RANDOMLEX: EXACT RANDOMIZED RATIOS FOR THE UNIFORM k-RESOURCE PROBLEM AND DEPENDENCY CACHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Prefix/KV caches must retain prerequisite states along with each reusable computation. Physical capacity can therefore greatly exceed the number of independent caching choices. We determine the exact randomized complexity of the uniform k-resource problem, which isolates this distinction. RANDOMLEX samples one coordinate order and follows the lexicographically maximal work-function minimizer; against oblivious adversaries on an n-point uniform metric, it is $H_{\min\{k,n-1\}}$-competitive, matching the known lower bound. The proof represents normalized work functions as distance cones over M-convex minimizer families: free updates slice the family, paid updates perform one-unit rank truncations, and a Shapley-weighted potential gives the matching harmonic charge. A nonexpansive legality retraction and a finite-chain invariant extend the result to uniform-cost, no-bypass dependency caching. For posets of width at most ℓ, the exact ratio is $H_{\min\{k,\ell-1\}}$ for $\ell \geq 2$ and one for chains. Dependency width determines the effective dimension of online choice.

1 INTRODUCTION

Online memory managers often exploit dependencies: a cached computation is usable only when its prerequisites are present. In large-language-model serving, prefix KV states form paths in a radix tree; systems such as SGLang reuse these states across requests, while vLLM manages the underlying memory in pages (Kwon et al., 2023; Zheng et al., 2024). These dependencies separate two quantities that coincide in ordinary paging: physical cache capacity and the number of *independent choices* competing for it. Recent work analyzes randomized KV-cache eviction through competitive analysis (Wu et al., 2026). We ask what determines online complexity when capacity may be much larger than the number of independent feasible directions.

We approach the question through the uniform k-resource problem. There are k indistinguishable resource units distributed across n coordinates; an integer demand vector of total mass at most k must be dominated at every time, and moving one unit between coordinates costs one. The model generalizes paging. The best previous randomized lower and upper bounds, respectively, were

$$H_{\min\{k,n-1\}} \quad \text{and} \quad \min\{H_k, 2H_{n-1}\},$$

leaving a factor-two gap in the unresolved regime $n - 1 < k$ (Bieńkowski and Kutyłowski, 2012). This regime is precisely where capacity is larger than the number of independent coordinates, so a capacity-only harmonic bound is no longer the right description. Our first result closes the gap and shows that the smaller quantity $\min\{k, n-1\}$ is the exact effective dimension.

RANDOMLEX samples one uniform permutation of the coordinates and reuses it throughout the request sequence. After each request, it selects the lexicographically maximal minimizer of the normalized extended work function. The analysis depends on the geometry of the evolving minimizer set. Every normalized work function is a distance cone over an M-convex family, and the random order selects one extreme base of that family. A free request slices the family, while a paid request removes one unit of polymatroid rank. Under a Shapley-style lower-rank potential, the free update has zero expected amortized cost and the paid update incurs the harmonic charge needed to match the lower bound.

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.

For dependency caching, the relevant topology parameter is the poset width. If the dependency poset has width p, Dilworth's theorem partitions it into p chains. A legal full cache can then be represented by the number of cached prefix elements on each chain, turning the problem into a p-coordinate resource instance. Two domain issues make this reduction nontrivial: arbitrary resource vectors need not correspond to legal ideals, and finite chains impose upper bounds on each coordinate. A Johnson-nonexpansive retraction maps arbitrary vectors to legal ideals, and an invariant keeps every reachable work-function minimizer inside the finite-chain box. The width is therefore the number of effective independent cache directions: a chain has width one and no online choice, while an antichain recovers ordinary paging.

Contributions.

1. We prove that RANDOMLEX is $H_{\min\{k,n-1\}}$-competitive, matching the known lower bound and determining the exact randomized ratio of the uniform k-resource problem.
2. We give an exact update calculus for M-convex minimizer sets. Free coordinate increases obey a lower-rank slice formula and have zero expected amortized cost. Paid increases are ordinary one-unit rank truncations, whose charge is bounded by the harmonic number of the support of one base.
3. We prove the exact width law for uniform-cost, no-bypass dependency caching: $H_{\min\{k,\ell-1\}}$ for width at most $\ell \geq 2$, and one for $\ell = 1$. The result also resolves a one-index discrepancy in the topology-sensitive lower bound of Dallot et al. (2024); Section H gives the cover-time calculation and a direct paging reduction establishing the tight $\ell - 1$ index.

Proof roadmap. Figure 1 summarizes the core argument. First, infimal convolution with the uniform movement metric preserves M-convexity and yields the exact distance-cone identity. Second, lower ranks convert lexicographic extreme bases into suffix differences along the sampled permutation. Third, a free slice changes those lower ranks by a 0–1 down-set, so its unique movement event is canceled by the potential. Fourth, a paid update is an ordinary rank truncation; grouping Shapley coefficients bounds its charge by $H_{|D|}$ for the support D of one residual base, hence by $H_{\min\{k,n-1\}}$. Telescoping gives the upper bound, and the known paging-type lower bounds make it exact. The dependency theorem then uses the same result after the chain-prefix reduction.

2 MODEL AND ALGORITHM

We first analyze the uniform k-resource problem. The dependency-caching model and its reduction appear in Section 4.

Let $E = [n]$, $n \geq 2$, and

$$X_{n,k} = \{x \in \mathbb{Z}^E_{\geq 0} : x(E) = k\}, \qquad d(x, y) = \tfrac{1}{2}\|x - y\|_1.$$

A request is $a \in \mathbb{Z}^E_{\geq 0}$ with $a(E) \leq k$; x serves a when $x \geq a$. Set $a_0 = 0$ and refine each transition $a_{t-1} \to a_t$, in a fixed coordinate order, into unit decreases followed by unit increases. This refinement does not change offline cost: the old configuration serves all decreases, the new configuration serves all increases, and the movement between them is performed once. Against an oblivious adversary, a randomized algorithm is c-competitive, for $c \geq 1$, if for some sequence-independent constant β,

$$\mathbb{E}[\mathrm{cost}_{\mathcal{A}}(\sigma)] \leq c\,\mathrm{OPT}(\sigma) + \beta$$

for every finite request sequence σ.

Fix an initial state x_0 and initialize

$$w_0(x) = d(x, x_0), \qquad V_0 = \{x_0\}.$$

The extended work function after a request prefix is the minimum service cost followed by arbitrary movement to endpoint x; the endpoint itself need not serve the latest request. If w is the previous normalized work function and the new feasible set is $F(a) = \{y \in X_{n,k} : y \geq a\}$, then

$$\widehat{w}^+(x) = \min_{y \in F(a)} \{w(y) + d(y, x)\}. \tag{1}$$

Define the updated normalized work function and minimizer family by

$$w^+(x) := \widehat{w}^+(x) - \min_z \widehat{w}^+(z), \qquad V^+ := \arg\min_x w^+(x).$$

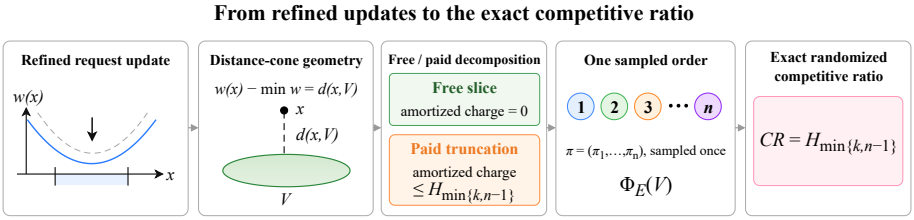


Figure 1: Proof architecture. Distance-cone geometry reduces each refined request to a free slice or a one-unit rank truncation. A sampled order and a lower-rank potential then give the exact harmonic ratio.

RANDOMLEX. Sample a uniform permutation $\pi = (\pi_1, \dots, \pi_n)$ once. For nonempty V, let

$$g_\pi(V) := \operatorname*{lex\,arg\,max}_{x \in V} (x_{\pi_1}, \dots, x_{\pi_n}),$$

and after every refined request output $x_t = g_\pi(V_t)$. The permutation is the algorithm's only randomness. We work in the oracle model: given the current work function, the oracle returns its lexicographically maximal minimizer.

3 ANALYSIS OF RANDOMLEX

Figure 1 summarizes the four interfaces in the argument.

3.1 WORK FUNCTIONS ARE EXACT M-CONVEX DISTANCE CONES

The core proof uses three standard discrete-convex interfaces: the M-convex/integral-base correspondence (Murota, 2003, Theorem 4.15), integer infimal-convolution closure (Murota, 2003, Theorem 6.13), and the M-optimality criterion (Murota, 2003, Theorem 6.26). Their hypotheses are recorded in Section J. The appendix directly proves the remaining ingredients: the distance kernel, coordinate restriction, minimizer-set closure, greedy suffix formula, and truncation augmentation.

Lemma 1 (Work-function cone)**.** *Every normalized extended work function is integer-valued, M-convex, and one-Lipschitz in d. Consequently,*

$$w(x) - \min w = d(x, V) \qquad (x \in X_{n,k}). \tag{2}$$

Proof sketch. Let $\varphi(z) = \sum_i \max\{z_i, 0\} + \delta_{\{z(E)=0\}}(z)$, which equals $\frac{1}{2}\|z\|_1$ on the zero-sum lattice and is M-convex. Restricting w to $F(a)$ is M-convex by Lemma 11; (1) is its infimal convolution with φ. The triangle inequality gives one-Lipschitzness. The M-exchange inequality also shows directly that $V = \arg\min w$ is an M-convex set. Finally, the M-optimality criterion (Murota, 2003, Theorem 6.26) provides a strictly decreasing unit exchange outside V. Integrality and one-Lipschitzness force every decrease to equal one; iterating and reversing the Lipschitz inequality proves (2). Full details are in Section B. □

Lemma 2 (Refinement and normalization accounting)**.** *The sum of normalization increments over the refined request sequence equals the offline optimum of the original sequence.*

Proof. Let W_t be the unnormalized extended work function after refined request t. Its minimum is the offline prefix optimum, so the normalization increment is $\min W_t - \min W_{t-1}$ and the increments telescope. Refinement preserves the offline optimum: for an original transition $a \to b$ through $a \wedge b$, every refined trajectory serves the original endpoints; conversely, hold the old endpoint through the decreases, move once to the new endpoint, and hold it through the increases. □

By (2), a unit demand increase has normalization increment $c = \operatorname{dist}(V, F(a + e_i)) \in \{0, 1\}$. When $c = 0$, the new minimizers are the feasible slice of V; when $c = 1$, they are exactly the feasible states one exchange away from V.

Table 1: Amortized accounting. Splitting the free case at the current minimizer threshold covers every update; the last column is movement plus potential change.

Update	ΔOPT	minimizer family	expected amortized cost
demand decrease	0	unchanged	0
trivial free increase	0	unchanged	0
nontrivial free increase	0	feasible slice	0
paid increase	1	rank truncation	$\leq H_{\min\{k,n-1\}}$

3.2 Lower ranks and the potential

For a nonempty M-convex set $V \subseteq X_{n,k}$ define

$$p_V(S) = \min_{x \in V} x(S).$$

By the M-convex/integral-base correspondence (Murota, 2003, Theorem 4.15), p_V is an integral supermodular lower-rank representation of V. By supermodularity, Lemma 13 shows that for suffixes $T_j = E \setminus \{\pi_1, \ldots, \pi_j\}$,

$$g_\pi(V)_{\pi_j} = p_V(T_{j-1}) - p_V(T_j). \tag{3}$$

Define

$$\Phi_E(V) = -\sum_{\emptyset \neq S \subseteq E} \frac{p_V(S)}{|S|\binom{n}{|S|}}. \tag{4}$$

The coefficient is exactly the probability that S is the suffix immediately before a distinguished element of S is deleted from a uniform permutation, the same random-order weighting underlying the Shapley value (Shapley, 1953).

A three-coordinate illustration. For $n = 3$, $k = 2$, take the M-convex family $V = \{(2,0,0),(1,1,0),(1,0,1)\}$ and raise the first-coordinate demand from one to two. This is a nontrivial free slice with $V^+ = \{(2,0,0)\}$. If coordinate 1 is first in the random order, the old lexicographic state is already $(2,0,0)$; otherwise it is one exchange away. Hence expected movement is $2/3$. Directly from (4), $\Phi(V) = -4/3$ and $\Phi(V^+) = -2$, so the potential change is $-2/3$ and the amortized charge is exactly zero. Thus movement and potential cancel in the smallest nontrivial case of Lemma 3.

3.3 Free and paid updates

3.3.1 Free updates conserve amortized cost

Let the current demand be a and put $\alpha = \min_{x \in V} x_i$. For a unit increase $a_i \to a_i + 1$, if $a_i + 1 \leq \alpha$, every minimizer already serves the new demand, so $V^+ = V$ and both movement and potential change are zero. Otherwise, feasibility of the old minimizers forces $a_i = \alpha$. The remaining free update is the nontrivial slice $V^+ = \{x \in V : x_i \geq \alpha + 1\} \neq \emptyset$.

Lemma 3 (Exact free-update calculus)**.** *For every $S \subseteq E$,*

$$p_{V^+}(S) = \begin{cases} p_V(S), & i \notin S, \\ \max\{p_V(S), p_V(S \setminus \{i\}) + \alpha + 1\}, & i \in S. \end{cases}$$

If $h = p_{V^+} - p_V$, then $h \in \{0,1\}$ and $h(A \cup \{i\}) = 1$ precisely when the marginal $p_V(A \cup \{i\}) - p_V(A)$ is minimal. These A form a down-set. Therefore, for a uniform π,

$$\mathbb{E}_\pi d(g_\pi(V), g_\pi(V^+)) + \Phi_E(V^+) - \Phi_E(V) = 0. \tag{5}$$

For each fixed permutation the movement is zero or one.

The lower-rank formula follows from a base simultaneously tight on the nested sets $A \subseteq A \cup \{i\}$ and one symmetric exchange. Along the random suffix chain, h has the form $0 \to 1 \to 0$; its discrete derivative in (3) is exactly the movement vector. Averaging the unique active suffix gives (5). See Section D.

3.3.2 PAID UPDATES ARE RANK TRUNCATIONS

For a paid increase at i, every old minimizer has $v_i = a_i$. Translate by the current demand a, delete coordinate i, and let

$$W \subseteq X_{m,K}, \qquad m = n-1, \quad K = k - a(E).$$

The new residual family is its lower shadow $W^- := \{w - e_j : w \in W,\ w_j > 0\} \subseteq X_{m,K-1}$.

Lemma 4 (Paid-update calculus)**.** *If $r_W(S) = \max_{w \in W} w(S)$, then*

$$r_{W^-}(S) = \min\{r_W(S), K-1\}, \qquad p_{W^-}(S) = \max\{p_W(S) - 1, 0\}.$$

Thus W^- is the base family of the ordinary rank truncation. For every fixed permutation, the old and new RANDOMLEX *states differ by exactly one resource move. Moreover,*

$$1 + \Phi_E(V') - \Phi_E(V) \le H_{\min\{K,n-1\}}. \tag{6}$$

To see why the truncation is exact, first note that every $u = w - e_j \in W^-$ satisfies $u(S) \le r_W(S)$ and $u([m]) = K-1$, so its upper rank is at most $\min\{r_W(S), K-1\}$. Conversely, take an integer vector u satisfying these truncated inequalities. If no coordinate could be increased while preserving them, then each coordinate would lie in a set tight for r_W; tight sets are closed under union by submodularity, making $[m]$ tight and contradicting $u([m]) = K-1 < r_W([m]) = K$. Thus some $u + e_j$ is a base of W, proving that no truncated base is missing. Along the greedy prefix chain, clipping total rank at $K-1$ then changes exactly one greedy coordinate.

Grouping the potential coefficients for A and $A \cup \{i\}$ gives

$$1 + \Delta\Phi = \sum_{\emptyset \ne S \subseteq [m]} \frac{\mathbf{1}\{p_W(S) > 0\}}{|S| \binom{m}{|S|}}.$$

Choose any $w \in W$ with support D. Positive lower rank implies $S \cap D \ne \emptyset$, and the beta-integral identity

$$\sum_{S : S \cap D \ne \emptyset} \frac{1}{|S| \binom{m}{|S|}} = H_{|D|}$$

gives (6), since $|D| \le \min\{K, n-1\}$. The appendix gives the full augmentation and coefficient calculation in Section E.

Why the dimension is $\min\{k, n-1\}$. Once the requested coordinate i is fixed, all residual movement occurs among the other $n-1$ coordinates. A residual base has total mass $K \le k$, so its support contains at most $\min\{K, n-1\}$ coordinates. The Shapley coefficients charge at most the harmonic number of this support. This same argument replaces n by dependency width in the caching result.

3.4 TELESCOPING THE EXACT RATIO

The potential obeys $-k \le \Phi_E(V) \le 0$, with equality $\Phi_E(\{x\}) = -k$ for a singleton. The normalization constants in (1) telescope to OPT: minimizing the unnormalized work function after each refined request gives the offline prefix optimum, and refinement preserves the original optimum.

Theorem 5 (Exact randomized k-resource ratio)**.** *For every $n \ge 2$, $k \ge 1$, and every fixed request sequence,*

$$\mathbb{E}[\mathrm{ALG}] \le H_{\min\{k,n-1\}} \mathrm{OPT}.$$

Hence RANDOMLEX *is $H_{\min\{k,n-1\}}$-competitive against an oblivious adversary. Together with the known lower bound, this is the exact randomized competitive ratio.*

Proof. Demand decreases and trivial free increases change neither the normalized work function nor the selected state. A nontrivial free increase has zero expected movement-plus-potential by Lemma 3. A paid increase raises the offline prefix optimum by one and has amortized cost at most $H_{\min\{k,n-1\}}$ by Lemma 4. For the fixed request sequence, every minimizer set is deterministic. Thus the per-update expectations apply to the single reused permutation and sum by linearity of expectation. After telescoping, the boundary term $\Phi_E(V_T) - \Phi_E(V_0)$ is nonnegative. The matching lower bound is due to Bieńkowski and Kutyłowski (2012). □

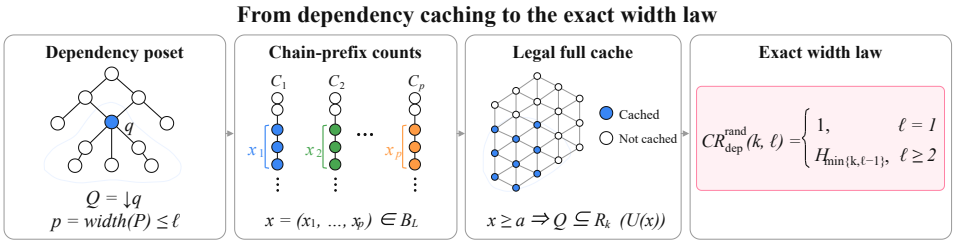


Figure 2: Dependency-caching transfer. A dependency poset induces chain-prefix counts. A nonexpansive retraction lifts these counts to legal full caches, yielding the exact width law.

4 Exact Width Law for Dependency Caching

Let P be a finite poset. A legal cache is an order ideal; serving object q requires its dependency closure $\downarrow q$. Cache capacity is at most k, evictions are free, every fetched object costs one, and bypass is disallowed. Every request is *capacity-feasible*:

$$|\downarrow q_t| \le k \qquad \text{for every } t. \tag{7}$$

Let ℓ bound the width of P, and assume $k \le |P|$ (otherwise replace k by $|P|$). For the exact upper bound, the online algorithm and comparator start from the same fixed legal ideal I_0 of size k, whose initial contents are uncharged. A separate corollary below gives an explicit additive term for any smaller common initial ideal.

4.1 A nonexpansive legality retraction

For a cover $u \prec v$, define the compression

$$C_{u,v}(S) = \begin{cases} S - \{v\} + \{u\}, & v \in S,\ u \notin S, \\ S, & \text{otherwise.} \end{cases}$$

Each compression is one-Lipschitz for Johnson distance $d_J(A,B) = \frac{1}{2}|A \triangle B|$. Repeated sweeps through all covers strictly lower a topological-rank sum whenever they change the set, and hence stabilize at a size-k ideal. Denote the stable map by R_k.

Lemma 6 (Legality retraction)**.** *The map R_k is Johnson-nonexpansive, fixes every legal size-k ideal, and preserves contained requests: if an ideal $Q \subseteq S$, then $Q \subseteq R_k(S)$.*

Proof. For one cover compression, if both or neither of A, B change, their distance is unchanged. If only A changes, replacing a violating successor v by its predecessor u cannot increase $|A \triangle B|$: the reason B does not change is exactly that either $v \notin B$ or $u \in B$. Hence every compression, their composition, and every iterate are nonexpansive. A fixed point has no violated cover and is therefore an ideal; ideals are fixed from the start. Finally, if $Q \subseteq S$ is an ideal, no compression can remove $v \in Q$, because its predecessor u also lies in $Q \subseteq S$ and blocks that compression. Thus Q remains contained throughout the sweep. □

Lemma 7 (Full-cache simulation)**.** *Let I_0 be a legal ideal of size at most k, and let $J_0 \supseteq I_0$ be any legal ideal of size k. Every trajectory of ideals of size at most k starting from I_0 can be lifted online to a size-k ideal trajectory starting from J_0, with no larger transition fetch cost. If an online algorithm itself starts at I_0 of size $s < k$, it can first reach such a J_0 using exactly $k - s$ fills.*

Proof. Maintain a full ideal K containing the current comparator ideal I. For a transition $I \to J$, process the new objects $J \setminus I$ in topological order. Before inserting a missing object z, remove a maximal element of $K \setminus J$; such an element exists because $z \notin K$ while $|K| = k \ge |J|$, and it is also maximal in K because J is downward closed. All predecessors of z lie in J and have already been inserted, so adding z preserves legality. Each object fetched by the comparator causes at most one exchange of the full cache, giving the transition-wise cost bound. Starting from size $s < k$, repeatedly adding a minimal outside object reaches a containing legal size-k ideal in exactly $k - s$ fills. □

Let $p = \mathrm{width}(P) \le \ell$. By Dilworth's theorem, partition P into p chains $C_1, \dots, C_p$ (Dilworth, 1950), put $L_r = |C_r|$, and define the finite-chain box

$$B_L := \{x \in X_{p,k} : 0 \le x_r \le L_r \text{ for every } r\}.$$

A requested closure $Q = \downarrow q$ gives lower bounds $a_r = |Q \cap C_r|$, with $a_r \le L_r$ and $a([p]) = |Q| \le k$ by (7). The following invariant ensures that the chain-prefix reduction stays within this domain.

Lemma 8 (Finite-chain box invariance)**.** *Run the unrestricted p-coordinate k-resource work-function process from the count vector of I_0. After every refined update, its minimizer family satisfies $V_t \subseteq B_L$. Consequently every* RANDOMLEX *state lies in B_L, so the union $U(x)$ of the first x_r elements of each C_r is always well-defined and has size k.*

Proof. Initially $V_0 = \{x(I_0)\} \subseteq B_L$. Suppose $V \subseteq B_L$ before one refined update. A demand decrease leaves V unchanged, while a free increase either leaves V unchanged or replaces it by a subset. At a paid increase of coordinate i, every old minimizer has $v_i = a_i$ and every new minimizer is $x = v + e_i - e_j$ for some $v \in V$ by Lemma 1. Since the refined request has $a_i + 1 \le L_i$, the increased coordinate satisfies $x_i = a_i + 1 \le L_i$; the donor only decreases and all other coordinates are unchanged. Hence $x \in B_L$, and induction proves the claim. □

For $x \in B_L$, expanding counts to chain prefixes preserves resource distance. Applying R_k restores all cross-chain dependencies without increasing movement or removing the request.

Theorem 9 (Exact dependency-width law)**.** *Under* (7)*, for uniform-cost, no-bypass dependency caching on the class of posets of width at most ℓ, with a common uncharged legal size-k initial cache, the optimal randomized competitive ratio against an oblivious adversary is*

$$\mathrm{CR}^{\mathrm{rand}}_{\mathrm{dep}}(k, \ell) = \begin{cases} 1, & \ell = 1, \\ H_{\min\{k,\ell-1\}}, & \ell \ge 2. \end{cases}$$

Upper and lower bounds. Let $p = \mathrm{width}(P)$. For $p \ge 2$, apply Theorem 5 to the p chain counts. By Lemma 8, every selected resource state has a valid finite prefix union; R_k maps it to a legal full cache, preserves the requested ideal, and cannot increase movement. Conversely, every full legal cache is exactly the union of its chain prefixes, so its count vector has the same movement cost. Lemma 7 lifts any at-most-k comparator trajectory from the common full initial cache without extra cost. Hence

$$\mathbb{E}[\mathrm{ALG}] \le H_{\min\{k,p-1\}}\mathrm{OPT} \le H_{\min\{k,\ell-1\}}\mathrm{OPT}.$$

If $\ell \ge k + 1$, ordinary paging on $k + 1$ incomparable objects is a width-$\le \ell$ instance with lower bound H_k. If $2 \le \ell \le k$, take $\ell - 1$ isolated objects and one chain of length $k - \ell + 2$. By Lemma 7, restricting to full legal caches does not weaken the lower bound. There are exactly ℓ such caches: omit an isolate or omit the chain maximum. Requests to these ℓ distinguished objects therefore induce paging with cache $\ell - 1$, giving $H_{\ell-1}$; see Section H for the explicit equivalence. The width-one case has a unique full ideal. □

For a poset of width p, the construction gives

$$p = 1 : \ 1, \qquad 2 \le p \le k : \ H_{p-1}, \qquad p \ge k + 1 : \ H_k.$$

Capacity determines how much state can be retained, while width determines how many independent cache directions compete online. Once $p \le k$, increasing capacity does not increase the worst-case online choice penalty. The trace measurements below quantify this distinction.

Corollary 10 (Smaller initial caches)**.** *Let the algorithm and comparator start from a common legal ideal of size $s \le k$. With $H = H_{\min\{k,\ell-1\}}$ for $\ell \ge 2$ (and $H = 1$ for $\ell = 1$),*

$$\mathbb{E}[\mathrm{ALG}] \le H\,\mathrm{OPT} + (k - s).$$

Hence the exact ratio of Theorem 9 also holds under the standard additive-constant definition, including an empty initial cache.

Proof. Extend the initial ideal to a containing legal size-k ideal using $k - s$ fills. By Lemma 7, any comparator trajectory lifts from the same full initial ideal without larger transition cost; Theorem 5 then gives the displayed bound. □

Table 2: Exact adversarial search at horizon 300.

n	k	states	edges	best ratio	bound
3	2	27	84	$227/152$	$3/2$
4	3	368	1,752	$277/153$	$11/6$

5 Computational and Trace Evidence

The experiments test two parts of the theory: the proof identities through exact arithmetic, and width-limited structure in public prefix workloads.

Exact adversarial search and replay. We enumerate the reachable normalized work-function graph. At every step, an exact-rational dynamic program chooses a demand that maximizes the finite-horizon ratio. Table 2 gives exact finite-horizon ratios close to both nontrivial bounds. In a separate replay over 2,020 random request sequences with $n \le 5$ and $k \le 4$, the companion code enumerates all $n!$ coordinate orders and checks the endpoint dynamic program, refinement, normalization accounting, distance-cone identity, and competitive inequality.

Check	Cases / states	Updates / edges	Pair checks
All M-convex sets	254 sets	659 free, 164 paid	–
Reachable graph $(4, 4)$	1,606	8,196	exact rational
Reachable graph $(4, 5)$	5,774	30,908	exact rational
Poset retraction $n \le 5$	407 posets	2,378 maps	92,918
Low-width family	28 (k, ℓ) pairs	exact counts	–
Prefix-width utility	JSONL input	prefix trie	leaves = width

Table 3: Verification coverage included in the companion code.

The reachable-graph arrays store the ordinary work function on endpoints feasible for the latest request and $+\infty$ elsewhere. On feasible endpoints these values equal the metric-extended work function used in the proof, so the minimizer sets and RandomLex states coincide. The graph check covers these states and the edgewise amortized inequality. The full distance-cone lemma is proved analytically in Section B.

Complete-block trace measurements. For each workload, we serialize 100 duplicate-free samples with the Qwen2.5-7B-Instruct tokenizer (Qwen Team, 2024): ShareGPT V3, UltraChat (Ding et al., 2023), and the QMSum and NarrativeQA tasks (Zhong et al., 2021; Kočiský et al., 2018) distributed through LongBench (Bai et al., 2024). After discarding incomplete final blocks, we build the resulting prefix forest and measure its leaf width ℓ. To separate the structural effect from the finite number of requests, we report

$$G_{\mathrm{total}} = \frac{H_k}{B(\ell)}, \qquad G_{\mathrm{struct}} = \frac{H_{N-1}}{B(\ell)}, \qquad B(\ell) = \begin{cases} 1, & \ell = 1, \\ H_{\ell-1}, & \ell \ge 2. \end{cases}$$

The quantity G_{total} measures the direct improvement over the capacity-only H_k law. The quantity G_{struct} removes the finite-request contribution to isolate topology. In these expressions, N counts requests and k counts cache blocks. Table 4 uses 16-token blocks and $k = 8192$; all shown instances satisfy the no-bypass feasibility condition that every request closure fits in the cache.

The traces show two structural regimes. Shared-document QA creates broad branching, whereas multi-turn conversations create deeper chains and yield 1.20–$1.23\times$ structural improvements across the resamples. Prefix length controls the cost of a miss, while dependency width controls online choice complexity.

When dependency width improves the bound. A radix-tree KV cache is a dependency poset in which retaining a node requires its prefix ancestors. After discretization, equal-size token blocks and uniform recomputation cost fit this structure. The width law is most informative for a fixed session, shard, tenant, or other natural namespace with chain-like extension, where width can be much smaller

Table 4: Topology statistics over 100 duplicate-free samples per workload at 16-token blocks. Parentheses give IQR; brackets give the bootstrap 95% CI of median G_{struct}.

Workload	N	Width ℓ (IQR)	G_{total}	G_{struct} [95% CI]
ShareGPT multi-turn	512	144 (12)	1.730	1.229 [1.226, 1.234]
UltraChat multi-turn	512	162.5 (6.2)	1.693	1.203 [1.201, 1.205]
QMSum shared-doc. QA	100	89 (3)	1.895	1.023 [1.021, 1.023]
NarrativeQA shared-doc. QA	100	95 (2)	1.871	1.010 [1.010, 1.010]

than capacity. In this regime, the guarantee is controlled by dependency width rather than cache capacity alone.

6 Related Work

Competitive analysis provides the standard framework for online service and caching (Borodin and El-Yaniv, 1998). The k-server problem was introduced by Manasse et al. (1990); the work-function algorithm is $(2k-1)$-competitive (Koutsoupias and Papadimitriou, 1995), and randomized algorithms achieve polylogarithmic bounds on arbitrary finite metrics (Bansal et al., 2011). The uniform k-resource problem generalizes both k-server and paging; Bieńkowski and Kutyłowski (2012) gave asymptotically tight bounds but left the exact randomized upper bound open. Exact randomized paging has ratio H_k (Fiat et al., 1991; McGeoch and Sleator, 1991).

Our proof draws on M-convex work functions and integral base polyhedra (Murota, 2003; Fujishige, 2005). Polymatroid rank and greedy structure originate with Edmonds (1970). We use Murota's Theorems 4.15, 6.13, and 6.26. The potential applies the random-order coefficients of the Shapley value (Shapley, 1953) to lower ranks instead of cooperative-game payoffs.

Dallot et al. (2024) studied dependency-aware caching on arbitrary DAGs, including no-bypass and bypass variants. Their full version states a $2H_{\min\{k,\ell\}}$ upper bound and an $H_{\min\{k,\ell\}}$ lower bound; Section H gives the exact cover-time calculation and direct paging embedding that clarify the tight $H_{\ell-1}$ index. Modern KV-cache systems exploit shared-prefix structure (Kwon et al., 2023; Zheng et al., 2024), and Wu et al. (2026) study randomized competitive KV eviction with learned routing. Our theorem gives an exact law in capacity and topology for the corresponding uniform dependency abstraction. Section A gives additional systems context.

7 Conclusion

We determined the exact randomized competitive ratio of the uniform k-resource problem. RandomLex samples one coordinate order and repeatedly selects the lexicographically maximal work-function minimizer, attaining $H_{\min\{k,n-1\}}$. The analysis represents each normalized work function as an M-convex distance cone: free updates slice its minimizer family, paid updates truncate one unit of rank, and a lower-rank potential accounts for both operations. Combined with the chain-prefix reduction and a nonexpansive legality retraction, this argument gives the exact dependency-caching ratio $H_{\min\{k,\ell-1\}}$.

The result separates storage capacity from online choice complexity. Ordinary paging recovers the H_k penalty, whereas a dependency chain has no competitive loss; the intermediate cases are governed by poset width. Our trace study measures this capacity-width gap on public prefix workloads. Natural extensions include nonuniform object sizes and fetch costs, bypass policies, and dependency structures that change over time. A central question is whether these models admit an equally sharp structural parameter and a comparably simple randomized policy.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of ACL*, pages 3119–3137, 2024.

Nikhil Bansal, Niv Buchbinder, Aleksander Madry, and Joseph Naor. A polylogarithmic-competitive algorithm for the k-server problem. In *Proceedings of FOCS*, pages 267–276, 2011.

Marcin Bieńkowski and Jarosław Kutyłowski. The k-resource problem in uniform metric spaces. *Theoretical Computer Science*, 459:42–54, 2012.

Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

Julien Dallot, Amirmehdi Jafari Fesharaki, Maciej Pacut, and Stefan Schmid. Dependency-aware online caching. In *IEEE INFOCOM*, pages 871–880, 2024.

Robert P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51 (1):161–166, 1950.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of EMNLP*, pages 3029–3051, 2023.

Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and Their Applications*, pages 69–87, 1970.

Amos Fiat, Richard M. Karp, Michael Luby, Lyle A. McGeoch, Daniel D. Sleator, and Neal E. Young. Competitive paging algorithms. *Journal of Algorithms*, 12(4):685–699, 1991.

Satoru Fujishige. *Submodular Functions and Optimization*. Elsevier, second edition, 2005.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt Cache: Modular attention reuse for low-latency inference. In *Proceedings of Machine Learning and Systems*, volume 6, pages 325–338, 2024.

Chao Jin, Zili Zhang, Xuanlin Jiang, Fangyue Liu, Shufan Liu, Xuanzhe Liu, and Xin Jin. RAGCache: Efficient knowledge caching for retrieval-augmented generation. *ACM Transactions on Computer Systems*, 44(1):2:1–2:27, 2025.

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.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

Elias Koutsoupias and Christos H. Papadimitriou. On the k-server conjecture. *Journal of the ACM*, 42(5):971–983, 1995.

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 ACM Symposium on Operating Systems Principles*, pages 611–626, 2023.

Yucheng Li, Huiqiang Jiang, Qianhui Wu, Xufang Luo, Surin Ahn, Chengruidong Zhang, Amir H. Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and Lili Qiu. SCBench: A KV cache-centric analysis of long-context methods. In *International Conference on Learning Representations*, 2025.

Yuhan Liu, Hanchen Li, Yihua Cheng, Siddhant Ray, Yuyang Huang, Qizheng Zhang, Kuntai Du, Jiayi Yao, Shan Lu, Ganesh Ananthanarayanan, Michael Maire, Henry Hoffmann, Ari Holtzman, and Junchen Jiang. CacheGen: KV cache compression and streaming for fast large language model serving. In *Proceedings of ACM SIGCOMM*, pages 38–56, 2024.

Mark S. Manasse, Lyle A. McGeoch, and Daniel D. Sleator. Competitive algorithms for server problems. *Journal of Algorithms*, 11:208–230, 1990.

Lyle A. McGeoch and Daniel D. Sleator. A strongly competitive randomized paging algorithm. *Algorithmica*, 6:816–825, 1991.

Kazuo Murota. *Discrete Convex Analysis*. SIAM, 2003.

Kazuo Murota. On infimal convolution of M-convex functions. *RIMS Kôkyûroku*, 1371:20–26, 2004.

Ruoyu Qin, Zheming Li, Weiran He, Jialei Cui, Feng Ren, Mingxing Zhang, Yongwei Wu, Weimin Zheng, and Xinran Xu. Mooncake: Trading more storage for less computation—a KVCache-centric architecture for serving LLM chatbot. In *23rd USENIX Conference on File and Storage Technologies (FAST)*, pages 155–170, 2025.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Lloyd S. Shapley. A value for n-person games. In *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, 1953.

Vikranth Srivatsa, Zijian He, Reyna Abhyankar, Dongming Li, and Yiying Zhang. Preble: Efficient distributed prompt scheduling for LLM serving. In *International Conference on Learning Representations*, 2025.

Fangzhou Wu, Sandeep Silwal, and Qiuyi Zhang. Randomization boosts KV caching, learning balances query load: A joint perspective. In *International Conference on Learning Representations*, 2026.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. CacheBlend: Fast large language model serving for RAG with cached knowledge fusion. In *Proceedings of EuroSys*, pages 94–109, 2025.

Lu Ye, Ze Tao, Yong Huang, and Yang Li. ChunkAttention: Efficient self-attention with prefix-aware KV cache and two-phase partition. In *Proceedings of ACL*, pages 11608–11620, 2024.

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*, volume 37, pages 62557–62583, 2024.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of NAACL-HLT*, pages 5905–5921, 2021.

Code availability statement. The companion package includes selected exact-arithmetic verification scripts and replay checks. The full code will be made public.

AI use statement. Generative AI tools assisted with manuscript drafting, language editing, and polishing; the authors verified the scientific content and final manuscript.

A Additional KV-Cache Systems Context

Modern KV-cache systems exploit shared-prefix structure through PagedAttention, RadixAttention, ChunkAttention, Prompt Cache, Hydragen, CacheBlend, RAGCache, CacheGen, Mooncake, and Preble (Kwon et al., 2023; Zheng et al., 2024; Ye et al., 2024; Gim et al., 2024; Juravsky et al., 2024; Yao et al., 2025; Jin et al., 2025; Liu et al., 2024; Qin et al., 2025; Srivatsa et al., 2025); SCBench evaluates the KV-cache lifecycle (Li et al., 2025). Together, these systems illustrate implementations of shared-prefix dependencies.

B Full Work-Function Proof

We prove Lemma 1 and the offline accounting identities used in the main argument.

Lemma 11 (Lower-bound restriction). *If f is M-convex and $B = \{x : x_i \geq a_i \ \forall i\}$, then $f + \delta_B$ is M-convex whenever its domain is nonempty.*

Proof. Take $x, y \in \operatorname{dom} f \cap B$ and i with $x_i > y_i$. The M-exchange axiom chooses j with $x_j < y_j$ such that

$$f(x) + f(y) \geq f(x - e_i + e_j) + f(y + e_i - e_j).$$

Because $x_i \geq y_i + 1 \geq a_i + 1$, lowering x_i preserves its bound. Because $y_j \geq x_j + 1 \geq a_j + 1$, lowering y_j preserves its bound. All other changed coordinates increase. Both exchanged points remain in B, proving the restricted exchange axiom. □

Full proof of Lemma 1. Put

$$\varphi(z) = \sum_{i \in E} \max\{z_i, 0\} + \delta_{\{z(E)=0\}}(z).$$

On the zero-sum hyperplane, $\varphi(z) = \frac{1}{2}\|z\|_1$. To check M-convexity directly, take $x_i > y_i$ and choose any j with $x_j < y_j$ (one exists because the sums agree). The discrete increment of $g(t) = \max\{t, 0\}$ is nondecreasing, so moving x_i, y_j down and x_j, y_i up by one cannot increase the two-function sum; this is exactly the M-exchange inequality. Hence $w_0(x) = \varphi(x - x_0)$ is M-convex. If w is M-convex, then $f = w + \delta_{F(a)}$ is M-convex by Lemma 11. The update is

$$\widehat{w}^+(x) = (f \,\square\, \varphi)(x) = \inf_{y+z=x} \{f(y) + \varphi(z)\}.$$

Both functions are proper M-convex and nonnegative, so the convolution never takes $-\infty$; Murota's integer convolution theorem (Murota, 2003, Theorem 6.13) therefore applies. Subtracting a constant preserves M-convexity, and all values are integral.

For any x, x' and feasible y, $w(y) + d(y, x) \leq w(y) + d(y, x') + d(x', x)$. Minimize and swap x, x' to obtain $|w^+(x) - w^+(x')| \leq d(x, x')$.

Let $V = \arg\min w$. First, V is M-convex: for $x, y \in V$, the M-exchange inequality produces exchanged points whose two values sum to at most $2 \min w$; neither can be below $\min w$, so both are again minimizers. If $x \notin V$, the M-optimality criterion (Murota, 2003, Theorem 6.26) yields a unit exchange $x' = x - e_i + e_j$ with $w(x') < w(x)$. Integrality and one-Lipschitzness force $w(x') = w(x) - 1$. Repeating reaches a minimizer after exactly $w(x) - \min w$ exchanges, so $d(x, V) \leq w(x) - \min w$. Conversely, one-Lipschitzness gives $w(x) - \min w \leq d(x, v)$ for every $v \in V$. Minimize over v. □

A demand decrease leaves the normalized work function unchanged. Indeed, old minimizers remain feasible and, using the distance cone,

$$\min_{y \in F_{\text{larger}}} \{d(y, V) + d(y, x)\} = d(x, V)$$

by the triangle inequality and the choice $y \in V$.

C Lower-Rank Representation and Greedy Extreme Bases

Lemma 12 (Lower-rank representation). *For every nonempty M-convex $V \subseteq X_{n,k}$, let $r_V(S) = \max_{x \in V} x(S)$ and $p_V(S) = \min_{x \in V} x(S)$. Then r_V is integral submodular, $p_V(S) = k - r_V(E \setminus S)$ is integral supermodular, and*

$$V = \{x \in \mathbb{Z}^E : x(E) = k,\ x(S) \geq p_V(S)\ \forall S \subseteq E\}.$$

Proof. Murota's M-convex/integral-base correspondence (Murota, 2003, Theorem 4.15) gives the upper-rank representation $V = \{x \in \mathbb{Z}^E : x(E) = k,\ x(S) \leq r_V(S)\ \forall S\}$ with integral submodular r_V. Complementing S in the fixed-sum equality $x(E) = k$ gives the stated lower representation and supermodularity of p_V. □

Lemma 13 (Suffix greedy formula). *For a permutation π and suffixes $T_j = E \setminus \{\pi_1, \dots, \pi_j\}$, the vector*

$$z_{\pi_j} := p_V(T_{j-1}) - p_V(T_j)$$

belongs to V and is its lexicographically maximal element in order π. Consequently $z = g_\pi(V)$ and (3) *holds. Moreover, by choosing an order whose suffix chain contains nested sets $A \subseteq S$, one obtains a base simultaneously tight on A and S.*

Proof. Clearly $z(E) = p_V(E) - p_V(\emptyset) = k$. For any $S \subseteq E$, list its elements in the order in which they are deleted from the suffix chain. Supermodularity says that the marginal $p(B \cup \{e\}) - p(B)$ can only increase as the context B grows. Each marginal contributing to $z(S)$ is therefore at least the corresponding marginal when the context is restricted to the later elements of S. Telescoping gives $z(S) \geq p_V(S)$, hence $z \in V$ by Lemma 12. The equality $z(T_j) = p_V(T_j)$ holds for every suffix. Since $x(E) = k$, maximizing the first j coordinates lexicographically is equivalent, successively, to minimizing these suffix sums; the tight suffix equalities therefore make z lexicographically maximal. The final statement follows by placing $E \setminus S$ first, then $S \setminus A$, then A in the deletion order. □

D Full Free-Update Proof

Let $\alpha = \min_{x \in V} x_i$ and $V^+ = \{x \in V : x_i \geq \alpha + 1\} \neq \emptyset$.

Lemma 14 (Exact lower rank). *For all $S \subseteq E$,*

$$p_{V^+}(S) = \begin{cases} p_V(S), & i \notin S, \\ \max\{p_V(S), p_V(S \setminus \{i\}) + \alpha + 1\}, & i \in S. \end{cases}$$

Proof. The right-hand sides are immediate lower bounds. If $i \notin S$, choose $x \in V$ tight on S. If $x_i = \alpha$, choose $y \in V^+$ and apply symmetric exchange to increase x_i, with donor j satisfying $x_j > y_j$. The donor cannot lie in S, otherwise the exchanged point in V would have value $p_V(S) - 1$. Thus the exchanged point lies in V^+ and remains tight on S.

Now write $S = A \cup \{i\}$. By Lemma 13, choose a base x simultaneously tight on $A \subseteq S$, so $x_i = p_V(S) - p_V(A) =: m_i(A)$. Supermodularity implies $m_i(A) \geq m_i(\emptyset) = \alpha$. If $m_i(A) \geq \alpha + 1$, then $x \in V^+$ and is tight on S. If $m_i(A) = \alpha$, exchange with any $y \in V^+$ to increase x_i. The donor cannot lie in A, otherwise the lower bound on A is violated; therefore it lies outside S, and the exchanged base has value $p_V(S) + 1$ on S. This matches $p_V(A) + \alpha + 1$. □

Let $h = p_{V^+} - p_V$. The lemma gives $h \in \{0, 1\}$, $h(S) = 0$ for $i \notin S$, and

$$h(A \cup \{i\}) = 1 \iff p_V(A \cup \{i\}) - p_V(A) = \alpha.$$

The marginal on the right is nondecreasing with A by supermodularity, so its minimum-level family is a down-set.

Full proof of free conservation. Using (3),

$$g_\pi(V^+)_{\pi_j} - g_\pi(V)_{\pi_j} = h(T_{j-1}) - h(T_j).$$

While the suffix contains i, the remaining set $T_j \setminus \{i\}$ shrinks. The down-set property therefore makes h change from zero to one at most once. Deleting i sets h back to zero. Since $h(E) = h(\emptyset) = 0$, the derivative either vanishes or contains one $+1$ and one -1, so movement is zero or one.

Let S_i be the suffix immediately before i is deleted. Movement occurs exactly when $h(S_i) = 1$. For fixed $S \ni i$,

$$\Pr[S_i = S] = \frac{(|S| - 1)!(n - |S|)!}{n!} = \frac{1}{|S| \binom{n}{|S|}}.$$

Thus

$$\mathbb{E}d(g_\pi(V), g_\pi(V^+)) = \sum_{S \ni i} \frac{h(S)}{|S| \binom{n}{|S|}} = -\big(\Phi_E(V^+) - \Phi_E(V)\big).$$

□

E Full Paid-Update Proof

Let $W \subseteq X_{m,K}$ and $W^- = \{w - e_j : w \in W, w_j > 0\}$.

Lemma 15 (Integral truncation)**.** *W^- is exactly the integer base family of the rank $r^-(S) = \min\{r_W(S), K-1\}$.*

Proof. Every $u = w - e_j \in W^-$ satisfies $u(S) \le r_W(S)$ and $u(S) \le K-1$. Conversely, the maximum of $u(S)$ over W^- is $r_W(S)$ when $r_W(S) \le K-1$ (delete a positive unit outside S from a maximizing base), and is $K-1$ when $r_W(S) = K$ (delete inside S). Hence the lower shadow has rank r^-.

It remains to show, by direct augmentation, that no integer base of r^- is missing. Let $u \ge 0$ have $u([m]) = K-1$ and $u(S) \le r_W(S)$ for every $S \subseteq [m]$. If no coordinate could be increased while preserving these inequalities, then for every $j \in [m]$ there would be a set $S_j \ni j$ tight at u: $u(S_j) = r_W(S_j)$. Tight sets are closed under union: modularity of u, submodularity of r_W, and the inequalities $u \le r_W$ force equality on $A \cup B$ whenever A and B are tight. Hence $[m] = \bigcup_{j\in[m]} S_j$ would be tight, contradicting $u([m]) = K-1 < r_W([m]) = K$. Thus some $u+e_j$ is independent; its size is K, so it is a base in W and $u \in W^-$. Finally, $r^-(S) = \min\{r_W(S), K-1\}$ is submodular: if one of two sets has rank at least $K-1$, monotonicity reduces the submodular inequality to the smaller set/intersection inequality, while if both ranks are below $K-1$ it follows directly from submodularity of r_W. Thus W^- is exactly the integral base family of the ordinary rank truncation. □

Because $p_W(S) = K - r_W(\bar{S})$, $p_{W^-}(S) = \max\{p_W(S) - 1, 0\}$. Along any greedy prefix chain, clipping the cumulative upper rank at $K-1$ removes exactly one unit from exactly one greedy coordinate. Reinserting the newly fixed unit at coordinate i proves unit movement.

For the potential, write the old and new minimizer families as $a + W$ and $a + e_i + W^-$. Translation by a cancels. The fixed new unit at i contributes -1 because $\sum_{S\ni i} 1/(|S|\binom{n}{|S|}) = 1$. Also, for nonempty residual A,

$$\frac{1}{|A|\binom{n}{|A|}} + \frac{1}{(|A|+1)\binom{n}{|A|+1}} = \frac{1}{|A|\binom{n-1}{|A|}}.$$

Therefore

$$1 + \Delta\Phi = \sum_{\emptyset \ne S \subseteq [m]} \frac{\mathbf{1}\{p_W(S) > 0\}}{|S|\binom{m}{|S|}}.$$

For any $w \in W$, let $D = \operatorname{supp}(w)$. If $p_W(S) > 0$, then every base intersects S, so $S \cap D \ne \emptyset$. Finally,

$$\begin{aligned}\sum_{S: S\cap D \ne \emptyset} \frac{1}{|S|\binom{m}{|S|}} &= \int_0^1 \frac{1-(1-t)^{|D|}}{t}\, dt \\ &= H_{|D|} \le H_{\min\{K,m\}}.\end{aligned}$$

This completes Lemma 4.

F Potential Boundary and Main Telescoping

For any $x \in V$, $p_V(S) \le x(S)$, so

$$-\Phi_E(V) \le \sum_{S\ne\emptyset} \frac{x(S)}{|S|\binom{n}{|S|}} = \sum_i x_i \sum_{S\ni i} \frac{1}{|S|\binom{n}{|S|}} = k.$$

Lower ranks are nonnegative, giving $\Phi_E(V) \le 0$. For a singleton all inequalities are equalities. Applying Lemma 2, free conservation, and paid charge at every refined update proves Theorem 5, including the boundary term.

G Dependency Retraction and Transfer Details

The proofs of Lemmas 6 and 7 in the main text establish the nonexpansive legality restoration and full-cache simulation. Here we record the remaining metric interface for the chain-prefix reduction.

Let $p = \mathrm{width}(P)$ and fix a Dilworth partition $C_1, \dots, C_p$. Write $L_r = |C_r|$ and $B_L = \{x \in X_{p,k} : 0 \le x_r \le L_r\}$. Every ideal intersects each chain in a prefix, so the count vector $x(I) = (|I \cap C_r|)_r$ of a full ideal belongs to B_L. For a requested ideal Q, let $a_r = |Q \cap C_r|$. Capacity feasibility gives $a([p]) = |Q| \le k$, and trivially $a_r \le L_r$.

For $x \in B_L$, let $U(x)$ be the union of the first x_r objects of each chain. This union is well-defined and has exactly k objects. If $x \ge a$, then $Q \subseteq U(x)$ and hence $Q \subseteq R_k(U(x))$. Furthermore,

$$d_J(U(x), U(y)) = \tfrac{1}{2} \sum_r |x_r - y_r| = d(x, y),$$

and retraction can only reduce this distance. Conversely, every full legal ideal is itself exactly the union of its chain prefixes, so mapping a full offline cache to its count vector preserves transition cost. With Lemma 7, these facts prove the upper-bound transfer used in Theorem 9; Corollary 10 records the initialization additive term.

H Prior Lower-Bound Index: Cover Time and Tight Paging Reduction

The full version of Dallot et al. (2024) states an $H_{\min\{k,\ell\}}$ randomized lower bound for no-bypass dependency caching. For $\ell < k$, its construction consists of $\ell - 1$ isolated objects and one chain of length $k - \ell + 2$, and its phase analysis identifies the distinguished requests with a simple random walk on the complete graph K_ℓ. The proof then uses ℓH_ℓ as a lower bound on the expected time to visit all vertices. The exact cover time instead has harmonic index $\ell - 1$.

Lemma 16 (Cover time of the distinguished walk)**.** *Let $(R_t)_{t \ge 0}$ be the simple random walk on K_ℓ, started from an arbitrary vertex, and let T be the number of transitions until every vertex has been visited. Then*

$$\mathbb{E}[T] = (\ell - 1) H_{\ell - 1}.$$

Equivalently, counting the initial visit, the expected number of walk positions up to cover is $1 + (\ell - 1) H_{\ell - 1}$.

Proof. After exactly s distinct vertices have been seen, with $1 \le s < \ell$, the next step is uniform over the $\ell - 1$ vertices other than the current one. Exactly $\ell - s$ of them are unseen, so the probability that the next transition discovers a new vertex is $(\ell - s)/(\ell - 1)$. The expected number of transitions needed to increase the number of seen vertices from s to $s + 1$ is therefore $(\ell - 1)/(\ell - s)$. Summing over $s = 1, \dots, \ell - 1$ gives

$$\mathbb{E}[T] = \sum_{s=1}^{\ell - 1} \frac{\ell - 1}{\ell - s} = (\ell - 1) \sum_{j=1}^{\ell - 1} \frac{1}{j} = (\ell - 1) H_{\ell - 1}.$$

□

The isometric paging embedding below gives the exact lower bound directly and explains the $\ell - 1$ index.

Proposition 17 (Exact paging embedding for $2 \le \ell \le k$)**.** *Let P consist of isolated objects $u_1, \dots, u_{\ell - 1}$ and a chain $c_1 \prec \cdots \prec c_m$ of length $m = k - \ell + 2$. With cache capacity k, the no-bypass dependency-caching instance on requests to*

$$D = \{u_1, \dots, u_{\ell - 1}, c_m\}$$

is isometric to ordinary paging on ℓ pages with cache size $\ell - 1$ once caches are taken full. Consequently its randomized competitive ratio is at least $H_{\ell - 1}$ against an oblivious adversary.

Proof. The poset has width ℓ. Since $|P| = k + 1$, every legal full cache has size k and omits exactly one object. Downward closure allows precisely ℓ possibilities: one may omit any isolate u_i, or omit the chain maximum c_m; omitting a lower chain element while keeping a successor is illegal. Hence the legal full caches are in bijection with subsets of D of size $\ell - 1$, according to which distinguished object is absent.

A request to u_i requires exactly that distinguished object, while a request to c_m requires the whole chain and is feasible exactly when c_m is present. Thus service feasibility under the bijection is identical to paging. For two full legal caches A, B, the fetch cost of the transition is $|B \setminus A| = d_J(A, B)$; between distinct states in this construction this is the same unit replacement cost as paging. Lemma 7 shows that any at-most-k online trajectory can be lifted to a full trajectory with no larger cost. The same lemma applied to an offline comparator, together with the fact that full trajectories are already feasible in the at-most-k model, shows equality of the two offline optima. The classical randomized paging lower bound on ℓ pages with cache size $\ell - 1$ is $H_{\ell-1}$ (Fiat et al., 1991; McGeoch and Sleator, 1991), proving the claim. □

This embedding is independent of the cover-time calculation and matches the upper bound in Theorem 9. Together with the $k + 1$-element antichain for $\ell \geq k + 1$, it yields the lower-bound side of the exact law $H_{\min\{k,\ell-1\}}$.

I Computational Verification Protocol and Reported Counts

In the companion artifact, run from its root:

```
python code/exhaustive_mconvex_structural_audit.py \
  --output results/mconvex_reproduced.json
python code/randomlex_reachable_graph_audit.py 4,4 4,5 \
  --output results/graphs_reproduced.json
python code/dependency_retraction_audit.py --max-n 5 \
  --output results/retraction_reproduced.json
python code/dependency_parameter_audit.py --max-k 8 \
  --output results/width_family_reproduced.json
python code/exact_adversarial_search.py 3,2,300 4,3,300 \
  --output results/exact_adversarial_search.json
pytest -q code/test_block_simulator.py
python code/replay_audit.py \
  --output results/replay_reproduced.json
```

The companion package includes a trace utility that builds a token-block prefix trie and reports its width. For a rooted prefix tree or forest ordered by ancestry, the leaves form an antichain and the root-to-leaf paths form a chain cover, so the width equals the number of leaves. The same prefix-width computation is used for the trace statistics in Table 4.

J External DCA Interfaces and Hypotheses

The core proof uses three external DCA interfaces. Table 5 lists the exact source labels and checks the hypotheses at each point of use.

Interface	Exact source and required hypotheses	Verification at the point of use
M-convex set $\leftrightarrow$ integral base polyhedron	Murota (2003, Theorem 4.15). Requires a nonempty integer set satisfying the M-exchange axiom.	Every $V = \arg\min w$ is nonempty and M-convex by the direct minimizer exchange argument in Section B; $V \subseteq X_{n,k}$ is finite. Hence Lemma 12 obtains integral upper/lower ranks.
Integer infimal convolution preserves M-convexity	Murota (2003, Theorem 6.13); independently Murota (2004, Theorem 2). Requires both operands to be M-convex and the convolution never to take $-\infty$.	The restricted work function is M-convex by Lemma 11; the distance kernel is proved M-convex directly in Section B. Both are proper and nonnegative on their effective domains, so their convolution never takes $-\infty$.
M-optimality / local-to-global criterion	Murota (2003, Theorem 6.26). Requires a proper M-convex function and a point in its effective domain.	Every normalized work function is integer-valued, proper, and finite on $X_{n,k}$. Thus a nonminimizer has an improving unit exchange; integrality plus one-Lipschitzness makes the decrease exactly one in Section B.

Table 5: Exact external theorem interfaces and hypothesis matching. Greedy suffix tightness and polymatroid augmentation are proved directly in Lemmas 13 and 15.