LARGE-SCALE BUCKET-FEASIBLE LSH-MINHASH DEDUPLICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

LSH-MinHash deduplication in DataTrove keeps one document per connected component of band collisions. Connectivity does not make every pair a near-duplicate: the largest ClueWeb22 component contains 28.8 million documents, of which one is kept. We instead retain at most one document per collision bucket, reusing DataTrove's four stages—signature generation, band matching, clustering, and filtering. Band-sharded Stage 2.5 pruning removes redundant incidences before single-node, memory-bound clustering. A multi-seed extension reruns the same stages under independent signatures with retuned banding, suppressing low-similarity collisions at a chosen high-similarity recall target; processing survivors sequentially satisfies every round's constraints without materializing a joint bucket family. Within a prior strong independent set framework, intrinsic bucket weights connect exact reductions, selection, and optimality certificates, and an observed-history certificate bounds final retention with a sharp information limit. On 1.62B ClueWeb22 and 3.98B HPLT documents, on the same buckets as legacy unionization, bucket-feasible selection retains 32M/34M more documents; four retuned seeds raise the gain to 64M/65M and lower Stage 3 MaxRSS by 48.36%/5.70%. Greedy retention reaches at least 99.43% of the puncturing upper bound in all 20 active seed rounds. Within the 28.8M-document component, the strongest local run retains 1,474,462 documents; exhaustive MinHash comparison of all survivor pairs finds 20 with estimated similarity ≥ 0.75.

1 INTRODUCTION

Deduplication reduces repeated training data, memorization, and wasted compute in language-model curation (Lee et al., 2022; Penedo et al., 2023; Weber et al., 2024). Lee et al. (2022) report reduced memorization and fewer training steps to comparable or better accuracy. RefinedWeb reports improved average zero-shot accuracy after deduplication across C4, OSCAR, and The Pile (Penedo et al., 2023, Table 5).

Nemotron-CC emphasizes the data-quality/quantity tradeoff for long-horizon pretraining (Su et al., 2025), combining global fuzzy and exact-substring deduplication while preserving high-quality tokens. Here we study which documents to retain under collision-bucket constraints.

MinHash and LSH make candidate generation practical at corpus scale (Broder, 1997; Charikar, 2002; Penedo et al., 2026). The subsequent retention rule determines how far local collision evidence propagates through overlapping buckets.

DataTrove forms connected components of cross-band collision links and retains one document per component (Penedo et al., 2026). In the reported ClueWeb22 run, the largest legacy component contains $28{,}812{,}792$ documents: the legacy rule removes $28{,}812{,}791$ and keeps one. Reprocessing that component with bucket-feasible multi-seed selection yields $1{,}474{,}462$ total survivors in the strongest reported local run. Exhaustive MinHash comparison of all survivor pairs finds 20 with estimated character-20-gram similarity at least 0.75, including 11 strictly above it (Section 6.2). For HPLT, the strongest local run retains $297{,}095$ of the $574{,}714$ documents in its largest legacy component, or 51.69% (Table 9).

Component connectivity does not require every pair to be near-duplicates. For example, buckets $\{a, b\}$ and $\{b, c\}$ permit retaining a, c together, whereas unionization retains only one. At the

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.

observed scale, this transitive exclusion makes representative selection a consequential curation decision. We treat two documents as duplicates exactly when they share a band, and seek the largest retained set with no such pair; downstream training effects are outside our scope.

Multi-seed extension through pipeline reuse. We run the same single-seed bucket-feasible pipeline—signature generation, band matching, clustering, and filtering—with independent seeds, reusing its pruning and selection routines without new clustering primitives. Larger per-band signatures suppress low-similarity collisions; additional rounds recover recall at a chosen high-similarity operating point (Section 5). Later rounds process only survivors, so the final output satisfies every seed's bucket constraints without jointly materializing all seed families, although it need not equal joint optimization. Proposition 3 certifies final retention from observed history and establishes its information limit.

Offline deduplication prioritizes retention and memory feasibility over minimizing passes. Stage 3 runs on a single node, where resident memory is the binding resource; Stage 2.5 removes redundant incidences before indexing, and weight-one preclusion shrinks residual selection.

Why the distinction matters theoretically. Every maximal bucket-feasible output matches or exceeds legacy retention and is unchanged by a legacy rerun on the same evidence. Conversely, even two bands with two-document buckets allow an unbounded gap between unionization and the bucket-feasible optimum (Section 2.3).

Block weight connects reduction, selection, and certification. Bucket-feasible retention is maximum strong independent set on the bucket hypergraph. In the framework of Wu & Cong (2026), the intrinsic quantity $w(B) = \min_{v \in B} d(v)$, with $d(v)$ counting incident buckets in the current family, connects three tasks: weight-one roots permit exact preclusion; weight layers organize residual selection; and reciprocal weights yield computable upper certificates. Feasible retention and these bounds together quantify the remaining cardinality gap.

Contributions.

- A complete-bucket interface for DataTrove's four stages, with retention dominance over unionization and an unbounded two-band gap (Section 2).
- Band-sharded Stage 2.5 containment pruning in expected linear time per shard, exploiting within-band disjointness (Section 3).
- A retuned multi-seed extension with sequential feasibility and observed-history certificates (Section 5).
- Evaluation on 1.62B ClueWeb22 and 3.98B HPLT documents: 32M/34M more retained documents on legacy's buckets, 64M/65M with retuned seeds, and a 99.43% minimum active-round bound ratio (Section 6).

2 Pipeline Setting and Retention Objective

2.1 DataTrove's four-stage workflow

DataTrove separates signature generation, band matching, clustering and filtering (Penedo et al., 2026). Legacy Stage 2 writes links between matching documents; Stage 3 uses union-find and retains one document per connected component. Figure 1 shows our replacement: complete buckets, Stage 2.5 pruning, and one-per-bucket selection. Signature generation and band matching retain their roles. Stage 4 filters using original document IDs, requiring the reader ordering and task assignment used during signature generation.

Bucket interface. Our input stores each complete nontrivial equal-band run as a bucket of document IDs; Stage 2.5 preserves this format. Public DataTrove instead serializes duplicate pairs. Treating its chain links as separate two-document constraints would allow multiple representatives per bucket. In the ordinary no-external-index workflow, connected components of consecutive equal-key links *within each band* reconstruct complete runs. Band identity must be preserved; direct

Per-round flow	Artifact and memory placement
$R_{t-1} \to$ Stage 1	Read surviving texts; generate signatures and sorted band files.
$\downarrow$ Stage 2	Group equal keys; write complete bucket rows with original IDs to external storage.
$\downarrow$ Stage 2.5	Load one target shard; stream immutable comparator rows; write pruned bucket files.
$\downarrow$ Stage 3	On a single node, build the pruned incidence index; preclude exact roots, then select on the residual. Resident memory is the bottleneck.
$\downarrow$ Stage 4 $\to R_t$	Stream documents through retained-ID filtering. Feed survivors into Stage 1 with the next seed.

Figure 1: Proposed execution and memory placement. Complete bucket files connect the stages; Stage 2.5 indexes target shards and Stage 3 must fit its incidence state on one node. Later seed rounds consume the preceding survivors.

bucket output avoids reconstruction. The summaries do not identify the experimental revision or serializer.

2.2 Bucket-feasible retention

For a document x with shingle set $S(x)$, a length-$m = br$ MinHash signature has coordinates $m_\ell(x) = \min_{s \in S(x)} \pi_\ell(s)$. Partitioning it into b bands of r rows defines a collision bucket for each nontrivial equal-band run. Under independent ideal minwise permutations, two documents of Jaccard similarity s collide with probability $1 - (1 - s^r)^b$. Signature values and the band-matching criterion are unchanged.

Let $\mathcal{B}_0$ be the emitted duplicate-bucket family and $\mathcal{B}$ the family after equality/subset pruning. Its active universe is $V = \bigcup_{B \in \mathcal{B}} B$; documents outside all duplicate buckets are retained automatically. The inherited retention objective is

$$\alpha(\mathcal{B}) = \max\{|R| : R \subseteq V,\ |R \cap B| \leq 1 \quad \forall B \in \mathcal{B}\}. \tag{1}$$

Bucket feasibility asserts no duplicate equivalence beyond this family.

2.3 Theoretical comparison with unionization

Let $V_1, \ldots, V_m$ be the disjoint document domains of the bucket-overlap components of $\mathcal{B}$, and let $\mathcal{B}_i$ be the buckets in component i. Components remain useful for decomposition:

$$\alpha(\mathcal{B}) = \sum_{i=1}^{m} \alpha(\mathcal{B}_i).$$

Unionization adds the restriction $|R \cap V_i| \leq 1$ for every i, rather than optimizing within each component. The following comparison specializes the component and maximality properties of Wu & Cong (2026) to the DataTrove retention rule.

For a fixed bucket family, let $\mathcal{F}_{\text{bucket}}$ be the feasible region in (1) and let $\mathcal{F}_{\text{union}} = \{R \subseteq V : |R \cap V_i| \leq 1\ \forall i\}$. Since each bucket lies within one component domain,

$$\mathcal{F}_{\text{union}} \subseteq \mathcal{F}_{\text{bucket}}.$$

For pairwise disjoint buckets, the regions coincide and one representative per bucket is optimal. With overlaps, inclusion can be strict, as Proposition 1 shows.

Theorem 1 (Idempotence under legacy rerun and retained-count dominance)**.** *For any bucket-feasible $R \subseteq V$, every induced bucket in $\mathcal{B}|_R = \{B \cap R : B \in \mathcal{B},\ B \cap R \neq \varnothing\}$ has size at most one. Consequently, rerunning legacy connected-component unionization on R with the same bucket evidence leaves R unchanged. If R is also maximal feasible, it contains at least one document in each of the m bucket-overlap components, and therefore*

$$|R| \geq m = |R_{\text{union}}|,$$

where R_{union} keeps one document per component. In particular, the inherited layered greedy root set R_G is idempotent under such a legacy rerun and never retains fewer documents than unionization on the original bucket family.

The proof is given in Appendix B.3.

Theorem 1 also applies to a simple ordered maximal scan; it does not distinguish the layered heuristic from that baseline.

The next construction shows that the inclusion can be strict by an arbitrarily large factor, even under the within-band disjointness required by the pipeline.

Proposition 1 (Unbounded retention gap with two bands)**.** *For every $k \geq 2$, there is a two-band bucket family with bucket size two and maximum document incidence two for which unionization retains one document while $\alpha(\mathcal{B}) = k$. Thus unionization has no positive constant approximation ratio for bucket-feasible cardinality.*

The proof is given in Appendix B.3.

The gap is a limitation of component contraction for this objective, not a hardness result or an approximation guarantee for greedy. The retained-count and rerun comparisons hold for fixed signatures, banding, and bucket evidence; new seeds or changed banding can introduce new conflicts. Documents outside all duplicate buckets are retained by both methods and add equally to their whole-corpus counts. Equality/subset pruning preserves both the feasible region and the component domains, so the comparison also applies across Stage 2.5.

Two-band instances also admit an exact bipartite-matching baseline (Appendix B.4), making the gap above a limitation of unionization even in a tractable case.

3 Bucket-Native Pipeline

3.1 Band-sharded redundancy pruning

Different bands can emit identical or nested buckets, adding storage and processing work without new constraints. Stage 2.5 removes these rows before Stage 3 builds its incidence structure, without discarding documents. For rows $A \subseteq B$, $|R \cap B| \leq 1$ implies $|R \cap A| \leq 1$; equal rows use a deterministic bucket-UID tie-break. The reduction is inherited; implementing it without a global containment search is the systems challenge.

Write $I_0 = \sum_{B \in \mathcal{B}_0} |B|$ and $I = \sum_{B \in \mathcal{B}} |B| \leq I_0$. Pruning removes $I_0 - I$ incidence entries from the subsequent index while preserving the active document union. Its direct memory benefit concerns rows and incidences; document removal occurs in Stage 3 preclusion. Appendix B.1 separates these storage costs from the single-node clustering requirement.

The implementation is nontrivial because exact subset suppression is expensive in a generic bucket family. Given arbitrary sets $B_1, \ldots, B_M$, naive discovery can require quadratically many set-containment tests, while a global inverted index can generate large candidate lists in dense regions. LSH-MinHash buckets have additional structure: each document has exactly one key per band, so distinct nontrivial buckets within a band are disjoint under Stage 2 maximal-run output. Containment witnesses therefore come from other bands. Algorithm 1 indexes only a host-band shard and uses this single-valued document-to-target map to detect cross-band containments by hit counting, without same-band pair searches.

Time and memory. For a shard with σ_A target incidences and σ_C comparator incidences, Algorithm 1 takes expected $O(\sigma_A + \sigma_C)$ time with hash lookup and $O(\sigma_A)$ working memory when comparator IDs are streamed. Each comparator ID hits at most one target, and only touched counters are inspected and reset. Thus large buckets do not induce all-pairs searches or large resident candidate lists. For b bands and total incidence I, one shard per band costs $O(bI)$ overall; finer sharding reduces per-worker memory but repeats comparator I/O (Appendix B.1). This memory/rescan tradeoff suits offline execution: more passes need not be a disadvantage when resident memory is the binding constraint.

Algorithm 1 (Band-sharded Stage 2.5 subset/equality pruning)**.** For each host band h and assigned target-file shard S:

1. Load target buckets A in S. Store each bucket's length, deterministic UID, and local index a. Build the single-valued map $\mathrm{target}(v) = a$ for $v \in A$.
2. Stream comparator buckets B from all other bands. For each B, count hits by looking up each $v \in B$ and record the touched target indices.
3. Inspect only touched targets. If the hit count for A equals $|A|$, then $A \subseteq B$. Mark A removed when $|A| < |B|$; for equal lengths remove only the deterministic UID loser. Reset the touched counters before the next B.
4. Rewrite the host shard without marked rows, preserving all unmarked rows and the binary format.

Hit counts assume that a bucket row lists each document once. Comparator inputs must remain a consistent snapshot during parallel pruning, and equality tie-breaking must be global rather than worker-local. A deleted proper subset remains implied by a maximal superset; the deterministic equality survivor prevents both copies from being removed.

Pruning preserves feasibility but changes incidence counts and therefore the scheduling and certificate quantities. It is applied consistently before the reported bucket-native routines. The summary logs do not establish the unit of their aggregate drop counter, so no removed-row percentage is inferred from it.

3.2 Two-stage clustering: preclusion and layered selection

Stage 3 applies the two-phase solver of Wu & Cong (2026). For the current bucket family, define

$$I(v) = \{i : v \in B_i\}, \qquad d(v) = |I(v)|, \qquad w(B) = \min_{v \in B} d(v).$$

Within a seed round, each document occurs in at most one bucket per band: $d(v) \leq b$, hence $w(B) \leq b$. Thus there are at most b weight layers; block weight is not bucket cardinality. Compress nonempty-support incidence twins first, retaining original-document lists for output.

Phase 1: weight-one preclusion. Retain a degree-one root per weight-one bucket and assign each other member to one incident root. Remove all these buckets' members from the residual, keeping the roots in the output. This is exact for cardinality: each root conflicts only within its own bucket, so replacing a selected member by that root introduces no conflict. An optimum can contain all these roots (Wu & Cong, 2026).

Phase 2: residual layered greedy. After residual equality/subset pruning, recompute degrees and *freeze them for the layered pass*. Process live layers from low to high. At weight w, prefer rows with more remaining degree-w classes, then choose a degree-w root minimizing its distinct equal-degree collision neighborhood. Assign unassigned members of its incident buckets, delete those rows, and update layer state. Every live bucket incident to a lowest-layer root has weight w, so the local index reaches all remaining conflicts. The inherited feasibility and maximality guarantees give Theorem 1; Appendix B.1 gives implementation bookkeeping.

3.3 Source-Rank-Aware Retention

Bucket feasibility leaves a choice of representatives. Quality-oriented recipes such as Nemotron-CC (Su et al., 2025) motivate source preferences. Following Wu & Cong (2026), let $\pi(v)$ rank sources, smaller preferred.

Choose the smallest $(\pi(v), \mathrm{id}(v))$ within each nonempty-support incidence-twin class and among the degree-one roots of a weight-one bucket. For layered selection, after choosing the layer and bucket, minimize $(c_w(v), \pi(v), \mathrm{id}(v))$ among eligible roots; $c_w(v)$ counts distinct unassigned degree-w classes sharing a live bucket. Source priority therefore resolves structural ties without overriding eligibility.

Twin-exchange guarantee. Replacing a selected document by its preferred twin preserves feasibility and cardinality, and never worsens source rank: identical nonempty supports allow at most one class member, and the replacement changes no bucket occupancy. Thus every feasible selection of twin classes admits its preferred representatives.

When logical IDs encode source order, $(c_w(v), \mathrm{id}(v))$ suffices; otherwise store π separately. Preserve original IDs and class-to-root maps for Stage 4 (Appendix B.1). All bucket constraints and maximality guarantees remain in force. Equal-score non-twins are not interchangeable: their tie-break can change later choices and final cardinality. This policy supplies local source preference, not a global quality optimum.

4 From Block Weights to Optimality Certificates

The block weights used for exact preclusion and layered selection also yield the upper certificates of Wu & Cong (2026). For a family $\mathcal{F}$ covering its active documents, the basic value is

$$\alpha(\mathcal{F}) \le U_{\mathrm{cf}}(\mathcal{F}) := \sum_{B \in \mathcal{F}} \frac{1}{w_{\mathcal{F}}(B)}. \tag{2}$$

Certificate weights use degrees intrinsic to the evaluated family $\mathcal{F}$: each retained document distributes one unit over its incident buckets, and feasibility limits a bucket's received charge to $1/w_{\mathcal{F}}(B)$.

Weight-one preclusion fixes roots. Puncturing rebuilds degrees on bucket-deleted families, with twin compression, preclusion, and Layered Superset Sharpening (LSS). LSS applies support dominance within certificate search, not clustering. With q fixed roots, residual nonempty-support family $\mathcal{C}$, and z surviving uncharged isolates, the valid accounting is

$$\alpha(\mathcal{B}) \le q + z + U_{\mathrm{cf}}(\mathcal{C}). \tag{3}$$

The reference contract defers newly exposed weight-one rows to preclusion, preserves isolates individually, and evaluates exact rational values before integer rounding.

Covering evaluates intrinsic weights on subfamilies covering the same live universe. Values need not decrease; retain the minimum valid checkpoint. Appendix B supplies verification rules, a triangle regression, and reference code. Corpus tables report original summary logs, not corpus reruns of that checker.

On the same seed-round instance, $|R| \le \alpha \le U$ brackets the optimum: $U - |R|$ bounds cardinality headroom, and $|R| = U$ proves optimality. The ratio $|R|/U$ requires a matched universe and valid reductions. Corpus totals add documents outside duplicate buckets. These certificates concern retention under observed constraints, not semantic duplicate precision or training quality.

Exact roots versus residual difficulty. If preclusion fixes q roots, its exact reduction gives $\alpha = q + \alpha_{\mathrm{res}}$. The residual certificate fraction is $(G-q)/(U-q)$ when $U > q$, while absolute headroom remains $U - G$. A large exact part can therefore coexist with a smaller residual fraction. Lemma 1 bounds q from intrinsic weights and separates this effect from the choice of residual ordering.

5 Multi-Seed Extension

Multi-seed processing extends the pipeline from selecting representatives under one signature family to allocating collision evidence across independent rounds. A larger number of rows per band suppresses accidental low-similarity collisions; additional seed rounds recover recall at a chosen high-similarity operating point. Each round reuses the signature, bucket, pruning, selection, and filtering stages.

5.1 MERGED EVIDENCE AND SEQUENTIAL FILTERING

Let $\mathcal{B}^{(t)}$ denote the buckets that seed t would emit on the original documents, with parameters (b_t, r_t). Under independent ideal MinHash rows, a fixed pair of similarity s has no-match probability

$$Q_T(s) = \prod_{t=1}^{T} (1 - s^{r_t})^{b_t}, \qquad P_T(s) = 1 - Q_T(s). \tag{4}$$

For constant (b, r), $Q_T(s) = (1 - s^r)^{Tb}$: the candidate evidence is equivalent to Tb independent bands in a merged signature. This probabilistic equivalence does not require storing all rounds together.

One option optimizes over $\bigcup_t \mathcal{B}^{(t)}$. The sequential option used in the study starts from $R_0 = V_0$ and applies round t to R_{t-1}, producing $R_t \subseteq R_{t-1}$. Only survivors are processed again, reducing later-round input size. This is a distinct optimization schedule: it need not return the same representatives or cardinality as one joint solve.

Proposition 2 (Sequential feasibility and unionization dominance)**.** *If each round returns a maximal bucket-feasible subset of its input, retaining documents outside its active buckets, then R_T is feasible for the union of all round families on V_0. It is unchanged by a legacy rerun on that evidence and retains at least one document in every component of the aggregate collision graph.*

Proof. Later rounds only delete documents, so earlier bucket constraints remain satisfied. Every removed document has a bucket witness to a root retained in its removal round. If that root is later removed, follow its witness to a later root. Time strictly increases along this chain, which ends in R_T within the same aggregate component. Thus every component has a survivor. Feasibility gives rerun stability. □

Per-round certificates do not compose. Even two individually optimal rounds can retain only $1/k$ of the joint optimum for arbitrary k (Appendix B.6). An optional reinsertion pass restores global maximality without reducing retention, but needs every discarded document's original-round incidences (Appendix B.7); it is not part of the corpus measurements.

Proposition 3 (Observed-history certification and sharpness)**.** *Suppose each sequential output is feasible for the complete buckets on its input. Let $\mathcal{H}_t$ faithfully record those buckets on R_{t-1} with consistent original IDs, $\mathcal{H} = \bigcup_t \mathcal{H}_t$, and $\mathcal{F} = \bigcup_t \mathcal{B}^{(t)}$. For any valid history upper bound U_{hist} on V_0, including isolates,*

$$|R_T| \le \alpha_{V_0}(\mathcal{F}) \le \alpha_{V_0}(\mathcal{H}) \le U_{\text{hist}}. \tag{5}$$

In the unrestricted abstract band-key model, $\alpha_{V_0}(\mathcal{H})$ is attained by a full-evidence completion consistent with the observations. For every $k \ge 2$, some identical histories admit full optima 1 and k.

Appendix B.6 gives the proof, with private keys establishing sharpness. The latter is not a realizability claim for MinHash on fixed texts. The ratio $|R_T|/U_{\text{hist}}$ needs no missing future hashes; history alone cannot authorize safe reinsertion.

5.2 RETUNING THE COLLISION PROFILE

At fixed (b, r), extra seeds raise collision probability at every similarity and add constraints; they cannot enlarge the joint feasible region. The useful tradeoff therefore retunes r and the total band budget together. For integers $b, r \ge 1$, thresholds $0 < \tau_L < \tau_H < 1$, and a desired miss rate $0 < \delta < 1$, choosing

$$T = \left\lceil \frac{\log \delta}{b \log(1 - \tau_H^r)} \right\rceil \quad \Longrightarrow \quad Q_T(\tau_H) \le \delta, \quad P_T(\tau_L) \le \log(1/\delta)(\tau_L/\tau_H)^r + b\tau_L^r \tag{6}$$

makes the tradeoff explicit. The low-similarity bound follows from $P_T(s) \le Tbs^r$ and $-\log(1 - x) \ge x$. Increasing r suppresses the low-similarity tail at fixed target recall, at the cost of more rounds and signature work.

For the reported settings, $(14, 9) \times 1$ has $P_T(0.5) = 0.0270$ and $P_T(0.8) = 0.867$. Retuning to $(12, 14) \times 4$ changes these to 0.00293 and 0.885, respectively (Figure 4). At $s = 0.75$, however, recall falls from 0.66470 to 0.57809; the curves do not uniformly dominate. The 0.8 comparison is an illustrative operating point, not an independently validated duplicate threshold. The full signature budget increases from 126 to 672 row hashes per document. Sequential processing instead computes $\sum_t b_t r_t |R_{t-1}|$ document–coordinate pairs; actual work also depends on document lengths, bucket processing, and I/O. Appendix B.5 extends the fixed-pair analysis to a surviving-pair bound valid under adaptive selection. Multi-seed gains thus reflect changed evidence as well as selection; the single-seed rows of Section 6 give the same-bucket comparison.

6 Corpus-Scale Evaluation

6.1 Setup and fixed-banding retention

We evaluate preprocessed English ClueWeb22 (1,620M documents, 10.2 TB) and HPLT v2 (3,980M, 18.8 TB) (Overwijk et al., 2022; Burchell et al., 2025). The pipeline uses 16 CPU-only Linux nodes, each with two 32-core x86_64 sockets, 128 logical CPUs and 2.32 TB RAM; Stage 3 runs on one node. All settings use 20-grams, equality/subset pruning, incidence-twin compression and weight-one preclusion. Notation $(b, r) \times T$ denotes T sequential, independently seeded rounds. Table 1 reports Stage 3 MaxRSS, maximized across rounds, with matched corpora, node/task configurations and reporting methods.

Table 1: Full-corpus retention and single-node Stage 3 MaxRSS (maximum across rounds, K as reported). Single-seed rows share legacy's signatures and buckets; multi-seed rows retune evidence.

	ClueWeb22 (1,620M)		HPLT v2 (3,980M)	
Method	Retained (M)	MaxRSS (K)	Retained (M)	MaxRSS (K)
Legacy $(14, 9)$	1,255	182,961,552	2,574	561,285,112
Greedy $(14, 9) \times 1$	1,287	174,594,256	2,608	550,997,016
Greedy $(12, 11) \times 2$	1,300	129,788,924	2,619	537,920,460
Greedy $(10, 12) \times 3$	1,306	100,062,300	2,626	531,991,872
Greedy $(12, 14) \times 4$	1,319	94,481,532	2,639	529,300,888

Legacy and single-seed greedy $(14, 9, 20)$ use the same hash seeds and Stage 1 configuration, and so select from the same buckets. Their 32M/34M gap on ClueWeb/HPLT therefore isolates the retention rule. Four retuned seeds raise the gains to 64M/65M, but also change the bucket evidence.

Single-node memory footprint. Four-seed Stage 3 MaxRSS falls 48.36%/5.70% versus legacy and 45.89%/3.94% versus single-seed greedy (ClueWeb/HPLT). These configuration-level reductions include retuning and survivor filtering; they do not isolate Stage 2.5's effect.

Figures 2 and 3 and Tables 4–5 give the full summaries. Legacy connected components and greedy root-assigned clusters are different partitions; their maximum sizes describe output geometry, not duplicate quality.

6.2 Retention and pairwise signature diagnostics

The corrected ClueWeb three-seed rerun loads all $n = 1{,}474{,}462$ survivors and compares every unordered pair without sampling: $n(n-1)/2 = 1{,}087{,}018{,}357{,}491$ comparisons. The histogram records 4, 7 and 9 pairs at $\Delta = 28, 29, 30$, respectively, and none below 28. With $m = 120$ coordinates, Δ counts disagreements and $\widehat{J} = 1 - \Delta/m$ estimates character-shingle Jaccard similarity. For ideal MinHash, $\Pr[h(A) = h(B)] = J(A, B)$; a finite observed agreement fraction need not equal $J(A, B)$ (Broder, 1997). Exhaustive pair coverage therefore does not establish exact-set separation.

Appendix A.1 specifies shingle units, normalization, empty-input handling and exhaustive traversal. The authors reran all twelve local diagnostics overnight on the cluster with corrected ID alignment (Table 10). HPLT's four-seed result has 41 pairs with $\widehat{J} \geq 0.75$, including 26 strictly above it; Table 2 gives the ClueWeb counts.

Table 2: Strongest local ClueWeb retention and exhaustive signature evaluation. The corrected rerun covers all survivors without sampling. The implementation checks the histogram total, including overflow, against $n(n-1)/2$.

Quantity	Value
Original legacy component	28,812,792
Legacy survivors	1
Final survivors / loaded documents n	1,474,462
Sampling mode	None (all pairs)
Compared unordered pairs $n(n-1)/2$	1,087,018,357,491
Rerun pairs with $\widehat{J} \geq 0.75$ / $\widehat{J} > 0.75$	20 / 11

6.3 Certificate tightness and ordering ablation

Puncturing is the strongest reported bound in all 20 full-corpus seed-round rows: the minimum G_t/U_t^{act} is 99.43% (rounded), with 16 gaps within 0.5% and all within 0.6%. It takes 2–16 iterations on ClueWeb and 2–10 on HPLT. Ratios concern active per-round optima, counting documents in at least one non-singleton bucket and excluding singleton-only documents. Appendix B.6 derives conditional $(12, 14) \times 4$ final whole-corpus ratios of 95.23%/96.95% using nominal initial sizes and including inactive documents.

Matched ordering ablation. With identical per-round inputs and preprocessing, random within-layer ordering reaches a minimum bound ratio of 99.42% (rounded). Secondary ordering improves 19 of 20 counts, ties once, and gains less than 0.01% throughout (Appendix A.2; single-seed counts in Table 14). These dependent comparisons do not estimate random-order variability, expected superiority or runtime.

7 Related Work and Limitations

Foundations include MinHash/LSH (Broder, 1997; Charikar, 2002), scalable near-duplicate filtering (Broder, 2000; Manku et al., 2007; Khan et al., 2024; Penedo et al., 2026), and hypergraph independent sets (Halldórsson & Losievskaja, 2009). The incidence-weight framework, local indexes and source-compatible selection follow Wu & Cong (2026). RabbitTClust compares single-linkage and representative MinHash clustering for genomes (Xu et al., 2023); our target is document retention.

Limitations. (i) *Evidence across configurations.* Single-seed rows share legacy's buckets and isolate the retention rule; retuned multi-seed rows also change the evidence and measure the complete pipeline. Local totals are reconstructed from per-round notes and include inactive documents; historical survivor-ID artifacts are not packaged. (ii) *Duplicate definition.* Documents sharing a band are duplicates by definition; downstream training effects are outside our scope. The corrected diagnostics reuse selection signatures and report nonzero close-pair counts (Appendix A.1), not exact-Jaccard separation. (iii) *Systems costs.* Stage 3 MaxRSS is measured; isolated Stage 2.5 memory savings, temporary storage and elapsed time remain open, and offline passes still incur costs. (iv) *Ablation scope.* The ordering ablation retains layering, whose advantage over unlayered maximal greedy remains unmeasured. These questions are distinct from same-instance cardinality certificates; Appendix B's reference checks do not reproduce historical execution.

8 Conclusion

Bucket-feasible LSH-MinHash deduplication avoids transitive exclusions within DataTrove's four-stage pipeline: from a 28.8M-document legacy component that unionization reduces to one representative, the local ClueWeb study retains $1{,}474{,}462$ documents. Retuned multi-seed rounds lower single-node Stage 3 MaxRSS by 48.36%/5.70% on ClueWeb/HPLT, and block weights connect exact reductions, selection and certificates, with a 99.43% minimum active-round bound ratio. Next steps are to demonstrate the benefit of the additional retained data for LLM pretraining and to characterize how the similarity threshold affects model performance.

AI USE STATEMENT

Generative AI tools assisted manuscript writing, LaTeX formatting, organization, reference formatting, and figure typesetting. They also assisted comparison with related work and technical consistency checks, including feedback on mathematical assumptions, algorithm specification, and interpretation of reported measurements. They also assisted the accompanying reference code and small-instance tests. They did not generate the reported corpus experiments. The authors take responsibility for the final manuscript and its claims.

REPRODUCIBILITY STATEMENT

Section 3 specifies band-sharded pruning and implementation bookkeeping. Section 3.2 summarizes the solver conventions; Wu & Cong (2026) provide the full layered algorithm and its general guarantees. Appendix A records the reported hardware, banding, retention counts, and certificate tables. The accompanying manuscript source package builds this document and includes reference certificate checks, a simple maximal greedy baseline, an exact two-band baseline, and small-instance tests. These are not the production solver or a rerun of the corpus experiments.

REFERENCES

Andrei Z. Broder. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences 1997*, pages 21–29. IEEE, 1997.

Andrei Z. Broder. Identifying and filtering near-duplicate documents. In *Combinatorial Pattern Matching (CPM 2000)*, volume 1848 of LNCS, pages 1–10. Springer, 2000.

Laurie Burchell et al. An expanded massive multilingual dataset for high-performance language technologies (HPLT). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17452–17485. arXiv:2503.10267, 2025. URL https://arxiv.org/abs/2503.10267.

Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 380–388. ACM, 2002.

Magnús M. Halldórsson and Elena Losievskaja. Independent sets in bounded-degree hypergraphs. *Discrete Applied Mathematics*, 157(8):1773–1786, 2009.

John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973. doi: 10.1137/0202019.

Arham Khan, Robert Underwood, Carlo Siebenschuh, Yadu N. Babuji, Aswathy Ajith, Kyle Hippe, Ozan Gökdemir, Alexander Brace, Kyle Chard, and Ian T. Foster. LSHBloom: Memory-efficient, extreme-scale document deduplication. *arXiv preprint arXiv:2411.04257*, 2024.

Katherine Lee et al. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, 2022.

Gurmeet Singh Manku, Arvind Jain, and Anish Das Sarma. Detecting near-duplicates for web crawling. In *Proceedings of the 16th International Conference on World Wide Web*, pages 141–150. ACM, 2007.

Arnold Overwijk, Chenyan Xiong, Xiao Liu, Cameron VandenBerg, and Jamie Callan. ClueWeb22: 10 billion web documents with visual and semantic information. arXiv preprint arXiv:2211.15848, 2022. URL https://arxiv.org/abs/2211.15848.

Guilherme Penedo, Hynek Kydlíček, Alessandro Cappelli, Thomas Wolf, and Mario Sasko. DataTrove: large scale data processing. Software, Zenodo, 2026. URL https://github.com/huggingface/datatrove.

Guilherme Penedo et al. The RefinedWeb dataset for Falcon LLM: Outperforming curated corpora with web data, and web data only. arXiv preprint arXiv:2306.01116, 2023. URL https://arxiv.org/abs/2306.01116.

Dan Su et al. Nemotron-CC: Transforming Common Crawl into a refined long-horizon pretraining dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2459–2475, Vienna, Austria. Association for Computational Linguistics, 2025.

Maurice Weber et al. RedPajama: an Open Dataset for Training Large Language Models. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024 Datasets and Benchmarks Track)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d34497330b1fd6530f7afd86d0df9f76-Abstract-Datasets_and_Benchmarks_Track.html.

Yingquan (Cody) Wu and Jason Cong. Maximum strong independent sets in hypergraphs: Reductions, bounds, and greedy certificates. *arXiv preprint arXiv:2609.17951*, 2026. URL https://arxiv.org/abs/2609.17951.

Xiaoming Xu, Zekun Yin, Lifeng Yan, Hao Zhang, Borui Xu, Yanjie Wei, Beifang Niu, Bertil Schmidt, and Weiguo Liu. RabbitTClust: enabling fast clustering analysis of millions of bacteria genomes with MinHash sketches. *Genome Biology*, 24:121, 2023. doi: 10.1186/s13059-023-02961-6. URL https://doi.org/10.1186/s13059-023-02961-6.

A DATA STUDIES: CLUEWEB AND HPLT

Accounting for local results. For round input size N_{t-1}, active-document count A_t, and selected root count G_t, inactive documents remain, giving

$$N_t = N_{t-1} - A_t + G_t. \tag{7}$$

Table 9 reports total survivors reconstructed from the per-round active-document and selected-root counts in Tables 11–13, using Equation (7). Selected-root counts are the source of record; documents outside the active buckets remain in the output. For example, the first ClueWeb local run has $28{,}812{,}792 - 28{,}763{,}822 = 48{,}970$ inactive documents and $689{,}602$ selected roots, hence $738{,}572$ total survivors. These are accounting-derived totals, not new corpus runs or independently replayed end-of-run counts. The accompanying CSV records the derivation and the superseded summary entries. The signature diagnostic is separate from count reconciliation: a reconstructed survivor count does not verify which signature rows were compared. Updated low-distance entries from the corrected cluster rerun are reported below.

The case studies compare retention under local bucket constraints with connected-component retention on two large corpora. The tables preserve the reported measurements; retuned multi-seed configurations change the bucket evidence as well as the retention stage.

Stage 2.5 uses the band-sharded pass in Section 3.1. The appendix tables report incidence-derived certificates; the measured Stage 3 MaxRSS appears in Table 1. Neither isolates the pruning pass's causal effect. Undefined aggregate drop counters are preserved only in the historical diagnostics.

Incidence-twin compression and weight-1 preclusion from Wu & Cong (2026) are used in the bound and clustering routines. LSS is used only inside puncturing to sharpen residual certificates; the general dominance-closure and conditional-optimality results are not additional components evaluated in these case studies.

Section 6 specifies the corpus sizes and hardware. From the rounded totals, the fixed-banding gains correspond to about 1.98 and 0.85 percentage points of the ClueWeb22 and HPLT inputs, or 2.55% and 1.32% relative to legacy retained counts. These quantify document yield, not independently measured useful-token yield.

Notation (b, r, g) gives bands, rows per band, and n-gram length, with $g = 20$. In $(b, r, 20; t)$, t is the round ordinal, not a documented numeric seed. Later-round inputs depend on earlier outputs. Legacy and single-seed greedy $(14, 9, 20)$ use the same hash seeds and Stage 1 configuration, so they share signatures and buckets; the available notes do not fix an immutable experimental revision. The supplied implementation uses character shingles, case-preserving normalization and no signature for empty shingle sets, as specified in Appendix A.1; its settings should not be replaced by current public DataTrove defaults.

Table 3: All 20 reported full-corpus certificate comparisons. Each $(b, r, 20; t)$ row gives seed round t for both corpora. Ratios use the original underlying counts; iteration counts are those reported for puncturing. The full bound and count tables remain in Appendix A.

	ClueWeb22		HPLT v2	
Parameters	Greedy/U_{punct}	Iterations	Greedy/U_{punct}	Iterations
$(14, 9, 20; 1)$	99.43%	15	99.87%	10
$(12, 11, 20; 1)$	99.43%	14	99.90%	6
$(12, 11, 20; 2)$	99.80%	7	99.92%	3
$(10, 12, 20; 1)$	99.46%	14	99.91%	5
$(10, 12, 20; 2)$	99.83%	7	99.93%	2
$(10, 12, 20; 3)$	99.94%	5	99.97%	2
$(12, 14, 20; 1)$	99.46%	16	99.91%	6
$(12, 14, 20; 2)$	99.81%	8	99.93%	4
$(12, 14, 20; 3)$	99.93%	3	99.97%	2
$(12, 14, 20; 4)$	99.96%	2	99.98%	2

Tables 4 and 5 expand the retention and cluster-size summaries. Their reported maxima do not specify the measurement stage; they are not measurements of Stage 2.5 memory savings. The

Table 4: ClueWeb22 case study (1,620M documents) (Overwijk et al., 2022). The ratio column compares greedy retention to the strongest per-round structural bound, which is U_{punct} in all reported full-corpus rows.

Clustering method	Reported max bucket	Reported max cluster
Legacy/Unionized $(14, 9, 20)$	3,865,885	28,812,792
Greedy $(14, 9, 20)$	3,865,885	2,389,951
Greedy $(12, 11, 20) \times 2$	2,907,278	2,044,416
Greedy $(10, 12, 20) \times 3$	2,183,054	1,681,157
Greedy $(12, 14, 20) \times 4$	2,183,769	1,859,458

Clustering method	Retained (M)	Greedy-to-Bound Ratio (%)
Legacy/Unionized $(14, 9, 20)$	1255 / 1620	–
Greedy $(14, 9, 20)$	1287 / 1620	99.43%
Greedy $(12, 11, 20) \times 2$	1300 / 1620	99.43%, 99.80%
Greedy $(10, 12, 20) \times 3$	1306 / 1620	99.46%, 99.83%, 99.94%
Greedy $(12, 14, 20) \times 4$	1319 / 1620	99.46%, 99.81%, 99.93%, 99.96%

Table 5: HPLT v2 case study (3,980M documents) (Burchell et al., 2025). The ratio column compares greedy retention to the strongest per-round structural bound, which is U_{punct} in all reported full-corpus rows.

Clustering method	Reported max bucket	Reported max cluster
Legacy/Unionized $(14, 9, 20)$	38,980	574,714
Greedy $(14, 9, 20)$	38,980	38,896
Greedy $(12, 11, 20) \times 2$	13,730	10,960
Greedy $(10, 12, 20) \times 3$	12,625	7,867
Greedy $(12, 14, 20) \times 4$	3,783	3,783

Clustering method	Retained (M)	Greedy-to-Bound Ratio (%)
Legacy/Unionized $(14, 9, 20)$	2574 / 3980	–
Greedy $(14, 9, 20)$	2608 / 3980	99.87%
Greedy $(12, 11, 20) \times 2$	2619 / 3980	99.90%, 99.92%
Greedy $(10, 12, 20) \times 3$	2626 / 3980	99.91%, 99.93%, 99.97%
Greedy $(12, 14, 20) \times 4$	2639 / 3980	99.91%, 99.93%, 99.97%, 99.98%

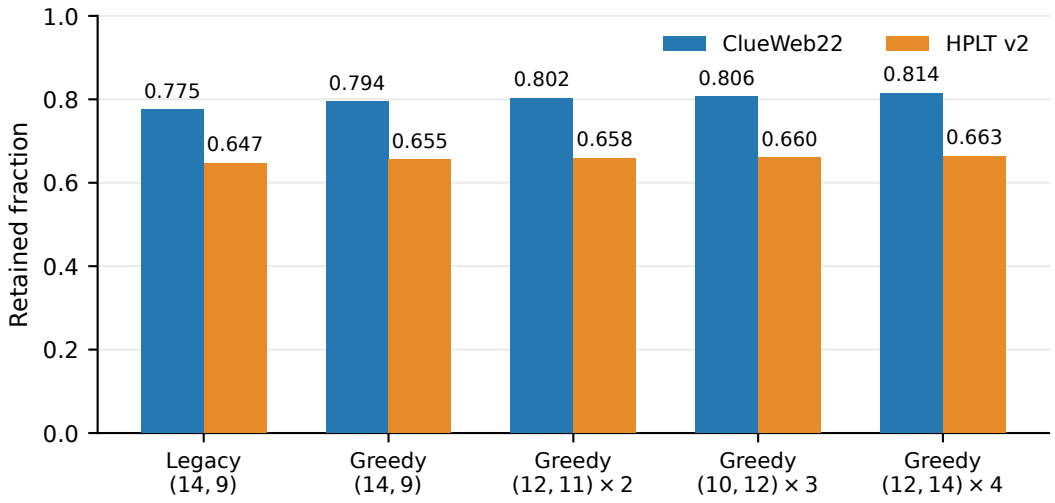


Figure 2: Retained-document fraction across methods for ClueWeb and HPLT.

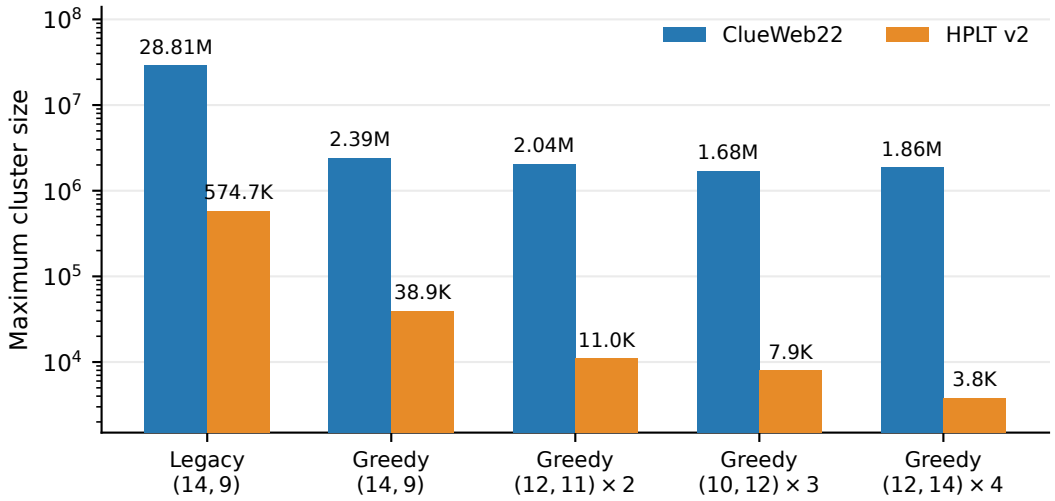


Figure 3: Reported maximum cluster sizes (log scale): legacy connected components and greedy root-assigned clusters describe different partitions. Retuned configurations also change the bucket evidence.

single-seed comparison shares legacy's signatures and buckets; retuned multi-seed rows also change evidence.

The ratio column compares the greedy retained count with the strongest structural bound available for that seed round; in all 20 full-corpus rows, that strongest value is the converged puncturing certificate U_{punct}. In Tables 4 and 5, comma-separated ratios are per-round values in seed order.

Table 1 separately reports matched Stage 3 MaxRSS. End-to-end timing remains unmeasured; memory reductions do not establish a runtime advantage over unionization.

Tables 6 and 7 give the structural certificates behind the ratio column. Puncturing is the strongest reported full-corpus certificate: U_{punct} is strictly smaller than the final covering bound U_{cover} in every row of both tables. It converges in 2–16 rounds on ClueWeb and 2–10 rounds on HPLT. The larger iteration counts reflect a long tail rather than sustained large improvements.

For example, ClueWeb $(10, 12, 20; 1)$ moves from $U_{\mathrm{w1}} = 96{,}580$K to $96{,}472$K after LSS, $95{,}681$K after the first puncturing pass, and $94{,}283$K after 14 passes. Later passes certify smaller residual improvements.

The final greedy outputs are correspondingly close to this reported target. Across all 20 full-corpus rows, the minimum ratio of layered greedy retention to U_{punct} rounds to 99.43%; the reported greedy gap is within 0.5% in 16 rows and within 0.6% in every row. Iterated covering remains a useful independent diagnostic: although it is never the best full-corpus bound in Tables 6–7, its reported converged value stays close to U_{punct}.

The active counts and certificates in Tables 6 and 7 exclude singleton-only documents. The undefined aggregate drop counter does not establish a removed-row count or explain changes between the certificate checkpoints.

Table 6: ClueWeb22 per-round bound comparison. Parentheses in U_{punct} and U_{cover} give the number of iterations used; U_{cover} reports the final iterated-covering value. K-valued columns are rounded; percentages use underlying whole counts. The smallest bound in each row is bold.

Params	Clustered docs (K)	U_{cf} (K)	U_{w1} (K)
$(14, 9, 20; 1)$	445,059	125,549	117,278
$(12, 11, 20; 1)$	380,630	110,622	104,182
$(12, 11, 20; 2)$	66,172	27,210	26,407
$(10, 12, 20; 1)$	348,072	102,163	96,580
$(10, 12, 20; 2)$	63,924	25,754	25,006
$(10, 12, 20; 3)$	35,055	14,473	14,262
$(12, 14, 20; 1)$	324,708	97,634	92,451
$(12, 14, 20; 2)$	57,422	24,075	23,391
$(12, 14, 20; 3)$	31,719	13,418	13,210
$(12, 14, 20; 4)$	22,152	9,326	9,234

Params	U_{punct} (K; iters)	U_{cover} (K; iters)	Greedy (K; % best)
			111,993
$(14, 9, 20; 1)$	**112,639** (15)	114,767 (2)	(99.43%)
			100,398
$(12, 11, 20; 1)$	**100,973** (14)	102,479 (2)	(99.43%)
			26,242
$(12, 11, 20; 2)$	**26,294** (7)	26,353 (2)	(99.80%)
			93,774
$(10, 12, 20; 1)$	**94,283** (14)	95,371 (2)	(99.46%)
			24,873
$(10, 12, 20; 2)$	**24,916** (7)	24,965 (2)	(99.83%)
			14,239
$(10, 12, 20; 3)$	**14,248** (5)	14,258 (2)	(99.94%)
			89,335
$(12, 14, 20; 1)$	**89,822** (16)	91,121 (2)	(99.46%)
			23,254
$(12, 14, 20; 2)$	**23,299** (8)	23,350 (2)	(99.81%)
			13,186
$(12, 14, 20; 3)$	**13,195** (3)	13,205 (2)	(99.93%)
			9,226
$(12, 14, 20; 4)$	**9,229** (2)	9,233 (2)	(99.96%)

Table 8 illustrates changes between reported certificate checkpoints. These decreases are not counts of fixed roots or a causal pruning ablation. The complete 20-row decomposition is preserved in the accompanying CSV, including its original numeric precision; some differences inherit rounded checkpoint values. Within-layer ordering gains are reported separately in Table 15; they do not measure incidence-twin compression's effect.

Table 7: HPLT v2 per-round bound comparison. Parentheses in U_{punct} and U_{cover} give the number of iterations used; U_{cover} reports the final iterated-covering value. K-valued columns are rounded; percentages use underlying whole counts. The smallest bound in each row is bold.

Params	Clustered docs (K)	U_{cf} (K)	U_{w1} (K)
$(14, 9, 20; 1)$	2,143,149	775,677	773,269
$(12, 11, 20; 1)$	2,055,131	749,252	747,841
$(12, 11, 20; 2)$	103,213	51,384	51,257
$(10, 12, 20; 1)$	2,011,604	735,841	734,787
$(10, 12, 20; 2)$	97,063	48,500	48,415
$(10, 12, 20; 3)$	53,820	26,702	26,676
$(12, 14, 20; 1)$	1,985,194	728,199	727,283
$(12, 14, 20; 2)$	84,986	42,681	42,621
$(12, 14, 20; 3)$	46,905	23,415	23,397
$(12, 14, 20; 4)$	32,480	16,146	16,138

Params	U_{punct} (K; iters)	U_{cover} (K; iters)	Greedy (K; % best)
			771,381
$(14, 9, 20; 1)$	**772,416** (10)	773,645 (2)	(99.87%)
			746,589
$(12, 11, 20; 1)$	**747,373** (6)	748,288 (2)	(99.90%)
			51,204
$(12, 11, 20; 2)$	**51,245** (3)	51,286 (2)	(99.92%)
			733,829
$(10, 12, 20; 1)$	**734,471** (5)	735,207 (2)	(99.91%)
			48,374
$(10, 12, 20; 2)$	**48,408** (2)	48,441 (2)	(99.93%)
			26,667
$(10, 12, 20; 3)$	**26,675** (2)	26,682 (2)	(99.97%)
			726,294
$(12, 14, 20; 1)$	**726,959** (6)	727,738 (2)	(99.91%)
			42,585
$(12, 14, 20; 2)$	**42,615** (4)	42,644 (2)	(99.93%)
			23,390
$(12, 14, 20; 3)$	**23,396** (2)	23,403 (2)	(99.97%)
			16,135
$(12, 14, 20; 4)$	**16,138** (2)	16,140 (2)	(99.98%)

Table 8: Illustrative $(14, 9, 20; 1)$ certificate decreases, rounded thousands. LSS is the Layered Superset Sharpening of Wu & Cong (2026). These are bound changes, not numbers of roots committed by exact preclusion.

Corpus	Weight-one	LSS	First puncture	Later punctures
ClueWeb22	8,271	156	1,769	2,714
HPLT v2	2,408	38	165	651

Two local $(12, 11, 20)$ maximum-bucket fields exceed the corresponding full-corpus value without established equality of evidence. Their table entries are marked "not established"; raw values and the undefined drop counters remain in the historical diagnostics, outside the scientific comparisons.

The local reruns apply the proposed pipeline within the largest components produced by legacy unionization. They explore how bucket-feasible retention and retuned seeds change selection inside a component. Their per-round root counts and reconstructed total survivors do not establish whether the original component is a coherent semantic duplicate class.

Table 9: Local rerun summaries. Total survivors are reconstructed from the per-round counts, retaining inactive documents.

ClueWeb: legacy $(14, 9, 20)$ **component**

Re-dedup method	Reported max bucket	Max cluster (1st)	Final survivors
Legacy baseline	3,865,885	28,812,792	1
Greedy $(14, 9, 20)$	3,865,885	2,672,081	738,572
Greedy $(12, 11, 20) \times 2$	*not established*	1,804,265	1,299,912
Greedy $(10, 12, 20) \times 3$	2,183,052	1,681,156	1,474,462
Greedy $(12, 14, 20) \times 4$	2,183,768	1,859,271	1,455,583

Table 9: Local rerun summaries (continued). The same reconstruction qualifications apply.

ClueWeb: legacy $(12, 11, 20)$ **component**

Re-dedup method	Reported max bucket	Max cluster (1st)	Final survivors
Legacy baseline	2,907,278	27,213,166	1
Greedy $(14, 9, 20)$	3,865,731	2,672,072	682,828
Greedy $(12, 11, 20) \times 2$	*not established*	1,804,265	450,154
Greedy $(10, 12, 20) \times 3$	2,182,988	1,681,124	908,141
Greedy $(12, 14, 20) \times 4$	2,183,719	1,859,271	886,228

Table 9: Local rerun summaries (continued). The same reconstruction qualifications apply.

HPLT: legacy $(14, 9, 20)$ **component**

Re-dedup method	Reported max bucket	Max cluster (1st)	Final survivors
Legacy baseline	38,980	574,714	1
Greedy $(14, 9, 20)$	38,980	21,762	41,673
Greedy $(12, 11, 20) \times 2$	10,820	8,329	250,531
Greedy $(10, 12, 20) \times 3$	10,556	6,249	279,584
Greedy $(12, 14, 20) \times 4$	2,710	2,103	297,095

Table 9 gives reconstructed total survivors within the largest legacy components. For the ClueWeb source component of size 28,812,792, the per-round notes yield 738,572 total survivors under the single-signature method and 1,474,462 under the three-signature variant.

Legacy clustering contributes one document from this source component, whereas single-seed selection on the same buckets retains 738,572. The retuned multi-seed totals exceed one million survivors, but this difference combines changed bucket evidence with a different retention rule. Bucket feasibility alone does not establish semantic distinctness or the value of the additional documents for language-model training.

For the HPLT source component of size 574,714, the corresponding reconstructed totals are 41,673 survivors under the single-signature variant and 297,095 under the four-signature variant. The latter retains 51.69% of the source component under the stated inactive-document accounting.

The local summaries use total survivors derived from the per-round root counts and inactive-document accounting. For multi-seed rows, maximum root-assigned cluster size is first-round geometry $(b, r, 20; 1)$, while total survivors is the final count after all listed rounds. The historical maximum-bucket fields do not specify their measurement stage.

A.1 PAIRWISE SIGNATURE EVALUATION: METHOD AND SCOPE

Shingles and normalization. The supplied `minhash.py` implements `get_shingles` with sliding windows over a Python string: the units are characters, with $n = 20$. It first deletes ASCII characters in `string.punctuation`, then strips leading/trailing whitespace and replaces each run matching `\s+` by one space. Case, accents and non-ASCII punctuation are preserved; the lowercasing statement is commented out. Neither the constructed word tokenizer nor `TextNormConfig` is used in this path. Each character window is hashed; the supplied configuration uses 64-bit hashes, with xxHash as the default family. These are code-level settings of the supplied version, not a recovered immutable run manifest.

Empty shingle sets and input scope. After normalization, fewer than 20 characters produce no shingles. Stage 1 writes no signature record for such a document; the analyzer does not assign an empty-set Jaccard value. For these local studies the signature folder is restricted to one original legacy component. The Stage 3 call passes `remove_keys`, and the analyzer excludes those IDs from the folder's signature records. Thus its intended universe is the component's surviving documents that have signatures. Signature-bearing documents outside nontrivial buckets are included; signatureless documents are absent even if retained by filtering. In the reported ClueWeb three-seed rerun, all $1{,}474{,}462$ final survivors were loaded and evaluated; none was omitted for lacking a signature. Coverage concerns this complete survivor set, not all pairs among the original 28.8M documents before removal.

Metric and threshold. `analyze_cluster_signature` reads saved MinHash coordinates, not original shingle sets. With $m = br$ coordinates in the analyzed round, its intended metric is

$$\Delta(x, y) = \sum_{k=1}^{m} \mathbf{1}[h_k(x) \neq h_k(y)], \qquad \widehat{d}_J(x, y) = \Delta(x, y)/m, \qquad \widehat{J} = 1 - \widehat{d}_J.$$

The histogram has exact integer bins $0, \ldots, q$, where $q = \lfloor m/4 \rfloor$, and an overflow bin for $\Delta > q$. Comparison stops early only after entering overflow; low-bin counts are unaffected by this early exit when signatures are correctly assembled. For $m = 120$, bins $0, \ldots, 30$ count $\widehat{J} \geq 0.75$, while bins $0, \ldots, 29$ count $\widehat{J} > 0.75$. This inclusive integer-bin cutoff differs from a strict $d_J < 0.25$ label. The corrected routine labels overflow $\Delta > q$; the inspected earlier version printed an off-by-one label without changing the integer bins. An exact comparison of finite signatures still estimates original-set Jaccard (Broder, 1997). The same round's selection signatures are reused, so the histogram is not an independent held-out similarity measurement.

Exhaustive enumeration and sampling. The routine partitions loaded rows into blocks (default size 32). Disjoint worker ranges cover every outer block index; each outer block visits all later blocks. Within a diagonal block it compares only $i < j$; between different blocks it compares their full Cartesian product. Consequently, each unordered loaded-row pair is visited exactly once. The corrected code requires the histogram sum, including overflow, to equal $N(N-1)/2$ and raises an error on a mismatch. For the ClueWeb three-seed rerun, $N = 1{,}474{,}462$ and this total is $1{,}087{,}018{,}357{,}491$. The supplied kernel matches a direct pair loop in 768 small-instance checks. Above `max_docs_exact`=10,000,000 loaded rows, the default routine instead samples that many documents without replacement (seed 42), compares all sampled pairs and scales the histogram. "Exact" in this parameter refers to pair enumeration, not exact Jaccard. The ClueWeb rerun used all-pairs mode: its complete survivor population is below the cap, so no document or pair sampling occurred. Early exit after entering overflow shortens a pair's coordinate comparison without skipping the pair itself.

Cross-band identity requirement. Each signature row must concatenate coordinates for the same `(read_id,doc_id)` in every band. Stage 1 sorts each band file by its own hash values. The inspected version joined retained records by position, which can suppress close-pair counts despite matching file counts. The corrected loader establishes a fixed ID-to-row map within each read shard, then joins every band to those rows. It rejects duplicate, missing or unexpected retained IDs and logs pair totals and sampling mode. Twelve synthetic regression tests, including 40 band-order permutations, pass using the actual function bodies with a sequential Numba stand-in. This

verifies the loader logic, not corpus execution or parallel performance. The authors subsequently reran the twelve local diagnostics on the cluster and supplied the updated histograms below. This post-selection diagnostic does not change retention counts; it remains a finite-signature evaluation.

Corrected cluster-rerun results. Table 10 reports all twelve updated low-distance entries. Their integers describe signature-difference bins; a suffix such as 120 records the signature length. In the compact notation, `[0-a: 0, c1, c2, ...]` records zero counts through bin a, followed by counts for successive integer bins. Every updated entry covers bins $0, \ldots, q$; overflow is omitted. The first ClueWeb row has zeros through bin 25 and six counts for bins 26–31. For ClueWeb $(10, 12, 20) \times 3$, $4 + 7 + 9 = 20$ pairs have $\widehat{J} \geq 0.75$; the nine at bin 30 are exactly at the estimated cutoff, leaving 11 strictly above it. For HPLT $(12, 14, 20) \times 4$, the corresponding counts are $3 + 6 + 7 + 10 + 15 = 41$ and $3 + 6 + 7 + 10 = 26$. The rerun thus reports nonzero residual close-pair counts. These estimates characterize the compared signatures; exact lexical separation and downstream training value require different evidence.

Table 10: Local signature-distance diagnostics from the corrected ID-aligned cluster rerun. Counts refer to integer MinHash disagreements, not exact set-Jaccard distances.

ClueWeb: legacy $(14, 9, 20)$ **component**

Re-dedup method	Reported signature-difference bins
Legacy baseline	–
Greedy $(14, 9, 20)$	`[0-25: 0, 4, 5, 8, 12, 45, 82]_126`
Greedy $(12, 11, 20) \times 2$	`[0-30: 0, 2, 4, 7]_132`
Greedy $(10, 12, 20) \times 3$	`[0-27: 0, 4, 7, 9]_120`
Greedy $(12, 14, 20) \times 4$	`[0-36: 0, 1, 3, 6, 11, 19, 30]_168`

Table 10: Corrected cluster-rerun signature-distance diagnostics (continued); the same metric and bin conventions apply.

ClueWeb: legacy $(12, 11, 20)$ **component**

Re-dedup method	Reported signature-difference bins
Legacy baseline	–
Greedy $(14, 9, 20)$	`[0-30: 0, 3]`
Greedy $(12, 11, 20) \times 2$	`[0-30: 0, 1, 3, 2]`
Greedy $(10, 12, 20) \times 3$	`[0-25: 0, 1, 2, 4, 9, 10]`
Greedy $(12, 14, 20) \times 4$	`[0-38: 0, 5, 6, 9, 20]`

Table 10: Corrected cluster-rerun signature-distance diagnostics (continued); the same metric and bin conventions apply.

HPLT: legacy $(14, 9, 20)$ **component**

Re-dedup method	Reported signature-difference bins
Legacy baseline	–
Greedy $(14, 9, 20)$	`[0-21: 0, 2, 3, 5, 8, 8, 8, 19, 15, 26, 36]`
Greedy $(12, 11, 20) \times 2$	`[0-30: 0, 4, 7, 10]`
Greedy $(10, 12, 20) \times 3$	`[0-26: 0, 2, 1, 4, 8]`
Greedy $(12, 14, 20) \times 4$	`[0-37: 0, 3, 6, 7, 10, 15]`

The same re-dedup runs also show how tightly the greedy clustering algorithm tracks the strongest available structural bound within a single legacy union cluster. Tables 11, 12, and 13 report the re-dedup max-cluster geometry, the weight-1 bound U_{w1}, the puncturing bound U_{punct}, the final covering bound U_{cover}, the greedy retained count, and the resulting saturation ratio.

The local bound pattern aligns with the full-corpus studies. Puncturing is strongest in every ClueWeb re-dedup row and either strongest or tied in HPLT. The hardest ClueWeb rows reach 98.33% and

97.97% of the best puncturing bound; every later ClueWeb multi-seed residual round is above 98%. On HPLT, every local row is at least 99.94% of the best bound, with several rows rounding to 100.00%.

Table 11: ClueWeb re-dedup subinstances from the legacy $(14, 9, 20)$ source cluster. Parentheses give convergence iterations; the smallest reported structural bound in each row is bold.

Re-dedup method	Docs	Max cluster
Greedy $(14, 9, 20; 1)$	28,763,822	2,672,081
Greedy $(12, 11, 20; 1)$	27,350,821	1,804,265
Greedy $(12, 11, 20; 2)$	1,683,053	8,565
Greedy $(10, 12, 20; 1)$	26,280,411	1,681,156
Greedy $(10, 12, 20; 2)$	2,167,213	11,167
Greedy $(10, 12, 20; 3)$	1,002,968	9,128
Greedy $(12, 14, 20; 1)$	25,937,783	1,859,271
Greedy $(12, 14, 20; 2)$	1,865,797	8,782
Greedy $(12, 14, 20; 3)$	1,082,607	3,210
Greedy $(12, 14, 20; 4)$	790,362	7,260

Re-dedup method	U_{w1}	U_{punct}	U_{cover}	Greedy (% best)
				689,602
Greedy $(14, 9, 20; 1)$	1,023,456	**701,283** (8)	746,749 (2)	(98.33%)
				1,157,326
Greedy $(12, 11, 20; 1)$	1,426,918	**1,173,541** (6)	1,212,066 (2)	(98.62%)
				363,668
Greedy $(12, 11, 20; 2)$	377,247	**365,544** (4)	368,646 (2)	(99.49%)
				1,382,017
Greedy $(10, 12, 20; 1)$	1,588,395	**1,395,763** (7)	1,430,085 (2)	(99.02%)
				487,641
Greedy $(10, 12, 20; 2)$	498,038	**489,003** (4)	490,982 (2)	(99.72%)
				242,604
Greedy $(10, 12, 20; 3)$	244,512	**242,874** (3)	243,228 (2)	(99.89%)
				1,355,439
Greedy $(12, 14, 20; 1)$	1,622,050	**1,371,088** (7)	1,417,159 (2)	(98.86%)
				468,991
Greedy $(12, 14, 20; 2)$	474,301	**469,702** (3)	470,756 (2)	(99.85%)
				295,821
Greedy $(12, 14, 20; 3)$	298,224	**296,179** (3)	296,667 (2)	(99.88%)
				199,089
Greedy $(12, 14, 20; 4)$	199,857	**199,225** (2)	199,386 (2)	(99.93%)

Table 12: ClueWeb re-dedup subinstances from the legacy $(12, 11, 20)$ source cluster. Parentheses give convergence iterations; the smallest reported structural bound in each row is bold.

Re-dedup method	Docs	Max cluster
Greedy $(14, 9, 20; 1)$	27,005,410	2,672,072
Greedy $(12, 11, 20; 1)$	27,213,166	1,804,265
Greedy $(12, 11, 20; 2)$	802,890	3,949
Greedy $(10, 12, 20; 1)$	25,524,067	1,681,124
Greedy $(10, 12, 20; 2)$	1,814,082	9,618
Greedy $(10, 12, 20; 3)$	764,692	8,992
Greedy $(12, 14, 20; 1)$	25,250,147	1,859,271
Greedy $(12, 14, 20; 2)$	1,593,878	7,671
Greedy $(12, 14, 20; 3)$	879,426	3,101
Greedy $(12, 14, 20; 4)$	583,098	5,553

Re-dedup method	U_{w1}	U_{punct}	U_{cover}	Greedy (% best)
				475,072
Greedy $(14, 9, 20; 1)$	791,478	**484,913** (9)	528,931 (2)	(97.97%)
				1,073,528
Greedy $(12, 11, 20; 1)$	1,337,664	**1,088,998** (6)	1,125,828 (2)	(98.58%)
				179,516
Greedy $(12, 11, 20; 2)$	189,980	**181,179** (4)	183,772 (2)	(99.08%)
				1,214,762
Greedy $(10, 12, 20; 1)$	1,417,261	**1,228,011** (6)	1,261,248 (2)	(98.92%)
				398,564
Greedy $(10, 12, 20; 2)$	408,535	**399,910** (5)	402,106 (2)	(99.66%)
				184,490
Greedy $(10, 12, 20; 3)$	186,275	**184,732** (3)	185,075 (2)	(99.87%)
				1,194,612
Greedy $(12, 14, 20; 1)$	1,456,739	**1,209,830** (7)	1,255,304 (2)	(98.74%)
				397,535
Greedy $(12, 14, 20; 2)$	402,647	**398,234** (3)	399,247 (2)	(99.82%)
				238,674
Greedy $(12, 14, 20; 3)$	240,951	**239,002** (3)	239,481 (2)	(99.86%)
				148,790
Greedy $(12, 14, 20; 4)$	149,505	**148,911** (3)	149,055 (2)	(99.92%)

Table 13: HPLT re-dedup subinstance from the legacy $(14, 9, 20)$ source cluster. Parentheses give convergence iterations; the smallest reported structural bound in each row is bold.

Re-dedup method	Docs	Max cluster
Greedy $(14, 9, 20; 1)$	574,714	21,762
Greedy $(12, 11, 20; 1)$	319,997	8,329
Greedy $(12, 11, 20; 2)$	103,978	3,750
Greedy $(10, 12, 20; 1)$	300,156	6,249
Greedy $(10, 12, 20; 2)$	58,231	3,462
Greedy $(10, 12, 20; 3)$	40,547	1,065
Greedy $(12, 14, 20; 1)$	260,330	2,103
Greedy $(12, 14, 20; 2)$	64,098	2,022
Greedy $(12, 14, 20; 3)$	38,788	2,112
Greedy $(12, 14, 20; 4)$	34,721	1,097

Re-dedup method	U_{w1}	U_{punct}	U_{cover}	Greedy (% best)
				41,673
Greedy $(14, 9, 20; 1)$	42,113	**41,699** (2)	41,741 (2)	(99.94%)
				86,092
Greedy $(12, 11, 20; 1)$	86,247	**86,097** (2)	86,120 (2)	(99.99%)
				13,700
Greedy $(12, 11, 20; 2)$	13,707	**13,701** (1)	13,702 (2)	(99.99%)
				87,437
Greedy $(10, 12, 20; 1)$	87,526	**87,440** (1)	87,449 (2)	(100.00%)
				9,746
Greedy $(10, 12, 20; 2)$	9,749	**9,746** (1)	9,747 (1)	(100.00%)
				6,621
Greedy $(10, 12, 20; 3)$	6,622	**6,621** (1)	6,622 (1)	(100.00%)
				96,379
Greedy $(12, 14, 20; 1)$	96,451	**96,382** (1)	96,388 (1)	(100.00%)
				11,142
Greedy $(12, 14, 20; 2)$	11,147	**11,142** (1)	11,144 (1)	(100.00%)
				6,792
Greedy $(12, 14, 20; 3)$	**6,792**	**6,792** (1)	**6,792** (1)	(100.00%)
				6,005
Greedy $(12, 14, 20; 4)$	6,006	**6,005** (1)	6,006 (1)	(100.00%)

Table 14: Matched single-seed $(14, 9, 20; 1)$ active-subinstance bounds and retention. Counts are rounded thousands; ratios use underlying whole counts.

Quantity	ClueWeb22	HPLT v2
Closed-form bound U_{cf}	125,549	775,677
Weight-one bound U_{w1}	117,278	773,269
Puncturing U_{punct}	112,639	772,416
Covering U_{cover}	114,767	773,645
Random-order layered	111,982	771,361
Ordered layered	111,993	771,381
Random-order / U_{punct}	99.42%	99.86%
Ordered / U_{punct}	99.43%	99.87%

A.2 Matched within-layer ordering ablation

We compare the complete ordering of Section 3.2 with random ordering within each weight layer, on identical per-round bucket inputs and preprocessing. Both retain weight-one preclusion and increasing weight layers; the random-order variant omits the secondary bucket priority and collision-score ordering. Table 15 uses paired retained-root counts from the result notes: the historical `old:` field gives the random-order count, and the corresponding `Number of effective clusters` gives the ordered count. The notes' `clustered docs` field counts active input documents, not roots. Both retained counts exclude singleton-only documents.

Secondary ordering increases the count in 19 of 20 recorded pairs and ties in one. Relative gains are below 0.01% throughout. Compared with the same reported U_{punct}, the random-order variant has minimum ratios of 99.42% on ClueWeb22 and 99.86% on HPLT (rounded). Thus most of the reported bound saturation already occurs without secondary ordering; that ordering adds a small increment in these runs. The accompanying CSV preserves exact counts, gains, and both bound ratios. Randomization is within layers; the summaries do not specify its generator or the sampling rule among eligible roots.

Table 15: Matched within-layer ordering ablation. Counts are retained roots on the active subinstance. Relative gain is $100(G_{\mathrm{ord}} - G_{\mathrm{rand}})/G_{\mathrm{rand}}$; $(b, r; t)$ identifies bands, rows per band, and round, with 20-grams throughout.

$(b, r; t)$	Random order	Secondary order	Added roots	Gain (%)
ClueWeb22				
$(14, 9; 1)$	111,982,361	111,993,089	10,728	0.009580
$(12, 11; 1)$	100,388,576	100,397,912	9,336	0.009300
$(12, 11; 2)$	26,241,890	26,242,129	239	0.000911
$(10, 12; 1)$	93,764,649	93,773,880	9,231	0.009845
$(10, 12; 2)$	24,872,749	24,873,017	268	0.001077
$(10, 12; 3)$	14,239,034	14,239,074	40	0.000281
$(12, 14; 1)$	89,326,948	89,335,395	8,447	0.009456
$(12, 14; 2)$	23,253,916	23,254,087	171	0.000735
$(12, 14; 3)$	13,185,553	13,185,591	38	0.000288
$(12, 14; 4)$	9,226,131	9,226,137	6	0.000065
HPLT v2				
$(14, 9; 1)$	771,361,355	771,380,669	19,314	0.002504
$(12, 11; 1)$	746,578,090	746,588,649	10,559	0.001414
$(12, 11; 2)$	51,203,738	51,203,759	21	0.000041
$(10, 12; 1)$	733,821,818	733,828,974	7,156	0.000975
$(10, 12; 2)$	48,373,977	48,373,989	12	0.000025
$(10, 12; 3)$	26,666,976	26,666,977	1	0.000004
$(12, 14; 1)$	726,288,174	726,294,427	6,253	0.000861
$(12, 14; 2)$	42,585,441	42,585,449	8	0.000019
$(12, 14; 3)$	23,389,825	23,389,826	1	0.000004
$(12, 14; 4)$	16,135,240	16,135,240	0	0.000000

These are paired comparisons on each observed round's fixed input, not separate end-to-end multi-seed executions; their gains must not be summed as a final-corpus improvement. The summaries provide no random-order variability or matched timing estimates. This ablation isolates secondary ordering while retaining the layer schedule and reductions; it does not compare against unlayered maximal greedy or assess duplicate quality.

A.3 Exact preclusion and residual headroom

The following diagnostic combines the exact preclusion of Wu & Cong (2026) with elementary block-weight accounting; it is not a new solver theorem. Weight-one roots belong to duplicate buckets and are distinct from the singleton-only documents excluded from active-instance ratios.

Lemma 1 (Exact-preclusion diagnostic)**.** *For a family on its active universe, let M be its number of nonempty rows, q its number of weight-one rows, and $C = \sum_B 1/w(B)$ its intrinsic reciprocal-weight sum. Choose one degree-one root from each weight-one row and delete the union of those rows from the residual universe. Keeping any surviving isolates,*

$$C \le q + \frac{M-q}{2}, \qquad q \ge \max\{0, \lceil 2C - M \rceil\}, \qquad \alpha = q + \alpha_{\mathrm{res}}.$$

If the reported feasible solution includes these roots and has size G, and $U \ge \alpha$, then $G - q$ is a feasible residual count and $U - q$ a valid residual upper bound. Their difference is $U - G$. For $G < U$ and any $0 \le q_L \le q$,

$$\frac{G-q}{U-q} \le \frac{G-q_L}{U-q_L}.$$

Proof. Non-weight-one rows have weight at least two, proving the first two inequalities. The chosen degree-one roots are pairwise compatible and conflict with no document outside the deleted union. A feasible set contains at most q documents in that union, since its intersection with each of the q rows has size at most one. Hence its size is at most $q + \alpha_{\mathrm{res}}$; the roots together with a residual optimum attain equality. Subtracting q gives the accounting identities. Finally $(G-x)/(U-x)$ has derivative $(G-U)/(U-x)^2 < 0$ for $x \le q$. □

The last inequality bounds the *certificate fraction*, not the actual residual approximation ratio: the residual solution may be optimal despite a loose residual certificate. If $G = U$, both problems are certified optimal; an empty residual needs no ratio. For numerical phase attribution, M, C, q, G must refer to the same preclusion-input family, and G must include its fixed roots. A rounded certificate field $\widehat{C}$ with error at most ε only justifies $q_L = \max\{0, \lceil 2(\widehat{C} - \varepsilon) - M \rceil\}$. The available summaries do not independently establish the correspondence between `Number of unique buckets` and `original-form`, so we do not present their inferred phase counts as measurements.

B VERIFICATION CONTRACTS AND LSH-SPECIFIC BASELINES

B.1 IMPLEMENTATION ACCOUNTING

Figure 1 separates external artifacts from resident state in the execution contract. Optional certificate search may retain observed rows across rounds; it is separate from survivor-only selection.

Source priority and document identity. Section 3.3 uses source rank only within the structural eligibility and tie rules. The implementation compresses identical supports by keeping their minimum key, chooses the minimum degree-one key in each weight-one bucket, and compares $(c_w(v), \mathrm{key}(v))$ in layered root selection. Its packed key is

$$\mathrm{key}(v) = 2^{32}\mathrm{read_id}(v) + \mathrm{doc_id}(v), \qquad 0 \le \mathrm{doc_id}(v) < 2^{32}.$$

Numeric key order therefore compares the logical reader rank before the within-reader document index. Assigning contiguous reader-rank ranges to sources in priority order implements the stated preference. Source metadata alone does not change these comparisons. If IDs do not encode that order, the explicit source-rank key in Section 3.3 is needed. Original IDs remain the identifiers used by filtering and output provenance.

Changing physical workers or scheduling waves must preserve logical reader ranks, the input manifest and within-reader enumeration. Source preference then remains independent of worker placement. Across seed rounds, preserve the source-priority policy even if the surviving documents are repartitioned; reconstructed reader ranks must respect that policy, or priorities must be carried explicitly.

The twin-exchange guarantee concerns the original feasible family: $\mathcal{R} = \{R : |R \cap B| \le 1 \text{ for every } B\}$. A feasible set contains at most one member of each nonempty-support twin class. Replace each selected member by the minimum (π, id) member of its class. Every bucket occupancy is unchanged, so the resulting set lies in $\mathcal{R}$, has the same size, and has no worse source rank for each replacement. This proves both cardinality preservation of the quotient and preferred materialization of any selected classes. Empty-support documents are retained individually, not compressed into one representative.

This exchange does not apply to roots that merely have equal collision scores. Source preference also does not override degree eligibility: with buckets $\{a, b\}$ and $\{b, c\}$, preclusion retains a, c even if b has the best source rank. Hence source-aware tie-breaking enforces all original bucket constraints while expressing local preferences; it is not an algorithm for globally maximizing a document-quality objective.

Algorithm 1: per-shard cost. A nonempty target shard has m_A rows and incidence $\sigma_A = \sum_{A \in S} |A|$; σ_C counts streamed comparator incidences. IDs, lengths, UIDs, and counters occupy machine words. Building the map and storing targets costs expected $O(\sigma_A)$ time and space, since nontrivial rows give $m_A \le \sigma_A/2$. For one comparator B, each ID requires one expected constant-time lookup and increments at most one counter. The number of touched counters is at most $\min\{m_A, |B|\}$. Inspecting and resetting only those counters therefore costs $O(|B|)$, including the containment and UID tests. Rewriting surviving target rows costs $O(\sigma_A)$. The total is expected $O(\sigma_A + \sigma_C)$ time, with no term proportional to the number of target/comparator bucket pairs. Clearing all m_A counters per comparator would invalidate this bound.

Target rows, the map, removal flags, counters, and the touched-index list require $O(\sigma_A)$ working memory. Comparator IDs can be consumed incrementally, counting $|B|$ if its length is not supplied, and checking hits at the row boundary. With a fixed I/O buffer, this needs no storage proportional to $|B|$. A reader that materializes each comparator instead uses $O(\sigma_A + K_C)$ memory, where K_C is its largest comparator row. Snapshot files and rewritten output require $O(I)$ external storage, where I is the total input incidence; they need not all reside in RAM. These bounds concern pruning after bucket construction.

Aggregate scans and the sharding tradeoff. Write I_h for incidence in band h, $I = \sum_{h=1}^{b} I_h$, and q_h for its number of nonempty target shards. If each shard scans all other bands, total expected

work and input/output volume in words are

$$O\left(I + \sum_{h=1}^{b} q_h(I - I_h)\right).$$

Target loading and rewriting contribute $O(I)$. One shard per band costs $O(bI)$; q shards per band cost $O((1 + q(b-1))I)$. For n documents, $d(v) \leq b$ and $I = \sum_v d(v) \leq bn$, giving $O(b^2 n)$ with one shard per band: linear in corpus size for fixed b. For merged seed evidence, the degree bound becomes $\sum_t b_t$. Per-worker memory depends on its largest target shard; concurrent workers' memory adds, and shared I/O bandwidth affects elapsed time. Sharding bounds resident index size at the cost of repeated scans. These are algorithmic bounds, not measured corpus speedups. A target row is indivisible in Algorithm 1, so the target-shard budget must accommodate the largest assigned row. Smaller shard files alone do not enforce an arbitrary memory cap for an oversized target.

Memory entering Stage 3. Removing a subset row preserves the document union because a surviving maximal superset still contains its members. It also preserves the conflict graph: every pair in a removed row co-occurs in a surviving superset. Thus Stage 2.5 neither reduces active-document count nor splits a giant component by itself. Its direct benefit is eliminating redundant row records and incidences before the Stage 3 index is built. With n active documents, M surviving rows and I incidences, the incidence representation, document-to-row index and root/row state use $O(n + M + I)$ words without materializing pairwise conflicts. This describes those structures, not a bound on every production allocation. Preclusion subsequently removes committed roots and their assigned documents from the residual search.

Stage 3 must run on a single node in the evaluated implementation; its clustering state cannot be effectively distributed across nodes. Arbitrary document sharding would omit cross-shard conflicts, and giant components prevent component partitioning from resolving the memory bottleneck. Stage 2.5 can parallelize over target shards with immutable comparator snapshots, but this does not distribute the subsequent clustering state. The Stage 3 measurements in Table 1 report $\max_{1 \leq t \leq T} \mathrm{MaxRSS}_{3,t}$ for each sequential T-round configuration, with matched corpus, node/task configuration and reporting method. Raw K values are preserved without unit conversion; the statistic is a maximum across rounds, not a sum of their peaks. The ten supplied values are included in `lsh_minhash_maxrss.json`. The reductions compare complete configurations and do not quantify the isolated pruning effect or incidence reduction $I_0 - I$. Offline processing makes repeated scans and seed passes a useful tradeoff for single-node memory feasibility; time, bandwidth and temporary storage remain operational costs.

Removing assigned documents changes a live row's minimum remaining degree, not the stored degree of a surviving document. The same local document-to-bucket index supports incidence lookup and updates to $n_{\min}(B) = |\{v \in B : d(v) = w\}|$ for layer-w rows. Collision scores and decrements count distinct document classes, not repeated incidences. No degree rebuild or constraint-row reduction is interleaved with this pass; certificate searches rebuild intrinsic degrees separately.

Layered selection cost. The incidence-local accounting follows Wu & Cong (2026). Let J_{res} be total incidence scanned over residual row states, T_{index} the index build/update work, and T_{score} the distinct-collision-score work. With integer-keyed queues, the coarse bound is

$$O(J_{\mathrm{res}} + T_{\mathrm{index}} + T_{\mathrm{score}})$$

apart from hash/sort preprocessing. Repeated row visits and score evaluation remain part of this cost. This accounting motivates implementation measurements but is not an empirical runtime comparison.

A direct score implementation can be quadratic even with linear incidence. Take one band with $B = \{v_1, \ldots, v_m\}$ and $C = \{u_1, \ldots, u_m\}$, and another with $A_i = \{v_i, u_i\}$, $m \geq 3$. All degrees are two, no rows are redundant, and no candidates are twins. Total incidence is $4m$, but scoring every v_i by separately scanning B and A_i reads $m(m+2)$ entries. Caching the common B contribution gives each closed-neighborhood score $m + 1$ in linear work for this snapshot. This identifies a repeated-scan risk, not a measured production cost or a general bound for cached score updates.

The selection output contract uses original document IDs: a retained-ID set, an assignment from each removed active document to a retained root sharing an original bucket, and unchanged inactive

IDs. Retained IDs must be unique, every original row must meet them at most once, and the assignment domains must partition the input. For sequential rounds, assignment links can be followed to final survivors; the resulting witness chains need not make every document directly adjacent to its final root. The reference baseline scans an ordered list, accepts a document iff all its incident rows are unoccupied, and marks those rows occupied. It supplies a simple same-objective control, not a reproduction of the layered solver.

B.2 Replayable retention certificates

The reference implementation `lsh_minhash_reference_checks.py` accepts a fixed document universe, rows with unique original bucket IDs, optional retained IDs, and a reduction ledger. It rejects duplicate IDs, out-of-universe memberships, invalid witnesses, and infeasible retained sets. Feasibility is checked against the original rows. Empty-support documents remain separate vertices throughout; a quotient may not merge them into one class.

A ledger state consists of uncharged surviving documents, surviving original row IDs with their induced memberships, and the number q of fixed roots. The checker reconstructs memberships and intrinsic degrees after every operation; it does not trust submitted degree counts. It permits the following witnessed operations:

1. *Row puncture:* delete a row only while its current weight exceeds one. This relaxes the constraints. Newly exposed weight-one rows wait for a rebuild/preclusion step; their stale original layer does not authorize further puncturing.
2. *Redundant-row removal:* delete a row contained in a distinct surviving witness row, including equality. Original row IDs identify both witnesses.
3. *Support dominance or twins:* delete x only with a surviving $y \neq x$ satisfying $\varnothing \neq I(y) \subseteq I(x)$. Any selected x can be replaced by y, so the cardinality optimum is preserved. Every LSS deletion must supply such a checked witness.
4. *Weight-one preclusion:* choose a current degree-one root, remove all members of its unique row from the residual universe, and increase q by one. The exchange argument of Wu & Cong (2026) justifies this exact cardinality reduction.

Thus the original optimum is bounded by q plus the residual optimum at each checkpoint. If z residual documents are isolated, the checker evaluates $q + z + \sum_B 1/w(B)$ using exact rational arithmetic. Only then may its floor be used as an integer upper bound. Taking the smallest valid checkpoint value preserves validity. This contract is an independently specified reference interface; equivalence to the historical corpus implementation requires its exported ledger.

For the triangle $\{a, b\}, \{b, c\}, \{a, c\}$, the initial value is $3/2$. Deleting $\{a, b\}$ yields value 2 and weight-one rows; their further puncturing is rejected by this contract. If one instead allowed arbitrary row deletion with correct isolate accounting, the values after two and three deletions would be $1 + 1 = 2$ and $3 + 0 = 3$, not 1 and 0. This regression distinguishes an unsafe reading of a schedule from a verified certificate and says nothing about which code produced the tables.

A second verification route accepts nonnegative rational weights y_B on original rows with $\sum_{B \ni v} y_B \geq 1$ for every active document. Then every feasible R satisfies

$$|R| \leq z + \sum_B y_B,$$

where z counts original isolates. This follows by summing the coverage inequalities over retained active documents and using $|R \cap B| \leq 1$. The choice $y_B = 1/w(B)$ is valid because each incident row has $w(B) \leq d(v)$. A sharper combinatorial certificate is not automatically a fractional cover of the same value; it must supply a proved lift or a replayable ledger. For a verified cover and feasible R containing all original isolates, the unrounded slack has the complementary-slackness decomposition

$$z + \sum_B y_B - |R| = \sum_B y_B(1 - |R \cap B|) + \sum_{v \in R_{\text{active}}} \left(\sum_{B \ni v} y_B - 1 \right).$$

Both terms are nonnegative: unused row capacity and overcoverage of selected documents. Adding and subtracting $\sum_B y_B|R \cap B|$ proves the identity. It localizes a cover's gap; an unlifted puncturing value cannot be substituted.

The verifier takes the strongest integer upper bound among all successfully verified routes before checking a submitted top-level claim. It rejects a claim without a supporting ledger or cover.

The included tests check all 128 hypergraph families on three labeled vertices, 987 valid single-step reductions, invalid-witness regressions, the triangle, and empty-support cases. These verify reference-code behavior against exact small-instance optima; they do not validate the corpus certificates.

B.3 Proofs of the unionization comparisons

Proof of Theorem 1.

Proof. Feasibility gives $|B \cap R| \leq 1$ for every bucket, so no induced bucket can link distinct retained documents in a rerun. If a maximal feasible R missed V_i, any document in V_i could be added: buckets in that component meet no retained document, and all other buckets are disjoint from V_i. This contradicts maximality, proving the count inequality. The final assertion uses feasibility and maximality of the layered solver (Wu & Cong, 2026). □

Proof of Proposition 1.

Proof. Use documents $x_1, \ldots, x_k, y_1, \ldots, y_{k-1}$. Band 1 has buckets $B_i = \{x_i, y_i\}$ and band 2 has buckets $E_i = \{y_i, x_{i+1}\}$, for $1 \leq i < k$; the remaining document in each band has a singleton key. Buckets within either band are disjoint. Together they form the path $x_1, y_1, x_2, \ldots, y_{k-1}, x_k$, so unionization keeps one document. All k documents x_i are bucket-feasible. Conversely, the $k-1$ disjoint buckets B_i cover every document except x_k, so a feasible set has size at most k. The retention ratio is therefore exactly $1/k$. □

B.4 Exact optimization with two bands

Proposition 4 (Two-band matching correspondence)**.** *For a fixed two-band bucket instance, bucket-feasible document sets correspond exactly to matchings in a bipartite multigraph. An exact maximum-cardinality retained set can therefore be found in polynomial time.*

Proof. Create one left vertex for each first-band bucket and one right vertex for each second-band bucket. Each document is an edge joining its two bucket vertices. If a document has no nontrivial bucket in a band, use a private endpoint on that side; a globally isolated document has two private endpoints and is always retainable. Edges sharing an endpoint are exactly documents sharing a bucket, so the one-per-bucket constraints are precisely the matching constraints. Parallel edges represent identical two-band supports and cannot both be selected. Maximizing retained documents is thus maximum bipartite matching. □

The correspondence is an application of classical matching (Hopcroft & Karp, 1973), consistent with the degree-two viewpoint of Wu & Cong (2026). It is not a new matching algorithm. The reference code uses simple augmenting paths and returns document IDs; it also supplies an ordered maximal greedy baseline for any number of bands. Tests compare matching with exhaustive optimization on all 225 pairs of partitions of four documents, plus three missing-incidence cases. For corpus analysis, the informative quantities would include residual incidence-degree distributions, effective band counts, component sizes, and the location of the greedy/certificate gap. Minimum bucket weight two is not the required maximum-incidence condition.

B.5 A duplicate-risk implication under adaptive selection

Proposition 5 (Surviving-pair bound under ideal MinHash)**.** *Fix the original document universe and its true shingle similarities. Let M_τ count its unordered pairs with similarity at least τ. Suppose round parameters (b_t, r_t) are fixed in advance and all MinHash rows are independent ideal minwise*

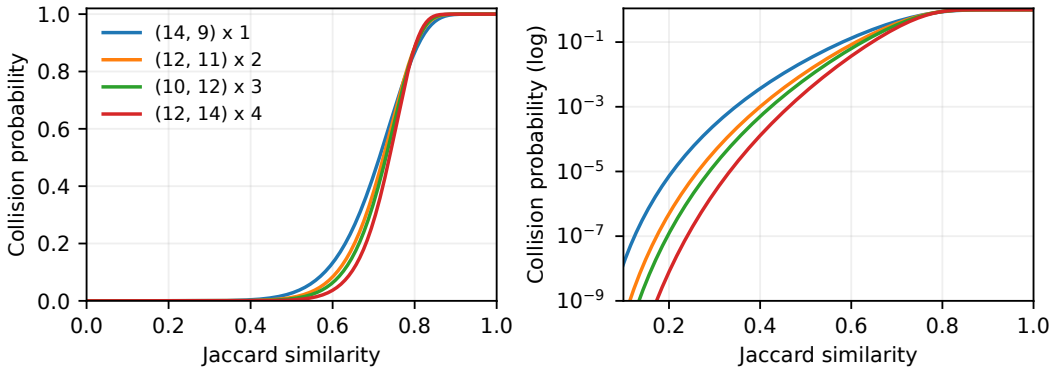


Figure 4: Ideal collision profiles for the evaluated single- and multi-seed configurations: linear scale (left) and low-similarity tail (right). These are analytic probabilities, not corpus precision measurements.

permutations. If R is the final output of any adaptive selection rule feasible for each round on its surviving documents, and $D_\tau(R)$ counts such pairs remaining in R, then

$$\mathbb{E}D_\tau(R) \le \sum_{\{u,v\}:s(u,v)\ge\tau} \prod_t (1 - s(u,v)^{r_t})^{b_t} \le M_\tau \prod_t (1 - \tau^{r_t})^{b_t}.$$

Consequently, $\Pr(D_\tau(R) \ge 1) \le \min\{1, M_\tau \prod_t (1 - \tau^{r_t})^{b_t}\}$.

Proof. A pair retained at the end survived every preceding round. Feasibility implies it had no band collision in any of them. For each fixed original pair, this event has probability $\prod_t (1 - s(u,v)^{r_t})^{b_t}$. Its simultaneous-retention event is a subset of that event, even when selection is adaptive. Sum the indicator expectations over the fixed original pair universe and use $s(u,v) \ge \tau$. Markov's inequality gives the final probability bound. No independence across different document pairs is needed. □

This is an elementary consequence of LSH collision probabilities and event containment, not a new concentration theorem or a measured quality guarantee. M_τ cannot be replaced by a pair count in an adaptively selected output. The bound can be vacuous at corpus scale; nonideal hashes and data-dependent parameter tuning need additional justification. Stage 2.5 preserves the original bucket constraints and therefore preserves the implication used in the proof.

B.6 Final-output certificates from observed history

Keep the original universe V_0 and the complete nontrivial bucket rows actually observed on each input R_{t-1}, using consistent document IDs and row identifiers namespaced by round. Their union is $\mathcal{H}$; documents outside its rows count as individual isolates. Each observed row is the restriction of a full-round bucket to R_{t-1}. Therefore every set feasible for the full family $\mathcal{F}$ is feasible for $\mathcal{H}$. Conversely, two final survivors sharing a full-round row would also share its observed restriction, since both were present in that round's input. Per-round feasibility rules this out, proving the lower side of Proposition 3 without requiring maximality. A replayed ledger or a verified original-row cover on $(V_0, \mathcal{H})$ yields U_{hist} without later hashes for earlier deletions. Completeness of the recorded round rows and faithfulness to execution remain input requirements; checking a supplied history cannot establish them.

A per-round bound U_t on *all* of R_{t-1}, including its isolates, can also be lifted safely as

$$\alpha_{V_0}(\mathcal{H}) \le |V_0 \setminus R_{t-1}| + U_t.$$

Indeed, split any history-feasible set inside and outside R_{t-1}. Thus one may minimize the combined-history certificate, these padded bounds, and $|V_0|$. Raw per-round minima, products of ratios, and sums of fixed roots do not provide this accounting. As observed rows accumulate, $\alpha_{V_0}(\mathcal{H})$ cannot increase; a minimum of valid prefix certificates preserves that monotonicity even if a raw bound worsens. The reference `-history` mode checks nested retained sets, row memberships,

per-round feasibility, and supplied within-band disjointness, then evaluates the combined relaxation and padded bounds.

Conditional calculation from summary counts. Let A_t be the number of documents in at least one non-singleton bucket of round t, G_t its selected active roots, and U_t^{act} a valid upper bound on that active subinstance. The reported Greedy-to-Bound ratios are G_t/U_t^{act}, excluding singleton-only documents from both counts. A singleton key in one band does not make a document inactive if it collides in another. Including round isolates gives $U_t = |R_{t-1}| - A_t + U_t^{\mathrm{act}}$, hence

$$U_{\mathrm{pad},t} = |V_0| - A_t + U_t^{\mathrm{act}}, \qquad |R_T| = |V_0| - \sum_{t=1}^{T}(A_t - G_t). \tag{8}$$

The second identity assumes inactive documents are retained with no additional filtering. Earlier deletions contribute padding to the bound, not retained documents to the numerator.

Using the nominal initial sizes $\widehat{N}_0 = 1{,}620{,}000{,}000$ and $3{,}980{,}000{,}000$, the logged $(12, 14) \times 4$ counts give the following *conditional whole-corpus calculations*. Hats mark values dependent on those nominal sizes. The first-round padded bound is the smallest of the four for each corpus.

Corpus	$\widehat{N}_4$	$\widehat{U}_{\mathrm{pad},1}$	$\widehat{N}_4/\widehat{U}_{\mathrm{pad},1}$
ClueWeb22	1,319,001,086	1,385,114,609	95.23%
HPLT	2,638,839,655	2,721,764,809	96.95%

These are neither new corpus runs nor independently validated certificates. Exact original counts, output accounting, and validity of the same-instance logged bounds are prerequisites for presenting certified corpus ratios. They include retained inactive documents and must not be compared directly with the active-subinstance ratios. The accompanying summary calculator reproduces all eight configurations from the twenty logged round records.

The additive headroom cancels the initial-size term. Since round one minimizes the padded bound here,

$$U_{\mathrm{pad},1} - |R_4| = (U_1^{\mathrm{act}} - G_1) + \sum_{t=2}^{4}(A_t - G_t).$$

The logged counts give $66{,}113{,}523$ and $82{,}925{,}154$ documents for ClueWeb22 and HPLT. Conditional on the same assumptions, these are upper bounds on the final optimum-minus-output gap, not measured losses. They give a concrete baseline for evaluating how much a verified combined-history certificate tightens the conservative padded bound.

Sharp information limit. Consider abstract band-key completions with unrestricted unused keys. Given consistent observed per-band partitions, give every unobserved document a distinct private key in each later band. This completion creates no new nontrivial rows, so its full family has exactly the feasible sets of $\mathcal{H}$. Consequently $\alpha_{V_0}(\mathcal{H})$ is the tightest uniform upper bound on the optimum implied by those observations alone. This is an information limit for abstract keys, not a claim that every completion is realizable by ideal MinHash on fixed known texts.

The limit can be arbitrarily large. Let round one contain the k disjoint pairs $\{x_i, y_i\}$ and retain all y_i. Round two observes one bucket $\{y_1, \ldots, y_k\}$ and retains y_1. Both rounds are optimal on their supplied inputs. If the unseen second-round keys of all x_i are private, the full optimum is k, witnessed by all x_i. If instead all x_i share the observed y key, the second full-round bucket contains all $2k$ documents and the full optimum is 1. The two completions have the same observed history and final output. Reinsertion against $\mathcal{H}$ would add $x_2, \ldots, x_k$, violating the latter completion. Thus history certifies the existing output but does not safely determine which discarded documents to reinsert.

The reference tests exhaust all $49{,}408$ maximal two-round selection sequences on three documents and check the ordering in Equation (5) against exact optima. They also verify the two indistinguishable completions for $k = 2, \ldots, 7$ and reject malformed histories. These are mathematical and reference-code checks; the conditional summary calculation above does not independently validate a historical corpus certificate.

B.7 COMPLETING THE SEQUENTIAL OUTPUT

Let $\mathcal{F} = \bigcup_t \mathcal{B}^{(t)}$ contain every round's buckets on the original universe V_0. Start with the sequential survivors R_T and mark their incident buckets occupied. Scan $V_0 \setminus R_T$ in a fixed order. Reinsert a document precisely when none of its incident buckets is occupied, then mark those buckets occupied. The reference function `complete_feasible` implements this pass.

The output R^+ contains R_T and remains feasible by induction. A rejected document conflicts with a selected document that is never subsequently removed. Thus every unselected document remains blocked, and R^+ is maximal for $\mathcal{F}$. With supplied incidences and a scan order, work is $O(|V_0| + \sum_{B \in \mathcal{F}} |B|)$, without a pairwise expansion. The reference function's default sorts IDs for determinism, adding $O(|V_0| \log |V_0|)$ work. Full-round bucket identities for discarded documents must be retained or recomputed; the signature and I/O cost of obtaining them is additional.

Write $L = \sum_t b_t \geq 1$. The classical maximal-packing charging argument gives $|R^+| \geq \alpha(\mathcal{F})/L$: assign each optimum document outside R^+ to a conflicting selected document and assign common documents to themselves. A selected document outside the optimum has at most L incident buckets, each containing at most one optimum document. A selected document in the optimum, or an isolated document, receives only its own charge. This is a coarse guarantee from bounded incidence, not a new approximation theorem or a substitute for a merged-instance certificate.

In the private-key example of Appendix B.6, a survivor y_j blocks all other y_i and x_j, but every x_i with $i \neq j$ can be reinserted. Completion therefore raises retention from 1 to the joint optimum k. Reversing the two rounds also retains k: the first keeps all x_i and one y_j, and the second removes just one of x_j, y_j. The reference tests verify these values for $k = 2, \ldots, 7$. They also check completion from every feasible initial set on all three-document hypergraph families.

B.8 SCOPE OF REFERENCE VALIDATION

The reference files specify small-instance verification, not a required corpus-scale release. The CSV reconciles all twelve local totals from per-round active/root counts, retaining inactive documents and recording superseded values. This resolves arithmetic, not historical certificate invariants.

Remaining empirical scope. Matched unlayered-selection, joint/sequential/completed-processing and Stage 2.5 memory studies would isolate their respective mechanisms. Such studies would hold original evidence fixed, separate selection from optional certificate search, and report peak memory, I/O and elapsed time. The signature histogram in Section 6.2 is an existing diagnostic reported by the authors; the supplied small-instance tools do not reproduce its corpus histogram or add production timings or training results.