j \nu_j + \sum_i c_i y_i. \end{aligned} \tag{5}$$

Here r and c are the unnormalized left and right marginals of the implied pair distribution. Values are combined once these marginals are available, avoiding a separate value multiplication for each component.

Theorem 1 (Exact composable product state)**.** *If $W = AB^\top$, Equation* (5) *computes the exact prefix numerator and partition for every finite query. Thus $(N/Z, \log Z)$ gives exact attention after any exact tail. With dense stored arrays, its prediction cost is $O((m+s)(d+p+R))$ arithmetic operations, and its payload is*

$$S_{\text{product}} = (m+s)(d+p+R) \tag{6}$$

real scalars plus one coverage integer, excluding the separately retained tail.

Why the contraction is exact. Expanding Ab recovers each left dictionary row's total occupied weight; Ba does the same on the right. Both marginals sum to the pair partition. Additive values can then be summed through these marginals, sharing the dictionary vectors across all components. Appendix A.5 gives the calculation and cost count.

Dense-occupancy control. Storing W directly is also exact. Compute $r = u \odot (Wt)$ and $c = t \odot (W^\top u)$, then the same value sums. This costs $O((m+s)(d+p) + ms)$ operations and $(m+s)(d+p)+ms$ scalars. Positive factors reduce the occupancy array precisely when $R(m+s) < ms$. They reduce the declared payload relative to n separate dense token rows precisely when

$$(m+s)(d+p+R) < n(d+p). \tag{7}$$

These criteria separate two savings. Sharing vectors replaces the $n(d+p)$ original coordinates by $(m+s)(d+p)$ dictionary coordinates; factorization changes the occupancy cost from ms to $R(m+s)$. A reusable dictionary can therefore be worthwhile even when factoring its occupancy is not. Conversely, a small component count does not by itself imply a smaller total state. The criteria compare dense prediction payloads at equal precision. Fitting, factor verification and discarded assignments belong to construction costs.

Stable evaluation. The reference implementation normalizes the log weights $q^\top \mu_j + \log A_{j\ell}$ separately within each component, and does the same for the right dictionary. It then merges component log partitions and uses their normalized probabilities to obtain the two marginals. Zero factors have log weight $-\infty$ and empty components contribute nothing. Finally, it merges the prefix state with the exact tail using a common log scale. Independently subtracting dictionary maxima without retaining scale is insufficient: the two maxima can correspond to an absent pair, causing all occupied pairs to underflow. Appendix F gives an explicit check. The reference implementation uses float64 arithmetic.

4 Occupancy determines exact positive complexity

The nonnegative rank $\operatorname{rank}_+(W)$ is the minimum number of nonnegative rank-one summands of W (Cohen & Rothblum, 1993). This quantity characterizes the specified cache interface.

Theorem 2 (Minimum positive component count)**.** *Suppose all ms combined keys $\mu_j + x_i$ are distinct and the query domain contains a nonempty open subset of $\mathbb{R}^d$. Among fixed-dictionary positive product representations, the least number of components that preserves attention after every exact tail is* $\operatorname{rank}_+(W)$.

Why occupancy, rather than output alone, determines the rank. The tail probes first identify Z_A. With distinct combined keys, equality of partitions on an open query set identifies every coefficient of their exponential sum: restriction to a suitable line gives a nonsingular Vandermonde system. Hence $W = AB^\top$, and nonnegative rank is necessary. The exact contraction supplies sufficiency (Appendix A.3).

Exact composition forces preservation of the occupancy coefficients, beyond the normalized response. The tail distinguishes these representations: with all prefix values zero, every occupancy has the same standalone output, although their required composable representations differ.

Three occupancy patterns illustrate the component count. Full occupancy $\mathbf{1}\mathbf{1}^\top$ needs one positive component. Diagonal occupancy I_r needs r, already by ordinary rank. The four-cycle

$$W_\circ = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \tag{8}$$

has ordinary rank three but positive rank four. Appendix C.1 gives the support argument. This is the classical fooling-set mechanism; robust versions are established in positive-factorization theory (Braun & Pokutta, 2013).

Four dense positive components cost 32 occupancy scalars for $W_\circ$, while its dense matrix needs 16. Its fourth column equals its first minus its second plus its third, so a signed rank-three contraction is another exact control. Positivity enables componentwise normalization at a representational cost. If combined keys collide, the partition identifies only the sum of occupancies at each key; the theorem's rank necessity need not survive. If the query domain lies in a proper subspace, distinctness must hold after projection onto that subspace.

What matching assignment marginals preserves. Matching assignment marginals preserves the anchor output but need not preserve its response to a query change. Appendix E.1 gives the covariance calculation.

The derivative also yields a finite diagnostic. Let J be the value–key covariance difference between W and its independent marginals, and let λ_0 be the prefix mass fraction at query zero. For a fixed direction q and fixed exact tail, define $\Delta o(t) = o_W(tq) - o_{\mathrm{ind}}(tq)$. If M_v bounds value distances from a common center and S_q is the score range over all reconstructed pairs and tail tokens, then

$$\|\Delta o(t) - t\lambda_0 Jq\| \le \tfrac{1}{2} M_v S_q^2 t^2, \qquad 0 \le t \le 1. \tag{9}$$

Appendix E proves this by differentiating the two complete attention outputs. With dictionaries fixed, it predicts a linear occupancy defect and a quadratic remainder near zero.

5 A SEPARATION FROM ORIGINAL-VALUE SUBSETS

We next compare the exact product state with a fixed subset of original values. We allow its decoder arbitrary query-dependent nonnegative mixing weights, a stronger control than a fixed weighted softmax.

Theorem 3 (Uniform raw-subset separation)**.** *For every $r \ge 2$, there is a cache of r^2 unit-norm keys and unit-norm values in $\mathbb{R}^{2r}$ with an exact one-component product state using $2r$ dictionary rows. Every fixed proper subset of its original values, decoded as any convex combination of those values, has worst-case output error at least $1/4$ on*

$$\|q\| \le 2\log(3(r-1)). \tag{10}$$

The product state is exact for all queries and exact tails. Keeping all original rows attains zero error.

The omitted pair supplies the witness. For any omitted original value, the construction chooses a query making each of its two dictionary coordinates have probability $3/4$. The exact output projects to $3/4$ along the omitted value, whereas every retained value, and every convex mixture of them, projects to at most $1/2$. The gap is the stated $1/4$ (Appendix B.1).

At any uniform tolerance below $1/4$, all r^2 original rows are therefore necessary within this subset class, whereas the product state uses two dictionaries and one component. Its dense scalar ratio is $2/r + 1/(2r^2)$ when $d = p = 2r$; it is below one for $r \ge 3$. Figure 2 also retains the small $r = 2$ case, where the factor payload is larger. The proof constructs the query analytically.

This construction differs from general attention coreset results (Kochetkova et al., 2025; Liberty et al., 2026). Both dimension and query radius grow with r, and the instance has a highly structured value dictionary. A decoder that creates new value vectors, uses negative weights, or retrieves omitted values from external storage is outside the subset class. Offloaded retrieval systems such as PQCache fall outside this access model (Zhang et al., 2025a). Additive *values* matter as much as additive keys: arbitrary values for all r^2 pairs would reintroduce storage not counted by the two dictionaries.

6 A FINITE-QUERY OBSTRUCTION FOR APPROXIMATE OCCUPANCY

Exact identifiability does not determine the approximation error of a smaller positive state. For the four-cycle, a finite set of queries and exact tails gives a robust obstruction. Ordinary tokens realize the required tail masses.

Theorem 4 (Sixteen probes require four positive components)**.** *Take $m = s = 4$, $\mu_j = e_j/\sqrt{2}$, $x_i = f_i/\sqrt{2}$ in $\mathbb{R}^8$, zero scalar value dictionaries and occupancy $W_\circ$. There are sixteen specified query/tail pairs, each with $\|q\| = 2\log 5$, at most 40 zero-key tail tokens and values in $[0,1]$, on which every fixed-dictionary positive product representation with $R \le 3$ has maximum combined-output error greater than $1/502$. Four positive components attain zero error.*

From a matrix obstruction to an attention error. A robust positive-rank inequality forces every three-component occupancy to differ from $W_\circ$ by more than $1/32$ in some entry. The sixteen queries

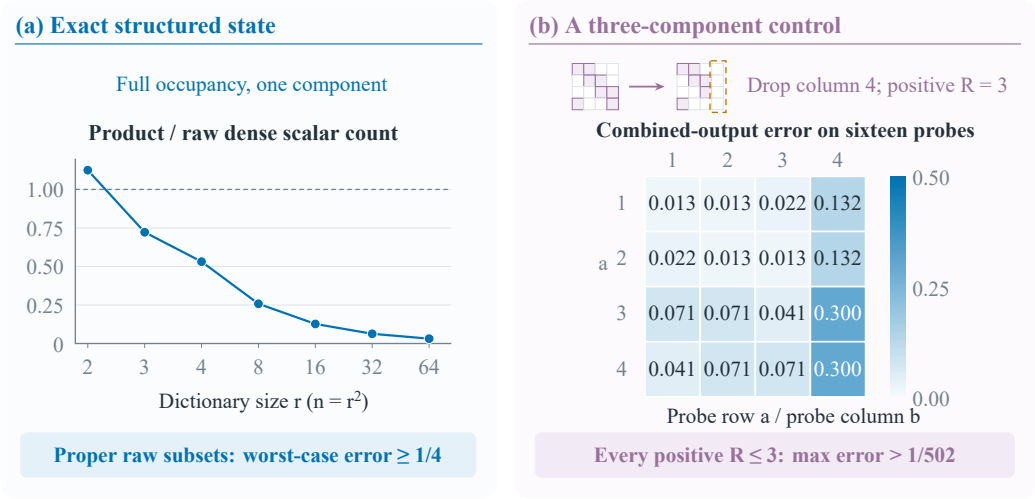

Figure 2: Dictionary reuse and occupancy impose different limits on compression in the two representation classes shown. (a) A full product of r^2 original rows admits an exact one-component state with $2r$ dictionary rows. The curve gives its dense scalar cost relative to all original rows in Theorem 3; coverage is common. Every proper raw subset has worst-case error at least $1/4$ on the stated, size-dependent query ball. (b) Dropping the fourth column of the four-cycle gives one explicit three-positive-component control. The heatmap shows its combined-output errors on the sixteen query/tail probes. These are individual control results, not the universal bound: Theorem 4 gives a maximum error greater than $1/502$ for every three-component representation in the specified positive class.

give an invertible measurement of these entries, forcing a partition error greater than $8/25$. A tail of known mass converts this error to the output gap $1/502$. The probes are fixed before seeing the approximation (Appendix C.2).

This lower bound holds at finite radius and nonzero tolerance. Composition is essential: every standalone prefix output is exactly zero, regardless of occupancy. The tail turns a partition error into an observable output error. The representation class fixes the dictionaries and requires positive product states. Dense occupancy and signed contractions belong to other classes.

7 Approximate Dictionaries and Missing Combinations

Construction assigns every token to a dictionary pair. Let W record the assigned weights (counts for ordinary tokens) and $\widehat{W} = AB^\top$ be the stored occupancy. Suppose each actual key and value differs from its assigned sum by at most ε_k and ε_v. The intermediate cache uses these weights with reconstructed dictionary pairs. The assignments cover every token. The represented combined mass $\widehat{D} = \widehat{Z}_A + Z_T$ must be positive for its normalized output to be defined. For a guarantee that includes an empty tail, this requires $\widehat{W} \neq 0$; an all-zero approximate occupancy is not a valid standalone prefix state.

Store row discrepancies $s_j = \sum_i |W_{ji} - \widehat{W}_{ji}|$, a right-key center x_c with radius $a_x \geq \max_i \|x_i - x_c\|$, and a value center c_v with $R_0 \geq \max_{ji} \|\nu_j + y_i - c_v\|$. The latter includes pairs absent from the original cache but present in the approximation. At a query, define

$$H = \|q\|, \qquad \widehat{D} = \widehat{Z}_A + Z_T, \qquad \widehat{\lambda} = \widehat{Z}_A/\widehat{D},$$
$$E_{\mathrm{occ}} = e^{q^\top x_c + H a_x} \sum_j s_j e^{q^\top \mu_j}, \qquad t_{\mathrm{occ}} = \min(1, E_{\mathrm{occ}}/\widehat{D}), \qquad t_k = \tanh(H\varepsilon_k/2). \tag{11}$$

Let $R_T = \max_{t \in T} \|v_t - c_v\|$, zero for the empty tail; set $R_{\mathrm{rec}} = \max(R_0, R_T)$, $R_{\mathrm{true}} = \max(R_0 + \varepsilon_v, R_T)$ and $\lambda^+ = \min(1, \widehat{\lambda} + t_{\mathrm{occ}})$.

Table 1: All 540 fixed-cache cases. Lower error is better. Bytes count dense float64 prediction arrays and coverage; tail storage is common. The metric is head-output error. The ordinary clustering controls outperform both product constructions.

Representation	Prediction bytes	RMS error	Mean error
Product: independent	14,456	0.2083	0.1584
Product: dense occupancy	14,736	0.1858	0.1338
Joint centroids	15,488	0.1272	0.0747
Local rank 0	16,008	0.1035	0.0670
Local rank 1	14,920	0.1247	0.0795
Local rank 2	15,928	0.1242	0.0772
Top anchor subset	16,392	0.2552	0.1753
Uniform subset	16,392	0.2284	0.1799

Theorem 5 (Joint perturbation bound)**.** *For every finite query and exact tail with $\widehat{D} > 0$, the actual and represented combined outputs satisfy*

$$\|o - \widehat{o}\| \le 2R_{\mathrm{rec}} t_{\mathrm{occ}} + 2R_{\mathrm{true}} t_k + \lambda^{+} \varepsilon_v. \tag{12}$$

Evaluation uses only the representation, the stored discrepancy/radius fields, the query and the exact tail. Exact dictionaries and occupancy make the bound zero.

Three terms, three construction choices. Insert the reconstructed dictionary cache with its true assignment counts between the actual and represented caches. Changing occupancy gives the first term, changing keys gives the second, and changing values gives the third. The common exact tail cancels from the occupancy discrepancy but remains in the denominator, so its mass moderates that contribution. Appendix D.1 proves the bound, including missing pairs.

The metadata contains $m + d + p + 4$ additional real scalars with these choices of centers and radii. Construction computes them by scanning the original cache and the proposed occupancy. Prediction retains no original token assignments. The numerical diagnostic evaluates a nominal float64 bound.

8 Mechanism checks and a fixed-cache assessment

Mathematical mechanism checks. Exact rational checks cover the contraction, positive-rank witnesses and finite-query tail constants. An independent expansion of attention checks the numerical contraction, perturbation bounds and absent-pair normalization failure. Appendix F gives the test counts, arithmetic conventions and reproduction details. The saved-cache assessment tests these mechanisms on model-derived inputs.

Saved-cache construction. We reuse all 540 first-layer head/query cases from a fixed three-context GPT-2 (Radford et al., 2019) integration study: three 30-token prefixes, twelve heads, and fifteen continuation queries with their growing exact tail. This assessment uses the saved model records. The cache keys already include attention scaling; all summary/subset calculations use float64. Each prefix independently fits two seven-entry dictionaries by ordinary product quantization of joint key/value coordinates (Jégou et al., 2011). Key and value blocks are each divided by their prefix RMS scalar magnitude, and even/odd coordinates form the two subspaces. Deterministic farthest-point initialization and twenty Lloyd iterations use no continuation queries. Appendix G gives all construction details.

We compare two occupancies for the same dictionaries: the full 7×7 count matrix, and one positive product formed from its assignment marginals. The product occupancy assumes independent assignments. Controls are a fifteen-centroid joint quantizer with counts, the previously fixed local covariance summaries of ranks zero through two, and sixteen-token top-anchor and uniform subsets. Every prediction representation fits the same 16,392-byte array-payload budget, including counts, factors and one coverage integer. Certificate metadata is additional.

Occupancy error with fixed dictionaries. At query tq, independent and dense occupancy agree at zero and separate with the predicted covariance derivative. Across all 540 fixed query/tail pairs,

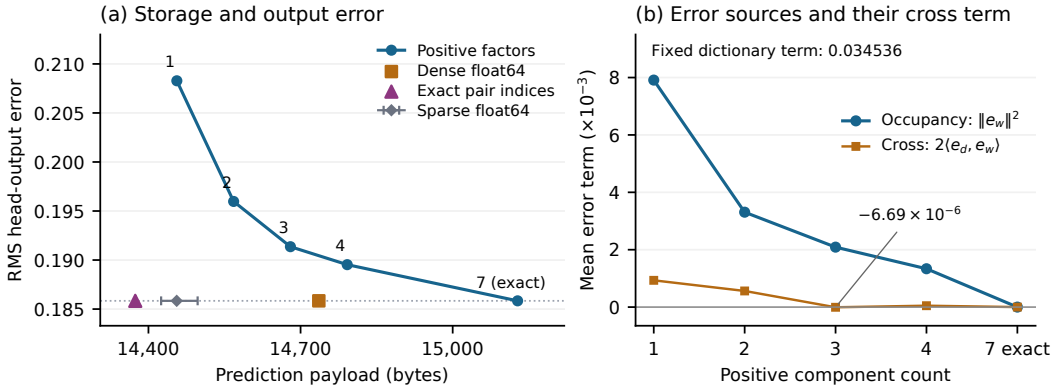

Figure 3: Fixed dictionaries, queries and tails isolate occupancy costs. (a) Labels show component count: $R = 2$–4 fit occupancy alone, $R = 1$ is independence, and $R = 7$ is exact $WI^\top$. All payloads include 14,336 dictionary bytes and eight coverage bytes. Sparse storage shows its statewise mean and range; exact indices use one byte per prefix token. (b) Adding the occupancy squared error and cross term to the fixed dictionary MSE 0.034536 gives total MSE. The negative rank-three cross term is retained.

the RMS norm of $\lambda_0 Jq$ is 0.1810; remainder RMS values 3.55×10^{-9} at $t = 10^{-4}$ and 3.55×10^{-7} at $t = 10^{-3}$ match local quadratic order. All 3,240 comparisons satisfy Equation (9). At $t = 1$, remainder RMS is 0.1651, so the derivative is not an accurate full-query correction (Appendix H).

At the original queries, set $e_d = o_{\text{raw}} - o_W$ and $e_w = o_W - o_{\text{ind}}$. Their squared-error identity is $\|e_d + e_w\|^2 = \|e_d\|^2 + \|e_w\|^2 + 2\langle e_d, e_w\rangle$. The mean terms are 0.034536, 0.007911 and 0.000933, summing to 0.043381. Exact occupancy improves original-cache error in 464 of 540 cases. In the remaining cases, correcting occupancy can remove a favorable cancellation with dictionary error. Once occupancy is exact, changing its factorization cannot alter the dictionary reconstruction term.

When do more components repay their storage? We keep all 36 fitted dictionary states fixed and fit $R = 2, 3, 4$ positive components using only occupancy counts, with eight fixed initializations and KL-loss selection. Rank one reproduces independence; $R = 7$ uses the exact factorization $A = W, B = I$. Figure 3 shows RMS errors $0.2083, 0.1960, 0.1914, 0.1895, 0.1858$ at $R = 1, 2, 3, 4, 7$. Each added component costs 112 bytes. Rank three saves only 56 bytes relative to dense occupancy; rank four costs 56 bytes more while retaining error. Exact pair indices instead match dense occupancy at error 0.1858 in 14,374 bytes, fewer than every tested factorization; ranks one through four also have higher aggregate error. These short integer-count matrices favor explicit pair coding over the fitted factors. Appendix H gives the complete storage accounting and error decomposition.

Comparison with the controls. Table 1 shows that exact occupancy improves the product construction's RMS Euclidean error from 0.2083 to 0.1858. Both are below the raw subsets' aggregate errors, 0.2552 and 0.2284, but above joint centroids at 0.1272 and the anchor-weighted rank-zero summary at 0.1035. Against joint centroids, the independent and dense product constructions win 16 and 27 of 540 cases.

Appendix I reports the earlier local-summary NLL comparison. The present assessment measures head-output error on fixed caches.

Independent expanded-array reductions reproduce all original method outputs and certificate values, and all 4,860 new sweep outputs decoded from serialized payloads (Appendices G and H).

9 Related Work

Product structure and positive factors. Nonnegative rank and robust fooling-set methods are established tools (Cohen & Rothblum, 1993; Braun & Pokutta, 2013). Sum-product models pro-

vide a broader setting for tractable partition evaluation (Poon & Domingos, 2011). Our contraction applies this algebra to a compressed prefix. The characterization and finite-query construction determine what exact composition requires. Product-key memories use Cartesian keys for efficient top-k lookup with a separate value table (Lample et al., 2019). Here both keys and values are additive and the full occupancy-weighted partition is evaluated. Tensor Product Attention instead factors each token's head-by-feature activations in a trained architecture (Zhang et al., 2025b). We study fixed-cache representations rather than that trained architecture.

Approximation and retrieval. We fit the dictionaries by product quantization (Jégou et al., 2011). PQCache uses product codes for token selection and then fetches original key/value pairs (Zhang et al., 2025a); its prediction access differs from a stored subset. Streaming and nearly optimal attention coresets address general bounded-norm caches (Kochetkova et al., 2025; Liberty et al., 2026), whereas our subset separation exploits a particular structured family with growing query radius. Universal analytic moment compression (Wang et al., 2026b) and sparse-plus-linear attention (Wang et al., 2026a) are further foundations for replacing explicit token evaluation.

10 Discussion

Two sources of storage savings. Dictionary reuse and occupancy compression save different fields. At $r = 16$, the exact construction stores 2,080 dense scalars versus 16,384 original scalars and 2,304 with dense occupancy. Shared vectors yield most of the saving; positive factors add a further 224 scalars. A supplied factorization suffices to verify exactness and storage without finding the minimum nonnegative rank. In the saved count matrices, adding positive components reduces occupancy error but quickly exhausts the small remaining storage saving. Exact one-byte pair indices match dense occupancy error at lower payload than every tested factorization (Figure 3). Choosing an encoding is therefore part of the compression problem, not merely an implementation of a rank choice.

Choose the representation before optimizing its contraction. The results suggest a concrete construction-time comparison. Fit or supply additive dictionaries, retain their true occupancy, and compare the dense occupancy payload with a proposed positive factorization and the original rows. If exact occupancy is affordable, either exact contraction gives the same reconstructed attention. If occupancy must be approximated, evaluate its error separately from dictionary reconstruction and retain the metadata needed by the joint bound. At prediction time, merge the prefix numerator and partition with the exact tail; the coverage count remains a separate record of the original tokens represented.

The fixed-cache results also distinguish what a better occupancy fit can repair. Exact occupancy leaves dictionary reconstruction error; improving that term requires different dictionaries or another representation. Total error must retain the cross term, since restoring occupancy can remove favorable cancellation.

Scope of the engineering conclusion. The theorems concern fixed caches, explicit representation classes and exact arithmetic. They do not compare generated histories or improve a general worst-case coreset rate. Positive rank counts positive product components, not all compressed encodings; sparse occupancy and indexed dictionaries can beat the dense controls. Positional rotations can destroy additive structure, while dependent occupancy can eliminate the factor saving. Construction, array headers and device allocation are outside the declared payload; no kernel, device-memory or latency advantage is established.

Float64 evaluation rejects overflowing dictionary dot products. Its nominal bounds are not outward-rounded enclosures; large query radii or occupancy discrepancies can make them uninformative at a useful tolerance.

The structural opportunity is nevertheless precise: many full tokens can share a small set of key *and* value components. This reuse yields a useful composable cache only when their pairings can also be represented economically and their unnormalized mass remains available for future merges.

REPRODUCIBILITY STATEMENT

The anonymous artifact contains the reference contraction, rational checkers, fixed saved-cache construction, independent expanded-array reductions, figure/table builders and the unchanged numerical records of both original model runs. Complete derivations accompany the manuscript source. The default offline reproduction checks the finite results, original recorded predictions and nominal bounds, and the occupancy sweep without downloading or running a model. The source package separately compiles the paper. Exact-arithmetic results are recorded separately from nominal floating-point checks. The saved-cache study reports every declared head/query pair and all controls.

DISCLOSURE OF AI USE

AI tools assisted with mathematical review, manuscript preparation and language editing, and code development. The human authors take responsibility for the mathematical claims, the experimental design and execution, the code and reported results, and their interpretation and reproducibility. No AI tool is listed as an author.

REFERENCES

Gábor Braun and Sebastian Pokutta. Common information and unique disjointness. Electronic Colloquium on Computational Complexity, TR13-056, revision 2, 2013. URL https://eccc.weizmann.ac.il/report/2013/056/revision/2/download/.

Joel E. Cohen and Uriel G. Rothblum. Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. *Linear Algebra and its Applications*, 190:149–168, 1993. doi: 10.1016/0024-3795(93)90224-C.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.08691.

Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011. doi: 10.1109/TPAMI.2010.57.

Ekaterina Kochetkova, Kshiteej Sheth, Insu Han, Amir Zandieh, and Michael Kapralov. Streaming attention approximation via discrepancy theory. *arXiv preprint arXiv:2502.07861*, 2025. URL https://arxiv.org/abs/2502.07861.

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Large memory layers with product keys. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1907.05242.

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

Maxim Milakov and Natalia Gimelshein. Online normalizer calculation for softmax. *arXiv preprint arXiv:1805.02867*, 2018. URL https://arxiv.org/abs/1805.02867.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, 2011. URL https://homes.cs.washington.edu/~pedrod/papers/uai11a.pdf.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language-models.pdf.

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. *arXiv preprint arXiv:2504.09936*, 2025. URL https://arxiv.org/abs/2504.09936.

Bailin Wang, Dan Friedman, Tao Lei, and Chong Wang. SPLA: Block sparse plus linear attention for long context modeling. *arXiv preprint arXiv:2601.22379*, 2026a. URL https://arxiv.org/abs/2601.22379.

Hong-Yi Wang, Di Luo, Tomaso Poggio, Isaac L. Chuang, and Liu Ziyin. A universal compression theory for lottery ticket hypothesis and neural scaling laws. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2510.00504.

Hailin Zhang, Xiaodong Ji, Yilin Chen, Fangcheng Fu, Xupeng Miao, Xiaonan Nie, Weipeng Chen, and Bin Cui. PQCache: Product quantization-based KVCache for long context LLM inference. *Proceedings of the ACM on Management of Data*, 3(3), 2025a. doi: 10.1145/3725338.

Yifan Zhang, Yifeng Liu, Huizhuo Yuan, Zhen Qin, Yang Yuan, Quanquan Gu, and Andrew Chi-Chih Yao. Tensor product attention is all you need. In *Advances in Neural Information Processing Systems*, 2025b. URL https://arxiv.org/abs/2501.06425.

A Exact representation: proof and access details

The positive product class fixes the two key dictionaries, the two value dictionaries and nonnegative factor columns independent of the query. The prefix state is evaluated from these objects; the exact tail is then merged by Equation (2). The class does not include an auxiliary oracle giving the original partition. Supplying such an oracle would change the information restriction behind the necessity theorem.

A.1 Normalizer identifiability

Correct composition with every exact tail identifies the prefix partition at each query. Append one zero-key token, first with value zero and then with any nonzero vector z. The difference between the two combined outputs is $z/(Z_A+1)$. Correctness for both choices therefore identifies Z_A, and either output then identifies N_A. In contrast, duplicating all prefix tokens preserves the entire function $q \mapsto o_A(q)$ and every derivative while changing Z_A. The argument identifies the information that composition requires.

A.2 Tail probes identify the unnormalized state

At a fixed query, a zero-key tail token has mass one. Let F_0 and F_z denote the combined outputs for tail values zero and $z \neq 0$. Then

$$F_z - F_0 = \frac{z}{Z_A + 1}. \tag{13}$$

Equality of both probe outputs for two candidate prefixes forces the same Z_A. Their zero-tail-value outputs $N_A/(Z_A + 1)$ then force the same numerator. This works in any positive output dimension. The two tails need not be considered simultaneously; the guarantee in Theorem 2 quantifies over every exact tail.

By contrast, if every original token is duplicated c times, both numerator and partition are multiplied by c, so their ratio and all its derivatives agree on the entire query space. The example is exact and independent of a local Taylor approximation. It excludes an output-only interface, not a state storing enough information to reconstruct the partition.

A.3 Proof of Theorem 2

Proof. Theorem 1 gives sufficiency. For necessity, the two-tail argument in Appendix A.1 identifies the partition at every query, so

$$\sum_{ji} (W_{ji} - (AB^\top)_{ji}) e^{q^\top(\mu_j + x_i)} = 0 \tag{14}$$

on the open set. Choose a direction having distinct inner products with all combined keys and restrict queries to a short line segment in that set. Derivatives of orders zero through $ms - 1$ give a

Vandermonde system with nonzero determinant. Every coefficient in Equation (14) is zero. Hence $W = AB^\top$ and $R \geq \operatorname{rank}_+(W)$. The next subsection supplies the direction and boundary details. □

A.4 Linear independence of the relevant exponentials

Write the distinct combined keys as $z_1, \ldots, z_M$, $M = ms$. For every pair $a \neq b$, the directions v with $v^\top(z_a - z_b) = 0$ form a proper hyperplane. A finite union of such hyperplanes cannot cover $\mathbb{R}^d$. Choose v outside this union, so the numbers $\gamma_a = v^\top z_a$ are distinct. Choose q_* in the open query set. A sufficiently short segment $q_* + tv$ remains inside it.

If $\sum_a h_a e^{q^\top z_a} = 0$ throughout the set, differentiating along that segment at $t = 0$ gives

$$\sum_{a=1}^{M} h_a e^{q_*^\top z_a} \gamma_a^k = 0, \qquad k = 0, \ldots, M-1. \tag{15}$$

The coefficient matrix has determinant $\prod_{a<b}(\gamma_b - \gamma_a) \neq 0$. All $h_a e^{q_*^\top z_a}$, and therefore all h_a, vanish. Substituting $h_{ji} = W_{ji} - (AB^\top)_{ji}$ proves the necessity of exact occupancy.

If combined keys collide, group equal exponents before applying this argument. It identifies the total occupancy of each group, not each pair separately. The numerator additionally identifies the corresponding summed weighted values, but those conditions need not imply $W = AB^\top$. Similarly, when queries range over a proper affine subspace, only the projected exponent differences can be distinguished. The distinct-key hypothesis must be formulated for that domain. No claim about necessity on a finite arbitrary query set follows from this Vandermonde proof; Section 6 supplies a separate finite construction.

A.5 Proof of Theorem 1

Proof. Expand $r_j = u_j \sum_i W_{ji} t_i$ and $c_i = t_i \sum_j W_{ji} u_j$. Their sums equal the pair partition, and their two value sums equal $\sum_{ji} W_{ji} u_j t_i (\nu_j + y_i)$. The matrix-vector products in Equation (5) use $O(R(m+s))$ operations; dictionary logits and the final value sums use $O((m+s)(d+p))$. Counting the shared arrays gives Equation (6). □

A.6 Cost, zero components and stronger representations

The four shared dictionary arrays contain $(m + s)(d + p)$ scalars. The two factors contain $(m + s)R$. One coverage integer records the number of original tokens represented by this state; it is not inferred from an approximate factor sum. Prediction also needs temporary scalar marginals and factor reductions. In a straightforward implementation the temporary per-component normalized dictionary probabilities occupy $O(R(m + s))$ scalars. This workspace is distinct from the stored cache payload and does not contain original tokens.

A column with zero mass on either side contributes zero and can be removed. A nonzero cache has at least one active column. Nonnegative scaling $A_{:\ell} \mapsto cA_{:\ell}$ and $B_{:\ell} \mapsto B_{:\ell}/c$ for $c > 0$ preserves occupancy; the reference checks this invariance under large reciprocal rescalings. The common factor scaling remains part of the log-partition arithmetic.

Dense occupancy stores ms scalars and avoids factor construction. Sparse occupancy may store only nonzeros plus indices and row pointers. An indexed dictionary representation stores token assignments rather than a dense occupancy matrix. These representations provide alternative storage controls. Positive rank determines the number of fixed positive product components. The accounting comparison uses the explicitly named dense arrays at common precision, including counts and coverage. The reported payload counts the declared arrays.

B THE SUBSET SEPARATION IN FULL DETAIL

B.1 PROOF OF THEOREM 3

Proof. Let $e_1, \dots, e_r, f_1, \dots, f_r$ be an orthonormal basis and set $k_{ji} = v_{ji} = (e_j + f_i)/\sqrt{2}$ with full occupancy. For an omitted pair (j, i), take $q = \sqrt{2}\beta(e_j + f_i)$, $\beta = \log(3(r-1))$. The exact two dictionary distributions each assign $3/4$ probability to the named coordinate. Its output therefore has projection $3/4$ onto the unit vector $(e_j + f_i)/\sqrt{2}$. Every retained value has projection at most $1/2$, as does every convex combination of them. The projected gap proves the Euclidean lower bound. The query has norm 2β and the empty tail suffices for the lower bound. Exact factorization proves the upper statement. □

B.2 CONSTRUCTION AND DETAILED VERIFICATION

In Theorem 3, choose dictionaries

$$\mu_j = \nu_j = e_j/\sqrt{2}, \qquad x_i = y_i = f_i/\sqrt{2}, \qquad W_{ji} = 1. \tag{16}$$

All combined keys and values have norm one. Fix any proper subset S of the r^2 original values and an omitted pair (j, i). At $q = \sqrt{2}\beta(e_j + f_i)$, a pair (a, b) has score $\beta(\mathbf{1}_{a=j} + \mathbf{1}_{b=i})$. Its unnormalized weight is a product of two positive factors, each equal to e^β at the selected coordinate and one elsewhere. Thus the exact output is

$$o(q) = \frac{1}{\sqrt{2}} \left(\sum_a P_a e_a + \sum_b Q_b f_b \right), \quad P_j = Q_i = \frac{e^\beta}{e^\beta + r - 1}. \tag{17}$$

With $e^\beta = 3(r-1)$, these probabilities are $3/4$. The unit witness $w = (e_j + f_i)/\sqrt{2}$ has $w^\top o = 3/4$. For every retained row $(a, b) \in S$,

$$w^\top v_{ab} = \tfrac{1}{2}(\mathbf{1}_{a=j} + \mathbf{1}_{b=i}) \le \tfrac{1}{2}. \tag{18}$$

This inequality holds for any nonnegative mixing weights summing to one, even if an arbitrarily powerful decoder chooses them as a function of q. Cauchy–Schwarz gives $\|o - \widehat{o}\| \ge w^\top(o - \widehat{o}) \ge 1/4$. The subset is fixed at construction; choosing a different retained set after reading an external original cache is outside the access model.

The lower bound uses an empty tail. The upper bound is exact with arbitrary tails by Theorem 1. Hence the construction separates the two representations even before requiring the stronger compositional operation. For $d = p = 2r$, the factor state costs $2r(4r + 1)$ dense scalars and all original rows cost $4r^3$, with equal coverage overhead. At $r = 2$ the ratio is $9/8$; at $r \ge 3$ it is less than one. An arbitrary compressed bit encoding can exploit sparsity or repeated basis structure and is not lower-bounded by the raw-row count.

C FINITE-QUERY POSITIVE-RANK OBSTRUCTION

C.1 POSITIVE RANK OF THE FOUR-CYCLE

Each nonnegative rank-one support is a rectangle. Covering two diagonal entries would force both cross entries to be positive, while at least one is zero. Four diagonal entries therefore require four components, and the column factorization attains four.

C.2 PROOF OVERVIEW FOR THEOREM 4

The proof transfers a quantitative matrix obstruction to attention. For every nonnegative square matrix M with positive trace,

$$\operatorname{rank}_+(M) \ge \frac{(\operatorname{tr} M)^2}{\sum_{ji} M_{ji} M_{ij}}. \tag{19}$$

Given $M = AB^\top$, form $G = VV^\top$ with $V_{j\ell} = \sqrt{A_{j\ell} B_{j\ell}}$. Then $\operatorname{tr} G = \operatorname{tr} M$, $\operatorname{rank} G \le R$ and Cauchy–Schwarz gives $G_{ji}^2 \le M_{ji} M_{ij}$. Applying the trace/rank inequality to positive semidefinite G proves Equation (19). This is a positive-factorization argument, not an ordinary-rank bound.

If $\|M - W_\circ\|_{\max} \le \delta$, the diagonal entries are at least $1 - \delta$ and every off-diagonal pair has one entry at most δ. Thus

$$\operatorname{rank}_+(M) \ge \frac{4(1-\delta)^2}{(1+\delta)(1+4\delta)}. \tag{20}$$

At $\delta = 1/32$ this is $961/297 > 3$. We use this explicit sufficient threshold; we do not claim it is the largest possible one.

At $q_{ab} = \sqrt{2}\log 5(e_a + f_b)$, the matrix of partitions is $S = UW_\circ U^\top$, where $U = 4I + \mathbf{1}\mathbf{1}^\top$. A represented occupancy M gives $\widehat{S} = UMU^\top$. Every row of U^{-1} has ℓ_1 norm $5/16$, so

$$\|M - W_\circ\|_{\max} \le \frac{25}{256}\|\widehat{S} - S\|_{\max}. \tag{21}$$

Consequently any $R \le 3$ representation has a partition error greater than $8/25$ at one of the sixteen queries. Since $S_{ab} = 24 + 16(W_\circ)_{ab}$, append exactly $S_{ab} \in \{24, 40\}$ zero-key, unit-value tokens at that query. The true output is $1/2$, whereas the represented output is $S_{ab}/(\widehat{S}_{ab} + S_{ab})$. A mass error exceeding $8/25$ implies an output error exceeding $(8/25)/(2(80 + 8/25)) = 1/502$. The following subsections check both signs of the mass error and all constants.

C.3 A robust matrix witness

Assume $M = AB^\top$ with nonnegative factors and R columns. Let $V_{j\ell} = \sqrt{A_{j\ell}B_{j\ell}}$ and $G = VV^\top$. Then G is positive semidefinite, has rank at most R, and

$$\operatorname{tr} G = \sum_{j\ell} A_{j\ell}B_{j\ell} = \operatorname{tr} M. \tag{22}$$

For every ordered pair (j, i), Cauchy–Schwarz implies

$$\begin{aligned} G_{ji}^2 &= \left(\sum_\ell \sqrt{A_{j\ell}B_{i\ell}}\sqrt{A_{i\ell}B_{j\ell}}\right)^2 \\ &\le \left(\sum_\ell A_{j\ell}B_{i\ell}\right)\left(\sum_\ell A_{i\ell}B_{j\ell}\right) = M_{ji}M_{ij}. \end{aligned} \tag{23}$$

The nonzero eigenvalues of G number at most R, so $(\operatorname{tr} G)^2 \le R \operatorname{tr}(G^2)$. Sum the entrywise bounds and rearrange to obtain Equation (19). The denominator is positive when the trace is positive because it includes the squares of all diagonal entries. The argument is a direct positive-factor trace bound; robust fooling-set methods such as Braun & Pokutta (2013) provide the broader prior context.

Suppose $\|M - W_\circ\|_{\max} \le \delta < 1$. The trace is at least $4(1 - \delta)$. Each diagonal denominator term is at most $(1 + \delta)^2$. For each of the twelve ordered off-diagonal pairs, at least one reference entry is zero. Nonnegativity and the entrywise approximation give $M_{ji}M_{ij} \le \delta(1 + \delta)$. Therefore

$$\operatorname{rank}_+(M) \ge \frac{16(1-\delta)^2}{4(1+\delta)^2 + 12\delta(1+\delta)} = \frac{4(1-\delta)^2}{(1+\delta)(1+4\delta)}. \tag{24}$$

At $\delta = 1/32$ this equals $961/297 > 3$. It follows that any matrix with nonnegative rank at most three must differ from $W_\circ$ by more than $1/32$ in at least one entry. Matrices not having positive trace cannot be close in this entrywise sense, so no omitted trace-zero case weakens the conclusion.

C.4 The sixteen fixed query/tail pairs

For the dictionaries of Theorem 4, $\exp(q_{ab}^\top \mu_j) = 5$ when $j = a$ and one otherwise, and the same holds on the right. Thus the partition matrix is $S = UW_\circ U^\top$ with $U = 4I + \mathbf{1}\mathbf{1}^\top$. Direct multiplication gives

$$U^{-1} = \frac{1}{4}\left(I - \frac{1}{8}\mathbf{1}\mathbf{1}^\top\right). \tag{25}$$

The absolute row sum is $7/32 + 3/32 = 5/16$. Applying this inverse on both sides of $\widehat{S} - S$ proves Equation (21), since each entry of the result is a bilinear combination whose absolute coefficients sum to $(5/16)^2$.

The four-cycle has total occupancy eight and every row and column sum two. Expanding $(\mathbf{1}\mathbf{1}^\top + 4I)W_\circ(\mathbf{1}\mathbf{1}^\top + 4I)$ yields

$$S_{ab} = 8 + 4(2+2) + 16(W_\circ)_{ab} = 24 + 16(W_\circ)_{ab}. \tag{26}$$

If every partition error were at most $8/25$, every occupancy entry error would be at most $(25/256)(8/25) = 1/32$, contradicting the positive-rank bound for $R \le 3$. At least one partition error therefore exceeds $8/25$.

At each query q_{ab} append S_{ab} zero-key, unit-value tokens. Both the original and represented prefix value dictionaries are zero. Write $z = S_{ab} \le 40$ and $\Delta = \widehat{S}_{ab} - S_{ab}$, with $\Delta \ge -z$. The exact and represented outputs differ by

$$\left|\frac{1}{2} - \frac{z}{2z+\Delta}\right| = \frac{|\Delta|}{2(2z+\Delta)}. \tag{27}$$

For $\Delta \ge 0$ this expression increases with Δ and decreases with z. For $-z \le \Delta < 0$, replacing Δ in the denominator by $|\Delta|$ can only decrease the expression. Therefore $|\Delta| > 8/25$ implies a gap strictly greater than $1/502$. The tail specification is fixed by $W_\circ$ and (a, b), not tuned after observing an approximate representation.

Four columns give the exact positive factorization $A = W_\circ, B = I_4$. Three signed columns also suffice: take the first three columns of $W_\circ$ as A and

$$B^\top = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \tag{28}$$

This is an exact stronger control outside the positive class. The plotted positive rank-three control instead sets the fourth column to zero. Its errors are explicit illustrative values; the proof above applies to every positive rank-three representation, including optimized ones.

D Complete perturbation proof

D.1 Proof overview for Theorem 5

The first term measures occupancy error, including missing and invented combinations. For nonnegative arrays with masses $A, B > 0$, their normalized distributions differ in total variation by at most their unnormalized ℓ_1 difference divided by $\max(A, B)$. The tail cancels from that difference, and E_{occ} bounds the remainder. In particular, the bound needs no subtraction from an estimated denominator. The second term uses the bounded-likelihood-ratio inequality for log-score perturbations in $[-H\varepsilon_k, H\varepsilon_k]$. The final term charges value error only to the intermediate prefix fraction. The detailed proof follows.

Let the reconstructed cache use the true assigned weights W (counts for ordinary tokens) and dictionary sums. Compare it first to the factor cache, with occupancy $\widehat{W}$ and the same dictionary sums. Both include the identical exact tail. They define nonnegative unnormalized weights on the union of all dictionary pairs and tail indices. The actual and reconstructed combined masses are positive because the original prefix is nonempty. The represented mass is positive by the hypothesis $\widehat{D} > 0$ in Theorem 5; the normalization comparisons below therefore have positive denominators.

D.2 Occupancy perturbation with zeros allowed

For any nonnegative arrays a, b with positive sums A, B, suppose first $A \ge B$. Then

$$\begin{aligned} \|a/A - b/B\|_1 &\le \|a-b\|_1/A + \|b\|_1|1/A - 1/B| \\ &= \big(\|a-b\|_1 + A - B\big)/A \le 2\|a-b\|_1/A. \end{aligned} \tag{29}$$

Interchanging the arrays covers $B \ge A$, and division by two gives

$$\mathrm{TV}(a/A, b/B) \le \frac{\|a-b\|_1}{\max(A,B)}. \tag{30}$$

No common-support assumption is required. The tail cancels from $a-b$. Since $q^\top x_i \le q^\top x_c + Ha_x$,

$$\|a-b\|_1 = \sum_{ji} |W_{ji} - \widehat{W}_{ji}| e^{q^\top(\mu_j + x_i)} \le E_{\mathrm{occ}}. \tag{31}$$

One of A, B is the factor combined denominator $\widehat{Z}_A + Z_T$, so total variation is at most t_{occ}. Center all pair and tail values at c_v. Their reconstructed norms are at most R_{rec}, hence the output discrepancy is at most $2R_{\mathrm{rec}}t_{\mathrm{occ}}$. The probability of any event differs by at most total variation, so the reconstructed prefix fraction is at most $\widehat{\lambda} + t_{\mathrm{occ}}$ and at most one.

D.3 Key and value reconstruction

Expand the reconstructed prefix back to its original token assignments. It now has the same token index set and weights as the actual prefix. The actual/reconstructed log-score difference is in $[-H\varepsilon_k, H\varepsilon_k]$ on the prefix and zero on the tail.

For completeness, a likelihood multiplier $X \in [a, b]$ under a baseline distribution, with mean $M > 0$, changes the normalized distribution by total variation $\mathbb{E}|X - M|/(2M)$. Convexity of $|X - M|$ bounds its expectation by the interpolation of the two endpoints. Consequently

$$\mathrm{TV} \le \frac{(M-a)(b-M)}{M(b-a)} \le \frac{\sqrt{b} - \sqrt{a}}{\sqrt{b} + \sqrt{a}}. \tag{32}$$

The second maximum occurs at $M = \sqrt{ab}$, as can be seen by writing the numerator divided by M as $a + b - M - ab/M$. For $a = b$, the distributions coincide. Substitution of $a = e^{-H\varepsilon_k}, b = e^{H\varepsilon_k}$ gives $t_k = \tanh(H\varepsilon_k/2)$, including the zero-error boundary by continuity.

Let π and p be the actual and reconstructed combined probabilities, V the actual values, and $\widetilde{V}$ the reconstructed values. Insert pV:

$$\pi V - p\widetilde{V} = (\pi - p)V + p(V - \widetilde{V}). \tag{33}$$

After centering at c_v, actual values have norm at most R_{true}, so the first term is at most $2R_{\mathrm{true}}t_k$. Values differ by at most ε_v only on the prefix; the second term is at most the reconstructed prefix fraction times ε_v, bounded by $\lambda^+\varepsilon_v$. Add the occupancy discrepancy to obtain Equation (12).

One construction of the stored radius is $R_0 = \max_j \|\nu_j - c_L\| + \max_i \|y_i - c_R\|$ with $c_v = c_L + c_R$. This includes every pair, whether occupied or not. The reference uses dictionary means as centers and computes the error radii by a full construction-time scan.

E An occupancy derivative with a finite remainder

E.1 Assignment marginals and the output derivative

Replacing W by the product of its row and column marginals preserves total count, mean key and mean value at query zero. It therefore preserves both the anchor numerator and partition and the partition's first derivative. It need not preserve the output derivative. Under the probability distribution $W/\sum W$, expand the key–value covariance:

$$\mathrm{Cov}(\mu + x, \nu + y) = \mathrm{Cov}(\mu, \nu) + \mathrm{Cov}(x, y) + \mathrm{Cov}(\mu, y) + \mathrm{Cov}(x, \nu). \tag{34}$$

The independent product keeps the first two terms and removes both cross terms. Since the anchor states and denominator derivatives agree, the combined output Jacobian defect is their transpose times the prefix mass fraction. Matching assignment marginals therefore preserves the anchor output but need not preserve its response to a query change.

E.2 Finite-remainder proof

Use the same dictionaries for W and $W_{\mathrm{ind}} = ab^\top$, where $a = W\mathbf{1}$ and $b = W^\top\mathbf{1}/n$ with $n = \sum_{ji} W_{ji} > 0$. Both occupancies have mass n, and their row and column marginals agree. Consequently their mean keys and values agree. At $q = 0$, adding the same fixed tail preserves

equality of both the numerator and denominator, and their denominator derivatives also agree. With covariance oriented value-by-key, let

$$J = \sum_{ji} \frac{W_{ji} - (W_{\mathrm{ind}})_{ji}}{n} (\nu_j + y_i)(\mu_j + x_i)^\top.$$

The shared means cancel, so this is exactly the covariance difference. Differentiating the combined numerator and denominator gives $\Delta o(0) = 0$ and $\Delta o'(0) = \lambda_0 Jq$, where $\lambda_0 = n/(n + |T|)$ for ordinary tail tokens. Arbitrary nonnegative tail weights replace $|T|$ by their sum. The same tq is applied to the prefix and tail throughout.

To bound the remainder, include every reconstructed pair and tail token in one finite support, including combinations invented by independent occupancy. Set $s_a = q^\top k_a$, let $S_q = \max_a s_a - \min_a s_a$, and choose any common center c with $\|v_a - c\| \le M_v$. For either complete attention output $f(t)$, direct differentiation of the normalized exponential weights yields

$$f''(t) = \mathbb{E}_t[(v - \mathbb{E}_t v)(s - \mathbb{E}_t s)^2].$$

Because $\|v - \mathbb{E}_t v\| \le 2M_v$ and $\mathrm{Var}_t(s) \le S_q^2/4$, one output has $\|f''(t)\| \le M_v S_q^2/2$. For completeness, the variance bound follows from $(s - \min s)(\max s - s) \ge 0$, followed by maximizing $(\mathbb{E}s - \min s)(\max s - \mathbb{E}s)$. Subtracting the two outputs bounds $\|\Delta o''(t)\|$ by $M_v S_q^2$. Integrating this bound twice from zero proves Equation (9).

The diagnostic protocol was fixed after inspecting the earlier product assessment and before running the mechanism checks. It preserves all 36 dictionary pairs, 540 query/tail cases and six stated multipliers. The calculation expands attention directly, without importing product-contraction code. It independently checks the derivative by differentiating each full normalized sum, validates all original-query outputs against their stored records, and checks the squared-error identity case by case. The original-query reference, dictionaries and assignments are unchanged. All numerical checks use float64 and a 2×10^{-11} absolute guard for the Taylor inequality. The maximum observed remainder-to-bound ratio is 0.02109 after upward rounding. The small-t agreement tests local first-order behavior.

F Numerical mechanism and finite arithmetic

The artifact's rational tests compare numerator and partition before normalization, then merge three exact tail masses. They use nonnegative rational factors with zeros and nonuniform rational dictionary weights. The analytic raw-subset projection is checked at $e^\beta = 3(r - 1)$, so its $3/4$ projection and $1/4$ gap are rational. The finite-query checker uses only integer/rational matrices. Its exact assertions include the rank-three signed attainment and the distinction between 16 dense occupancy scalars and 32 dense rank-four factor scalars.

The float64 checks separately compare factored and dense occupancy contraction with explicit expansion into occupied pairs. They vary dictionary sizes, positive ranks, nonempty and empty tails, and key/value perturbations according to a fixed declared generator. These generated fixtures test the specified mechanisms. Maximum coordinate discrepancy from expanded attention is below 1.56×10^{-15}; all 270 nominal perturbation bounds cover the observed Euclidean errors.

For the absent-pair numerical case, take scalar key dictionaries $(1000, 0)$ on both sides and occupancy $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. At query one the two occupied pairs both have score 1000, while the absent high/high pair would have score 2000. Multiplying separately shifted dictionary exponentials can underflow both occupied products after selecting 1000 as each dictionary shift. Normalizing within each positive component avoids this issue and returns log partition $1000 + \log 2$. With left values $(1, -2)$ and right values $(4, 3)$ the output is three. Reciprocal factor scales 10^{200} and 10^{-200} preserve this state within the stated numerical tolerance.

The reference checks that inputs are finite and rejects overflowing dictionary dot products. Each checker records its arithmetic.

The artifact checks exact rational numerator/partition equality in 36 merges with three tail masses, including zero. It enumerates all positive support rectangles of the four-cycle and checks its real-rank-three and positive-rank-four witnesses. The finite-query checker verifies U^{-1}, all sixteen integer tail masses, the $961/297$ rank threshold and the $1/502$ output constant using rational arithmetic.

Independently expanded float64 attention agrees with the product and dense contractions to maximum coordinate error 1.56×10^{-15} in 105 comparisons. All 270 declared occupancy/dictionary perturbation cases obey the nominal bound. An absent-pair example checks the need for componentwise normalization.

G SAVED-CACHE PROTOCOL AND COMPLETE COMPARISON

We use the fixed three-context GPT-2 record from the earlier corrected integration study. Each context supplies one thirty-token prefix, twelve first-layer heads of key/value width 64, fifteen subsequent queries and the keys/values of their growing exact tail. This makes $3 \cdot 12 \cdot 15 = 540$ head/query cases. We fit one representation per prefix/head using the prefix alone. We use all original cases, without adding contexts, selecting heads or fitting to continuation queries.

For each prefix, let $s_K = \sqrt{\mathrm{mean}(K^2)}$ and $s_V = \sqrt{\mathrm{mean}(V^2)}$, replacing either by one only if it is zero. Concatenate K/s_K and V/s_V. Even and odd feature coordinates form two disjoint subspaces. Each uses seven centroids, with the first initialized to the point closest to its subspace mean and each subsequent centroid to the point farthest from its nearest existing centroid. Ties choose the first index. Twenty Lloyd iterations alternate nearest-centroid assignments and nonempty-cluster means; an empty centroid is retained. A final assignment determines occupancy counts. Embed the two fitted codebooks into their disjoint coordinates and undo the two scales to obtain additive key/value dictionaries.

The dense-occupancy budget determines dictionary size independently of observed error: choose the largest $m = s$ satisfying $8[2m(d+p) + m^2] + 8 \le 16{,}392$. At $d = p = 64$ this is seven. Embedded zero entries remain charged as dense storage. The product candidate sets $A = W\mathbf{1}$ as a single column and $B = W^\top \mathbf{1}/n$ as the other. This retains total mass and both assignment marginals at query zero while discarding dependence. The dense control uses the same dictionaries and exact count matrix.

The joint-centroid control fits the whole normalized concatenated vector with the same deterministic initialization and twenty iterations. Its count is $\lfloor (16{,}392-8)/(8(64+64+1)) \rfloor = 15$, since centroid masses must also be stored. A sixteen-centroid comparison would exceed the budget. Top-anchor and uniform subsets use their original sixteen indices without refitting. Local covariance summaries reuse their original fitted arrays. Each prediction combines its prefix representation with the same exact tail for that head and query.

Table 2: RMS Euclidean head-output error by original context. Each column contains 180 cases, and every method uses the same cases.

Representation	Text 1	Text 2	Text 3
Product: independent	0.1982	0.2296	0.1953
Product: dense occupancy	0.1800	0.2055	0.1703
Joint centroids	0.1139	0.1546	0.1080
Local rank 0	0.1048	0.1014	0.1043
Local rank 1	0.1221	0.1296	0.1224
Local rank 2	0.1208	0.1272	0.1244
Top anchor subset	0.2736	0.2592	0.2309
Uniform subset	0.2159	0.2273	0.2414

Independent occupancy beats joint centroids in 16 cases and loses in 524; exact occupancy beats it in 27 and loses in 513. Against top-anchor, the respective win/loss counts are 255/285 and 282/258. Against uniform they are 323/217 and 361/179. None is a tie at tolerance 10^{-12}. The matrix of occupancies, reconstruction error radii and every per-query nominal bound are retained in the artifact, including bounds that do not admit useful tolerances.

The independent verifier uses neither the product module nor the local summary module. It explicitly expands all positive dictionary pairs, adds the exact tail, and performs separate scalar-sum weighted reductions. It recomputes all method errors, payload identities, original assignment counts and both certificate formulas from the saved arrays. All 4,320 method outputs and 1,080 certificate values agree within the declared floating tolerances. The verifier operates on the same saved records.

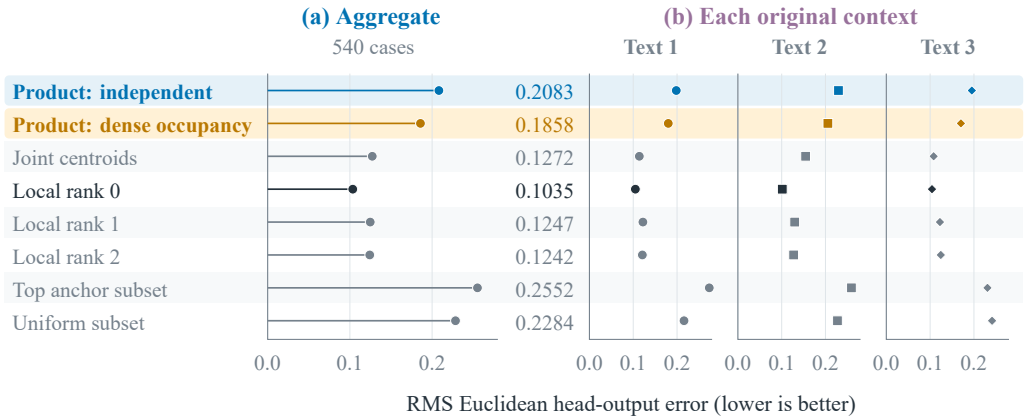

Figure 4: The fixed-cache assessment compares eight representations on the same 540 head/query cases. The cases come from three original contexts, each with a 30-token prefix, twelve heads, and fifteen continuation queries with growing exact tails. (a) Aggregate RMS Euclidean head-output error over all cases. (b) The same metric for each original context, shown in aligned panels with distinct marker shapes. Lower error is better. Blue denotes independent occupancy and orange its exact-occupancy control; clustering and raw subsets are neutral. These existing-cache results use no new model inference, seed selection, or fitted query distribution. The stronger clustering controls retain their lower aggregate errors.

H FIXED-DICTIONARY OCCUPANCY COMPONENT SWEEP

The sweep reuses exactly the 36 seven-by-seven count matrices, four dictionaries per state, 540 query/tail pairs and original-output references described in Appendix G. No dictionary is refitted. All ranks, initializations, fitting criteria and storage conventions are fixed before evaluating the new factors. The original independent and dense outputs are first reproduced, including payloads 14,456 and 14,736 bytes and RMS errors 0.208280 and 0.185840.

Fitting uses only construction information. For $R = 2, 3, 4$, minimize the generalized KL loss

$$D(W\|AB^\top) = \sum_{ji:W_{ji}>0} W_{ji} \log \frac{W_{ji}}{(AB^\top)_{ji}} - \sum_{ji} W_{ji} + \sum_{ji} (AB^\top)_{ji}.$$

We use eight gamma-initialized mixture-of-products EM runs for each state and rank. Initialization $s = 0, \ldots, 7$ uses seed $20260926 + 100000u + 1000R + s$, where $u = 0, \ldots, 35$ indexes context then head. Initially A has total mass 30 and each column of B sums to one. At each iteration, assign responsibilities proportional to $A_{jr}B_{ir}$, multiply them by W_{ji}, set A_{jr} to their sum over i, and set B_{ir} to their sum over j divided by the component mass. These updates preserve the total mass and both occupancy marginals, up to rounding. Every ten iterations, stop if the KL improvement is at most $10^{-11}(1 + D_{\text{previous}})$, with a maximum of 10,000 iterations. Choose the smallest final occupancy KL, breaking exact ties by initialization order. All 864 runs met the stopping rule; the largest iteration count was 6,350. Continuation queries, output errors and tolerances never enter fitting or selection. These are fitted factors, not globally optimal rank-R solutions.

Rank one is the original marginal product, not a newly selected solution. Rank seven is the exact endpoint $A = W, B = I_7$, not an estimated minimum nonnegative rank. All fitted factors preserve both marginals within 2×10^{-12} and hence preserve the reconstructed output at query zero, including the same exact tail.

Actual stored arrays. All methods store the same four dense float64 dictionaries, totaling 14,336 bytes, including their embedded zeros, and one eight-byte coverage integer. Dense factors store $2 \cdot 7R$ float64 entries, or $112R$ occupancy bytes. Dense float64 occupancy stores 392 bytes. The sparse exact control stores a one-byte linear pair index and one float64 weight per occupied pair;

the 36 states have 9–17 occupied pairs, requiring 81–153 bytes (mean 111.75). A count-specific indexed control stores 30 one-byte pair indices, one for each prefix token; repeated indices encode multiplicity. A second integer control stores all 49 counts as uint8. Both are exact because there are only 49 possible pairs and every count is at most 30. These integer encodings do not claim to represent arbitrary real-weight occupancies. Each payload is serialized and independently decoded; the table counts its actual bytes, excluding array headers and the common exact tail. No certificate metadata is used or charged in this output-error sweep.

Table 3: Fixed-dictionary component and encoding comparison on all 540 cases. Each total includes eight coverage bytes in addition to the displayed dictionary and occupancy fields. Sparse bytes are means across the 36 states; its total range is 14,425–14,497. Occupancy RMS measures $o_W - o_R$, whereas RMS error compares with the original uncompressed prefix and the same exact tail. The rank-seven factor and all four exact encodings agree up to rounding.

Representation	Dict. bytes	Occ. bytes	Total bytes	RMS error	Occ. RMS
Positive factors, $R = 1$	14336	112	14456	0.20828	0.08895
Positive factors, $R = 2$	14336	224	14568	0.19597	0.05751
Positive factors, $R = 3$	14336	336	14680	0.19136	0.04571
Positive factors, $R = 4$	14336	448	14792	0.18953	0.03656
Exact factors, $R = 7$	14336	784	15128	0.18584	0.00000
Dense float64 occupancy	14336	392	14736	0.18584	0.00000
Sparse float64 occupancy	14336	111.75	14455.75	0.18584	0.00000
Exact pair indices (uint8)	14336	30	14374	0.18584	0.00000
Exact dense counts (uint8)	14336	49	14393	0.18584	0.00000

Error and storage answer different questions. The aggregate occupancy RMS decreases from 0.08895 at rank one to 0.05751, 0.04571 and 0.03656 at ranks two, three and four. The fixed dictionary MSE remains 0.034536. The corresponding mean cross terms are 0.000933, 0.000562, -0.00000669 and 0.00004823; they are included in every total. Exact occupancy has lower aggregate error, but ranks one through four beat it on 76, 114, 143 and 150 individual queries, respectively, through favorable cancellation with dictionary error. Increasing component count reduces aggregate error in this sweep without creating a useful storage advantage over the exact pair-index encoding. Rank three saves 56 bytes against dense float64 occupancy (0.38% of its total payload), whereas rank four already exceeds it. This empirical comparison does not establish that a different fitting algorithm cannot find better factors.

The independent verifier reads all 324 serialized prediction payloads, reconstructs each occupancy and computes attention by explicit expansion and scalar summation. All 4,860 representation/query outputs reproduce the saved sweep, with maximum coordinate discrepancy 6.67×10^{-16}, and all reported RMS values and error decompositions agree. The portable artifact retains the factors, all initialization losses, statewise bytes and casewise outputs. Reproduction makes no model calls.

I THE PRESERVED LOCAL-SUMMARY STUDY

The earlier corrected experiment evaluated teacher-forced GPT-2 continuation NLL after replacing first-layer prefix attention. It retained the three original contexts and compared native execution, a full reconstruction control, local summaries, and two raw-token subsets. Its native and full reconstruction computations used the model's float32 arithmetic; summary/subset operations used float64 before entering subsequent model computation. Two model runs reproduced all 1,220 saved arrays exactly. The product assessment uses those existing first-layer caches.

The local construction partitions the prefix into clusters at anchor q_0. With anchor-weighted cluster mass Z_j, means $\bar{k}_j, \bar{v}_j$, centered variables x_i, y_i and covariance $C_j = \mathbb{E} x_i y_i^\top$, it stores a rank-$r$ matrix $C_{j,r}$. At displacement $\delta = q - q_0$ it predicts mass $w_j = Z_j e^{\delta^\top \bar{k}_j}$ and value $\bar{v}_j + C_{j,r}^\top \delta$. The combined anchor Jacobian defect is

$$\sum_j \frac{Z_j}{Z_A(q_0) + Z_T(q_0)} (C_j - C_{j,r})^\top. \tag{35}$$

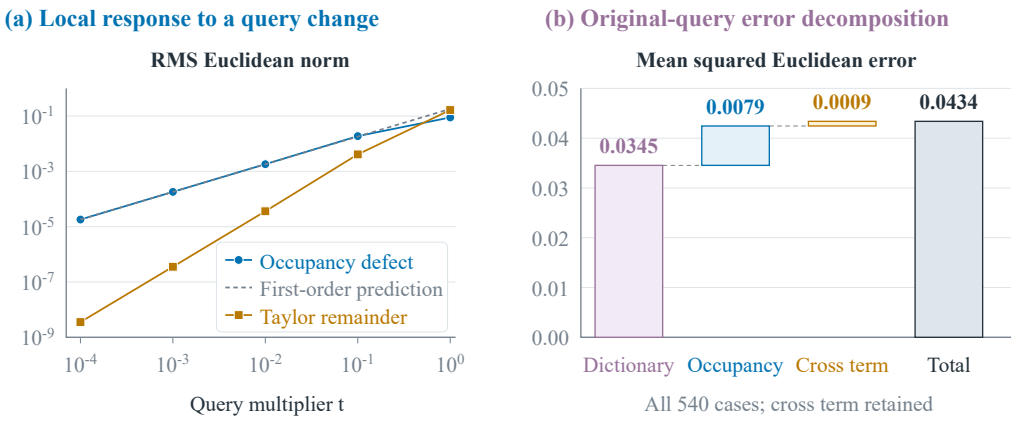

Figure 5: Fixed-dictionary comparisons separate occupancy error from dictionary reconstruction error on all 540 saved cases. The error definitions distinguish the original output o_{raw}, the reconstructed-pair output o_W, and the independent-occupancy output o_{ind}, with $e_d = o_{\mathrm{raw}} - o_W$ and $e_w = o_W - o_{\mathrm{ind}}$. (a) With the same dictionaries and exact tail at query tq, the first-order prediction tracks the occupancy output difference locally. The Taylor remainder is second order near zero and substantial at $t = 1$; zero is omitted from the logarithmic axis. (b) At the original queries, the mean dictionary squared error, occupancy squared error, and cross term sum to the mean total squared error. The cross term is retained; the two error sources need not add without it.

Table 4: Preserved end-to-end integration result, with 45 target predictions per method across the three contexts. These are not measurements of the new product state.

Original integration method	Mean target NLL
Native	0.931584
Rank zero, scale discarded	1.067264
Rank zero, retained partition	0.970857
Rank one, retained partition	0.983321
Rank two, retained partition	0.999335
Top-anchor raw subset	0.936119
Uniform raw subset	0.940988

It depends on the weighted aggregate, so independently truncating each covariance need not be optimal even at first order. An aggregate low-rank correction can also be placed in a single cluster with an appropriate rescaling. That observation alone does not establish a representation lower bound or the product-state result in this paper.

The corrected local finite bound uses $h = \|\delta\|$, cluster radii $a_j = \max\|x_i\|$, moments $S_j = \mathbb{E}\|x_i\|^2$, $T_j = \mathbb{E}[\|x_i\|^2\|y_i\|]$, residuals $e_j = \|C_j - C_{j,r}\|_{\mathrm{op}}$, and

$$\phi_j = (e^{ha_j} - 1 - ha_j)/a_j^2, \qquad u_j = \phi_j S_j, \qquad v_j^{\mathrm{rem}} = \phi_j T_j, \tag{36}$$

with $\phi_j = h^2/2$ at $a_j = 0$. Writing $p_j = w_j/(\sum_j w_j + Z_T)$, its combined output error is bounded by

$$\sum_j p_j \left[u_j\|\bar{v}_j - \widehat{o}\| + he_j + v_j^{\mathrm{rem}}\right]. \tag{37}$$

To see this, let $M_j = \mathbb{E}e^{\delta^\top x_i} \geq 1$ by centering and Jensen. The true cluster numerator is $w_j[M_j\bar{v}_j + C_j^\top\delta + R_j]$ with $\|R_j\| \leq v_j^{\mathrm{rem}}$ and $0 \leq M_j - 1 \leq u_j$. Subtract the approximate combined output times the true denominator from its numerator. Exact-tail terms cancel, and the true denominator is at least $\sum_j w_j + Z_T$. Triangle inequalities yield the displayed bound. Differentiation at zero gives the stated Jacobian defect. All of these statements concern exact arithmetic and the fixed cache.

Earlier integrations that discarded common log-partition offsets or advanced a coverage count without retaining the rebuilt summary were superseded by this corrected study. Their drift or rebuild

claims are not used here. The source artifact retains the corrected records, their independent reductions and the explicit scope of the new saved-cache comparison.