APPROXIMATION-PRESERVING CONSISTENT CLUSTERING WITH OPTIMAL DELETION GUARANTEES

Anonymous authors
Paper under double-blind review

ABSTRACT

Online clustering must adapt as new points arrive. Even a single insertion can trigger many center replacements. Consistent clustering seeks a good solution at every step while limiting center recourse, the total number of new centers introduced over time. For insertion-only (k, z)-clustering, including k-median and k-means, Chan et al. (2025) convert an offline α-approximation into an online algorithm, but the approximation factor becomes $(1 + \varepsilon)\alpha^2$. We improve this factor to $(1 + \varepsilon)\alpha$ for arbitrarily small ε, preserving the quality of the offline solver while retaining controlled recourse. The central difficulty is freeing room for new centers before knowing which old centers a future solution can reuse. Our BUDGETDELETE routine frees capacity within a prescribed cost budget, without an additional clustering oracle, and provides a deletion guarantee that holds for every comparison solution. This guarantee is tight in its budget and geometric separation parameters, up to constants depending on z. Local cost comparisons at multiple distance scales then certify when old centers can replace newly computed ones with little additional cost. On a common weighted-epoch interface, the exponent of $1/\varepsilon$ in the proved recourse bound decreases from 6 to 3 for k-median and from 14 to 5 for k-means. To isolate the cost of accuracy, we consider a single center with points arriving at consecutive integer locations on the real line. Even in this elementary setting, maintaining near-optimal squared loss requires accuracy-dependent recourse, for which we establish matching upper and lower bounds. On three UCI datasets with $k \in \{5, 10, 20\}$, our practical variant uses one shared parameter setting and reduces accumulated smoothed coreset cost by 1.9% to 10.1% and center recourse by 4.0% to 41.0% relative to the released Chan et al. (2025) heuristic, with lower recorded update-loop computation.

1 INTRODUCTION

k-means clustering is a cornerstone of unsupervised learning, organizing large datasets around a small set of representative centers. When data arrive continuously, maintaining this representation as the dataset grows becomes an online clustering problem. Rerunning an offline algorithm after every insertion preserves its quality guarantee, but the resulting centers can be highly sensitive to new input. Even a single new point may cause all k centers to be replaced. These replacements incur additional costs when the centers support persistent prototypes, indices, or cached computations. Consistent clustering addresses this tension by requiring good clustering cost at every step while limiting center recourse, the number of new centers introduced over time (Lattanzi & Vassilvitskii, 2017).

We study the broader (k, z)-clustering objective, which sums distances to the nearest center raised to the power z and includes k-median for $z = 1$ and k-means for $z = 2$. An offline α-approximation keeps this cost within a factor of α of the optimum. Our goal is to preserve this guarantee up to arbitrarily small multiplicative overhead with controlled center recourse.

Chan et al. (2025) give an offline-to-online conversion with controlled recourse, but the approximation factor becomes $(1 + \varepsilon)\alpha^2$. The extra factor of α comes from an additional approximate clustering call used to free capacity for future arrivals. Using a separate constant-factor solver gives

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.

$O(\alpha)$ quality, which still permits a constant multiplicative loss relative to the supplied offline approximation.

The difficulty is deciding which centers to remove. Enough centers must be removed to make room for future arrivals without exhausting the cost budget. Yet the next offline solution, which determines which old centers can be reused, is still unknown when these deletions are chosen.

Our contributions. We obtain $(1+\varepsilon)\alpha$ approximation at every step for arbitrarily small ε, with controlled recourse. The conversion accepts positively weighted points, with ordinary arrivals as the unit-weight case.

To achieve this guarantee, our deletion rule, BUDGETDELETE, frees room for arrivals within a chosen cost allowance. Its guarantee on the number of deletions holds for every later offline solution, avoiding another clustering approximation. This deletion guarantee is tight in its cost budget and geometric separation parameters, up to constants depending on z. When the next offline solution arrives, low recourse also requires reusing old centers. We maintain cost comparisons at multiple distance scales to certify when those centers can replace newly computed ones at little additional cost.

The deletion guarantee bounds the number of new centers needed after reuse in terms of arrivals between offline calls. We then account for the center changes needed to maintain the cost comparisons. On a common weighted-epoch interface (Section 3), the conversion also reduces the power of $1/\varepsilon$ in the recourse bound from 6 to 3 for k-median and from 14 to 5 for k-means.

These conversion guarantees extend to continuous Euclidean k-means under explicit bounds on the data domain, total weight, and initial optimum. We also account for the updates and resets needed to combine the conversion with streaming summaries.

For center updates, matching bounds for squared loss on consecutive integer arrivals show that accuracy-dependent recourse is unavoidable even for a single center on the real line, with all arrivals known in advance.

The same tradeoff between quality and stability guides our practical variant. Direct cost tests defer recomputation. An output stabilization rule reuses reported centers within a cost allowance without increasing cumulative recourse over the internal sequence of solutions. On released coreset streams from three UCI datasets with $k \in \{5, 10, 20\}$, the full method at one shared setting reduces smoothed coreset-cost AUC by 1.9% to 10.1% and recourse by 4.0% to 41.0% relative to the released Chan et al. (2025) heuristic, with lower recorded update-loop computation.

2 RELATED WORK

Center recourse and consistent clustering. Lattanzi & Vassilvitskii (2017) introduced consistent clustering and proved an $\Omega(k \log n)$ recourse barrier for constant approximation. Subsequent work obtained nearly linear dependence on k for constant-approximation metric k-median (Fichtenberger et al., 2021) and deterministic local search with outlier extensions (Guo et al., 2021). Within the phase-and-reuse architecture of Chan et al. (2025), our simultaneous deletion certificate replaces restricted clustering, improving $(1+\varepsilon)\alpha^2$ to $(1+\varepsilon)\alpha$ quality with the same offline solver.

Allowing deletions as well as insertions motivates fully dynamic algorithms that study update time alongside recourse for facility location (Cohen-Addad et al., 2019), k-median (Bhattacharya et al., 2024; 2025a), k-center (Łącki et al., 2024; Forster & Skarlatos, 2025; Bhattacharya et al., 2025b), and Euclidean k-means (Bhattacharya et al., 2026). Łącki et al. (2024) give a deterministic constant approximation for fully dynamic k-center with constant worst-case center recourse. Under resource augmentation, Buchbinder et al. (2025) compare against a minimum-recourse offline trajectory satisfying a quality constraint. Our one-mean experiment uses this benchmark, while our general conversion bounds recourse over positive insertions.

Offline approximation and streaming summaries. Our conversion connects two interfaces. Offline approximation supplies the solution quality, with guarantees from seeding and local search (Arthur & Vassilvitskii, 2007; Choo et al., 2020), primal-dual algorithms (Ahmadian et al., 2017), and dual fitting (Charikar et al., 2026). The theorem specifies a weighted finite-facility ora-

cle, while the practical implementation uses k-means++ seeding followed by weighted Lloyd iterations (Arthur & Vassilvitskii, 2007; Lloyd, 1982).

Streaming summaries supply weighted inputs. Coresets represent clustering costs with small weighted datasets (Har-Peled & Mazumdar, 2004; Chen, 2009; Feldman & Langberg, 2011; Cohen-Addad et al., 2021b) and support memory-efficient streaming algorithms (Ailon et al., 2009; Ackermann et al., 2012). For recourse, their evolution matters alongside their size. Woodruff et al. (2023) construct an online coreset valid for all prefixes whose sampled points are never removed. Our composition accounts for positive records and resets, connecting summary updates to center recourse.

Related objectives and requirements. Other formulations constrain assignments or evaluate a different loss. Moseley et al. (2024) study irrevocable assignments given an upper bound on the final optimum, while Chakraborty et al. (2025) and Gadekar et al. (2026) constrain label changes relative to an existing clustering. Our center-recourse guarantee leaves assignment changes unrestricted. Online facility-location and clustering algorithms can charge loss on arrival (Meyerson, 2001; Liberty et al., 2016), while other formulations bound cumulative loss or regret (Cohen-Addad et al., 2021a; Bhattacharjee et al., 2023). Our guarantee controls the clustering cost of all points seen so far at every prefix.

Privacy can also be combined with consistency. Epasto et al. (2026) study continual private release, and Cohen et al. (2026) combine private semi-coreset streams with consistent clustering. Their guarantees include additive privacy error alongside multiplicative approximation.

3 Model and main results

Let $(\mathcal{X}, d)$ be a metric space and let $\mathcal{F} \subseteq \mathcal{X}$ be a fixed finite set of $f \geq k \geq 2$ feasible centers. Demand locations may lie anywhere in $\mathcal{X}$. Define

$$\sigma = \min_{x \neq y \in \mathcal{F}} d(x, y), \quad D = \max_{x,y \in \mathcal{F}} d(x, y), \quad L = 1 + \log_2(D/\sigma).$$

A weighted dataset P is a multiset of positive records (p, w) with $p \in \mathcal{X}$. For a nonempty center set $C \subseteq \mathcal{F}$,

$$F_P(C) = \sum_{(p,w) \in P} w\, d(p, C)^z, \qquad \mathrm{OPT}(P) = \min_{1 \leq |C| \leq k} F_P(C), \qquad z \geq 1.$$

After initialization and m positive weighted records, the outputs are $C_0, \ldots, C_m$, each of size at most k. With a streaming summary, m counts summary records and n counts all original points, including initialization. Total center-addition recourse is

$$\Gamma = |C_0| + \sum_{t=1}^{m} |C_t \setminus C_{t-1}|\,.$$

Fix a nearest-facility map $\pi_{\mathcal{F}}(p) \in \arg\min_{f \in \mathcal{F}} d(p, f)$. The endpoint oracle is the offline solver used to recompute centers. It returns a nonempty set of at most k facilities of cost at most $\alpha\, \mathrm{OPT}(P)$, with $\alpha \geq 1$. Its output is padded to k distinct facilities before restoration. Oracle correctness is deterministic unless a simultaneous success event is stated. Zero-cost prefixes are served exactly until the first positive-optimum prefix initializes the conversion (Appendix C).

An implicit grid extends the guarantee to continuous Euclidean k-means under the scale assumptions in Corollary G.3. The deletion budget β controls the cost of freeing slots. The reuse budget η controls both substitution with old centers and restoration of their local certificates. The theorem bounds the recourse of the conversion as these budgets shrink and its approximation guarantee approaches that of the offline oracle.

Theorem 3.1 (Two-parameter online conversion). *Fix $0 < \beta < 1$ and $0 < \eta \le 1/20$. Suppose the initial dataset $P^{(0)}$ has positive optimum and m positive weighted records arrive. Put*

$$H = 1 + \log_2 \frac{\mathrm{OPT}(P^{(m)})}{\mathrm{OPT}(P^{(0)})}.$$

There is an online algorithm that, conditional on successful endpoint-oracle calls, maintains at most k centers and, for every prefix t, satisfies

$$F_{P^{(t)}}(C_t) \le (1+\beta)(1+\eta)^2 \alpha \,\mathrm{OPT}(P^{(t)}), \tag{1}$$

with

$$\Gamma = O_z\left(kHL + \frac{\alpha m L}{\beta \eta^{2z}} + mL^2\right). \tag{2}$$

There are at most $m+1$ endpoint-oracle calls, including initialization. The deletion routine makes none. The remaining optimization consists of nearest-facility queries and exact weighted one-center minimization over $\mathcal{F}$.

Corollary 3.2 (Near-oracle point). *For $0 < \varepsilon \le 1$, choosing $\beta = \varepsilon/4$ and $\eta = \varepsilon/20$ gives $(1+\varepsilon)\alpha$ quality and*

$$\Gamma = O_z\left(kHL + \frac{\alpha m L}{\varepsilon^{2z+1}} + mL^2\right).$$

The displayed precision power is 3 for k-median and 5 for k-means.

The three recourse terms account for initialization and growth restarts, fresh centers and their maintenance, and insertion-triggered certificate updates. Section 5 explains this accounting, with proofs in Appendix C and total-time bounds in Appendix H.

Table 1: Guarantees on the common weighted-epoch interface defined below. Both methods use certified endpoint calls. The preceding conversion also uses certified restricted clustering calls. Constants may depend on z.

Conversion	Prefix quality	Epoch recourse
Chan et al. (2025)	$(1+\varepsilon)\alpha^2$	$O_z(k + \alpha m L^2/\varepsilon^{8z-2})$
This work	$(1+\varepsilon)\alpha$	$O_z(kL + \alpha m L/\varepsilon^{2z+1} + mL^2)$

Comparison on a common input interface. Table 1 compares both conversions over the same interval of weighted insertions, called an epoch, with $\mathcal{F} = [\Delta]^d$, $\Delta \ge 2$, and integer $z \ge 1$. The initial dataset is supported on $\mathcal{F}$ and has positive optimum. Each of the next m records introduces a previously unsupported facility location with fixed positive weight, and the optimum grows by a factor of at most two. Then $L = 1 + \log_2(\sqrt{d}(\Delta - 1))$. Lemma C.6 of Chan et al. (2025), with a constant rescaling of its accuracy parameter, gives the first row. Both rows include initialization. When a streaming summary supplies the weighted records, its construction and updates incur the additional costs in Appendix E.

Corollary C.7 optimizes the allocation of a prescribed quality budget to first order.

4 BUDGETED DELETION

Deletion must free slots while keeping reassignment targets available. BUDGETDELETE makes compatible deletions within a cost budget, then certifies their number against every future offline solution.

Let $U \subseteq \mathcal{F}$ consist of k distinct centers with $F_P(U) > 0$, and assign each record to a closest $u \in U$, obtaining P_u. Let $a(u)$ be a nearest center in $U \setminus \{u\}$ and define the reassignment increment

$$q_u = F_{P_u}(\{a(u)\}) - F_{P_u}(\{u\}) \ge 0.$$

The undirected graph with edges $\{u, a(u)\}$ has at most one cycle in each component, so it is three-colorable. Within each color class, sort centers by q_u and take the longest prefix whose total increment is at most $\beta F_P(U)$. We call the rule that returns the largest of these three prefixes BUDGETDELETE. Deleted centers never point to one another, so every reassignment target is retained.

Fix any k-center comparison set V. For $0 < \delta < 1/2$, a pair $(u, v) \in U \times V$ is δ-separated if

$$d(u, v) \le \delta d(u, U \setminus \{u\}), \qquad d(u, v) \le \delta d(v, V \setminus \{v\}).$$

These pairs match centers that are close relative to the other centers in either solution. They form a matching of size $t_\delta(U, V)$, leaving $k - t_\delta(U, V)$ centers unmatched in each solution.

Theorem 4.1 (Deletion certificate). *Let $A_z = 2^{z-1}5^z$. The set $S = \text{BUDGETDELETE}(P, U, \beta)$ satisfies*

$$F_P(U \setminus S) \le (1 + \beta)F_P(U), \tag{3}$$

and, simultaneously for every k-center set V and every $0 < \delta < 1/2$,

$$|S| + 1 \ge \frac{\beta\delta^z}{6A_z[1 + F_P(V)/F_P(U)]}\big(k - t_\delta(U, V)\big). \tag{4}$$

The algorithm uses $O(Nk + k^2)$ distance evaluations on N supported locations and $O(k \log k)$ additional work.

Rearranging (4) bounds unmatched centers through deletion capacity and the cost ratio $F_P(V)/F_P(U)$. This connects deletions to fresh introductions at phase endpoints.

Proof idea. Equation (3) is the cost of the explicit reassignment. For the size bound, let $r_u = d(u, U \setminus \{u\})$ and $s_u = d(u, V)$. A packing argument shows that at least $(k - t_\delta)/2$ centers satisfy $r_u \le 4s_u/\delta$. On these centers, triangle inequality gives aggregate increment

$$\sum q_u \le A_z\delta^{-z}\big(F_P(U) + F_P(V)\big).$$

One color contains at least $(k - t_\delta)/6$ such centers, and its cheapest budget-feasible prefix gives (4), as proved in Appendix A. The set S is chosen before V and δ enter the analysis. This allows the same deletions to be compared with the endpoint solution returned after future arrivals.

Any budget-feasible set with at least $|S|$ deletions inherits this certificate (Appendix A).

5 CONVERSION AND RECOURSE ANALYSIS

A phase starts from a robust snapshot U, which stores centers and their local reuse certificates, called witnesses. BUDGETDELETE frees slots in a separate working set C while preserving the snapshot. Arrivals fill slots with their nearest facilities when absent. The first arrival after C becomes full triggers the endpoint oracle, followed by reuse and witness restoration.

Algorithms 2 and 3 specify the subroutines. Deletion prepares the next phase and is reported at the subsequent insertion.

Multiscale witnesses. Even a small center displacement can greatly increase the relative cost of a tight cluster. Following Fichtenberger et al. (2021); Chan et al. (2025), witnesses store local one-center comparisons on balls whose radii double at each level. Low-cost balls use exact local optimization. On high-cost balls, the existing cost absorbs small displacement. Set

$$\tau = \eta/(100z), \qquad \delta = \tau^2/64 = \Theta_z(\eta^2). \tag{5}$$

For a δ-separated pair, a witness ball at the scale needed for the comparison fits inside the endpoint cluster. If that cluster contains no new record, the relevant local comparisons remain valid and certify the substitution.

Theorem 5.1 (Clean separated substitution). *Let U be a robust snapshot for P, let P' extend P by positive records, and let V be an endpoint solution. If (u, v) is δ-separated for* (5) *and the cluster $P'[V, v]$ assigned to v by V contains no new record, then*

$$F_{P'[V,v]}(\{u\}) \le (1 + \eta)F_{P'[V,v]}(\{v\}).$$

All clean separated pairs may be substituted simultaneously, giving a center set of cost at most $(1 + \eta)F_{P'}(V)$.

Algorithm 1: Our two-parameter online conversion for (k, z)-clustering

$P \leftarrow P^{(0)}$; $V \leftarrow \mathcal{O}_\alpha(P)$, padded to k distinct facilities
$U \leftarrow \textsc{Restore}(P, V)$ with invalid initial witnesses
$C_0 \leftarrow U$; $P_0 \leftarrow P$; $S \leftarrow \textsc{BudgetDelete}(P_0, U, \beta)$; $C \leftarrow U \setminus S$
for $t = 1, \ldots, m$, *when positive record* (p_t, w_t) *arrives* **do**
  $P \leftarrow P \uplus \{(p_t, w_t)\}$
  if $|C| < k$ **then**
    $f_t \leftarrow \pi_{\mathcal{F}}(p_t)$
    if $f_t \notin C$ **then**
      $C \leftarrow C \cup \{f_t\}$
    $C_t \leftarrow C$
  else
    $U \leftarrow \mathrm{TerminalUpdate}(P_0, U, P, \mathcal{O}_\alpha)$
    $C_t \leftarrow U$
    $P_0 \leftarrow P$; $S \leftarrow \textsc{BudgetDelete}(P_0, U, \beta)$; $C \leftarrow U \setminus S$
return $C_0, \ldots, C_m$

At a normal endpoint, reuse old centers in clean separated pairs and introduce unmatched or dirty endpoint centers. Restoration contributes another factor $1 + \eta$. Comparison with the new oracle solution resets these losses at every phase. Deletion then contributes $1 + \beta$, while nearest facilities serve new records at their minimum feasible cost. Restart from the oracle solution if its cost exceeds 2α times the phase-start snapshot cost. Each restart more than doubles the optimum, giving at most $H - 1$ restarts.

Fresh centers. Let P_0 be the phase-start data, U its robust snapshot, and P_+ the data including the terminating record. Let V be the padded endpoint solution, $s = |S|$ the number of deletions, and h the number of records inserted during this phase. At a normal endpoint, the failed growth-restart test and positive insertions give

$$F_{P_0}(V) \le F_{P_+}(V) \le 2\alpha F_{P_0}(U).$$

Deletion frees s slots, each record fills at most one, and the next record after the working set becomes full triggers the endpoint. Hence $s + 1 \le h$, with repeated nearest facilities only extending the phase. Apply the simultaneous certificate of Theorem 4.1 to P_0, U and the subsequently returned V to obtain

$$k - t_\delta(U, V) \le \frac{6A_z(1 + 2\alpha)}{\beta\delta^z}(s + 1) \le \frac{6A_z(1 + 2\alpha)}{\beta\delta^z}\, h.$$

A V-cluster is dirty if, under fixed tie-breaking, it contains a record inserted during this phase. Each new record belongs to one cluster, so at most h clusters are dirty. This counts clusters touched by new records, regardless of how old records are reassigned. Thus the number B_{phase} of fresh introductions before restoration satisfies

$$B_{\mathrm{phase}} \le k - t_\delta(U, V) + h = O_z\left(\frac{\alpha h}{\beta\delta^z}\right) = O_z\left(\frac{\alpha h}{\beta\eta^{2z}}\right), \tag{6}$$

where (5) gives $\delta = \Theta_z(\eta^2)$. Summing over disjoint phase records gives $O_z(\alpha m/(\beta\eta^{2z}))$, plus k per growth restart.

Amortized witness regeneration. A lineage tracks a retained center's versions. Insertions invalidate a witness only within its influence ball, which encloses its construction balls. Live balls containing the same insertion have radii growing more than fourfold from newest to oldest, giving $O(L)$ charges per record. Between data invalidations, a lineage regenerates only as isolation (distance to the nearest other center) grows. Each regeneration more than quadruples the stored radius, allowing $O(L)$ events. With B fresh lineages, the retained-lineage regeneration count satisfies

$$T = O\big((k + B + mL)L\big). \tag{7}$$

Initialization contributes k additions. Opening centers within phases and reintroducing temporarily deleted snapshot centers each contribute at most m. Thus $\Gamma \le k + 2m + B + T$, and (6) and (7) give (2).

6 ACCURACY AND THE COST OF CONSISTENCY

Accuracy limits both deletion capacity and center stability. We first characterize the deletion interface, then isolate the update cost of accuracy on a single-center stream.

Theorem 6.1 (Deletion-interface tightness). *For every $z \geq 1$, $k \geq 4$, $0 < \delta \leq 1/4$, and $0 < \beta < 1$, there is a finite metric instance with unit-weight data P and k-center sets U, V such that $F_P(U) = F_P(V)$ and $t_\delta(U, V) = 0$, but every proper $S \subsetneq U$ satisfying $F_P(U \setminus S) \leq (1+\beta)F_P(U)$ obeys*

$$|S| \leq \frac{\beta\delta^z}{(1-\delta)^z - \delta^z}\, k = O_z(\beta\delta^z k).$$

The construction uses a cycle metric with unit-spaced centers and a demand at distance δ from each. Its original cost is $k\delta^z$, but deleting a center forces its demand to travel at least $1-\delta$. Each deletion consumes $\Omega_z(1)$ of the budget $\beta k\delta^z$. An offset comparison set has the same cost and no separated pairs. This matches Theorem 4.1 up to constants depending on z and the additive one in $|S|+1$.

For center stability, consider the equally spaced stream $x_t = t \in \mathbb{R}$. Its steadily moving mean isolates how accuracy constrains the time between updates.

Theorem 6.2 (One-mean precision barrier). *Let $n \geq 2$ and $0 < \varepsilon \leq 1/3$. On the fixed stream $x_t = t \in \mathbb{R}$, every trajectory of a single center with $(1+\varepsilon)$ optimal squared cost on every prefix has*

$$\Gamma = \Omega\left(\frac{1}{\sqrt{\varepsilon}} \log(1 + n\sqrt{\varepsilon})\right).$$

The bound applies even with advance knowledge of the stream, and holds for every feasible realization of a randomized trajectory.

The quality constraint confines the center to an interval around the prefix mean. This mean advances by $1/2$ per arrival, while the allowed displacement is $O(t\sqrt{\varepsilon})$. A fixed center can therefore serve a limited range of prefixes. Counting these ranges gives the lower bound.

On this stream, LAZYMEAN starts at x_1 and retains its reported mean until its cost exceeds $(1+\varepsilon)$ times the current optimum, when it outputs the current mean.

Corollary 6.3 (Tight precision dependence on the linear stream). *For $x_t = t \in \mathbb{R}$, $n \geq 2$, and $0 < \varepsilon \leq 1/3$,* LAZYMEAN *has*

$$\Gamma_n = O\left(\varepsilon^{-1/2} \log(1 + n\sqrt{\varepsilon})\right).$$

Together with Theorem 6.2, this determines the minimum recourse on this stream up to absolute constants.

If the last refresh was at prefix a, retaining its mean at prefix t adds $t(t-a)^2/4$ to the optimal cost $t(t^2-1)/12$. The allowed excess gives the same spacing scale between updates that drives the lower bound. Appendix D.1 proves the matching rate and gives a dimension-independent guarantee for positive weighted Euclidean streams.

7 EMPIRICAL EVALUATION

We evaluate cost, center recourse, and recorded computation on coreset streams from three UCI datasets, Skin, Shuttle, and Covertype, with $k \in \{5, 10, 20\}$ and one shared parameter setting. Chan et al. (2025) evaluate both an implementation of their theoretical clustering routine and a single-swap heuristic. We compare with their released heuristic, C+25-H. Controlled studies examine the original BUDGETDELETE routine and single-center recourse.

7.1 PRACTICAL IMPLEMENTATION AND EVALUATION SETUP

To reduce update overhead, OURS replaces the deletion and multiscale witness machinery of Theorem 3.1 with direct cost tests. The core calls a clustering solver when its cost crosses a refresh threshold and accepts the candidate only with sufficient cost improvement. The solver uses two

k-means++ initializations followed by weighted Lloyd iterations, with a fixed seed. OURS-H adds *output stabilization*, reusing reported centers within a cost allowance.

After every weighted record, OURS-H retains the previous output if its cost is at most $(1+\xi)^2$ times the internal cost. Otherwise, it tests mixtures of old and new centers within factor $1+\xi$, falling back to the internal solution. A recourse check ensures that cumulative output recourse never exceeds the core's at any prefix, regardless of the solver's approximation guarantee (Appendix F).

Here β, η set the refresh threshold, β also sets the required improvement, and ξ sets the output cost allowance (Appendix I.2). Methods receive the same positive weighted records in one fixed order per dataset. The shared parameters $(\beta, \eta) = (0.01, 0.12)$ and $\xi = \eta/3 = 0.04$ were selected on this benchmark collection.

Ratios to C+25-H measure accumulated smoothed coreset cost (Q_{AUC}), final center additions (R_Γ), and recorded update-loop time (R_T). Weighted-record costs and cumulative recourse are aligned to raw insertion time, retaining initialization and all additions at repeated timestamps. Full-method time adds the core loop and separately timed output stabilization replay on given weighted streams (Appendices I.2 and I.3).

7.2 JOINT IMPROVEMENTS IN COST, RECOURSE, AND COMPUTATION

At the shared setting, OURS-H reduces smoothed coreset-cost AUC by 1.9% to 10.1% and recourse by 4.0% to 41.0% across all nine settings. Its core and output stabilization loops together use 0.024 to 0.653 of recorded baseline time (Figure 1 and Table 2).

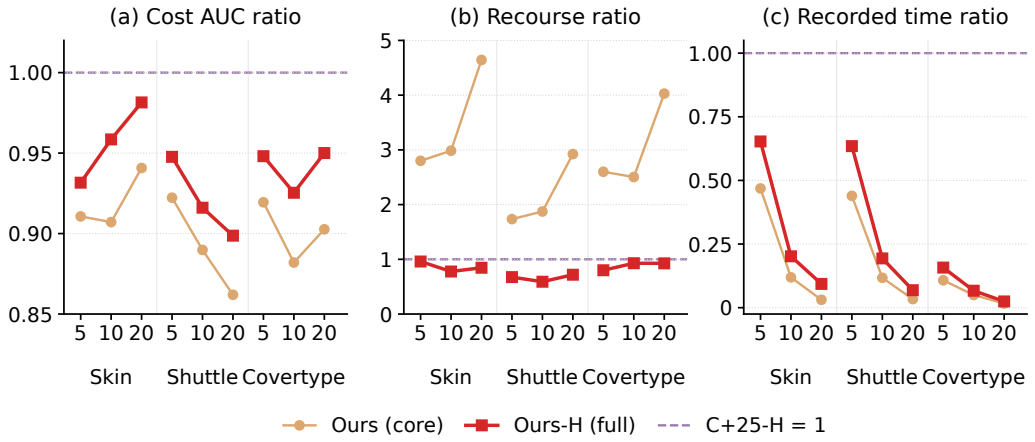

Figure 1: Comparison across the nine settings, with $k = 5, 10, 20$ within each dataset. Ratios are relative to C+25-H and smaller is better. OURS-H adds output stabilization to the same OURS core. Timer scopes appear in Appendix I.3.

Table 2: Equal-weight mean ratios to C+25-H and counts below one across nine settings. Joint requires all three ratios below one. Time uses two-repetition means.

	Mean ratio			Settings better than C+25-H			
Method	Q_{AUC}	R_Γ	R_T	Cost AUC	Recourse	Time	Joint
C+25-H (reference)	1.000	1.000	1.000	—	—	—	—
Ours-H (full)	0.940	0.802	0.233	9/9	9/9	9/9	9/9
Ours (core)	0.904	2.899	0.154	9/9	0/9	9/9	0/9

The core improves cost and recorded time in all nine settings, but its cost-only test allows multiple center replacements. Output stabilization reuses centers on the same internal trajectories, reducing

recourse by 61.1% to 81.8% with a 2.3% to 5.7% increase in cost AUC over the core. Cost and time remain below the baseline, yielding $9/9$ joint improvements versus $0/9$ for the core alone.

7.3 Operating tradeoffs

Figure 2 varies (β, η) over a shared 7×7 grid at $k = 10$, with $\xi = \eta/3$. Moving from lowest cost to lowest recourse on each sampled frontier reduces recourse by 38.6% to 43.6% while raising cost AUC by 4.0% to 6.3%. Here η jointly adjusts refresh and output stabilization.

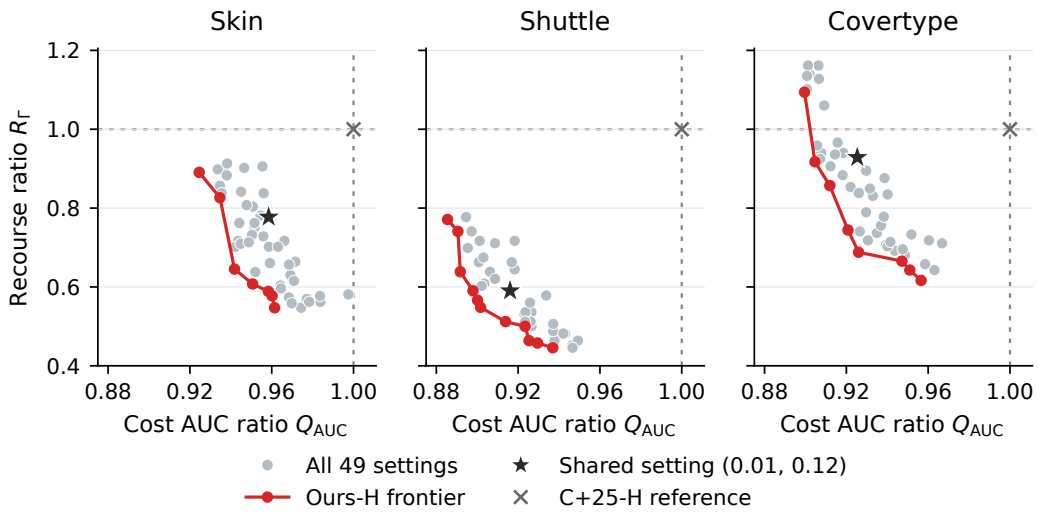

Figure 2: All 147 settings of Ours-H at $k = 10$, with 49 per dataset. Lines connect sampled nondominated points, stars mark the shared parameters $(0.01, 0.12)$, and dotted lines mark C+25-H. Both axes use common-evaluation ratios.

The component comparison in Table 2 isolates output stabilization on a fixed internal trajectory. Varying the budgets reveals the balance between clustering cost and center replacements.

7.4 Controlled studies of the theoretical components

We use fixed snapshots to measure how much deletion capacity original BudgetDelete captures within its quality budget. On 45 snapshots at five budgets, its mean deletion-capacity ratio $s/s^\star$ is 0.685 over 162 positive-capacity settings. Here s is its deletion count and $s^\star$ is the maximum under the same budget, found by exact enumeration. The rule reaches exact capacity on paired-redundant instances but can be conservative on other geometries, including 31 cases with feasible but uncertified deletions. All 225 reported cost and reassignment-budget checks hold (Appendix I.1).

The linear stream $x_t = t$ permits an exact comparison with minimum offline recourse. This comparator knows the entire stream but must satisfy the same quality bound at every prefix. With $n = 100{,}000$, the online LazyMean rule satisfies every prefix constraint using 1.65 to 1.84 times this minimum across 20 accuracies (Figure 7 and Appendix I.4).

8 Conclusion

Budgeted deletion and multiscale reuse preserve offline approximation up to $1 + \varepsilon$ with controlled recourse and tight deletion guarantees. Matching bounds on a linear stream show that maintaining near-optimal squared loss requires accuracy-dependent recourse even for a single center on the real line. Our practical variant combines cost tests with output stabilization that never exceeds its internal trajectory's cumulative recourse. At the shared setting on the evaluated UCI streams, it jointly improves smoothed coreset-cost AUC, recourse, and recorded update-loop computation over the released C+25-H.

AI USE STATEMENT

Generative AI tools assisted with manuscript organization, language editing, LaTeX preparation, preliminary proof checking, experimental analysis and presentation, and preparation of reproducibility scripts. The authors developed the theoretical results and are responsible for the mathematical claims, references, experimental evidence, and final text.

REPRODUCIBILITY STATEMENT

The supplementary package contains the implementation, released baseline, weighted inputs, archived measurements, and reconstruction scripts. Its README gives separate instructions for reconstructing figures and tables from recorded results, rerunning the algorithms on the supplied weighted streams, and replaying output stabilization on generated trajectories. Complete proofs appear in the appendices, and Appendix I specifies the experimental protocols, metrics, and timer scopes.

REFERENCES

Marcel R. Ackermann, Marcus Märtens, Christoph Raupach, Kamil Swierkot, Christiane Lammersen, and Christian Sohler. StreamKM++: A clustering algorithm for data streams. *ACM Journal of Experimental Algorithmics*, 17, 2012. doi: 10.1145/2133803.2184450.

Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and Euclidean k-median by primal-dual algorithms. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science*, pp. 61–72, 2017.

Nir Ailon, Ragesh Jaiswal, and Claire Monteleoni. Streaming k-means approximation. In *Advances in Neural Information Processing Systems 22*, 2009.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, 2007.

Robi Bhattacharjee, Jacob Imola, Michal Moshkovitz, and Sanjoy Dasgupta. Online k-means clustering on arbitrary data streams. In *Proceedings of the 34th International Conference on Algorithmic Learning Theory*, volume 201 of *Proceedings of Machine Learning Research*, pp. 204–236, 2023. URL https://proceedings.mlr.press/v201/bhattacharjee23b.html.

Sayan Bhattacharya, Martín Costa, Naveen Garg, Silvio Lattanzi, and Nikos Parotsidis. Fully dynamic k-clustering with fast update time and small recourse. In *Proceedings of the 65th IEEE Symposium on Foundations of Computer Science*, pp. 216–227, 2024. doi: 10.1109/FOCS61266.2024.00023. URL https://arxiv.org/abs/2408.01325.

Sayan Bhattacharya, Martín Costa, and Ermiya Farokhnejad. Fully dynamic k-median with near-optimal update time and recourse. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pp. 1166–1177, 2025a. doi: 10.1145/3717823.3718176. URL https://arxiv.org/abs/2411.03121.

Sayan Bhattacharya, Martín Costa, Ermiya Farokhnejad, Silvio Lattanzi, and Nikos Parotsidis. Almost optimal fully dynamic k-center clustering with recourse. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4196–4209, 2025b. URL https://proceedings.mlr.press/v267/bhattacharya25a.html.

Sayan Bhattacharya, Martín Costa, Ermiya Farokhnejad, Shaofeng H.-C. Jiang, Yaonan Jin, and Jianing Lou. Fully dynamic Euclidean k-means. In *Proceedings of the 67th IEEE Symposium on Foundations of Computer Science*, 2026. URL https://arxiv.org/abs/2507.11256. To appear. Full version: arXiv:2507.11256.

Niv Buchbinder, Roie Levin, and Yue Yang. Competitively consistent clustering. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 5793–5810, 2025. URL https://proceedings.mlr.press/v267/buchbinder25a.html.

Diptarka Chakraborty, Hendrik Fichtenberger, Bernhard Haeupler, Silvio Lattanzi, Ashkan Norouzi-Fard, and Ola Svensson. Clustering with label consistency. *arXiv preprint arXiv:2512.19654*, 2025. URL https://arxiv.org/abs/2512.19654.

T.-H. Hubert Chan, Shaofeng H.-C. Jiang, Tianyi Wu, and Mengshi Zhao. Online clustering with nearly optimal consistency. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b37aa1d677970f2f56d0d17410c52b3b-Abstract-Conference.html.

Moses Charikar, Vincent Cohen-Addad, Ruiquan Gao, Fabrizio Grandoni, Euiwoong Lee, and Ernest van Wijland. A $(4+\epsilon)$-approximation for Euclidean k-means via non-monotone dual-fitting. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, 2026. doi: 10.1145/3798129.3800894. URL https://people.idsia.ch/~grandoni/Pubblicazioni/CCGGLW26stoc.pdf.

Ke Chen. On coresets for k-median and k-means clustering in metric and Euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009. doi: 10.1137/070699007. URL https://epubs.siam.org/doi/10.1137/070699007.

Davin Choo, Christoph Grunau, Julian Portmann, and Václav Rozhoň. k-means++: Few more steps yield constant approximation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1909–1917, 2020. URL https://proceedings.mlr.press/v119/choo20a.html.

Edith Cohen, Vadym Doroshenko, Badih Ghazi, Pritish Kamath, Alexander Knop, Ravi Kumar, Ethan Leeman, Pasin Manurangsi, Adam Sealfon, and Marika Swanberg. Online differentially private consistent clustering. *arXiv preprint arXiv:2608.27896*, 2026. URL https://arxiv.org/abs/2608.27896.

Vincent Cohen-Addad, Niklas Oskar D. Hjuler, Nikos Parotsidis, David Saulpic, and Chris Schwiegelshohn. Fully dynamic consistent facility location. In *Advances in Neural Information Processing Systems 32*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/fface8385abbf94b4593a0ed53a0c70f-Abstract.html.

Vincent Cohen-Addad, Benjamin Guedj, Varun Kanade, and Guy Rom. Online k-means clustering. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1126–1134, 2021a. URL https://proceedings.mlr.press/v130/cohen-addad21a.html.

Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. A new coreset framework for clustering. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 169–182, 2021b. doi: 10.1145/3406325.3451022. URL https://arxiv.org/abs/2104.06133.

Alessandro Epasto, Tamalika Mukherjee, and Peilin Zhong. Differentially private clustering in data streams. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 1360–1368, 2026. URL https://proceedings.mlr.press/v300/epasto26a.html.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pp. 569–578, 2011. doi: 10.1145/1993636.1993712.

Hendrik Fichtenberger, Silvio Lattanzi, Ashkan Norouzi-Fard, and Ola Svensson. Consistent k-clustering for general metrics. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms*, pp. 2660–2678, 2021. doi: 10.1137/1.9781611976465.158. URL https://arxiv.org/abs/2011.06888.

Sebastian Forster and Antonis Skarlatos. Dynamic consistent k-center clustering with optimal recourse. In *Proceedings of the 2025 ACM-SIAM Symposium on Discrete Algorithms*, pp. 212–254, 2025. doi: 10.1137/1.9781611978322.7. URL https://arxiv.org/abs/2412.03238.

Ameet Gadekar, Aristides Gionis, and Thibault Marette. Label-consistent clustering for evolving data. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 1204–1215, 2026. doi: 10.1145/3770855.3817933. URL https://arxiv.org/abs/2512.15210.

Xiangyu Guo, Janardhan Kulkarni, Shi Li, and Jiayi Xian. Consistent k-median: Simpler, better and robust. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1135–1143, 2021. URL https://proceedings.mlr.press/v130/guo21a.html.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pp. 291–300, 2004. doi: 10.1145/1007352.1007400. URL https://sarielhp.org/p/03/kcoreset/.

Jakub Łącki, Bernhard Haeupler, Christoph Grunau, Rajesh Jayaram, and Václav Rozhoň. Fully dynamic consistent k-center clustering. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 3463–3484. Society for Industrial and Applied Mathematics, 2024. doi: 10.1137/1.9781611977912.124. URL https://epubs.siam.org/doi/10.1137/1.9781611977912.124.

Silvio Lattanzi and Sergei Vassilvitskii. Consistent k-clustering. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1975–1984, 2017. URL https://proceedings.mlr.press/v70/lattanzi17a.html.

Edo Liberty, Ram Sriharsha, and Maxim Sviridenko. An algorithm for online k-means clustering. In *Proceedings of the 18th Workshop on Algorithm Engineering and Experiments*, pp. 81–89, 2016. doi: 10.1137/1.9781611974317.7.

Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982. doi: 10.1109/TIT.1982.1056489.

Adam Meyerson. Online facility location. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pp. 426–431, 2001. doi: 10.1109/SFCS.2001.959917.

Benjamin Moseley, Heather Newman, and Kirk Pruhs. Online k-median with consistent clusters. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2024)*, volume 317 of *Leibniz International Proceedings in Informatics*, pp. 20:1–20:22, 2024. doi: 10.4230/LIPIcs.APPROX/RANDOM.2024.20. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.APPROX/RANDOM.2024.20.

David P. Woodruff, Peilin Zhong, and Samson Zhou. Near-optimal k-clustering in the sliding window model. In *Advances in Neural Information Processing Systems 36*, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/476ab8f369e489c04187ba84f68cfa68-Abstract-Conference.html.

A BUDGETED DELETION ALGORITHM AND PROOF

Algorithm 2: BudgetDelete, on weighted set P, k distinct centers U, and budget $0 < \beta < 1$

compute clusters P_u, nearest-center targets $a(u)$, and increments q_u
properly three-color the graph with edges $\{u, a(u)\}$
foreach *color class* I **do**
    sort I by nondecreasing q_u with fixed tie-breaking
    let S_I be the longest prefix with $\sum_{u \in S_I} q_u \leq \beta F_P(U)$
return *a largest* S_I

To see that the graph is a pseudoforest, orient every edge from u to $a(u)$, so each vertex has outdegree one and every component has at most one cycle. Every induced subgraph has a vertex of degree at most two. Deleting such vertices and coloring in reverse order gives three colors.

Proof of Theorem 4.1. All deleted centers share a color, hence $a(u) \notin S$ for every $u \in S$. Reassign P_u to $a(u)$ for $u \in S$ and keep all other assignments. The increase is at most $\sum_{u \in S} q_u \le \beta F_P(U)$, proving (3). In particular, at least one center is retained.

Fix a comparison set V and scale δ only for the analysis. Write $t = t_\delta(U, V)$, $r_u = d(u, U \setminus \{u\})$, $s_u = d(u, V)$, and

$$E = \{u \in U : r_u \le 4s_u/\delta\}.$$

Separated pairs form a matching. We claim $|E| \ge (k - t)/2$. For $u \notin E$, a nearest $v \in V$ lies in the open ball $B^\circ(u, r_u/2)$. If u is unpaired, the first separation inequality holds and the second fails, so some $v' \ne v$ also lies in this ball. These open balls are pairwise disjoint because both radii are at most half the distance between their U-centers. At most t balls are paired, hence $k \ge 2\,|U \setminus E| - t$.

For $u \in E$ and $p \in P_u$,

$$d(p, a(u)) \le d(p, u) + r_u \le \frac{5}{\delta}\big(d(p, u) + d(p, V)\big).$$

Using $(x + y)^z \le 2^{z-1}(x^z + y^z)$ gives

$$\sum_{u \in E} q_u \le T := A_z \delta^{-z}\big(F_P(U) + F_P(V)\big).$$

If $k = t$, the claim is trivial. Otherwise, one color contains $b \ge (k - t)/6$ members of E. Its r cheapest such members cost at most rT/b. The integer $r = \lfloor \beta F_P(U) b/T \rfloor$ is feasible, and searching the full color class can only enlarge the returned prefix. Therefore $|S| + 1 \ge \beta F_P(U) b/T$, which is (4). The construction of S never used V or δ, proving simultaneity. The work bound follows by computing assignments and pairwise center distances, followed by three sorts. □

Inheritance under budget-feasible strengthening. Let $S_{\mathrm{BD}} = \textsc{BudgetDelete}(P, U, \beta)$ be the exact-arithmetic output. Any proper subset $\widehat{S} \subsetneq U$ satisfying

$$\left|\widehat{S}\right| \ge |S_{\mathrm{BD}}|, \qquad F_P(U \setminus \widehat{S}) \le (1 + \beta) F_P(U)$$

inherits the deletion certificate. The displayed feasibility condition gives (3), and $\left|\widehat{S}\right| + 1 \ge |S_{\mathrm{BD}}| + 1$ gives (4) simultaneously for every comparison set and separation parameter. Inclusion $S_{\mathrm{BD}} \subseteq \widehat{S}$ is unnecessary.

B Robust witnesses and substitution

Fix $0 < \eta \le 1/20$ and set

$$\tau = \frac{\eta}{100z}, \qquad \chi = 4\tau, \qquad \delta = \frac{\tau^2}{64}, \qquad R_i = \frac{\sigma}{128} 2^i. \tag{8}$$

All balls are closed. For a weighted set B, let $\mathrm{wt}(B)$ be its total weight. Write $Q \preceq P$ when Q is a weighted submultiset of P, with no greater weight at any location.

Definition B.1 (Dyadic witness)**.** *A sequence $(c_0, \dots, c_t)$ is robust for P if for every $1 \le i \le t$, with $B_i = P \cap B(c_i, R_i)$, the following rule holds. (i) if B_i is empty, $c_{i-1} = c_i$. (ii) if $F_{B_i}(\{c_i\}) \ge \tau^z R_i^z \,\mathrm{wt}(B_i)$, again $c_{i-1} = c_i$. (iii) otherwise c_{i-1} is an exact minimizer of $F_{B_i}(\{c\})$ over $c \in \mathcal{F}$, preferring c_i on ties. Starting with $c_t = x$ and applying the rule backwards defines* $\textsc{BuildWitness}(P, x, t)$.

Lemma B.2 (Displacement and nesting)**.** *Every robust sequence satisfies*

$$d(c_{i-1}, c_i) \le 2\tau R_i, \qquad d(c_0, c_i) < \chi R_i,$$

$B_{i-1} \subseteq B_i$ for $i \ge 2$, and $F_{B_i}(\{c_{i-1}\}) \le F_{B_i}(\{c_i\})$.

Proof. Only a nonempty low-cost ball can move. Weighted Minkowski and exact minimization give

$$d(c_{i-1}, c_i) \le \left(\frac{F_{B_i}(\{c_{i-1}\})}{\mathrm{wt}(B_i)}\right)^{1/z} + \left(\frac{F_{B_i}(\{c_i\})}{\mathrm{wt}(B_i)}\right)^{1/z} < 2\tau R_i.$$

Summing the dyadic series gives $d(c_0, c_i) < 4\tau R_i$. Nesting follows from $R_{i-1} = R_i/2$ and $2\tau \le 1/2$. Cost decrease is part of the rule. □

Lemma B.3 (Cost domination). *If $Q \preceq P$ contains the entire weighted ball B_i of a robust sequence, then*

$$F_Q(\{c_0\}) \le (1+\chi)^z F_Q(\{c_i\}) \le (1+\eta) F_Q(\{c_i\}).$$

Proof. For $p \notin B_i$, Lemma B.2 gives $d(p, c_0) \le (1+\chi) d(p, c_i)$. On B_i, induct on i. Nesting allows the induction hypothesis through level $i-1$, and the local one-center step does not increase cost. The factor is applied once, not once per scale. Finally $(1+\chi)^z \le e^{\chi z} = e^{\eta/25} \le 1+\eta$. □

For a center set C, let $r_C(x) = d(x, C \setminus \{x\})$ and define

$$q_C(x) = \min\{i : R_i \ge r_C(x)/64\}, \qquad t_C(x) = \min\{i : R_i \ge r_C(x)/16\}. \tag{9}$$

A snapshot is robust if every center has a witness through its required index $q_C(x)$. The procedure RESTORE(P, C) repeatedly selects an invalid center x, replaces it by the first point of BUILDWITNESS$(P, x, t_C(x))$, and stores the generated sequence.

Lemma B.4 (Restoration). *One call to* RESTORE *generates at most one witness per input-center lineage, preserves cardinality and distinctness, and returns C' with*

$$F_P(C') \le (1+\eta) F_P(C).$$

Every generated witness has $O(L)$ levels and radius in $[\sigma/16, D/8)$.

Proof. Set $\lambda = 1/100$. If a center x of isolation r generates endpoint u at radius R_t, minimality and Lemma B.2 give

$$r/16 \le R_t < r/8, \qquad d(x,u) < \chi R_t < \lambda r. \tag{10}$$

The endpoint cannot collide with another current center because its displacement is less than r.

Suppose a lineage were processed twice, and consider the first repeated processing in the call. Let x be that lineage's location just before its first processing, let r be its isolation then, and let u be its generated endpoint. Choose a center y at distance r from x at that time. Before the proposed repeated processing, the lineage of y moves at most once. If it moves, then u is still present, so its isolation at that move is at most $d(y,u) \le (1+\lambda)r$. Its descendant y' therefore satisfies

$$d(u, y') \le (1+\lambda)r + \lambda(1+\lambda)r = (1+\lambda)^2 r < 4r.$$

This inequality also holds if y never moves. Write r' for the current isolation of u, and let q' be its required index. The centers are distinct facilities, so $\sigma \le r' \le D$. In particular, $r'/64 \ge \sigma/64 = R_1$, and the first dyadic radius at least $r'/64$ exists in the sequence of (8). Dyadic rounding and the stronger bound $r' \le (1+\lambda)^2 r$ give

$$R_{q'} < \frac{r'}{32} \le \frac{(1+\lambda)^2 r}{32} < \frac{r}{16} \le R_t.$$

Thus the stored witness is long enough. It remains robust because the data are fixed during the call, so the lineage cannot require a second processing, a contradiction.

Let $x_1, \dots, x_k$ be the input centers, let $r_j^0 = d(x_j, \{x_\ell : \ell \ne j\})$, and use their input Voronoi clusters P_j with fixed tie-breaking. Consider a lineage j that is processed. Until that processing its center remains x_j. Any other original center x_ℓ processed earlier had isolation at most $d(x_\ell, x_j)$ at its processing, because x_j was present. Its displacement is consequently at most $\lambda d(x_\ell, x_j)$. In particular, the current descendant of a nearest original neighbor of x_j lies within $(1+\lambda) r_j^0$ of x_j. The generation isolation r_j therefore satisfies $r_j \le (1+\lambda) r_j^0$, and the largest generation ball has radius

$$R_t < r_j/8 \le (1+\lambda) r_j^0/8 < r_j^0/2.$$

This ball is centered at the original x_j and lies strictly inside its original Voronoi cell, including for ambient demand points. Applying Lemma B.3 to $Q = P_j$ at the full generated level bounds the final descendant's cost on P_j by $(1+\eta)$ times the original center's cost. Unprocessed lineages have unchanged centers. Summing over the original clusters proves the objective bound without multiplying losses across lineages. Finally, every generation isolation lies in $[\sigma, D]$, which gives the stated radius range and $O(L)$ levels. □

Lemma B.5 (Witness locality)**.** *A full witness born at $u = c_0$ with largest radius R_t remains robust under positive insertions outside $B(u, (1+\chi)R_t)$.*

Proof. All construction balls lie inside $B(c_t, R_t)$, which lies in the displayed influence ball by Lemma B.2. Insertions outside change none of the memberships, weights, or one-center objectives used by the witness. □

Proof of Theorem 5.1. The case $u = v$ is immediate. Put $b = d(u, v) > 0$, and let $(u = c_0, c_1, \ldots)$ be the stored witness. Choose the least i with $R_i \geq b/\tau^2$. Separation implies that this index lies in the required robust prefix, and minimality gives

$$\tau^2 R_i/2 < b \leq \tau^2 R_i. \tag{11}$$

Let $\mathcal{B} = B(u, (1+\chi)R_i)$ and let $Q = P \cap \mathcal{B}$ denote its weighted contents. The ball $\mathcal{B}$ lies strictly inside the V-cell of v: for $p \in \mathcal{B}$, $d(p, v) < 2R_i$, whereas separation gives $d(v, V \setminus \{v\}) > 32R_i$. Moreover $B_i = B(c_i, R_i) \cap P \subseteq Q$. Cleanliness gives $P' \cap \mathcal{B} = Q \subseteq T$, where $T = P'[V, v]$ is the target weighted cluster.

There are two cases. If level i is empty or low-cost, the witness rule makes c_{i-1} an exact local minimizer (or keeps zero cost). By Lemma B.3,

$$F_{B_i}(\{u\}) \leq (1+\eta) F_{B_i}(\{v\}).$$

For $p \in T \setminus B_i$, displacement and (11) give

$$d(p, v) \geq \left(1 - \frac{\tau^2}{1-\chi}\right) d(p, u),$$

and the parameter choices make the inverse zth power at most $1 + \eta$.

If level i is high-cost, let $j < i$ be the largest lower level that moved, if one exists, and set $x = c_i$. Then $c_j = x$, $R_j \leq R_i/2$, and $d(u, x) < \chi R_j$. The moving rule gives $F_{B_j}(\{x\}) < \tau^z R_j^z \operatorname{wt}(B_j)$. For $p \in B_i \setminus B_j$, we have $d(p, x) > R_j$, so $d(p, u) \geq (1-\chi) d(p, x)$. Hence

$$\begin{aligned} F_{B_i}(\{u\}) &\geq (1-\chi)^z \big(F_{B_i}(\{x\}) - F_{B_j}(\{x\})\big) \\ &\geq (1-\chi)^z \tau^z R_i^z \big(\operatorname{wt}(B_i) - 2^{-z} \operatorname{wt}(B_j)\big) \\ &\geq \tfrac{1}{4} \tau^z R_i^z \operatorname{wt}(B_i). \end{aligned}$$

The last step uses $z \geq 1$, $B_j \subseteq B_i$, and $(1-\chi)^z \geq 1 - z\chi \geq 499/500$. If no lower level moved, $u = x$ and the same conclusion follows directly from the high-cost condition. For $p \in Q \setminus B_i$, where Q denotes the weighted contents of $B(u, (1+\chi)R_i)$, we have $d(p, x) > R_i$ and $d(u, x) < \chi R_i$, giving $d(p, u) > (1-\chi)R_i$. This contribution is at least $(\tau^z R_i^z/4) \operatorname{wt}(Q \setminus B_i)$, since $\tau < 1$. Combining the two parts yields

$$F_Q(\{u\}) \geq \frac{\tau^z R_i^z}{4} \operatorname{wt}(Q). \tag{12}$$

Weighted Minkowski, (11), and (12) imply

$$F_Q(\{v\})^{1/z} \geq F_Q(\{u\})^{1/z} - b \operatorname{wt}(Q)^{1/z} \geq (1 - 4\tau) F_Q(\{u\})^{1/z}.$$

Since $(1-\chi)^{-z} \leq 1 + \eta$, the comparison holds on Q. Outside Q the preceding pointwise argument applies. Separated pairs form a matching, and a replacement cannot collide with another V-center. Summing over the original V-assignment proves the simultaneous claim. □

C Online algorithm and proof of the main theorem

The algorithm stores a robust snapshot U and its witnesses separately from the working output C. Deleting u from C does not erase the version stored with U. A center that permanently leaves the snapshot loses its version. A later return to the same coordinate starts a new version.

Initialization pads the endpoint-oracle output to k distinct facilities and calls RESTORE with invalid witnesses. A phase starts from robust snapshot U and data P_0, computes $S = \textsc{BudgetDelete}(P_0, U, \beta)$, and sets $C = U \setminus S$. While $|C| < k$, the nearest facility $\pi_{\mathcal{F}}(p)$ of

Algorithm 3: TerminalUpdate, on stored (P_0, U, wit_U), terminal set P_+, and endpoint oracle $\mathcal{O}_\alpha$

$V \leftarrow \mathcal{O}_\alpha(P_+)$, padded to k distinct candidates
if $F_{P_+}(V) > 2\alpha F_{P_0}(U)$ **then**
  $W \leftarrow V$; discard all old witnesses `// growth restart`
else
  find all δ-separated pairs between U and V
  retain (u, v) only if $P_+[V, v]$ has no record inserted after P_0
  form W by replacing every retained v by u
  preserve witnesses at $W \cap U$ and invalidate the others
$U^+ \leftarrow \textsc{Restore}(P_+, W)$
return U^+

an arriving location p is added to C if absent. The first record arriving after C becomes full terminates the phase and is included in P_+. Deletion for a new phase is internal until the next reported insertion. Therefore, a completed phase with $s = |S|$ contains at least $s + 1$ records. Repeated nearest facilities can only make it longer.

Lemma C.1 (Approximation invariant)**.** *Every phase begins with $F_{P_0}(U) \le (1+\eta)^2\alpha\,\mathrm{OPT}(P_0)$, and every reported output satisfies* (1)*.*

Proof. Initialization costs at most $(1+\eta)\alpha$ by Lemma B.4. At a normal terminal update, Theorem 5.1 followed by restoration gives $(1+\eta)^2\alpha$. At a growth restart, restoration of the oracle solution gives only $(1+\eta)\alpha$. Thus losses reset instead of multiplying across phases.

For a nonterminal report, let $Q = P_t \setminus P_0$ and let $C_t^\star \subseteq \mathcal{F}$ attain $\mathrm{OPT}(P_t)$. Every $p \in Q$ has its nearest facility in the working set, so

$$F_Q(C_t) = \sum_{(p,w)\in Q} w\, d(p, \mathcal{F})^z \le F_Q(C_t^\star).$$

Writing $A = (1+\beta)(1+\eta)^2\alpha \ge 1$, deletion and the snapshot invariant give

$$\begin{aligned} F_{P_t}(C_t) &\le A\,\mathrm{OPT}(P_0) + F_Q(C_t^\star) \\ &\le AF_{P_0}(C_t^\star) + F_Q(C_t^\star) \le A\,\mathrm{OPT}(P_t). \end{aligned}$$

Terminal outputs have the smaller factor $(1+\eta)^2\alpha$. □

Lemma C.2 (Growth restarts)**.** *The number G of growth restarts is at most $H - 1$.*

Proof. At a restart,

$$\alpha\,\mathrm{OPT}(P_+) \ge F_{P_+}(V) > 2\alpha F_{P_0}(U) \ge 2\alpha\,\mathrm{OPT}(P_0).$$

The optimum is monotone under positive insertions, so it more than doubles between successive growth events. □

Lemma C.3 (Fresh introductions)**.** *Let B count fresh centers introduced into the pre-restoration sets W, with multiplicity and with all k centers fresh at a growth restart. Then*

$$B \le kG + O_z\left(\frac{\alpha m}{\beta\eta^{2z}}\right). \tag{13}$$

Proof. At a normal endpoint, the failed restart test gives $F_{P_0}(V) \le F_{P_+}(V) \le 2\alpha F_{P_0}(U)$. Let $t = t_\delta(U, V)$, $s = |S|$, and let the phase length be h. Since $s + 1 \le h$, Theorem 4.1 gives

$$k - t \le \frac{6A_z(1+2\alpha)}{\beta\delta^z}\,h.$$

At most h V-clusters contain a new record. Thus fresh centers are at most $k - t + h = O_z(\alpha h/(\beta\eta^{2z}))$ because $\delta = \Theta_z(\eta^2)$. Sum over normal phases and add k for each restart. □

A *version* is a center and the full witness created at its last restoration. Regeneration ends the version even when the coordinate is unchanged. Snapshot removal ends it. Temporary removal from the working output does not.

Lemma C.4 (Overlap of influence balls)**.** *At any insertion time, a point belongs to influence balls of at most $O(L)$ live versions.*

Proof. Order versions containing the point from newest to oldest. If u_i is newer than still-live u_j, then u_j was present when u_i was generated. If R_i is the generated radius, (10) implies $d(u_i, u_j) > (8-\chi)R_i$. Intersecting influence balls imply

$$(1+\chi)(R_i + R_j) > (8-\chi)R_i,$$

so $R_j > 4R_i$. Generated radii lie in an interval of ratio less than $2D/\sigma$, leaving only $O(L)$ versions in the chain. □

Lemma C.5 (Invalidation and radius growth)**.** *There are $O(mL)$ retained-version regenerations whose stored witness is long enough but no longer robust. If a retained witness of radius R is too short, its successor radius exceeds $4R$.*

Proof. By Lemma B.5, a long-enough witness becomes invalid only after an insertion enters its influence ball. Charge the version to one such insertion. It was live at that insertion and can be regenerated only once. Lemma C.4 gives $O(L)$ charges per insertion. For a short witness, required index greater than its stored index means current isolation r satisfies $r/64 > R$, while the new generated radius is at least $r/16 > 4R$. □

Lemma C.6 (Total retained-version regeneration)**.** *The number T of retained-version regenerations satisfies*

$$T = O\big((k + B + mL)L\big). \tag{14}$$

Proof. There is one lineage for each initial center and fresh introduction, hence at most $k+B$. Split a lineage after every data-invalidated event. Lemma C.5 creates $O(mL)$ extra blocks. Within a block, every remaining regeneration is caused by a short witness and multiplies radius by more than four. The radius range permits $O(L)$ such events per block. The split events themselves are absorbed because $L \geq 1$. □

Proof of Theorem 3.1. Within phases, opening nearest facilities costs at most m recourse. At a terminal report, relative to the previous working output C^-,

$$\left|U^+ \setminus C^-\right| \leq \left|U^+ \setminus U\right| + \left|U \setminus C^-\right| \leq \left|U^+ \setminus U\right| + |S|\,.$$

The second term sums to at most m. The first is bounded by fresh introductions plus retained-lineage regenerations, because Lemma B.4 changes each input lineage at most once per call. Including initialization,

$$\Gamma \leq k + 2m + B + T = O(kL + BL + mL^2).$$

Substitute Lemmas C.2, C.3 and C.6 to obtain (2). Lemma C.1 gives quality. Every completed phase contains a record, so there are at most m terminal oracle calls after initialization. BUDGETDELETE uses no endpoint call, and each local minimization is over $\mathcal{F}$ (by enumeration in a general metric, or implicit projection in Proposition G.2). □

Corollary C.7 (Pareto allocation)**.** *Let $0 < \rho \leq \rho_0(z)$ and $r = \log(1+\rho)$. Choosing*

$$\beta = e^{r/(2z+1)} - 1, \qquad \eta = e^{zr/(2z+1)} - 1 \tag{15}$$

gives approximation guarantee $(1+\rho)\alpha$ and a fresh-lineage term $O_z(\alpha mL/\rho^{2z+1})$. To first order, this allocation maximizes $\beta\eta^{2z}$ subject to $\beta + 2\eta \leq \rho$.

Proofs of Corollaries 3.2 and C.7. For Corollary 3.2,

$$(1+\varepsilon/4)(1+\varepsilon/20)^2 = 1 + \frac{7}{20}\varepsilon + \frac{11}{400}\varepsilon^2 + \frac{1}{1600}\varepsilon^3 \leq 1 + \varepsilon.$$

Substitution into (2) gives the stated power. For Corollary C.7, (15) yields

$$\log((1+\beta)(1+\eta)^2) = \frac{r+2zr}{2z+1} = r.$$

For fixed z and small ρ, $\beta\eta^{2z} = \Theta_z(\rho^{2z+1})$. The first-order maximizer follows by differentiating $(\rho - x)x^{2z}$ after setting $x = 2\eta$. □

Boundary cases. In a metric, $\mathrm{OPT}(P) = 0$ exactly when all supported locations belong to $\mathcal{F}$ and their number is at most k. Output these locations while this condition holds. At the first positive-optimum prefix, initialize the main algorithm. The transition and zero-cost prefix require $O(k)$ additions. This also covers a first arrival outside $\mathcal{F}$. For $k = 1$ with a general finite facility set, calling the endpoint oracle after each record gives at most $m+1$ additions. For continuous Euclidean one-mean clustering, Theorem D.1 gives the sharper lazy algorithm.

D Deletion tightness and the one-mean benchmark

Proof of Theorem 6.1. Take a circle of circumference k with its geodesic metric and retain, for indices modulo k,

$$u_i = i, \qquad p_i = i + \delta, \qquad v_i = i + 2\delta.$$

Let $U = \{u_i\}$, $V = \{v_i\}$, and put unit weight at each p_i. Both center sets have spacing one. Every v_j is farther than $\delta d(u_i, U \setminus \{u_i\})$ from u_i, so no pair is δ-separated. Since p_i is distance δ from both u_i and v_i,

$$F_P(U) = F_P(V) = k\delta^z.$$

If u_i is deleted, p_i is at distance at least $1 - \delta$ from every remaining center of U. Otherwise its nearest-center distance is δ. Hence

$$F_P(U \setminus S) \geq k\delta^z + |S|\left((1-\delta)^z - \delta^z\right).$$

Comparing with $(1+\beta)k\delta^z$ gives the theorem. □

Proof of Theorem 6.2. Fix any center trajectory satisfying all prefix constraints. For prefix $\{1, \ldots, t\}$, the optimal center and cost are

$$\mu_t = \frac{t+1}{2}, \qquad \mathrm{OPT}_t = \frac{t(t^2-1)}{12}.$$

The variance identity gives

$$\sum_{i=1}^{t}(i-c)^2 = \mathrm{OPT}_t + t(c-\mu_t)^2.$$

Thus $(1+\varepsilon)$ quality forces

$$|c_t - \mu_t| \leq \sqrt{\frac{\varepsilon(t^2-1)}{12}} \leq t\sqrt{\frac{\varepsilon}{12}}. \tag{16}$$

Put $s = \sqrt{\varepsilon/3}$. On any maximal interval of prefixes $a, \ldots, b$, $a \geq 2$, served by one center, applying (16) at both endpoints gives

$$\frac{b-a}{2} \leq (a+b)\sqrt{\frac{\varepsilon}{12}}, \qquad b \leq qa, \qquad q = \frac{1+s}{1-s}.$$

If a_j begins the jth constant-center interval, then $a_{j+1} \leq qa_j + 1$. With $K = 1/(q-1)$, this becomes $a_{j+1} + K \leq q(a_j + K)$. If r intervals cover prefixes $2, \ldots, n$, then

$$r \geq \frac{\log\left(1 + \dfrac{2s(n-1)}{1+3s}\right)}{\log\left(\dfrac{1+s}{1-s}\right)}.$$

For $s \leq 1/3$, the numerator is at least $\log(1 + s(n-1))$ up to an absolute constant and $\log q = 2\,\mathrm{arctanh}(s) \leq 3s$. Total center-addition recourse is at least r, proving the asymptotic bound. The argument uses only feasibility of the constant-center intervals and applies even when the trajectory is chosen with knowledge of the entire stream. □

D.1 LAZY WEIGHTED MEANS

For positive records $(x_i, w_i) \in \mathbb{R}^d \times (0, \infty)$, write

$$W_t = \sum_{i=1}^{t} w_i, \qquad \mu_t = \frac{1}{W_t} \sum_{i=1}^{t} w_i x_i, \qquad Q_t = \sum_{i=1}^{t} w_i \|x_i - \mu_t\|^2.$$

Here the feasible center space is all of $\mathbb{R}^d$. The variance identity gives, for every $c \in \mathbb{R}^d$,

$$F_t(\{c\}) = Q_t + W_t \|c - \mu_t\|^2. \tag{17}$$

Thus Q_t is the one-mean optimum. The algorithm LAZYMEAN outputs the common location while $Q_t = 0$, outputs μ_s at the first positive-variance prefix s, and subsequently retains its last reported mean until its cost exceeds $(1 + \varepsilon)Q_t$, when it outputs μ_t.

Theorem D.1 (Lazy weighted means)**.** *For any stream of positive weighted points in $\mathbb{R}^d$ and $0 < \varepsilon \leq 1$,* LAZYMEAN *has $(1 + \varepsilon)$ optimal cost at every prefix. If $Q_n = 0$, its recourse is one. Otherwise, with $s = \min\{t : Q_t > 0\}$,*

$$\Gamma_n \leq 2 + \frac{\log\big(W_n Q_n / (W_s Q_s)\big)}{2 \operatorname{arsinh} \sqrt{\varepsilon}}. \tag{18}$$

The bound is independent of dimension and requires no clustering oracle.

The algorithm needs only the current weight, mean, variance, and reported center. These quantities can be maintained in $O(d)$ arithmetic operations per insertion.

Proof of Theorem D.1. Positive weights imply $Q_t = 0$ exactly when all locations in the prefix coincide. Until the first positive-variance prefix, outputting the first location therefore has zero cost and costs one addition in total. If $Q_n = 0$, the claimed result follows. Otherwise, let $s = \min\{t : Q_t > 0\}$ and output μ_s, at a cost of at most one further addition. Subsequently, retaining a center only when its cost is at most $(1 + \varepsilon)Q_t$, and otherwise outputting the optimal mean, proves the prefix quality guarantee.

It remains to count refreshes after s. Fix two prefixes $s \leq a < t$. Denote the weight, mean, and variance of the intervening block by

$$W_B = W_t - W_a, \qquad \mu_B = \frac{1}{W_B} \sum_{i=a+1}^{t} w_i x_i, \qquad Q_B = \sum_{i=a+1}^{t} w_i \|x_i - \mu_B\|^2.$$

Weighted variance decomposition gives

$$Q_t = Q_a + Q_B + \frac{W_a W_B}{W_t} \|\mu_a - \mu_B\|^2, \tag{19}$$

$$\mu_t - \mu_a = \frac{W_B}{W_t}(\mu_B - \mu_a).$$

In particular, both W_t and Q_t are nondecreasing, and $Q_a > 0$. Since $Q_B \geq 0$,

$$W_t \|\mu_t - \mu_a\|^2 \leq \left(\frac{W_t}{W_a} - 1\right)(Q_t - Q_a). \tag{20}$$

Put $x = W_t / W_a \geq 1$, $y = Q_t / Q_a \geq 1$, and $p = xy$. Then

$$\begin{aligned} \frac{W_t \|\mu_t - \mu_a\|^2}{Q_t} &\leq (x - 1)(1 - 1/y) = (x - 1)(1 - x/p) \\ &\leq \frac{(p-1)^2}{4p} = \sinh^2\left(\frac{\log p}{2}\right). \end{aligned} \tag{21}$$

The second inequality follows by maximizing the quadratic in $x \in [1, p]$. Its maximum occurs at $x = (p + 1)/2$.

Suppose a is the previous refresh prefix and the next refresh occurs at t. The retained center is μ_a, and the strict refresh condition and (17) imply

$$W_t\|\mu_t - \mu_a\|^2 > \varepsilon Q_t.$$

Equation (21) therefore yields

$$\log \frac{W_t Q_t}{W_a Q_a} > 2 \operatorname{arsinh} \sqrt{\varepsilon}. \tag{22}$$

If the refresh prefixes after s are $t_1 < \cdots < t_r \le n$, set $t_0 = s$ and sum (22) over their consecutive intervals. For $r > 0$,

$$2r \operatorname{arsinh} \sqrt{\varepsilon} < \log \frac{W_{t_r} Q_{t_r}}{W_s Q_s} \le \log \frac{W_n Q_n}{W_s Q_s}.$$

For $r = 0$ the resulting non-strict bound also holds. Each refresh introduces at most one center, and initialization contributes at most two, proving (18). □

Proof of Corollary 6.3. For the unweighted stream $x_t = t$, one has $W_t = t$ and $Q_t = t(t^2-1)/12$. Put $u = \sqrt{\varepsilon} \le 1/\sqrt{3}$ and $b = \lceil 1/u \rceil \ge 2$.

If $n \le b$, the bound $\Gamma_n \le n$ suffices: with $x = nu \le 1 + u < 2$, the inequality $\log(1+x) \ge x/(1+x) \ge x/3$ gives

$$\Gamma_n \le n \le 3u^{-1} \log(1 + nu).$$

Now suppose $n > b$. There are at most b additions through prefix b. If no later refresh occurs, this already proves the claim. Otherwise, let $a > b$ be the first later refresh. Count this refresh once. Applying the telescoping argument above to the remaining refreshes, now starting from the reported mean μ_a, and using monotonicity, gives

$$\Gamma_n \le b + 1 + \frac{\log\big(W_n Q_n / (W_b Q_b)\big)}{2 \operatorname{arsinh} u}.$$

Since $b \ge 2$,

$$\frac{W_n Q_n}{W_b Q_b} = \frac{n^2(n^2-1)}{b^2(b^2-1)} \le \frac{4}{3}\left(\frac{n}{b}\right)^4.$$

Furthermore, $b \le u^{-1} + 1$, $n/b \le nu$, and $\operatorname{arsinh} u \ge u/\sqrt{2}$ for $0 < u \le 1$. Consequently,

$$\Gamma_n = O\big(u^{-1}[1 + \log(nu)]\big) = O\big(u^{-1} \log(1 + nu)\big),$$

where the final equality uses $nu > 1$. Combining this upper bound with Theorem 6.2 proves tightness up to absolute constants. □

E Composition with a streaming summary

Assumption E.1 (Positive-record strong coreset)**.** *Fix $0 \le \theta < 1$. For each raw prefix P_t, a causal routine supplies a positive weighted dataset Q_t supported in $\mathcal{X}$ such that, on one simultaneous success event,*

$$(1-\theta) F_{P_t}(C) \le F_{Q_t}(C) \le (1+\theta) F_{P_t}(C)$$

for every center set $C \subseteq \mathcal{F}$ of size at most k. Between declared resets, changes to Q_t arrive as positive records with no hidden weight reduction. A reset supplies a new initial snapshot and retires all witnesses. Let J be the number of nonempty epochs and M the total delivered records after epoch snapshots. For each epoch j, set $H_j = 1$ if its optimum stays zero. Otherwise set $H_j = 1 + \log_2(\mathrm{OPT}(Q_{j,\mathrm{last}})/\,\mathrm{OPT}(Q_{j,+}))$, where $Q_{j,+}$ is its first positive-optimum prefix (or its initial snapshot if already positive). Define $\mathcal{H} = \sum_j H_j \ge J$.

Theorem E.2 (Coreset composition)**.** *Under assumption E.1, running the conversion independently in each epoch yields*

$$F_{P_t}(C_t) \le (1+\beta)(1+\eta)^2 \frac{1+\theta}{1-\theta} \alpha\, \mathrm{OPT}(P_t)$$

and

$$\Gamma = O_z\left(k\mathcal{H}L + \frac{\alpha M L}{\beta \eta^{2z}} + ML^2\right).$$

Proof. Apply Theorem 3.1 to the reported coreset state. The strong inequalities transfer its cost to the raw prefix and transfer a raw optimum back to the coreset, giving the displayed factor. Sum the recourse theorem across epochs. Initialization and reset transitions contribute $O(k)$ per epoch and are absorbed by $k\mathcal{H}L$. If several coreset records occur during one raw update, suppressing intermediate outputs cannot increase recourse because $|C_b \setminus C_a| \le \sum_{i=a+1}^{b} |C_i \setminus C_{i-1}|$. □

Computational composition. Let $C_{\mathrm{summary}}(n)$ be the upstream total time, including the generation of reset snapshots. For epoch j, let m_j count its positive records after the snapshot, S_j bound every intermediate demand support processed by the converter, and Prep_j include initialization and any epoch-specific facility preprocessing. Applying Proposition H.1 separately to the J nonempty epochs gives

$$O_z\left(C_{\mathrm{summary}}(n) + \sum_{j=1}^{J} \left\{\mathsf{Prep}_j + m_j f_j + (m_j+1)[T_{\alpha,j}(S_j,k) + k^2 f_j S_j L_j]\right\}\right). \tag{23}$$

The sum includes zero-optimum warm starts, which require only support and nearest-facility checks until the positive-optimum initialization. With one shared fixed facility set and radius system, merged reset snapshots of at most S locations, $S_j \le S$, and $M = \sum_j m_j$, the cost becomes

$$O_z\big(C_{\mathrm{summary}}(n) + \mathsf{Prep} + Mf + (M+J)[T_\alpha(S,k) + k^2 fSL]\big). \tag{24}$$

Here shared facility preprocessing is charged once to Prep, and the $O(JS)$ cost of reading the merged snapshots and the $O(JkL)$ cost of discarding old witnesses are absorbed by the last term. If reset snapshots arrive as unconsolidated record lists, their reading and consolidation costs are additionally included in C_{summary} or the epoch initialization terms. Thus S bounds the converter's intermediate states, not just the externally reported summaries. Parameters M, J, S, and C_{summary} retain any dependence on accuracy and failure probability. When facilities or scale parameters differ across epochs, (23) retains that dependence. The bounds maintain the current center set, with copying k centers at every original time charged separately as in Appendix H.

Randomized components. If the coreset event fails with probability at most ξ_c, at most $Q_{\max}$ randomized oracle calls occur, and every call has conditional failure probability at most p given the history, a union bound gives joint success probability at least $1 - \xi_c - Q_{\max}p$. Independence is unnecessary. An expected-α oracle can be amplified at an enlarged factor. Markov's inequality gives cost at most $2\alpha\,\mathrm{OPT}$ with probability at least $1/2$, and the best of r conditionally independent runs fails this bound with probability at most 2^{-r}. The certified factor 2α, rather than the expectation α, is then used in the theorem and restart test.

F Certified zero-slack output stabilization

Theorem F.1 (Zero-slack output stabilization)**.** *Let $A_0, \dots, A_m$ be nonempty sets of at most k centers, with prefix costs F_t and cumulative recourse $\Gamma_{\mathrm{int}}(t)$. Initialize $C_0 = A_0$. At time t, accept a nonempty $D \subseteq C_{t-1} \cup A_t$, $|D| \le k$, only if*

$$F_t(D) \le F_t(A_t),$$
$$\Gamma_{\mathrm{out}}(t-1) + |D \setminus C_{t-1}| + |A_t \setminus D| \le \Gamma_{\mathrm{int}}(t).$$

An admissible choice always exists. Returning any such choice gives, at every prefix,

$$F_t(C_t) \le F_t(A_t), \qquad \Gamma_{\mathrm{out}}(t) \le \Gamma_{\mathrm{int}}(t).$$

The output stabilizer reads an internal trajectory $A_0, \dots, A_m$ and evaluates the current prefix cost F_t. Its state consists of the previous output set and cumulative output recourse. It leaves the internal trajectory unchanged. Define

$$\Gamma_{\mathrm{int}}(t) = |A_0| + \sum_{i=1}^{t} |A_i \setminus A_{i-1}|, \qquad \Gamma_{\mathrm{out}}(t) = |C_0| + \sum_{i=1}^{t} |C_i \setminus C_{i-1}|.$$

The mathematical acceptance rule in Theorem F.1 checks both cost and a recourse potential. The acceptance tests certify candidates proposed by the search procedure.

Algorithm 4: Certified zero-slack output stabilization

$C_0 \leftarrow A_0$; $\Gamma_{\mathrm{int}}(0) \leftarrow |A_0|$; $\Gamma_{\mathrm{out}}(0) \leftarrow |A_0|$
for $t = 1, \ldots, m$ **do**
  $\Gamma_{\mathrm{int}}(t) \leftarrow \Gamma_{\mathrm{int}}(t-1) + |A_t \setminus A_{t-1}|$
  $b_t \leftarrow F_t(A_t)$; $B_t \leftarrow \Gamma_{\mathrm{int}}(t) - \Gamma_{\mathrm{out}}(t-1)$
  if $F_t(C_{t-1}) \le b_t$ **then**
    $C_t \leftarrow C_{t-1}$
  else
    $C_t \leftarrow A_t$ `// always admissible fallback`
    foreach *proposed nonempty* $D \subseteq C_{t-1} \cup A_t$ *with* $|D| \le k$ **do**
      if $F_t(D) \le b_t$ *and* $|D \setminus C_{t-1}| + |A_t \setminus D| \le B_t$ **then**
        if $|D \setminus C_{t-1}| < |C_t \setminus C_{t-1}|$ **then**
          $C_t \leftarrow D$
  $\Gamma_{\mathrm{out}}(t) \leftarrow \Gamma_{\mathrm{out}}(t-1) + |C_t \setminus C_{t-1}|$
return $C_0, \ldots, C_m$

Candidate generation may be truncated arbitrarily, including proposing no mixed candidates. The fallback therefore makes the rule well defined without any search-success assumption. The strict improvement test in Algorithm 4 is one possible selection policy. The theorem permits any candidate passing both acceptance tests.

Proof of Theorem F.1. We prove the stronger invariant

$$\Gamma_{\mathrm{out}}(t) + |A_t \setminus C_t| \le \Gamma_{\mathrm{int}}(t). \tag{25}$$

It holds with equality at $t = 0$ because $C_0 = A_0$. Suppose it holds at $t - 1$. The inclusion

$$A_t \setminus C_{t-1} \subseteq (A_t \setminus A_{t-1}) \cup (A_{t-1} \setminus C_{t-1})$$

implies

$$\begin{aligned} |A_t \setminus C_{t-1}| &\le |A_t \setminus A_{t-1}| + |A_{t-1} \setminus C_{t-1}| \\ &\le \Gamma_{\mathrm{int}}(t) - \Gamma_{\mathrm{out}}(t-1) = B_t. \end{aligned} \tag{26}$$

Choosing $D = A_t$ has cost exactly $F_t(A_t)$, respects the cardinality constraint, and has

$$|D \setminus C_{t-1}| + |A_t \setminus D| = |A_t \setminus C_{t-1}| \le B_t.$$

Hence an admissible choice always exists. If the previous output set meets the current cost threshold, choosing $D = C_{t-1}$ also passes the potential test by (26), so the hold branch in Algorithm 4 is certified as well.

For every accepted choice $C_t = D$, the potential test and the definition of output recourse give

$$\Gamma_{\mathrm{out}}(t) + |A_t \setminus C_t| = \Gamma_{\mathrm{out}}(t-1) + |D \setminus C_{t-1}| + |A_t \setminus D| \le \Gamma_{\mathrm{int}}(t).$$

This proves the invariant by induction. Dropping the nonnegative final term yields recourse dominance, and the cost test yields prefix cost dominance. Both claims are deterministic for every internal trajectory and every sequence of candidate proposals. □

A configurable cost allowance. For any $\rho_t \ge 1$, replacing the cost test in Theorem F.1 by $F_t(D) \le \rho_t F_t(A_t)$ gives

$$F_t(C_t) \le \rho_t F_t(A_t), \qquad \Gamma_{\mathrm{out}}(t) \le \Gamma_{\mathrm{int}}(t).$$

The recourse-potential proof is unchanged, and $D = A_t$ remains an admissible fallback. The practical output stabilizer uses $\rho_t = (1+\xi)^2$ for retaining the previous output and the stricter factor $1 + \xi$ for mixed candidates. These conclusions use exact comparisons. Numerical tolerances are specified in Appendix I.2.

Mixed-set identity. For $A = A_t$, $C = C_{t-1}$, and every $D \subseteq A \cup C$,

$$|D \setminus C| + |A \setminus D| = |A \setminus C| + |(A \cap C) \setminus D|. \tag{27}$$

Indeed, $D \setminus C = (A \setminus C) \cap D$, while $A \setminus D$ is the disjoint union of $(A \setminus C) \setminus D$ and $(A \cap C) \setminus D$. Every member of $A \setminus C$ is therefore counted exactly once. By (26), preserving all common centers $A \cap C$ is sufficient to pass the potential test automatically. Dropping a common center consumes additional potential and must be checked against B_t. Exact cost acceptance alone does not imply this recourse condition.

Search and numerical evaluation. The practical candidate search is specified in Appendix I.2 and preserves the recourse guarantee by enforcing the acceptance tests. A numerical tolerance $\zeta_t \geq 0$ in the cost test gives $F_t(C_t) \leq \rho_t F_t(A_t) + \zeta_t$ instead of the exact cost inequality. The recourse certificate is an integer inequality and is unchanged. The exact rule composes with any internal quality and recourse guarantee, including Theorem 3.1, with the chosen cost factor ρ_t.

G Ambient demands and an implicit Euclidean grid

The demand domain and the facility set need not coincide. This distinction lets the conversion process the original Euclidean records while representing its permitted centers by an implicit grid. Grid snapping also appears in Cohen et al. (2026, Appendix A.2), where it controls aspect ratio with additive error. Here a positive optimum scale bounds the multiplicative rounding loss, and the Cartesian grid permits exact local one-mean minimization.

Proposition G.1 (Finite facilities with ambient demands)**.** *Let $(\mathcal{X}, d)$ be a metric space, let $\mathcal{F} \subseteq \mathcal{X}$ be finite with $|\mathcal{F}| \geq k \geq 2$, and allow positive weighted records at arbitrary points of $\mathcal{X}$. Define σ, D, L using $\mathcal{F}$, and define $\mathrm{OPT}(P)$ by minimization over centers in $\mathcal{F}$. In each nonterminal insertion of Algorithm 1, open*

$$f(p) \in \arg\min_{f \in \mathcal{F}} d(p, f)$$

if it is absent and a slot is available. Pad the initial and terminal oracle solutions to k distinct facilities. Then the quality, recourse, and endpoint-call bounds of Theorem 3.1 hold without requiring the demand locations to belong to $\mathcal{F}$.

Proof. Adding facilities cannot increase an oracle solution's cost, so padding preserves its guarantee. The deletion certificate uses only triangle inequalities between demand points and facilities. Its packing argument concerns centers in $\mathcal{F}$. The witness lemmas also remain valid. Every witness center lies in $\mathcal{F}$, so each center isolation lies in $[\sigma, D]$ and every generated witness still has $O(L)$ levels. The displacement, nesting, domination, and locality proofs apply to balls containing arbitrary demand points. Their induction starts with the identity comparison at level zero and does not require a demand-point separation bound. In the substitution proof, distinct facilities u, v satisfy $d(u, v) \geq \sigma$, ensuring that the selected comparison scale is above the base scale even when demand points are arbitrarily close together. The restoration and influence-ball arguments depend on facility isolation and are unchanged.

Only the nonterminal approximation argument needs a different inequality. Let P_0 be the current phase's initial dataset, let $P_t = P_0 \uplus P_{\mathrm{new}}$, and put $A = (1+\beta)(1+\eta)^2\alpha \geq 1$. At a nonterminal report, every record $(p, w) \in P_{\mathrm{new}}$ has its nearest facility $f(p)$ in the working set. For an optimal solution $C^\star \subseteq \mathcal{F}$ to P_t,

$$\begin{aligned} F_{P_t}(C_t) &\leq A\,\mathrm{OPT}(P_0) + \sum_{(p,w) \in P_{\mathrm{new}}} w\, d(p, \mathcal{F})^z \\ &\leq A F_{P_0}(C^\star) + F_{P_{\mathrm{new}}}(C^\star) \leq A\,\mathrm{OPT}(P_t). \end{aligned}$$

The terminal proof still compares the restored solution with the endpoint oracle. Each insertion opens at most one facility, a phase deleting s centers still requires at least $s+1$ records to terminate, and at most one endpoint cluster per new record is dirty. The fresh-center and witness-regeneration counts therefore remain unchanged. Nearest-facility queries are an additional operation. They can always be performed by enumeration and require no clustering-oracle call. □

Zero optimum. For positive weights in a metric space, $\mathrm{OPT}(P) = 0$ holds exactly when every supported location belongs to $\mathcal{F}$ and there are at most k such locations. Indeed, zero cost forces each demand location to coincide with a chosen facility. Thus the finite-facility warm start outputs the supported locations during zero-optimum prefixes and initializes the conversion at the first positive-optimum prefix, using $O(k)$ additional recourse. This characterization also handles a first demand outside $\mathcal{F}$: its finite-facility optimum is already positive. As usual, an empty initial prefix may report the empty set before the first record.

G.1 Grid construction and rounding

Suppose the Euclidean demand points lie in $B = [0, a]^d$, where $a > 0$, and write $D_E = a\sqrt{d}$. Let $W_{\max} > 0$ bound the total weight of every processed prefix. Let $\mathrm{OPT}_{\mathrm{E}}(P)$ denote the optimum with unrestricted Euclidean centers, and assume that every processed prefix satisfies $\mathrm{OPT}_{\mathrm{E}}(P) \geq \Lambda > 0$. Orthogonal projection of centers onto B cannot increase their distance to any demand in B, so both optimal centers and endpoint-oracle centers may be taken in B.

Fix $0 < \gamma \leq 1$ and set

$$h_\gamma = \big((1+\gamma)^{1/z} - 1\big)(\Lambda/W_{\max})^{1/z}.$$

Choose an integer $q \geq 1$ such that $q \geq D_E/(2h_\gamma)$ and $(q+1)^d \geq k$, and define

$$s = a/q, \qquad G = \{0, s, \ldots, qs\}^d.$$

The coordinatewise nearest-grid projection π satisfies

$$\|x - \pi(x)\|_2 \leq h_G := \frac{\sqrt{d}\, s}{2} \leq h_\gamma, \qquad \sigma_G = s, \quad D_G = D_E, \quad L_G = 1 + \log_2(q\sqrt{d}).$$

The set G is fixed before the processed stream and can be represented by (a, q, d) without enumerating its $(q+1)^d$ elements.

Proposition G.2 (Rounding and exact local optimization on an implicit grid)**.** *Under the preceding assumptions, every center set $C \subseteq B$ satisfies*

$$F_P(\pi(C))^{1/z} \leq F_P(C)^{1/z} + \mathrm{wt}(P)^{1/z} h_G. \tag{28}$$

Consequently, writing $\mathrm{OPT}_G(P)$ for the optimum over G,

$$\mathrm{OPT}_{\mathrm{E}}(P) \leq \mathrm{OPT}_G(P) \leq (1+\gamma)\,\mathrm{OPT}_{\mathrm{E}}(P).$$

A continuous α-approximation followed by projection onto B and then onto G is an $\alpha_G = \alpha(1+\gamma)$-approximation oracle over G.

For $z = 2$, the exact weighted one-center minimizer over G of any nonempty weighted set $S \subseteq B$ is a nearest-grid projection of its weighted mean. A nearest-facility query costs $O(d)$ arithmetic operations, and construction of a local one-center minimizer from N supported records costs $O(Nd)$ arithmetic operations, without enumerating G.

Proof. For each demand p, use a closest center of C and its rounded image to obtain $d(p, \pi(C)) \leq d(p, C) + h_G$. The weighted ℓ_z triangle inequality gives (28) for every real $z \geq 1$. Put $b_\gamma = (1+\gamma)^{1/z} - 1$. Since $\mathrm{wt}(P) \leq W_{\max}$ and $\mathrm{OPT}_{\mathrm{E}}(P) \geq \Lambda$,

$$\mathrm{wt}(P)^{1/z} h_G \leq b_\gamma\, \mathrm{OPT}_{\mathrm{E}}(P)^{1/z}.$$

Applying this bound to a continuous optimum proves the upper bound on OPT_G. The reverse inequality follows from the restriction of feasible centers. If C is an α-approximate continuous solution, then

$$\begin{aligned} F_P(\pi(C))^{1/z} &\leq (\alpha^{1/z} + b_\gamma)\,\mathrm{OPT}_{\mathrm{E}}(P)^{1/z} \\ &\leq \alpha^{1/z}(1+\gamma)^{1/z}\,\mathrm{OPT}_{\mathrm{E}}(P)^{1/z}, \end{aligned}$$

where the second inequality uses $\alpha \geq 1$. This gives cost at most $\alpha(1+\gamma)\,\mathrm{OPT}_{\mathrm{E}}(P)$ and hence at most $\alpha_G\,\mathrm{OPT}_G(P)$. Rounding may merge centers. Padding to k distinct grid points preserves the bound.

For the computational claim, let $\mu_S = \sum_{(p,w)\in S} wp/\operatorname{wt}(S)$. The squared-distance identity gives

$$F_S(\{c\}) = F_S(\{\mu_S\}) + \operatorname{wt}(S)\|c - \mu_S\|_2^2.$$

Thus a closest grid point to μ_S is an exact minimizer. The Cartesian grid permits independent coordinate rounding, and a current grid center is retained when it is a tied minimizer, as required by the witness rule. Computing the weighted mean costs $O(Nd)$ operations and its projection costs $O(d)$. The same projection gives the nearest facility to an arriving demand. □

Corollary G.3 (Continuous-demand guarantee through an implicit grid)**.** *Assume $P^{(0)}$ has positive continuous optimum, and let m subsequent positive weighted records satisfy the preceding box, weight, and optimum assumptions. Let a certified continuous α-approximation be available at every endpoint. The conversion with ambient demands, grid facilities, and the rounded endpoint oracle has*

$$F_{P^{(t)}}(C_t) \le (1+\beta)(1+\eta)^2(1+\gamma)^2\alpha\,\mathrm{OPT}_{\mathrm{E}}(P^{(t)})$$

on every processed prefix and total recourse

$$\Gamma = O_z\left(kH_G L_G + \frac{\alpha(1+\gamma)mL_G}{\beta\eta^{2z}} + mL_G^2\right), \qquad H_G = 1 + \log_2 \frac{\mathrm{OPT}_G(P^{(m)})}{\mathrm{OPT}_G(P^{(0)})}.$$

Moreover, if $H_{\mathrm{E}} = 1 + \log_2(\mathrm{OPT}_{\mathrm{E}}(P^{(m)})/\,\mathrm{OPT}_{\mathrm{E}}(P^{(0)}))$, then $H_G \le H_{\mathrm{E}} + \log_2(1+\gamma)$. For $0 < \varepsilon \le 1$, the choice

$$\beta = \varepsilon/4, \qquad \eta = \varepsilon/20, \qquad \gamma = \varepsilon/20$$

gives $(1+\varepsilon)\alpha$ approximation and

$$\Gamma = O_z\left(k(H_{\mathrm{E}}+1)L_G + \frac{\alpha mL_G}{\varepsilon^{2z+1}} + mL_G^2\right).$$

For $z = 2$, all nearest-facility and local one-center operations admit the implicit implementations in Proposition G.2.

Proof. Apply Proposition G.1 and Theorem 3.1 with facility set G and oracle factor α_G. Transfer the resulting objective bound using $\mathrm{OPT}_G \le (1+\gamma)\,\mathrm{OPT}_{\mathrm{E}}$. Applying the same inequality at the final prefix and $\mathrm{OPT}_G \ge \mathrm{OPT}_{\mathrm{E}}$ at initialization proves the bound on H_G. For the parameter choice,

$$\log\big((1+\beta)(1+\eta)^2(1+\gamma)^2\big) \le \frac{9}{20}\varepsilon \le \log(1+\varepsilon),$$

because $\log(1+\varepsilon) \ge \varepsilon/2$ on $0 \le \varepsilon \le 1$. The recourse estimate follows by substitution. □

Initialization from a certified oracle. The positive lower bound required by Corollary G.3 can be obtained from its certified continuous oracle. On the positive-optimum initialization data $P^{(0)}$, let A_0 be the oracle solution, projected onto the known box B, and compute $b_0 = F_{P^{(0)}}(A_0)$. On the oracle's success event, feasibility and its α guarantee give

$$0 < \Lambda := b_0/\alpha \le \mathrm{OPT}_{\mathrm{E}}(P^{(0)}) \le \mathrm{OPT}_{\mathrm{E}}(P^{(t)}).$$

The last inequality holds for all subsequent positive insertions. Construct G using this Λ, the box, and $W_{\max}$, then project the same A_0 onto G, remove duplicates, and pad to k distinct grid facilities. By Proposition G.2, this supplies the initialization endpoint with factor $\alpha_G = \alpha(1+\gamma)$ without a second oracle call. For randomized solvers, include this call in the same simultaneous success event as all subsequent endpoint calls. The grid remains fixed thereafter. A smaller valid Λ requires a finer grid, while an overestimate need not preserve the rounding guarantee.

A continuous zero-optimum prefix can be served exactly by its at most k distinct demand locations. If such a warm start is used, these temporary centers need not lie on G. At the first positive-continuous-optimum prefix, apply this initialization, adding at most k centers. After that prefix all outputs lie on the fixed grid. The construction still requires the box and the total-weight bound, and the estimate b_0/α requires a certified oracle guarantee. The initialization cost, including evaluation of b_0, is included in the computational accounting of Appendix H.

For $z = 2$, forming a witness ball, computing its mean, and projecting the mean require $O(Nd)$ arithmetic operations per level when all N supported locations are scanned. The complete accounting

in Proposition H.2 includes endpoint computation, witness validation, isolation distances, and maintenance of the weighted input. In the initialization of this appendix, evaluating b_0 costs $O(N_0kd)$ if the oracle does not already provide it, and is charged once to the initial endpoint or preprocessing. The bounds use arithmetic operations, with bit complexity depending additionally on numeric representations. For other z, exact one-center minimization over G is covered by the finite-facility accounting in Appendix H, with $f = |G|$.

H Computational accounting and proof dependencies

The following bounds account for a direct implementation of Algorithms 1, 2 and 3, including exact witness validation. They separate the computation of the conversion from its center recourse.

H.1 Interfaces and data representation

Let m be the number of positive records after initialization, N the maximum number of supported demand locations, and $f = |\mathcal{F}| \geq k \geq 2$. Let $Q_{\mathrm{end}} \leq m + 1$ count the actual endpoint calls, including initialization, and let $N_j \leq N$ be the support size at call j. Write $T_\alpha(N_j, k)$ for an upper bound on the full cost of the weighted finite-facility endpoint used by the algorithm. Dependence on f, dimension, success probability, and other solver parameters is retained in this notation. If amplification is used, T_α includes all repetitions, objective evaluations, and selection, and α denotes the resulting certified factor. A monotone upper envelope may replace T_α, so that

$$\sum_{j=1}^{Q_{\mathrm{end}}} T_\alpha(N_j, k) \leq Q_{\mathrm{end}} T_\alpha(N, k) \leq (m+1) T_\alpha(N, k). \tag{29}$$

The bounds use distance queries, fixed-z cost arithmetic, and comparisons as unit operations. Euclidean bounds below count coordinate operations explicitly. These are arithmetic operation bounds, with bit complexity depending additionally on the representations of coordinates and weights.

Store supported locations in an array with current weight $w(p)$ and phase-start weight $w_0(p)$. A linear search merges a repeated location in $O(N)$ operations. A newly encountered location has $w_0(p) = 0$, and each positive arrival increases $w(p)$. At an endpoint, a cluster is dirty exactly when it contains a location with $w(p) - w_0(p) > 0$. Thus aggregation preserves the new-record test even when a new record increases weight at an old location. Copying current weights into w_0 at each phase start costs $O(N)$. The stored phase-start objective, center set, and witness sequences retain the other information used by Algorithm 3.

Facilities have stable identifiers. Center sets and witnesses use these identifiers, so membership, padding, and set comparisons can use direct scans. Padding examines a fixed list of k distinct facilities and costs $O(k^2)$ without scanning all f facilities. Initial input reading, consolidation, facility setup, and radius setup are charged to Prep. In an explicit metric, computing σ and D by pairwise enumeration costs $O(f^2)$ distance queries. No distance table is required after this preprocessing.

H.2 Total time on finite facilities

Proposition H.1 (Direct implementation of the conversion)**.** *Under the interfaces above, the core conversion has total arithmetic time*

$$\mathsf{Time}_{\mathrm{core}} = O_z\Big(\mathsf{Prep} + mf + \sum_{j=1}^{Q_{\mathrm{end}}} T_\alpha(N_j, k) + (m+1)\big[Nk + k^2 + k\log k + k^2 fNL\big]\Big). \tag{30}$$

For $N \geq 1$, $L \geq 1$, and $f \geq k \geq 2$, this gives

$$\mathsf{Time}_{\mathrm{core}} = O_z\big(\mathsf{Prep} + mf + (m+1)\left[T_\alpha(N, k) + k^2 fNL\right]\big). \tag{31}$$

The implementation maintains the current center set and supports access to it. Copying a full output at every requested reporting time and saving a historical trajectory are additional output costs.

Table 3: Direct arithmetic costs for the core conversion. A restoration call includes unsuccessful validity checks as well as witness generation.

Operation	Cost	Frequency or scope
Endpoint solver	$T_\alpha(N_j, k)$	$Q_{\mathrm{end}} \le m+1$ calls, including initialization
Nearest facility	$O(f)$	At most once per arriving record
Record merge and center membership	$O(N+k)$	Per arriving record
BUDGETDELETE	$O(Nk + k^2 + k\log k)$	Once per phase start
Witness level	$O(fN)$	Ball scan, costs, exact one-center comparison
Witness generation or validation	$O(fNL)$	All generated levels or required stored prefix
RESTORE	$O(k^2fNL + k^3)$	At most k generations and $k+1$ scans of the current centers
Endpoint grouping and dirty flags	$O(Nk)$	One nearest-center assignment of each support location
Pair tests, padding, and set updates	$O(k^2 + kL)$	Per endpoint or phase start
Phase-start weight copy	$O(N)$	Per phase start

Proof. The deletion cost in Theorem 4.1 includes assignment of the N locations to k centers, all nearest-center targets, the increments, graph coloring, and sorting. Computing the objective of an endpoint solution and marking its dirty clusters takes $O(Nk)$ operations using the stored weights. Computing both sets of isolation distances and testing all separated pairs takes $O(k^2)$. Padding, constructing W, and comparing center identifiers also take $O(k^2)$, while copying at most k witness sequences takes $O(kL)$. Copying the weight snapshot costs $O(N)$.

For one witness level, scan all supported locations to form the specified ball and compute its mass and cost. On a nonempty low-cost ball, enumerate all f facilities and compute their weighted costs. This costs $O(fN)$ and implements the exact one-center rule of Definition B.1. Generation uses $O(L)$ levels by Lemma B.4. A stored witness is valid for the current set only if it is long enough and its prefix through the current required index obeys every rule in Definition B.1. Test that prefix by recomputing each ball and its costs. On a low-cost ball, check that the stored predecessor is an exact minimizer and that it equals the current level's center whenever the latter is a tied minimizer. A shorter required prefix does not require validation of unused higher levels. This preserves the stated validity semantics and costs $O(fNL)$ even when a test fails.

Implement RESTORE by scanning the current centers for an invalid one, processing that center, and restarting the scan. Lemma B.4 permits at most k processed input lineages, hence at most $k+1$ scans and $O(k^2)$ witness checks. Each scan recomputes isolation distances in $O(k^2)$ operations, giving $O(k^3)$ in total. Witness construction contributes $O(kfNL)$ and is absorbed by the $O(k^2fNL)$ validation cost. Thus one full call costs $O(k^2fNL + k^3)$. This implementation searches the stored witnesses directly. The influence-ball overlap bound used to analyze recourse is not used as a data structure for finding affected witnesses.

There are Q_{end} restoration calls and at most Q_{end} phase starts. The arriving records require at most mf nearest-facility work and $O(m(N+k))$ merge and membership work. Summing the preceding costs gives

$$O_z\Big(\mathsf{Prep} + mf + m(N+k) + \sum_j T_\alpha(N_j, k) + Q_{\mathrm{end}}[Nk + k^2 + k\log k + kL + k^3 + k^2fNL]\Big).$$

Since $f \ge k$ and $N, L \ge 1$, the k^3, kL, and per-record maintenance terms are absorbed by $(m+1)k^2fNL$. Substituting $Q_{\mathrm{end}} \le m+1$ proves (30). The remaining elementary terms and (29) give (31). □

H.3 Squared Euclidean clustering on an implicit grid

For the fixed Cartesian grid in Appendix G, let $T_{\mathrm{end}}(N_j, k, d)$ include the actual continuous endpoint solver, clipping to the box, grid projection, duplicate removal, and padding. Projection costs $O(kd)$. Direct duplicate checks and padding with the first k grid points cost $O(dk^2)$, using the grid coordinates without enumerating all of G. Repeated solver calls and objective evaluation for amplification are included when used. The endpoint's certified factor is $\alpha_G = \alpha(1+\gamma)$ as in Proposition G.2. The initial continuous call can also supply the scale and, after projection, the initial endpoint as described in Appendix G. Its objective evaluation is charged once, either within this initial endpoint cost or in Prep_G.

Proposition H.2 (Implicit-grid total time for $z = 2$)**.** *Under the fixed-grid assumptions of Corollary G.3, the squared-Euclidean implementation has total arithmetic time*

$$\mathsf{Time}_{\mathrm{core,grid}} = O\Big(\mathsf{Prep}_G + md + \sum_{j=1}^{Q_{\mathrm{end}}} T_{\mathrm{end}}(N_j, k, d) + (m+1)[dk^2 N L_G + dk^3]\Big). \tag{32}$$

With a monotone upper envelope for T_{end}, this is at most

$$O\big(\mathsf{Prep}_G + md + (m+1)[T_{\mathrm{end}}(N, k, d) + dk^2 N L_G + dk^3]\big). \tag{33}$$

Here Prep_G includes reading and consolidating the initial data and constructing the implicit grid representation. It does not enumerate grid pairs.

Proof. By Proposition G.2, a nearest-facility query costs $O(d)$. A witness level scans the N locations to find its ball and compute its weighted mean and costs, then projects the mean exactly onto G. It takes $O(Nd)$ operations for construction or validation, including the tie test. Thus a witness takes $O(NdL_G)$ and the at most $O(k^2)$ checks in a restoration call take $O(dk^2 N L_G)$. Recomputing all isolation distances on each of the at most $k+1$ scans contributes $O(dk^3)$. Endpoint grouping and deletion take $O(dNk + dk^2 + k\log k)$, snapshot copies take $O(N + dkL_G)$, and record merging takes $O(Nd)$ per arrival. These terms are absorbed by the displayed expression because $d, N, L_G \geq 1$ and $k \geq 2$. Summing over calls proves both bounds. The term dk^3 remains explicit because an ambient demand set can have $N < k$. It can be absorbed whenever $NL_G \geq k$, in particular when $N \geq k$. □

H.4 Raw streams, update time, and output costs

Corollary H.3 (Total time for n original points)**.** *Let $n \geq 1$ count all input points, including initialization. Without a coreset, $m \leq n$ and $N \leq n$, so Proposition H.1 gives*

$$O_z\big(\mathsf{Prep} + nT_\alpha(n, k) + k^2 f n^2 L\big). \tag{34}$$

For squared Euclidean clustering on the implicit grid, if $k \leq n$, Proposition H.2 gives

$$O\big(\mathsf{Prep}_G + nT_{\mathrm{end}}(n, k, d) + dk^2 n^2 L_G\big). \tag{35}$$

The facilities or fixed grid are specified causally before the conversion starts. The finite-facility statement does not select facilities from unseen future input locations.

Proof. Use $m + 1 \leq n + 1 \leq 2n$ and the monotone endpoint bounds. In (31), nf is absorbed by $k^2 f n^2 L$. In (33), nd and ndk^3 are absorbed by $dk^2 n^2 L_G$ when $k \leq n$. □

Total time and center recourse. The bounds above count the entire core run. Growth restarts occur only $O(H)$ times, but ordinary endpoints can occur $\Theta(m)$ times. If BudgetDelete deletes no center, the next record triggers an endpoint call, even when that call returns the same centers. The recourse bound therefore supplies neither $O(T_\alpha(N, k)\log m)$ total time nor an $O(\Gamma T_\alpha(N, k))$ charging rule. Dividing (31) by m for $m > 0$ gives an amortized bound including initialization. A single terminal update can incur $O_z(f + T_\alpha(N, k) + k^2 fNL)$ operations after preprocessing. The grid counterpart is $O(d + T_{\mathrm{end}}(N, k, d) + dk^2 N L_G + dk^3)$. These bounds distinguish worst-case update cost from cumulative recourse and make no latency improvement claim.

Outputs, storage, and practical searches. The core implementation stores the current state. Reporting all k facility identifiers at each of R requested times costs an additional $O(Rk)$, and copying Euclidean center coordinates costs $O(Rkd)$. A full saved trajectory requires the corresponding output storage. With direct scans, working storage outside the endpoint solver is $O(N + f + k^2 + kL)$ metric location references and scalar values. The $O(kL)$ term accounts only for witnesses. On the implicit grid it is $O(dN + dkL_G + k^2 + d)$ coordinate and scalar entries, with the grid stored implicitly. Solver workspace, stored distance tables if chosen, input representation, and historical outputs must be added when present.

The bounds concern the theoretical state machine and original BUDGETDELETE. The practical direct-refresh method has its own endpoint schedule. Each direct cost check of at most k centers costs $O(Nk)$ metric operations or $O(Nkd)$ Euclidean operations. Costs of the actual refresh schedule and its trigger evaluations must also be counted.

For an output stabilizer examining R reports and proposing K_{out} candidates in total, direct cost and set tests take $O((R + K_{\mathrm{out}})(Nk + k^2))$ metric operations, in addition to the time to generate those candidates. Euclidean coordinate implementations may charge $O((R+K_{\mathrm{out}})(Nkd+dk^2))$ instead. The fallback permits truncated search while preserving the output stabilization guarantee, but the cost of the search actually performed remains part of the implementation's time.

Metricity and positive weights suffice for Theorem 4.1. Finite separation and exact local one-center minimization enter the witness and restoration argument. Fresh-center accounting uses the simultaneous deletion certificate and restart test, while retained-lineage accounting uses overlap, data invalidation, and radius growth. Appendix E states how summary construction and resets enter the total time, and Appendix G states the fixed scales needed for the continuous transfer.

I EXPERIMENTAL DETAILS AND ADDITIONAL CURVES

I.1 BUDGETDELETE ON FIXED SNAPSHOTS

The mechanism experiment uses squared Euclidean cost on the real line. Each of three constructions is evaluated at $k \in \{4, 6, 8, 10, 12\}$ with three replicas, giving 45 snapshots with positive initial cost. Each snapshot is reused at $\beta \in \{0.005, 0.01, 0.02, 0.05, 0.1\}$.

For a snapshot (P, U), let s be the number of centers deleted by the original BUDGETDELETE routine and define

$$s^\star = \max\{|S| : S \subsetneq U,\ F_P(U \setminus S) \le (1+\beta)F_P(U)\}.$$

Exact search enumerates retained subsets in increasing cardinality and stops at the first feasible cardinality. We report $s/s^\star$ on settings with $s^\star > 0$. This is a comparison with maximum same-budget deletion capacity for these snapshots.

Snapshot constructions. The paired-redundant construction places center pairs at $10j$ and $10j+1$, with demands at $10j+0.18$ and $10j+0.32$, for $j = 0, \dots, k/2-1$. Their integer weights are sampled from $\{5, \dots, 14\}$ and $\{2, \dots, 7\}$. The graded-offset construction places centers at $2i$ and demands at $2i + 0.72 + 0.06((i + r) \bmod 4)$, for $i = 0, \dots, k-1$, with weights in $\{1, \dots, 9\}$. The long-line construction uses the same centers and demands at every integer from 0 through $3k - 1$, with weights in $\{1, \dots, 5\}$. Here $r \in \{0, 1, 2\}$ indexes the replica, and the weight generator uses seed $71000 + 97k + r$.

Capacity and budget checks. All 225 settings pass the post-deletion cost and reassignment-budget checks, with tolerance $10^{-10}\max\{1, \text{reference cost}\}$. The mean $s/s^\star$ is 0.6852 over 162 positive-capacity settings. This is a sample statistic. The separation-based cardinality guarantee is Theorem 4.1.

The paired-redundant construction reaches exact capacity throughout. The other constructions include 31 cases where the certificate deletes no center although budget-feasible deletions exist, quantifying its conservatism on these snapshots.

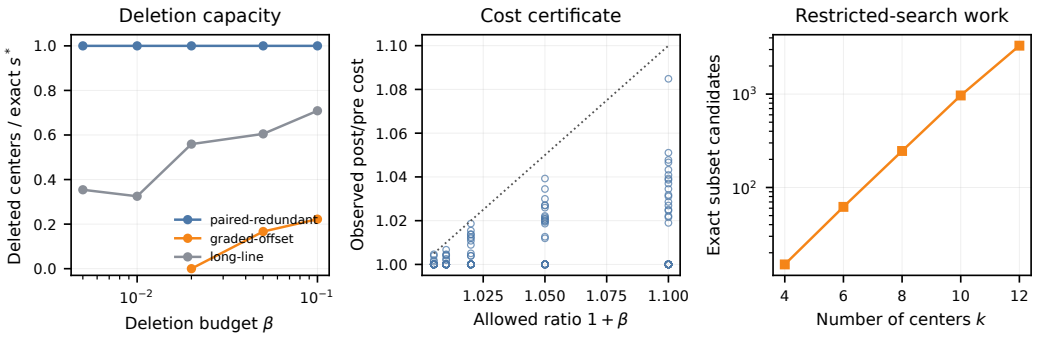

Figure 3: BUDGETDELETE on 45 fixed snapshots and five budgets per snapshot. Left, mean $s/s^\star$ within each construction and budget, restricted to settings with $s^\star > 0$. Middle, post-deletion cost divided by initial cost against the allowed ratio $1 + \beta$. Right, median number of subsets evaluated by exact search at each k. The experiment uses the original BUDGETDELETE routine and an exact same-budget capacity comparator.

Table 4: Deletion capacity by construction, over the settings with $s^\star > 0$. The last column counts cases where the original routine deletes no center although feasible deletions exist.

Construction	Positive capacity	Mean $s/s^\star$	$s = 0 < s^\star$
Paired-redundant	75	1.0000	0
Graded-offset	29	0.1609	21
Long-line	58	0.5402	10
All	162	0.6852	31

I.2 PRACTICAL REFRESH, OUTPUT STABILIZATION, AND EVALUATION

The released datasets contain 245,057 points with 3 features for Skin, 58,000 points with 7 features for Shuttle, and 581,012 points with 54 features for Covertype.

Internal update rule, OURS (core). The core initializes with distinct observed grid locations until k centers are available, with reference cost $b = 0$. Thereafter, it retains its current set A until the maintained cost exceeds $(1 + \beta)(1 + \eta)b$. The endpoint uses two k-means++ initializations with random seed 0, followed by weighted Lloyd iterations (scikit-learn 1.8.0). The solver selects a candidate by weighted inertia before grid projection, deduplication, and padding to k centers. The resulting set V is accepted only when $(1 + \beta)F_t(V) < F_t(A)$. After either acceptance or rejection, b is reset to the current cost. The lazy refresh strategy implements cost-budgeted updates by direct cost tests.

Controlled output stabilization, OURS-H (full). The full practical method applies output stabilization to the same internal trajectory after every weighted record. It retains the previous output set C if $F_t(C) \le (1 + \xi)^2 F_t(A)$, otherwise searches for a mixed set with cost at most $(1 + \xi)F_t(A)$, and falls back to A. Mixed search first greedily repairs the previous set to the inner cost budget and restores old centers when feasible. It then tests smaller new-center quotas in increasing order, using exact enumeration when a quota has at most 1024 candidates and deterministic forward selection otherwise. Common centers are preserved, and the recourse-potential condition in Appendix F is enforced. Thus the cost allowance controls reuse of the same internal trajectory, as formalized in Appendix F.

Output-stabilization cost tests allow additive tolerance $10^{-10} \max\{1, \text{compared costs}\}$. Recourse-potential checks use integer counts. The shared configuration is $(\beta, \eta) = (0.01, 0.12)$, with two k-means++ initializations and $\xi = \eta/3 = 0.04$. Its comparison is empirical. The certified quality guarantee in Theorem 3.1 applies to the deletion-and-witness converter with a certified oracle and $\eta \le 1/20$. The output stabilization guarantee compares reported centers with the internal trajectory, while the UCI metrics below compare them with the released heuristic.

Table 5: Complete quality and recourse results for Ours (core) and Ours-H (full) at the shared setting $(\beta, \eta) = (0.01, 0.12)$, with two k-means++ initializations and $\xi = 0.04$. Q_{AUC}, Q_{95}, and R_Γ use the common C+25-H evaluation as denominator, so the corresponding normalized baseline values are one. Γ_{set} is the absolute cumulative number of adjacent-set additions, including initialization.

Dataset/k	Method	Q_{AUC}	Q_{95}	Γ_{set}	R_Γ
Skin/5	C+25-H	1.0000	1.0000	125	1.0000
	Ours	0.9107	0.9378	350	2.8000
	Ours-H	0.9316	0.9649	120	0.9600
Skin/10	C+25-H	1.0000	1.0000	265	1.0000
	Ours	0.9071	0.9370	791	2.9849
	Ours-H	0.9585	0.9997	206	0.7774
Skin/20	C+25-H	1.0000	1.0000	415	1.0000
	Ours	0.9407	0.9948	1927	4.6434
	Ours-H	0.9815	1.0487	351	0.8458
Shuttle/5	C+25-H	1.0000	1.0000	83	1.0000
	Ours	0.9223	1.0150	144	1.7349
	Ours-H	0.9477	1.0023	56	0.6747
Shuttle/10	C+25-H	1.0000	1.0000	166	1.0000
	Ours	0.8898	0.9795	311	1.8735
	Ours-H	0.9161	0.9937	98	0.5904
Shuttle/20	C+25-H	1.0000	1.0000	284	1.0000
	Ours	0.8620	0.9505	830	2.9225
	Ours-H	0.8987	0.9717	204	0.7183
Covertype/5	C+25-H	1.0000	1.0000	130	1.0000
	Ours	0.9194	0.9478	338	2.6000
	Ours-H	0.9481	0.9909	104	0.8000
Covertype/10	C+25-H	1.0000	1.0000	266	1.0000
	Ours	0.8820	0.9138	666	2.5038
	Ours-H	0.9254	0.9609	247	0.9286
Covertype/20	C+25-H	1.0000	1.0000	421	1.0000
	Ours	0.9027	0.9168	1696	4.0285
	Ours-H	0.9500	0.9698	390	0.9264

Record updates and raw-time alignment. For weighted-record outputs C_t, center additions are counted by

$$\Gamma_{\mathrm{set}} = |C_0| + \sum_{t \geq 1} |C_t \setminus C_{t-1}|, \qquad R_\Gamma = \Gamma_{\mathrm{set}} / \Gamma_{\mathrm{set}}^{\mathrm{base}},$$

including initialization. Values are carried forward on the raw insertion axis. At repeated timestamps, the final record's cost and cumulative counter are retained, preserving all additions charged within that timestamp rather than recounting between the final sets.

The cost curves use a causal moving average of length $\lceil 0.01n \rceil$. Let c_s and b_s be the aligned, smoothed method and reference costs and define

$$\mathcal{T}_+ = \{s : b_s > 10^{-15} \max(1, \max_u b_u)\}, \qquad Q_{\mathrm{AUC}} = \frac{\sum_{s \in \mathcal{T}_+} c_s}{\sum_{s \in \mathcal{T}_+} b_s}.$$

The 95th percentile of c_s / b_s over $\mathcal{T}_+$ gives Q_{95}. Q_{AUC} is a ratio of accumulated costs, rather than an average of pointwise ratios. Both use weighted coreset-prefix costs. C+25-H's normalized quantities equal one by definition, and its absolute recourse in Table 5 uses the same external center-set counter as OURS and OURS-H.

Baseline and evidence scope. A read-only tracer records the released C+25-H centers, retaining the implementation's center-array alias to the first k input records. The archived audit matches its released saved costs and counters on all nine settings. Common evaluation then uses traced centers,

rather than mixing the released counter with externally counted recourse. The shared parameter setting was selected on the same benchmark collection. Each dataset has one fixed order and seed, so the comparison describes these nine runs. The timed repetitions repeat this fixed configuration, without providing independent quality trials. Full common-evaluation trajectories for $k = 10$ appear in Figures 4 and 5.

Parameter sweep. Figure 2 uses every pair in the common grid

$$\beta \in \{0.005, 0.0075, 0.01, 0.0125, 0.015, 0.02, 0.03\},$$
$$\eta \in \{0.08, 0.10, 0.12, 0.14, 0.16, 0.18, 0.20\},$$

with $k = 10$, two k-means++ initializations, and $\xi = \eta/3$. All 147 points are retained, including dominated settings and Covertype settings with recourse above the baseline. Frontier lines connect sampled nondominated points only. Across datasets, moving from minimum sampled cost to minimum sampled recourse reduces recourse by 38.6% to 43.6% while increasing cost AUC by 4.0% to 6.3%.

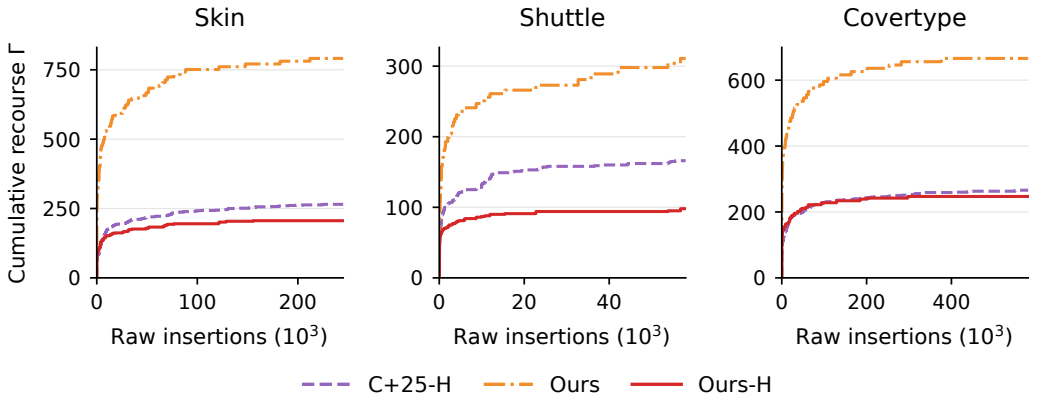

Figure 4: Cumulative center-addition recourse at $k = 10$ under common evaluation. OURS-H stabilizes the outputs of the same internal trajectory as OURS. All three methods count additions after each weighted record, including initialization, and retain every accumulated addition when mapped to raw timestamps.

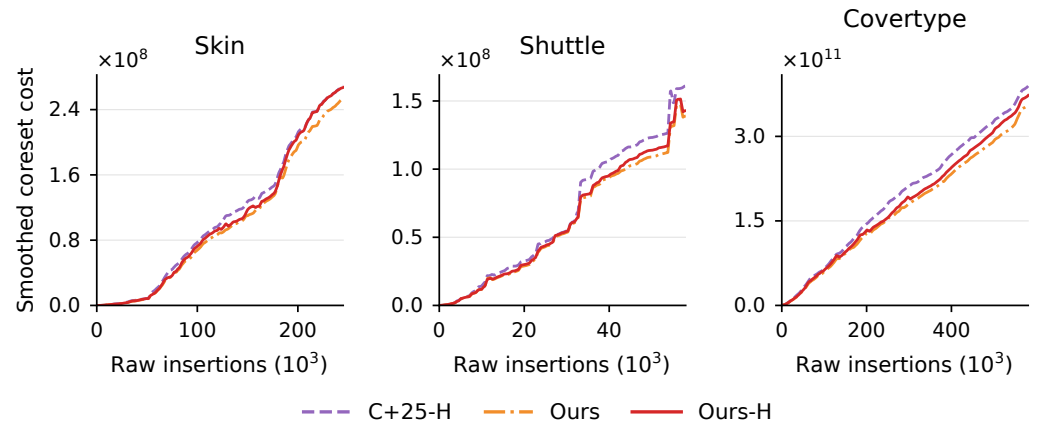

Figure 5: Weighted coreset-prefix costs at $k = 10$, aligned to raw insertion time and smoothed with a causal 1% window. All three curves use common evaluation. Output stabilization accepts a cost allowance relative to the internal OURS trajectory to reduce center changes.

I.3 RECORDED COMPUTATION

Protocol. Two full-stream repetitions per setting use alternating method order and a 48-record Shuttle warm-up. The recorded environment is Linux 6.8 x86_64, Python 3.12.9, and one OpenMP, MKL, and OpenBLAS thread. CPU model and memory were not recorded. Both repetitions match the selected quality run's final weighted cost and recourse on all nine settings.

Timer scopes. The OURS core timer covers its update loop, including endpoint calls, maintained costs, recourse, and report construction. It starts after stream extraction, grid construction, and initial setup. The OURS-H output stabilizer is replayed separately and timed after trajectory copying and setup, including candidate search and acceptance checks. Table 6 adds these two nonoverlapping loop times. For C+25-H, the measurement sums the released implementation's internal timer over updates after its first k records. This covers its candidate evaluations, center updates, and counters, while preprocessing, initial center selection, and outer progress-display overhead lie outside that timer.

Both comparisons start from given weighted streams. Input loading, coreset construction, file output, external common-metric evaluation, and plotting are excluded. The sum for OURS-H is core computation plus separate output stabilizer replay, rather than a measured online end-to-end wall clock. Figure 6 uses these same timer scopes.

Time ratios and joint comparisons. Let $\overline{T}^{(i)}_{\mathrm{core}}$, $\overline{T}^{(i)}_{\mathrm{out}}$, and $\overline{T}^{(i)}_{\mathrm{base}}$ be two-repetition means for setting i. Then

$$R^{(i)}_{T,\mathrm{core}} = \frac{\overline{T}^{(i)}_{\mathrm{core}}}{\overline{T}^{(i)}_{\mathrm{base}}}, \qquad R^{(i)}_{T,\mathrm{full}} = \frac{\overline{T}^{(i)}_{\mathrm{core}} + \overline{T}^{(i)}_{\mathrm{out}}}{\overline{T}^{(i)}_{\mathrm{base}}}.$$

Table 2 averages per-setting ratios equally and counts joint improvements by testing $Q_{\mathrm{AUC}} < 1$, $R_\Gamma < 1$, and $R_T < 1$ on each of the nine settings at the selected configuration.

Table 6: Recorded update-loop times in seconds on complete weighted streams. Ours is the practical refresh core and Ours-H adds output stabilization with $\xi = 0.04$. C+25-H and Ours-H entries are mean $\pm$ sample standard deviation over two repetitions with alternating method order. Ours-H sum adds the separately timed core and output stabilization replay loops. It is not a measured end-to-end online wall-clock time. Ratio is Ours-H sum divided by C+25-H. Timer scopes are specified in the text.

Dataset/k	C+25-H	Ours core	Stabilization	Ours-H sum	Ratio
Skin/5	0.841 ± 0.006	0.394	0.155	0.549 ± 0.005	0.653
Skin/10	6.881 ± 0.166	0.819	0.570	1.389 ± 0.007	0.202
Skin/20	60.650 ± 0.039	1.884	3.778	5.662 ± 0.080	0.093
Shuttle/5	0.382 ± 0.029	0.168	0.075	0.243 ± 0.002	0.635
Shuttle/10	3.176 ± 0.001	0.372	0.242	0.615 ± 0.003	0.194
Shuttle/20	26.918 ± 0.008	0.907	0.942	1.849 ± 0.021	0.069
Covertype/5	7.140 ± 0.096	0.767	0.357	1.125 ± 0.039	0.158
Covertype/10	74.981 ± 0.301	3.732	1.267	4.998 ± 1.881	0.067
Covertype/20	800.579 ± 2.790	13.160	6.448	19.608 ± 2.339	0.024

Observed computation. Output stabilization time exceeds core time on Skin at $k = 20$. Figure 6 preserves variation between repetitions, including the larger spread on Covertype at $k = 10$. These measurements describe the practical method, while Appendix H bounds the theoretical converter's operations.

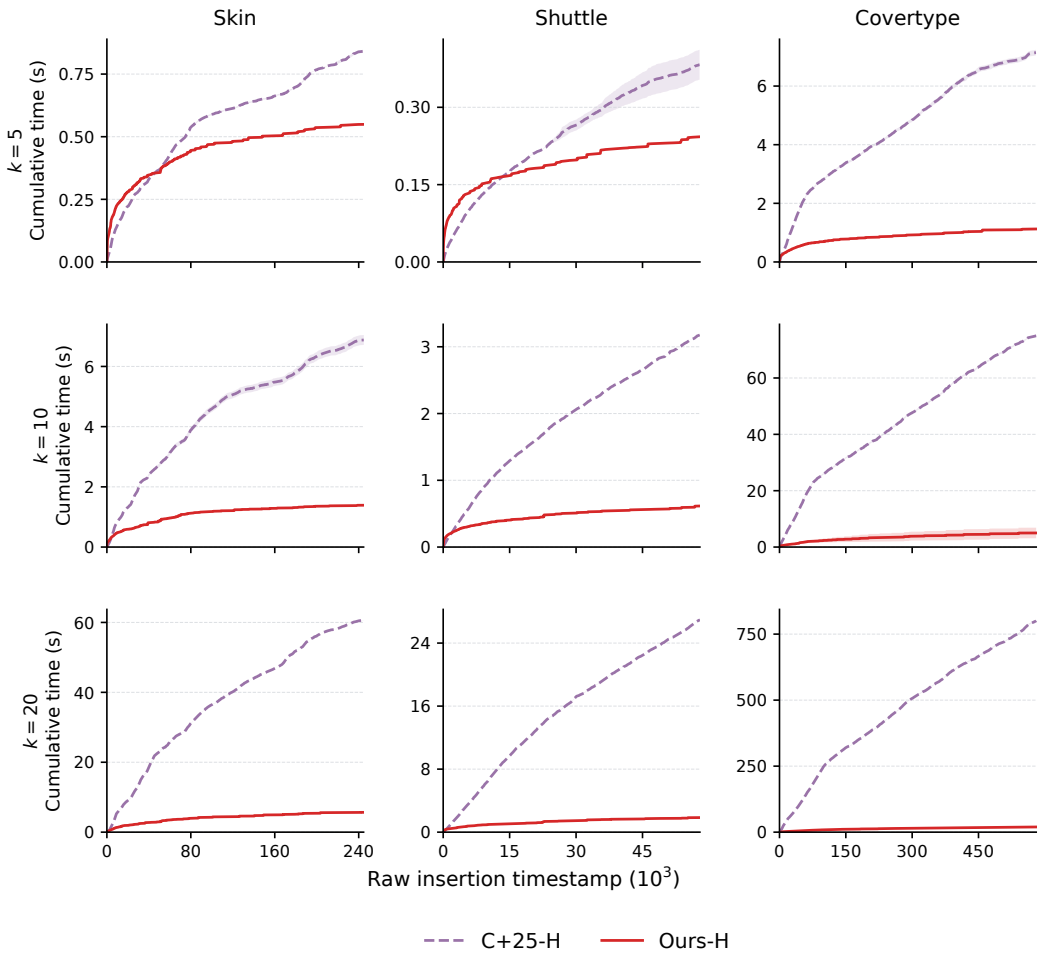

Figure 6: Cumulative recorded update-loop times for C+25-H and OURS-H on complete weighted streams. Columns identify datasets and rows identify k. Lines show means and shaded bands one sample standard deviation over two repetitions. The OURS-H curve adds the core loop and separate output stabilizer replay at each weighted record. The horizontal axis maps records to raw timestamps. Timer scopes are specified in Appendix I.3.

I.4 ONE-MEAN PRECISION BENCHMARK

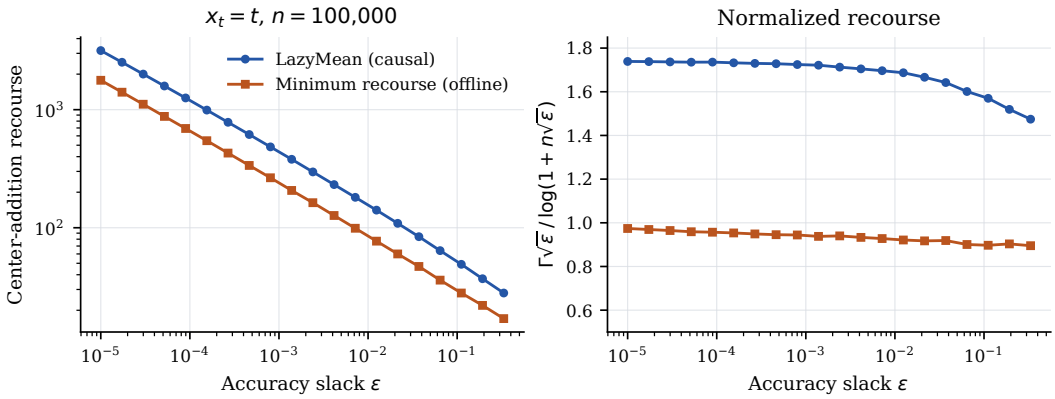

Figure 7: Deterministic one-mean experiment on $x_t = t$, $n = 100{,}000$. The left panel reports center-addition recourse, including initialization, for causal LAZYMEAN and the offline optimum. The right panel normalizes the counts by the rate in Corollary 6.3. Each point is a computed count at one of 20 accuracy settings, and lines connect observations.

Computing the one-mean comparator. On the fixed stream $x_t = t$, the feasible-center interval at prefix t is

$$I_t = \left[\frac{t+1}{2} - \sqrt{\frac{\varepsilon(t^2-1)}{12}},\ \frac{t+1}{2} + \sqrt{\frac{\varepsilon(t^2-1)}{12}}\right].$$

An offline constant-center block is feasible exactly when the intersection of its intervals is nonempty. Starting at the first uncovered prefix, take the longest feasible block and repeat. The greedy first block ends at least as late as the first block of any feasible partition. Removing its covered prefixes from an optimal partition deletes at least one block and only truncates, if necessary, the next surviving block. Truncation preserves feasibility. Induction therefore proves that the greedy block count is optimal. Selecting a center in each intersection realizes this count as center-addition recourse, including initialization.

The experiment evaluates 20 geometrically spaced values of $\varepsilon \in [10^{-5}, 1/3]$ with $n = 100{,}000$. It checks all prefix costs using the variance identity, and cross-checks the offline block count with segmentation dynamic programming for $n \in \{2, 5, 10, 20, 30\}$ at every accuracy setting. All 100 cross-checks pass. Policy decisions use double-precision arithmetic without tolerance. Cost validation allows relative roundoff of $2 \cdot 10^{-12}$ in the excess-cost slack. At $\varepsilon = 10^{-5}$, LAZYMEAN has recourse 3167 and the offline optimum has recourse 1774. At $\varepsilon = 1/3$, LAZYMEAN has recourse 28 and the offline optimum has recourse 17. The experimental supplement includes all 20 measurements and the plotting code.