DIFFERENTIALLY PRIVATE SAMPLING FOR WEIGHTED k-MEDIAN AND ROW SUBSET SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

A common approach to differentially private weighted optimization is to add noise to the private weights before running the optimization algorithm. Once the noisy weights are released, every computation based on them is post-processing, but randomized algorithms must then sample using perturbed rather than original weights. We study whether the private weights can remain inside the algorithm and be used directly for sampling. Direct sampling is difficult for zero and small weights because one record can change a sampling probability from zero to positive. The problem becomes more severe when sampling is repeated, since the privacy costs accumulate across steps and a fixed privacy budget requires larger minimum sampling weights. A private threshold provides this minimum weight, but a higher threshold can leave substantial total weight outside the sampling distribution. We address this issue by keeping the selected weights at their original locations and summing the other weights within public groups, so their total weight can still affect sampling through public representatives. Lower-bounding each group sampling weight by κ gives per-step privacy cost $\max\{\log(1+1/\lambda), \log(1+1/\kappa)\}$, which composes over adaptive sampling steps and also applies to joint weighted volume sampling. We bound the optimization error from replacing locations by their representatives and from adding weight to groups below κ. On the same family of 1-median instances, algorithms using only the selected weights incur $\Omega(\Delta M)$ cost on some input, while one sample from the grouped weights has expected cost $o(\Delta M)$ on every input. We apply these results to metric k-median and row subset selection. Experiments show that grouping the low-frequency weights improves solution quality over using only the selected weights and reduces running time on the tested instances.

1 INTRODUCTION

Differentially private optimization studies how to compute accurate solutions while limiting the influence of any individual record (Dwork et al., 2006; Bassily et al., 2014; Ghazi et al., 2020). We consider weighted objectives built from public loss functions. The private data determine the nonnegative integer weight assigned to each loss function. Thus the loss functions and feasible solutions are public, while the weights are private.

A common approach is to add noise to the private weights before running the optimization algorithm. Once the noisy weights are released, every computation based on them is post-processing and uses no additional privacy budget (Dwork & Roth, 2014; Hay et al., 2010). The same release can therefore be used throughout the optimization. The algorithm, however, computes its sampling probabilities and objective values from the noisy weights rather than the original ones.

Many randomized optimization algorithms use the weights mainly to define sampling probabilities (Arthur et al., 2007; Wang et al., 2006; Choo et al., 2020). This suggests keeping the weights inside the algorithm and using them directly when a sample is generated (Cohen et al., 2021). The released samples can then be analyzed for privacy without first releasing a noisy copy of every weight. We ask when this approach can satisfy pure differential privacy and still give an accurate solution for the original weighted objective.

The first problem appears at zero and small weights. Consider sampling a location x with probability proportional to $w_x a_x$, where $a_x \geq 0$ is fixed before the current sample. If $w_x = 0$, then x has zero

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.

sampling probability. Adding one record can increase w_x to one and give x positive probability. The likelihood ratio between the two neighboring inputs is then unbounded. For a positive weight, a unit increase changes the unnormalized sampling weight by a factor $1 + \frac{1}{w_x}$. A lower bound on the weights used in sampling can therefore control the privacy cost of one sampling step.

Repeated sampling makes the problem harder. After a sample is released, the algorithm can update its current solution. This update changes the coefficients used to define the next sampling distribution. The same private weights are therefore used several times with different sampling probabilities. The privacy costs of these steps accumulate under composition (Kairouz et al., 2015). With a fixed total sampling budget, increasing the number of sampling steps requires a larger lower bound on the weights used in each step.

A private threshold can provide this lower bound. It selects locations whose noisy weights are sufficiently large, and every selected sampling weight is at least a public value λ. This controls the zero-to-positive probability change for the selected locations. A larger λ, however, also raises the threshold. More private weight can then lie outside the selected set. If sampling uses only the selected weights, this weight no longer affects the sampling probabilities even though it remains part of the original objective.

Small individual weights can still have a large total weight. When public locations are close to one another, their weights can be summed within a public group and assigned to one public representative. The group total can then affect the sampling probabilities even when every individual weight in the group is below the threshold. This is useful even when the group is far from the selected set, since the approximation depends on the distance to the group representative.

The group totals introduce the same zero-to-positive probability change at the group level. A group can have total weight zero on one input and total weight one after a record is added. If its representative receives positive sampling probability only after this change, the likelihood ratio is again unbounded. We therefore assign every group that contains a location outside the selected set a sampling weight of at least κ. Once the group total reaches κ, the representative receives the exact group total.

Grouping also changes the weighted objective. A location outside the selected set is evaluated at its public representative instead of its original location. In addition, raising a group total to κ adds weight when the original group total is small. We bound the first change using the difference between the two loss-function values and the second change using the total added weight. These two terms allow the sampling algorithm to use low-frequency weight without charging every such record the worst possible loss.

Our contributions.

- We combine a private threshold with public groups. The selected weights remain at their original locations, while the other weights are summed at public representatives. Every selected weight is at least λ and every group weight is at least κ. For any threshold outcome, one record changes at most one sampling weight by one. This gives the per-step privacy cost $\epsilon_0 = \max\{\log(1 + 1/\lambda), \log(1 + 1/\kappa)\}$. With at most R sampling steps and additional private decisions of total conditional cost ϵ_a, the privacy cost is $\epsilon_g + R\epsilon_0 + \epsilon_a$. The same calculation applies to joint weighted volume sampling, so an entire sampled row set counts as one sampling step.
- We compare optimization with the grouped weights to the original weighted objective. Let $D_H(w)$ measure the change in loss-function values caused by replacing locations with their public representatives, and let $P_H(w)$ be the extra weight added to groups below κ. If $J_u(\widehat{\theta}) \le \alpha\,\mathrm{OPT}(u) + E$, then with high probability $J_w(\widehat{\theta}) \le \alpha\,\mathrm{OPT}(w) + E + (\alpha + 1)D_H(w) + \alpha B_{\mathrm{loss}} P_H(w)$. We also construct a family of metric instances on which algorithms using only the selected weights incur $\Omega(\Delta M)$ cost on some input, while sampling from the grouped weights has expected cost $o(\Delta M)$ on every input.
- We apply the sampling method to metric k-median and row subset selection. For k-median, grouped weights are used to initialize the centers and sample candidate centers, while the exponential mechanism (McSherry & Talwar, 2007) selects the swap. For row subset selection, a joint weighted volume sample chooses the initial rows and weighted projection-error sampling adds more rows.

The resulting error bounds depend on distances to public representatives and on the weight added by κ, rather than only on the total weight outside the selected set.

Organization. Section 2 discusses related work. Section 3 gives the weighted optimization model. Section 4 introduces the private threshold, public groups, and the privacy and objective guarantees for sampling with the resulting weights. Section 5 gives the k-median and row-subset-selection algorithms together with the separation result. Section 6 presents the experiments. The proofs and additional details are given in the appendix.

2 Related Work

Private weighted sampling. Cohen et al. (2021) study differentially private frequency-weighted sampling and estimation from the resulting samples. In our setting, sampled locations are used inside optimization algorithms and can affect the sampling probabilities in the next step. We therefore analyze repeated sampling from private weights and separately account for any private decision that also uses these weights. The selected weights and group totals remain private throughout the computation.

Private clustering and private coresets. Differentially private clustering has been studied through private combinatorial optimization, initialization, and local search (Gupta et al., 2010; Fan et al., 2023; La Tour et al., 2024). Private coreset methods instead release a private weighted summary and then run a non-private clustering algorithm as post-processing (Stemmer & Kaplan, 2018; Epasto et al., 2023). Our algorithms do not release the selected weights or the group totals. These weights are used only when sampling centers, rows, or other private choices made by the optimization algorithm. The separation result in Section 5.2 compares this use of grouped weights with algorithms that keep only (H, w_H) after thresholding.

Sampling for clustering and row subset selection. Our k-median analysis uses the exchange structure of local search (Arya et al., 2001) together with weighted D-sampling and sampled local search (Arthur et al., 2007; Lattanzi & Sohler, 2019; Choo et al., 2020). For row subset selection, we use volume sampling followed by sampling according to weighted squared residuals (Wang et al., 2006; Deshpande & Vempala, 2006; Deshpande & Rademacher, 2010; Guruswami & Sinop, 2012). We apply these sampling methods to the grouped weights and analyze the privacy of the sampled indices. In particular, the full distribution of a weighted volume sample satisfies the sampling condition in Section 4, so the sampled row set counts as one private sampling step.

3 Problem Setup

Let $\mathcal{X}$ be a fixed public set of $m \geq 1$ locations. The private weight $w_x \in \mathbb{Z}_{\geq 0}$ counts records at x; neighboring inputs satisfy $\|w - w'\|_1 = 1$. A mechanism $\mathcal{M}$ is pure ϵ-DP if $\mathbb{P}[\mathcal{M}(w) \in E] \leq e^{\epsilon}\mathbb{P}[\mathcal{M}(w') \in E]$ for every neighboring pair and measurable event E. The total weight $n = \sum_x w_x$ is private and appears only in error bounds, not as a public algorithm input. Geometry, losses, feasible sets, budgets, iteration limits, and tie-breaking rules are public. The privacy results assume exact sampling probabilities.

For a nonempty feasible set Θ, write $J_w(\theta) = g(\theta) + \sum_x w_x \ell_x(\theta), \mathrm{OPT}(w) = \inf_{\theta\in\Theta} J_w(\theta)$, where g is public and $0 \leq \ell_x(\theta) \leq B_{\mathrm{loss}}$. We assume the optimal values used below are finite and assume existence whenever a proof uses a minimizer; an approximate minimizer contributes its optimization error. For $\tau > 0$, let $L_\tau(w) = \sum_{x:w_x<\tau} w_x$. This is the total weight at locations with individual counts below τ. All logarithms are natural unless a base is shown.

4 Private Sampling with Grouped Weights

Many locations can have small individual weights even when their total weight is large. If the sampling rules use only the weights selected by a private threshold, the weights at all other locations no longer affect the sampling probabilities. We keep the selected weights at their original locations and sum the other weights within public groups. Each group is represented by one public location.

4.1 Thresholding and Grouping the Weights

Let $\lambda, \kappa \geq 1$ be public integers, with $\epsilon_g > 0$ and $\beta_g \in (0,1)$. Sample independent $\zeta_x \sim \mathrm{Lap}(1/\epsilon_g)$ and define

$$H = \{x : w_x + \zeta_x \geq 2\lambda\}, v_x = \mathbf{1}[x \in H] \max\{w_x, \lambda\}. \tag{1}$$

The set H is released with privacy cost ϵ_g by the Laplace mechanism (Dwork & Roth, 2014). The sampling steps use the weights v together with independent randomness.

The threshold has two useful consequences. Large weights are selected with high probability, and every selected weight used in sampling is at least λ. Let $\mathcal{E}_g = \{\max_x |\zeta_x| \leq \lambda\}$.

Lemma 4.1. *If $\lambda \geq \epsilon_g^{-1} \log(m/\beta_g)$, then $\mathbb{P}(\mathcal{E}_g) \geq 1 - \beta_g$. On $\mathcal{E}_g$, $\{x : w_x \geq 3\lambda\} \subseteq H \subseteq \{x : w_x \geq \lambda\}, v = w_H := w\mathbf{1}_H$.*

Thus, on $\mathcal{E}_g$, every selected location keeps its original weight. We next collect the weights at the other locations using a public partition.

Let $\mathcal{C}_1, \ldots, \mathcal{C}_K$ be a public partition of $\mathcal{X}$, chosen before accessing w, and let $r_j \in \mathcal{C}_j$ be a public representative of group j. Write $\pi(x) = r_j$ for $x \in \mathcal{C}_j$. For a threshold outcome H, define $I_H = \{j : \mathcal{C}_j \setminus H \neq \varnothing\}, t_j = \sum_{x \in \mathcal{C}_j \setminus H} w_x$. Here t_j is the total weight in group j contributed by locations outside H. The set I_H is determined by H and the public partition.

Each group in I_H contributes through its representative. We assign the representative weight at least κ and define

$$u_x = v_x + \sum_{\substack{j \in I_H \\ r_j = x}} \max\{t_j, \kappa\}, K_H = H \cup \{r_j : j \in I_H\}. \tag{2}$$

The definition of I_H includes groups with $t_j = 0$, so every representative in K_H receives positive sampling weight. Consequently, for any threshold outcome H, neighboring inputs use the same sampling set K_H.

This construction keeps the total weight of a group whenever $t_j \geq \kappa$. In particular, many locations with small individual weights can together contribute weight t_j through the representative r_j. The selected weights and the group totals are used directly in the sampling weights u.

Lemma 4.2. *For any threshold outcome H, neighboring inputs induce weight vectors u that differ in at most one coordinate, with magnitude at most one. The affected sampling weight is at least λ if the record change occurs at a location in H, and at least κ if it occurs at a location in $\mathcal{X} \setminus H$.*

For a record in H, the change occurs in its selected weight. For a record in $\mathcal{X} \setminus H$, the change occurs in the total of its public group and therefore at the representative of that group. These two cases give the lower bounds in Lemma 4.2.

4.2 Sampling with the Grouped Weights

Consider the candidate-sampling step in k-median. Given the current center set C, sample a location p with probability proportional to $u_p d(p, C)$. After a candidate is sampled and the center set changes, the distances used in the next round can change as well. We therefore analyze the privacy cost of each sampling step after fixing the outputs that have already been released.

For a threshold outcome H, let z denote the sampled indices and private decisions released before the current step. For each possible output $y \in \mathcal{Y}_{H,z}$, suppose its unnormalized probability has the form

$$Q_y(u; z) = \sum_{A \subseteq K_H} c_{y,A}(H, z) \prod_{i \in A} u_i, \mathbb{P}[Y = y \mid H, z] = \frac{Q_y(u; z)}{\sum_{a \in \mathcal{Y}_{H,z}} Q_a(u; z)}. \tag{3}$$

The sample space $\mathcal{Y}_{H,z}$ is finite, and all coefficients $c_{y,A}(H, z)$ are nonnegative. Given H and z, the coefficients and the sample space are determined by public information and the released outputs. If the normalizing sum is zero, the algorithm returns a fixed public symbol.

By Lemma 4.2, neighboring inputs change at most one coordinate of u. Suppose this coordinate changes from u_r to $u'_r = u_r + \delta$, where $0 \leq \delta \leq 1$. For every output y, separate the terms in Equation (3) according to whether they contain u_r: $Q_y(u; z) = A_y + u_r B_y, A_y, B_y \geq 0$. The

quantities A_y and B_y are the same on the two neighboring inputs. Therefore $Q_y(u;z) \le Q_y(u';z) \le \left(1+\frac{\delta}{u_r}\right) Q_y(u;z)$. Summing over all outputs gives the same inequalities for the normalizing sum. Hence, whenever the sampling probability is positive, $\frac{1}{1+\delta/u_r} \le \frac{\mathbb{P}_{u'}[Y=y|H,z]}{\mathbb{P}_u[Y=y|H,z]} \le 1+\frac{\delta}{u_r}$. If the changed record lies in H, then $u_r \ge \lambda$; otherwise $u_r \ge \kappa$. This gives the privacy cost of one sampling step.

Theorem 4.3. *For any H and z as above, a sampling step satisfying Equation* (3) *is pure ϵ_0-DP, where $\epsilon_0 = \max\{\log(1+1/\lambda), \log(1+1/\kappa)\}$. If the algorithm makes at most R such sampling steps and uses additional private decisions whose conditional privacy costs sum to ϵ_a, then the total privacy cost is*

$$\epsilon_g + R\epsilon_0 + \epsilon_a. \tag{4}$$

The sampling rule in each step may use public information, H, and the outputs already released. If the decision to continue or stop reads another private quantity, its privacy cost is included in ϵ_a.

For a public integer $R \ge 1$ and sampling budget $\epsilon_s > 0$, choose

$$\kappa = \left\lceil \max\left\{1, (e^{\epsilon_s/R}-1)^{-1}\right\}\right\rceil, \lambda = \left\lceil \max\left\{\kappa, \epsilon_g^{-1}\log(m/\beta_g)\right\}\right\rceil. \tag{5}$$

Then $R\epsilon_0 \le \epsilon_s$, so all sampling steps together use privacy budget at most ϵ_s. The choice of λ also gives the threshold guarantee in Lemma 4.1.

Joint volume sampling satisfies the same sampling rule. For a public row set, its unnormalized probability is $Q_S(u) = \det(A_S A_S^\top)\prod_{i\in S} u_i$. Thus the entire row set S counts as one sampling step, regardless of its cardinality. The bound $R\epsilon_0$ is tight for sampling rules of the form Equation (3) when all sampled outputs are released; the proof is given in Appendix A.

4.3 Comparing with the Original Objective

For each location $x \notin H$, the grouped weights evaluate its contribution at the representative $\pi(x)$. In addition, if a group has total weight $t_j < \kappa$, its representative receives an extra weight $\kappa - t_j$. We bound these two changes separately.

Let $d_x \ge 0$ be public bounds satisfying $|\ell_x(\theta) - \ell_{\pi(x)}(\theta)| \le d_x$for every $\theta \in \Theta$, and define

$$D_H(w) = \sum_{x\notin H} w_x d_x, P_H(w) = \sum_{j\in I_H} (\kappa - t_j)_+. \tag{6}$$

The quantity $D_H(w)$ bounds the total absolute change in the objective from replacing losses outside H by their representative losses. The quantity $P_H(w)$ is the total extra weight added to groups with $t_j < \kappa$. In particular, $P_H(w) \le K\kappa$.

On $\mathcal{E}_g$, the selected weights equal their original values. For every $\theta \in \Theta$, moving the other weights to their representatives changes the objective by at most $D_H(w)$. Raising small group totals to κ adds at most $B_{\text{loss}}P_H(w)$. Therefore

$$\begin{aligned} J_w(\theta) &\le J_u(\theta) + D_H(w), \\ J_u(\theta) &\le J_w(\theta) + D_H(w) + B_{\text{loss}}P_H(w). \end{aligned} \tag{7}$$

Taking the infimum over θ in the second inequality gives $\mathrm{OPT}(u) \le \mathrm{OPT}(w) + D_H(w) + B_{\text{loss}}P_H(w)$.

We can now convert a guarantee for J_u into a guarantee for the original objective.

Theorem 4.4. *Let λ and κ be chosen according to Equation* (5)*. Suppose that, for any threshold outcome H, an algorithm using independent randomness returns $\widehat{\theta}$ such that $J_u(\widehat{\theta}) \le \alpha\,\mathrm{OPT}(u) + E$ with probability at least $1-\beta_a$, where $\alpha \ge 1$ and $E \ge 0$ are independent of H. Then, with probability at least $1-\beta_g-\beta_a$, $J_w(\widehat{\theta}) \le \alpha\,\mathrm{OPT}(w) + E + (\alpha+1)D_H(w) + \alpha B_{\text{loss}}P_H(w)$.*

For metric losses, set $d_x = d(x,\pi(x))$. Hence a location can be far from H and contribute little to $D_H(w)$ when it is close to its public representative. Choosing more, smaller public groups can reduce these representative distances. At the same time, every group with $t_j < \kappa$ contributes $\kappa - t_j$ to $P_H(w)$. Section 5 applies these two terms to metric k-median and row subset selection.

5 Optimization with Grouped Weights

We now use the grouped weights u in two optimization problems. For metric k-median, u determines the sampled centers and candidate centers, and the swap is chosen privately from the weighted gains. We then compare this use of group totals with algorithms that keep only (H, w_H) after thresholding. For row subset selection, u determines a joint volume sample and the residual samples that enlarge its span.

5.1 Metric k-Median

Let $(\mathcal{X}, d)$ be a public metric of diameter $\Delta > 0$, and let $2 \le k \le m$. For a k-set C, write $\Phi_w(C) = \sum_x w_x d(x, C), \mathrm{OPT}_k(w) = \min_{|C|=k} \Phi_w(C)$. For the public grouping, define $T_H(w) = \sum_{x \notin H} w_x d(x, \pi(x))$. For every center set C, the triangle inequality gives $|d(x, C) - d(\pi(x), C)| \le d(x, \pi(x))$. Thus the metric loss in Equation (6) can use $d_x = d(x, \pi(x))$, giving $D_H(w) = T_H(w)$.

Sampled local search. Sample the first center proportionally to u_x. Once the center set C is nonempty, sample each new center proportionally to $u_x d(x, C)$ until $|C| = k$. Starting from this center set, perform T local-search rounds. In each round, sample a candidate p proportionally to $u_p d(p, C)$.

For each current center $q \in C$, define the gain from replacing q by p as $G(q) = \Phi_u(C) - \Phi_u\big((C \setminus \{q\}) \cup \{p\}\big)$. The swap gains use the private weights u, so we choose the action with the exponential mechanism. Set $U(q) = G(q)(q \in C), U(\perp) = b_{\mathrm{sw}}$, where $\perp$ keeps the current center set unchanged. In each round, sample $a \in C \cup \{\perp\}$ proportionally to $\exp\left(\frac{\epsilon_t U(a)}{2\Delta}\right)$.

Let the privacy budgets and failure probabilities satisfy $\epsilon_g + \epsilon_s + \epsilon_{\mathrm{sw}} \le \epsilon, \beta_g + \beta_i + \beta_E + \beta_p \le \beta \le \frac{1}{2}$. Let T be the public round count in Equation (10), and set $R = k + T, \epsilon_t = \frac{\epsilon_{\mathrm{sw}}}{T}, b_{\mathrm{sw}} = \frac{2\Delta T}{\epsilon_{\mathrm{sw}}} \log \frac{(k+1)T}{\beta_E}$. Use R in Equation (5). The full procedure is given in Algorithm 1.

The initialization and candidate proposals use at most $k + T$ sampling steps, so their total privacy cost is at most $\epsilon_g + \epsilon_s$ by Theorem 4.3. Given H, C, p, q, changing one record changes $G(q)$ by at most Δ. Each swap choice therefore has conditional privacy cost ϵ_t, and the T choices use total budget ϵ_{sw}.

Progress from a sampled candidate. For the grouped weight vector u, the initialized center set C_0 satisfies $\mathbb{E}\Phi_u(C_0) \le 4(1 + \mathsf{H}_{k-1}) \mathrm{OPT}_k(u)$, where $\mathsf{H}_j = \sum_{i=1}^{j} 1/i$ and $\mathsf{H}_0 = 0$. When the current cost is well above $\mathrm{OPT}_k(u)$, distance-weighted sampling has a constant probability of proposing a useful center.

Lemma 5.1. *Let $F = \Phi_u(C)$. If $F > 16\,\mathrm{OPT}_k(u)$, then a candidate sampled proportionally to $u_p d(p, C)$ permits a swap with gain greater than $F/(8k)$ with probability at least $5/32$.*

The exponential mechanism chooses an action whose utility is within b_{sw} of the largest utility in that round, except with probability β_E/T. On this event, the no-op action keeps the cost from increasing. If $\Phi_u(C) > \max\{16\,\mathrm{OPT}_k(u), 16kb_{\mathrm{sw}}\}$ and the event in Lemma 5.1 occurs, the chosen swap has gain greater than $\Phi_u(C)/(16k)$. Hence that round gives $\Phi_u(C_{\mathrm{new}}) \le \left(1 - \frac{1}{16k}\right) \Phi_u(C)$.

The round count in Equation (10) gives enough such candidate proposals with probability at least $1 - \beta_p$. Together with the initialization event and the accuracy of the private swap choices, this gives $\Phi_u(C_T) \le \max\{16\,\mathrm{OPT}_k(u), 16kb_{\mathrm{sw}}\}$ with failure probability at most $\beta_i + \beta_E + \beta_p$. The proof in Appendix B counts the successful candidate proposals before the swap action is chosen.

On $\mathcal{E}_g$, Equation (7) with $D_H = T_H$ and $B_{\mathrm{loss}} = \Delta$ gives $\Phi_w(C) \le \Phi_u(C) + T_H(w)$, and $\mathrm{OPT}_k(u) \le \mathrm{OPT}_k(w) + T_H(w) + \Delta P_H(w)$. Combining these inequalities with the bound on $\Phi_u(C_T)$ gives the following result.

Theorem 5.2. *Let $T, R, \epsilon_t, b_{\mathrm{sw}}$ be as above, and let λ, κ be chosen according to Equation* (5)*. Then, the algorithm satisfies pure ϵ-DP, and with probability at least $1 - \beta$, its weighted k-median cost*

satisfies

$$\Phi_w(C_T) \le \max\{16\,\mathrm{OPT}_k(w), 16kb_{\mathrm{sw}}\} + 17T_H(w) + 16\Delta P_H(w). \tag{8}$$

If every location outside H lies within distance ρ of its representative, then $T_H(w) \le \rho\sum_{x\notin H} w_x$, $P_H(w) \le K\kappa$. The algorithm evaluates weighted distances at the locations in K_H, whose size is at most $|H| + K$.

5.2 Why the Group Totals Matter

The bound in Theorem 5.2 depends on the distance from each location to its public representative. This can be much smaller than the distance from that location to the selected set H. The following family shows why this difference matters.

Let o be a public location and let $B_1, \dots, B_N$ be N public groups, each containing q locations. Distinct locations in the same group have distance δ, and all other distinct pairs have distance Δ. Let $r_i \in B_i$ be a public representative of group B_i.

On input j, assign weight $B = 3\lambda$ to o, weight $\lambda - 1$ to every location in B_j, and weight zero to all other locations. Let $M = q(\lambda - 1)$. On $\mathcal{E}_g$, every input gives $H = \{o\}, w_H(o) = B$. Hence all N inputs give the same pair (H, w_H).

Consider any algorithm that, after thresholding, receives (H, w_H) and fresh randomness and returns one public center. Conditional on $\mathcal{E}_g$, its output distribution is the same on all N inputs. Since the groups are disjoint, there is some j for which the algorithm returns a center in B_j with probability at most $1/N$. On input j, every center outside B_j incurs cost at least $M\Delta$ from the positive low-frequency weights.

The grouped weights contain more information about the same input. Use the public partition $\{o\}, B_1, \dots, B_N$. On input j and on $\mathcal{E}_g$, $u_o = B, u_{r_j} = M, u_{r_i} = \kappa(i \ne j)$. Therefore one sample proportional to u selects the representative of B_j with probability $\frac{M}{B+M+(N-1)\kappa}$. When M is large compared with $B + (N-1)\kappa$, this probability is close to one.

Theorem 5.3. *Let $A_0 \ge 1$, $N \ge 2$, $\Delta > 0$, and $\epsilon_g, \epsilon_s > 0$. For the family above, set $\beta_g = q^{-2}, \delta = \frac{\Delta}{q^2}$, and let λ, κ be given by Equation (5) with $m = 1 + Nq$ and $R = 1$. As $q \to \infty$, $L_{3\lambda}(w^{(j)}) = M, \mathrm{OPT}_1(w^{(j)}) = o(\Delta M)$ uniformly over j.*

For every algorithm that receives only (H, w_H) after thresholding and returns one center using fresh randomness, some input j satisfies $\mathbb{P}\left[\Phi_{w^{(j)}}(\widehat{c}) \ge A_0\,\mathrm{OPT}_1(w^{(j)}) + \frac{\Delta M}{2}\right] \ge 1 - \frac{1}{N} - o(1)$. On the same family, sampling one center proportionally to u is pure $(\epsilon_g + \epsilon_s)$-DP and, for every j, $\mathbb{E}\Phi_{w^{(j)}}(\widehat{c}) = o(\Delta M)$.

The finite bounds and parameter conditions are given in Theorem B.1. On $\mathcal{E}_g$, every positive low-frequency location lies at distance Δ from H, so $\sum_{x\notin H} w_x d(x, H) = M\Delta$. The same locations lie within distance δ of their representative, which gives $T_H(w) \le M\delta$. Thus the gain from grouping comes from the geometry inside each public group, rather than from proximity to the selected set.

5.3 Row Subset Selection

Let $A \in \mathbb{R}^{m\times d}$ be public, with rows $a_i^\top$ satisfying $\|a_i\|_2 \le \mathcal{R}$, and let $1 \le r \le d$ be the target rank. For a linear subspace L, write $\Phi_w(L) = \sum_i w_i\,\mathrm{dist}(a_i, L)^2$, $\mathrm{OPT}_r(w) = \min_{\dim L \le r} \Phi_w(L)$. For an index set S, define $\mathrm{RSS}_r(w; S) = \min_{\substack{L\subseteq \mathrm{span}\{a_i : i\in S\}\\ \dim L\le r}} \Phi_w(L)$. Each public representative is an index of an original row. Define $T_{2,H}(w) = \sum_{i\notin H} w_i \|a_i - a_{\pi(i)}\|_2^2$.

Joint and residual row sampling. Fix $0 < \eta \le 1$. In one repetition, first sample $\ell = 2r - 1$ rows jointly with probability proportional to $\det(A_V A_V^\top)\prod_{i\in V} u_i, |V| = \ell$. Let Q_0 be the span of these rows. Keeping Q_0 fixed, sample $b = \left\lceil \frac{16r}{\eta} \right\rceil$ additional rows independently with probabilities proportional to $u_i\,\mathrm{dist}(a_i, Q_0)^2$. Repeat this procedure $s = \left\lceil \log_2 \frac{1}{\beta_s} \right\rceil$ times with independent randomness and return the union of all sampled indices.

Let the privacy budgets and failure probabilities satisfy $\epsilon_g + \epsilon_s \le \epsilon, \beta_g + \beta_s \le \beta < 1$, and use $R = s(1+b)$ in Equation (5). The rank and zero-residual branches are given in Algorithm 2.

Each repetition uses one joint volume sample and at most b residual samples. The joint volume sample has the form in Equation (3), so the entire set of ℓ rows counts as one sampling step. Each residual sample also satisfies Equation (3). Hence Theorem 4.3 gives pure ϵ-DP.

For the grouped weight vector u, let D be the projection error after the joint volume sample. The volume-sampling bound gives $\mathbb{E}D \le 2\,\mathrm{OPT}_r(u)$ (Wang et al., 2006). Conditional on the sampled span, the b residual samples give expected excess at most $\frac{r}{b}D$ (Deshpande & Vempala, 2006). Therefore, for the excess X after one repetition, $\mathbb{E}X \le \frac{2r}{b}\,\mathrm{OPT}_r(u) \le \frac{\eta}{8}\,\mathrm{OPT}_r(u)$. For $\mathrm{OPT}_r(u) > 0$, Markov's inequality gives $\mathbb{P}\left[X \le \frac{\eta}{4}\,\mathrm{OPT}_r(u)\right] \ge \frac{1}{2}$. If one repetition satisfies this event, the final union contains its sampled span. The s independent repetitions therefore give, with probability at least $1-\beta_s$, $\mathrm{RSS}_r(u;S) \le (1+\eta/4)\,\mathrm{OPT}_r(u)$.

For any subspace L and any $\gamma > 0$, Young's inequality gives $\mathrm{dist}(a_i, L)^2 \le (1+\gamma)\,\mathrm{dist}(a_{\pi(i)}, L)^2 + (1+1/\gamma)\|a_i - a_{\pi(i)}\|_2^2$. Applying the same inequality with a_i and $a_{\pi(i)}$ exchanged compares the grouped optimum with the original optimum. Taking $\gamma = \eta/4$ and accounting for the extra group weight $P_H(w)$ gives the following bound.

Theorem 5.4. *Let $0 < \eta \le 1$, set $\ell = 2r-1$, $b = \lceil 16r/\eta \rceil$, $s = \lceil \log_2(1/\beta_s) \rceil$, and $R = s(1+b)$, and let λ, κ be given by Equation* (5)*. The row subset selection algorithm is pure ϵ-DP and returns at most $s(2r-1+b) = O\left(\frac{r}{\eta}\log\frac{2}{\beta_s}\right)$ distinct row indices. With probability at least $1-\beta$,*

$$\mathrm{RSS}_r(w;S) \le (1+\eta)\,\mathrm{OPT}_r(w) + \frac{13}{\eta}T_{2,H}(w) + 2\mathcal{R}^2 P_H(w). \tag{9}$$

The released output is the index set S. Its public span contains a rank-at-most-r subspace with the loss in Equation (9). A projector fitted using private weights requires an additional privacy guarantee. The cases $\mathrm{rank}(A_{K_H}) < \ell$ and $\mathrm{OPT}_r(u) = 0$, together with the coefficient calculation in Equation (9), are treated in Appendix C.

6 Experiments

We compare GW with the private clustering method of Fan et al. (2023), full-budget clipped histograms (HIST-CLIP), and histograms on GW's head/group support (HIST-GROUP). RSS also includes public pivoted QR and uniform row selection. Unless varied, $\epsilon = 1$ and $K = 128$. We report total original-weight objectives $\Phi_w(C)$ and $\mathrm{RSS}_r(w;S)$ on the preprocessed public inputs, with metric distances scaled to diameter one. Costs are not divided by the record count or an optimum. Results use 30 paired seeds and pointwise 95% bootstrap intervals. Algorithm time includes count processing and optimization; public setup is separate.

Comparison with private clustering. Figure 1a–b compares all four methods on Skin, Shuttle, and Census at $m = 2,048$, $k = 8$, $T = 64$, and Zipf-1 density $n/m = 100$. Fan et al.'s method uses private tree initialization and complete swap enumeration. GW takes 11.1–13.5 ms versus 3104.4–3116.9 ms, with higher mean costs. HIST-GROUP shows that the reduced support also benefits input perturbation. The full size grid and earlier Gupta comparisons are in Appendix D.13 and Table 8.

Row subset selection. At $r = 2$ and $n/m = 100$, GW's mean total reconstruction costs are 71,483 on Colon and 513,802 on Leukemia, compared with 74,031 and 555,837 for public QR (Figure 1c–d). USPS costs are close across methods. The output is a row index set, scored by the best rank-two reconstruction in its span. Realized row counts and additional settings appear in Appendix D.4.

Scaling with the public dictionary. On Census, increasing m from 8,192 to 131,072 at fixed density raises GW's median time from 17.7 to 44.1 ms (Figure 2). At the largest size, HIST-CLIP takes 3016.7 ms; their total costs are 3,476,976 and 3,337,089. These calls reuse public geometry; first-use costs are in Appendix D.12.

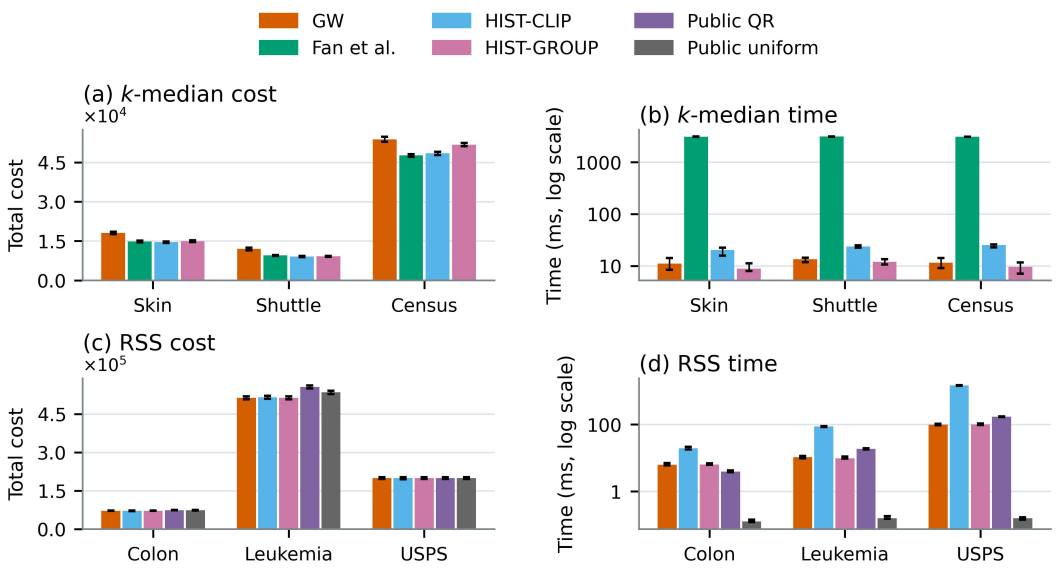


Figure 1: Mean total cost and median running time. (a,b) Matched k-median instances with dense distances. (c,d) RSS under Zipf-1 counts, with row caps 35, 35, and 140. Cost axes start at zero; time axes are logarithmic in milliseconds. Public QR time includes its reusable factorization. Intervals use 30 paired seeds.

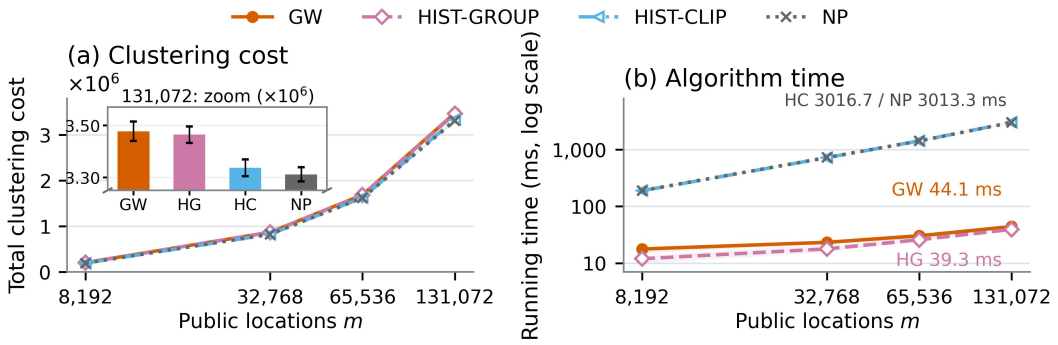


Figure 2: Census scaling with Zipf-1 density 100, $K = 128$, $k = 8$, and $T = 64$. Left: total cost and enlarged final-size range (HG/HC: grouped/full histogram). Right: median milliseconds, log scale. Labels mark final values; bands use 30 paired seeds. NP is nonprivate.

Group-total and threshold-only ablations appear in Figures 16 and 21. The practical $T = 64$ schedule differs from the longer schedule in Appendix B.

7 Conclusion

We studied differentially private weighted optimization using private weights that remain inside the algorithm. A private threshold keeps large weights at their original locations, while the remaining weights are summed within public groups and continue to affect the sampling steps. We gave privacy and optimization guarantees for metric k-median and row subset selection. On the same family of metric instances, our separation result shows that using only the selected weights can incur error of order $\Delta L_{3\lambda}(w)$, while grouped sampling can achieve $o(\Delta L_{3\lambda}(w))$ error.

AI USE STATEMENT

Generative AI tools were used only for language polishing of the manuscript and assistance in drafting validation and diagnostic code. All literature discovery was conducted by the authors, and all research ideas, problem formulation, algorithm design, theoretical results and proofs, experimental design, analysis, and substantive scientific content were developed by the authors. The authors independently reviewed and verified all AI-assisted material against the relevant literature, formal arguments, and executable diagnostics, and take full responsibility for the final manuscript, code, experiments, and results.

ETHICS STATEMENT

The privacy guarantees in this work assume a fixed public domain, geometry, loss functions, and grouping structure; if any of these are derived from private membership or other sensitive information, they require a separate privacy analysis. Thresholding and grouping control aggregate objective error but do not provide group-level fairness or representation guarantees, particularly for low-frequency locations. Applications should therefore assess such effects separately and avoid releasing intermediate private quantities beyond the analyzed outputs without an additional privacy argument.

REPRODUCIBILITY STATEMENT

All proofs for the main privacy, objective-transfer, separation, and application guarantees are included in the paper and appendices. The experimental appendix and accompanying artifacts specify the data preprocessing, algorithm parameters, seeds, timing boundaries, hardware, and evaluation procedures, and provide configurations, hashes, raw per-seed records, and analysis code. The artifacts also include theorem-to-code and implementation diagnostics for the principal privacy and objective guarantees.

REFERENCES

David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, volume 7, pp. 1027–1035, 2007.

Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pp. 21–29, 2001.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pp. 464–473. IEEE, 2014.

Ayoub Belhadji, Rémi Bardenet, and Pierre Chainais. A determinantal point process for column subset selection. *Journal of machine learning research*, 21(197):1–62, 2020.

Davin Choo, Christoph Grunau, Julian Portmann, and Václav Rozhon. k-means++: Few more steps yield constant approximation. In *International Conference on Machine Learning*, pp. 1909–1917. PMLR, 2020.

Edith Cohen, Ofir Geri, Tamas Sarlos, and Uri Stemmer. Differentially private weighted sampling. In *International Conference on Artificial Intelligence and Statistics*, pp. 2404–2412. PMLR, 2021.

Amit Deshpande and Luis Rademacher. Efficient volume sampling for row/column subset selection. In *2010 ieee 51st annual symposium on foundations of computer science*, pp. 329–338. IEEE, 2010.

Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pp. 292–303. Springer, 2006.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3-4):211–487, 2014.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Alessandro Epasto, Tamalika Mukherjee, and Peilin Zhong. Differentially private clustering in data streams. *arXiv preprint arXiv:2307.07449*, 2023.

Chenglin Fan, Ping Li, and Xiaoyun Li. K-median clustering via metric embedding: towards better initialization with differential privacy. *Advances in Neural Information Processing Systems*, 36: 73817–73838, 2023.

Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Differentially private clustering: Tight approximation ratios. *Advances in Neural Information Processing Systems*, 33:4040–4054, 2020.

Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1106–1125. SIAM, 2010.

Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1207–1214. SIAM, 2012.

Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially-private queries through consistency. In *36th International Conference on Very Large Databases (VLDB)*. Citeseer, 2010.

Junyu Huang, Qilong Feng, Ziyun Huang, Jinhui Xu, and Jianxin Wang. Near-linear time approximation algorithms for k-means with outliers. In *Forty-first International Conference on Machine Learning*, 2024.

Lingxiao Huang, Shaofeng H-C Jiang, and Jianing Lou. The power of uniform sampling for k-median. In *International Conference on Machine Learning*, pp. 13933–13956. PMLR, 2023.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pp. 1376–1385. PMLR, 2015.

Max Dupré La Tour, Monika Henzinger, and David Saulpic. Making old things new: a unified algorithm for differentially private clustering. *arXiv preprint arXiv:2406.11649*, 2024.

Silvio Lattanzi and Christian Sohler. A better k-means++ algorithm via local search. In *International Conference on Machine Learning*, pp. 3662–3671. PMLR, 2019.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103. IEEE, 2007.

Uri Stemmer and Haim Kaplan. Differentially private k-means with constant multiplicative error. *Advances in Neural Information Processing Systems*, 31, 2018.

Grant Wang, Santosh Vempala, and Luis Rademacher. Matrix approximation and projective clustering via volume sampling. *Theory of Computing*, 2006.

A Proofs for Sampling with Grouped Weights

We first prove the threshold and stability statements, followed by the privacy and objective guarantees for the grouped weights. The public partition and its representatives are chosen before accessing w, as in Section 4. All sampling steps and private decisions use fresh randomness after the threshold output is generated.

A.1 Thresholding and Grouping the Weights

Proof of Lemmas 4.1 and 4.2. For $\zeta_x \sim \mathrm{Lap}(1/\epsilon_g)$, we have $\mathbb{P}[|\zeta_x| > \lambda] = e^{-\epsilon_g \lambda}$. A union bound gives $\mathbb{P}[\mathcal{E}_g^c] \le m e^{-\epsilon_g \lambda} \le \beta_g$. On $\mathcal{E}_g$, every location with $w_x \ge 3\lambda$ satisfies $w_x + \zeta_x \ge 2\lambda$ and therefore belongs to H. Conversely, if $x \in H$, then $w_x \ge 2\lambda - \zeta_x \ge \lambda$. Hence $\{x : w_x \ge 3\lambda\} \subseteq H \subseteq \{x : w_x \ge \lambda\}$, and $v_x = w_x$ for $x \in H$, which gives $v = w_H$.

For the stability statement, consider the same threshold outcome H for two neighboring inputs. It is enough to consider adding one record at a location x. If $x \in H$, the group totals are unchanged and only $v_x = \max\{w_x, \lambda\}$ can change. Its increase is at most one, so only the coordinate u_x can change, and its sampling weight is at least λ. If $x \notin H$, then x belongs to a unique group $\mathcal{C}_j$. Only t_j changes, by one, so only the coordinate u_{r_j} can change. Since $\max\{t_j + 1, \kappa\} - \max\{t_j, \kappa\} \in [0, 1]$, the change has magnitude at most one, and the sampling weight at r_j is at least κ. This remains true when $r_j \in H$, because the selected weight at r_j is unchanged and is added to the group contribution. Deleting one record gives the same bound on the magnitude of the change. □

A.2 Privacy of Sampling with the Grouped Weights

Proof of Theorem 4.3. Consider any H, z and two neighboring inputs. If the corresponding weight vectors are equal, the sampling distributions are identical. Otherwise, after exchanging the two inputs if necessary, write $u'_r = u_r + \delta$ for the unique affected coordinate, where $0 < \delta \le 1$, and let $t = u'_r / u_r$.

For each output y, separate the terms in Equation (3) according to whether $r \in A$. Then $Q_y(u; z) = A_y + u_r B_y$ for some $A_y, B_y \ge 0$ that are the same on the two neighboring inputs. Therefore $Q_y(u; z) \le Q_y(u'; z) \le t Q_y(u; z)$. The same inequalities hold for $Z(u) = \sum_y Q_y(u; z)$ and $Z(u')$. Whenever the sampling probability is positive, it follows that $1/t \le \mathbb{P}_{u'}[Y = y \mid H, z] / \mathbb{P}_u[Y = y \mid H, z] \le t$. A numerator is zero on one input exactly when it is zero on the other. Moreover, every u_i with $i \in K_H$ is positive. Since all coefficients in Equation (3) are nonnegative, $Z(u) = 0$ if and only if all coefficients that can contribute to the normalizer are zero. This condition does not depend on the neighboring weight vector, so $Z(u') = 0$ as well and both inputs return the same public symbol.

By Lemma 4.2, $t \le 1 + 1/\lambda$ when the record change occurs at a location in H, and $t \le 1 + 1/\kappa$ when it occurs at a location in $\mathcal{X} \setminus H$. Thus one sampling step is pure ϵ_0-DP. The bound holds for every previously released sequence z.

For adaptive sampling, apply the same bound to each step after the previous outputs have been specified. At most R sampling steps contribute privacy cost $R\epsilon_0$, and the additional private decisions contribute at most ϵ_a by assumption. Branches determined by public information, H, and previously released outputs add no privacy cost. If the number of sampling steps is determined from these quantities and is bounded by R, the output sequence can be padded with a fixed public symbol to length R. If a continuation or stopping decision reads another private quantity, its privacy cost is included in ϵ_a. Composing these steps with the release of H, which has privacy cost ϵ_g, gives Equation (4). The final output is post-processing of these released quantities. Finally, the calibration in Equation (5) gives $\lambda \ge \kappa$ and $R \log(1 + 1/\kappa) \le \epsilon_s$.

To prove the tightness statement in the main text, consider ordinary sampling on two public locations and let both public groups be singletons. When both locations belong to H, the neighboring sampling weights (λ, M) and $(\lambda + 1, M)$ give a likelihood ratio for selecting the first location that approaches $1 + 1/\lambda$ as $M \to \infty$. When $H = \varnothing$ and the two public groups are singletons, neighboring counts producing sampling weights (κ, M) and $(\kappa + 1, M)$ give a ratio approaching $1 + 1/\kappa$. Using the construction associated with the larger of the two bounds and repeating it independently R times, the event that the first location is selected in every repetition has likelihood ratio approaching $e^{R\epsilon_0}$.

Hence no smaller uniform privacy parameter can hold for all sampling rules of the form Equation (3). This statement concerns the privacy cost $R\epsilon_0$ of the sampling steps, not the threshold cost or the privacy of any particular final solution. □

The guarantee applies to the released threshold set, samples, and private decisions analyzed above. Releasing a group total or an objective value requires an additional privacy argument. Likewise, the partition must be chosen from public information, as assumed in the main text.

A.3 Comparing with the Original Objective

Proof of Equation (7) *and Theorem 4.4.* On $\mathcal{E}_g$, the selected weights equal their original values. For every $\theta \in \Theta$, expanding the two objectives gives

$$J_u(\theta) - J_w(\theta) = \sum_{x \notin H} w_x\big(\ell_{\pi(x)}(\theta) - \ell_x(\theta)\big) + \sum_{j \in I_H} (\kappa - t_j)_+ \ell_{r_j}(\theta).$$

The absolute value of the first sum is at most $D_H(w)$. The second sum is nonnegative and at most $B_{\mathrm{loss}} P_H(w)$. These two bounds give Equation (7). Taking the infimum in the second inequality of Equation (7) gives $\mathrm{OPT}(u) \le \mathrm{OPT}(w) + D_H(w) + B_{\mathrm{loss}} P_H(w)$.

On the event that the algorithm returns $\widehat{\theta}$ with $J_u(\widehat{\theta}) \le \alpha\,\mathrm{OPT}(u) + E$, we have

$$\begin{aligned} J_w(\widehat{\theta}) &\le J_u(\widehat{\theta}) + D_H(w) \\ &\le \alpha\,\mathrm{OPT}(u) + E + D_H(w) \\ &\le \alpha\,\mathrm{OPT}(w) + E + (\alpha+1) D_H(w) + \alpha B_{\mathrm{loss}} P_H(w). \end{aligned}$$

For every threshold outcome H, the algorithmic guarantee fails with probability at most β_a. Combining this with $\mathbb{P}(\mathcal{E}_g^c) \le \beta_g$ proves the theorem. □

The remaining proofs are organized by application, beginning with metric k-median and then row subset selection.

B Proofs for Metric k-Median

We first state the k-median algorithm and then prove the initialization, candidate, swap, privacy, and utility bounds used in Section 5.1. The separation result is proved at the end.

B.1 Algorithm and Parameters

The public round count is

$$\begin{aligned} \rho = 5/32, \qquad M_0 &= 4(1 + \mathsf{H}_{k-1})/\beta_i, \\ N_0 &= \lceil 16k \max\{0, \log(M_0/16)\} \rceil, \\ T &= \left\lceil \frac{2N_0 + 8\log(1/\beta_p)}{\rho} \right\rceil. \end{aligned} \tag{10}$$

The quantities in Equation (10) depend only on k and the public failure probabilities. As in Section 5.1, set $R = k + T$, $\epsilon_t = \epsilon_{\mathrm{sw}}/T$, and $b_{\mathrm{sw}} = (2\Delta T/\epsilon_{\mathrm{sw}}) \log((k+1)T/\beta_E)$, and let λ, κ be given by Equation (5). Algorithm 1 gives the resulting procedure.

Algorithm 1 Private sampled local search with grouped weights

Input: Public metric $(\mathcal{X}, d)$ with diameter Δ; public groups $\mathcal{C}_1, \dots, \mathcal{C}_K$ with representatives $r_1, \dots, r_K$; a fixed public order on $\mathcal{X}$; public parameters $k, T, \epsilon_g, \epsilon_t, \lambda, \kappa, b_{\mathrm{sw}}$; private weight vector w.

Output: A center set $C_T \subseteq \mathcal{X}$ with $|C_T| = k$.

Sample independent $\zeta_x \sim \mathrm{Lap}(1/\epsilon_g)$ and compute H and v by Equation (1).
Set $I_H = \{j : \mathcal{C}_j \setminus H \neq \varnothing\}$ and $t_j = \sum_{x \in \mathcal{C}_j \setminus H} w_x$ for $j \in I_H$.
Compute u and K_H by Equation (2).
Sample $c_1 \in K_H$ proportionally to u_{c_1} and set $C \leftarrow \{c_1\}$.
while $|C| < k$ and there exists $x \in K_H$ with $d(x, C) > 0$ **do**
    Sample $p \in K_H$ with probability $u_p d(p, C)/\Phi_u(C)$ and set $C \leftarrow C \cup \{p\}$.
end while
if $|C| < k$ **then**
    Add the first $k - |C|$ locations of $\mathcal{X} \setminus C$ in the fixed public order to C.
end if
$C_0 \leftarrow C$.
for $t = 0, \dots, T-1$ **do**
    if $d(x, C_t) = 0$ for all $x \in K_H$ **then**
        $C_{t+1} \leftarrow C_t$.
    else
        Sample $p \in K_H$ with probability $u_p d(p, C_t)/\Phi_u(C_t)$.
        For $q \in C_t$, set $U_t(q) = \Phi_u(C_t) - \Phi_u((C_t \setminus \{q\}) \cup \{p\})$.
        Set $U_t(\perp) = b_{\mathrm{sw}}$.
        Sample $a \in C_t \cup \{\perp\}$ proportionally to $\exp(\epsilon_t U_t(a)/(2\Delta))$.
        If $a = \perp$, set $C_{t+1} = C_t$; otherwise set $C_{t+1} = (C_t \setminus \{a\}) \cup \{p\}$.
    end if
end for
return C_T.

B.2 Initialization

We prove the initialization bound for any nonnegative weight vector u with positive total weight. The argument is the weighted metric analogue of the sampling analysis used in Arthur et al. (2007).

Let O be an optimal k-set, with public tie-breaking used to assign each positive-weight location to one center in O. For a nonempty optimal cluster A with center o, write $W_A = \sum_{x \in A} u_x$ and $B_A = \sum_{x \in A} u_x d(x, o)$. If $c \in A$ is sampled proportionally to u_c, then the triangle inequality gives $\mathbb{E}[\sum_{x \in A} u_x d(x, c)] \le 2B_A$.

Now let C be the current center set, write $D(x) = d(x, C)$ and $F_A = \sum_{x \in A} u_x D(x)$, and suppose $F_A > 0$. Conditional on a distance-weighted sample falling in A, the sampled location c has probability $u_c D(c)/F_A$. Let $L_c = \sum_{x \in A} u_x \min\{D(x), d(x, c)\}$ be the cost contributed by A after adding c. Averaging $D(c) \le D(x) + d(x, c)$ over $x \in A$ gives $D(c) \le F_A/W_A + W_A^{-1} \sum_{x \in A} u_x d(x, c)$. Therefore

$$\begin{aligned}\mathbb{E}L_c &\le \frac{1}{W_A} \sum_{c \in A} u_c L_c + \frac{1}{W_A F_A} \sum_{c \in A} u_c \Big(\sum_{x \in A} u_x d(x, c) \Big) L_c \\ &\le 4B_A.\end{aligned}$$

For the last inequality, use $L_c \le \sum_{x \in A} u_x d(x, c)$ in the first term and $L_c \le F_A$ in the second. In each case, the triangle inequality through o gives $\sum_{c, x \in A} u_c u_x d(x, c) \le 2W_A B_A$.

We next bound the effect of the remaining distance-weighted samples. Suppose $q \ge 1$ optimal clusters have not yet been sampled. Let their current total cost and optimal cost be F_U and B_U, and let the current cost of the already sampled optimal clusters be F_C. After $0 \le t \le q$ further distance-weighted samples, the expected cost is at most

$$(F_C + 4B_U)(1 + \mathsf{H}_t) + \frac{q - t}{q} F_U. \tag{11}$$

If $F_C + F_U = 0$, the claim is immediate, so suppose the current cost is positive. The claim then follows by induction on t. The case $t = 0$ is immediate. If $q = t = 1$, the next sample either falls in an already sampled optimal cluster, in which case the cost is at most $F_C + F_U$, or falls in the remaining cluster, whose expected new contribution is at most $4B_U$. Weighting these two cases by their sampling probabilities gives at most $F_C + F_U(F_C + 4B_U)/(F_C + F_U) \le 2(F_C + 4B_U)$, which is the right-hand side of Equation (11). Now suppose $q > 1$ and $t \ge 1$, and apply the induction hypothesis after the first new sample. If the sample falls in an already sampled optimal cluster, the uncovered part of the induction bound contributes at most $(q - t + 1)F_U/q$. If it falls in an uncovered cluster A, the bound $\mathbb{E}L_c \le 4B_A$ above shows that the same term is at most $(q-t)(F_U-F_A)/(q-1)$, where $F_A = \sum_{x\in A} u_x D(x)$. Averaging over the first sample and using $\sum_A F_A^2 \ge F_U^2/q$ bounds these terms by $(q - t)F_U/q + F_C F_U/[q(F_C + F_U)]$. Since $t \le q$, the second term is at most $(F_C + 4B_U)/t$, which is exactly the additional contribution from $\mathsf{H}_t = \mathsf{H}_{t-1} + 1/t$. This proves Equation (11). If all positive-weight optimal clusters have already been sampled, further samples can only decrease the cost. If the current cost is zero, adding centers in the fixed public order preserves zero cost.

Let A be the optimal cluster containing the first sampled center. If there are $k' \le k$ positive-weight optimal clusters, apply Equation (11) with $q = t = k' - 1$. Conditional on the first sample falling in A, the expected contribution of A is at most $2B_A$, while the remaining optimal clusters have total optimal cost $B - B_A$, where $B = \mathrm{OPT}_k(u)$. Taking expectation over the first sample gives $\mathbb{E}\Phi_u(C_0) \le 4(1 + \mathsf{H}_{k-1})\,\mathrm{OPT}_k(u)$. For $k' = 1$, the first-sample bound alone is sufficient. Markov's inequality then gives $\Phi_u(C_0) \le M_0\,\mathrm{OPT}_k(u)$ except with probability β_i. If $\mathrm{OPT}_k(u) = 0$, nonnegativity implies $\Phi_u(C_0) = 0$ almost surely.

B.3 Candidate and Swap Analysis

Proof of Lemma 5.1. Let C be the current center set and let $o_1, \dots, o_k$ be optimal centers. Write X_i for the cluster of o_i, $A_i = \sum_{x\in X_i} u_x d(x, C)$, and $B_i = \sum_{x\in X_i} u_x d(x, o_i)$. Then $F = \sum_i A_i$ and $B = \sum_i B_i = \mathrm{OPT}_k(u)$.

Map each o_i to a nearest center $\phi(o_i) \in C$. For each i, let $q_i \in C$ be a center such that every current center appears as some q_i at most twice and $q_i \ne \phi(o_j)$ whenever $j \ne i$. Such a choice is possible as follows. A current center that is the image of exactly one optimal center is assigned to that optimal center. Suppose s remaining current centers are the images of r optimal centers, with at least two optimal centers mapped to each of the s centers. There are then $r - s$ current centers with no optimal center mapped to them. Since $r \ge 2s$, assigning at most two of the remaining r optimal centers to each of these $r - s$ centers is sufficient.

For each i, let $R_i = 2\sum_{x\notin X_i:\, x \text{ served by } q_i} u_x d(x, o(x))$, where $o(x)$ is the optimal center serving x. Because each current center is used as some q_i at most twice, $\sum_i R_i \le 4B$. For any $p \in X_i$, assign the points of X_i to p and reassign every other point previously served by q_i to $\phi(o(x))$. This gives

$$\Phi_u((C \setminus \{q_i\}) \cup \{p\}) \le F - A_i + \sum_{x\in X_i} u_x d(x, p) + R_i.$$

Indeed, if $x \notin X_i$ was served by q_i, then $\phi(o(x)) \ne q_i$ and remains in the center set. Writing $a = d(x, C)$ and $b = d(x, o(x))$, the definition of $\phi(o(x))$ gives $d(o(x), \phi(o(x))) \le d(o(x), q_i) \le a + b$, so $d(x, \phi(o(x))) \le a + 2b$.

Let $W_i = \sum_{x\in X_i} u_x$ and $P_i = \{p \in X_i : W_i d(p, o_i) \le 4B_i\}$. Every $p \in P_i$ satisfies $\sum_{x\in X_i} u_x d(x, p) \le 5B_i$. If $A_i > 5B_i$, more than half of A_i is contributed by locations in P_i. When $B_i = 0$, this is immediate. Otherwise, the total u-weight outside P_i is less than $W_i/4$. If $\delta_i = d(o_i, C)$, then $W_i\delta_i \le A_i + B_i$, and hence $\sum_{p\in X_i\setminus P_i} u_p d(p, C) \le B_i + (A_i + B_i)/4 < A_i/2$.

Consider the indices i for which $A_i - R_i - 5B_i > F/(8k)$. The sum of A_i over all remaining indices is at most $\sum_i R_i + 5B + F/8 \le 9B + F/8$. Since $F > 16B$, the indices satisfying the displayed condition account for more than $5F/16$ of the total weight used for distance-weighted sampling. For each such i, more than half of A_i is contributed by locations in P_i. Therefore the candidate lies in one of the corresponding sets P_i with probability greater than $5/32$. The preceding swap bound then gives gain greater than $F/(8k)$. Any candidate sampled with positive probability has $d(p, C) > 0$, so it is not already in C and the swap keeps exactly k centers. □

Proof of Theorem 5.2. For privacy, consider any threshold outcome H. Each sampling step has privacy cost at most ϵ_0 by Theorem 4.3. For a given center set, candidate p, and current center q, the gain is $G(q) = \sum_x u_x[d(x, C) - d(x, (C \setminus \{q\}) \cup \{p\})]$. The coefficient of every u_x lies in $[-\Delta, \Delta]$. By Lemma 4.2, neighboring inputs therefore change $G(q)$ by at most Δ. The exponential mechanism has privacy cost ϵ_t in each round. There are at most $k+T$ sampling steps and the T swap choices use total privacy budget ϵ_{sw}. The test $d(x, C) = 0$ for every $x \in K_H$ depends only on H, the public metric, and the sampled centers. Adding centers in the fixed public order and releasing the final center set are post-processing. Combining these costs with the threshold release gives pure ϵ-DP.

For utility, let $F_t = \Phi_u(C_t)$ and $B = \mathrm{OPT}_k(u)$. Let D_t be the event that the exponential mechanism in round t returns an action whose utility is within b_{sw} of the largest utility in that round. A round in which the algorithm does not sample a candidate satisfies this event automatically. By the definition of b_{sw}, $\mathbb{P}(D_t^c) \le \beta_E/T$, so $\mathcal{E}_E = \bigcap_t D_t$ has probability at least $1 - \beta_E$. On D_t, the selected utility is nonnegative because $U(\perp) = b_{\mathrm{sw}}$. Thus every real swap has nonnegative gain, and F_t is nonincreasing on $\mathcal{E}_E$.

Set $L = \max\{16B, 16kb_{\mathrm{sw}}\}$. If $F_t > L$, then by Lemma 5.1, with probability at least $\rho = 5/32$ the candidate permits a real swap with gain greater than $F_t/(8k)$. Since $F_t/(8k) > 2b_{\mathrm{sw}}$, on D_t the selected action cannot be $\perp$ and the chosen swap has gain greater than $F_t/(16k)$. Hence $F_{t+1} \le (1 - 1/(16k))F_t$.

To count these candidate events, define $Y_t = 1$ when $F_t \le L$, and when $F_t > L$ let Y_t be the indicator that the candidate permits a swap with gain greater than $F_t/(8k)$. Given all outputs before the candidate is sampled, $\mathbb{P}(Y_t = 1) \ge \rho$. Therefore, for $\tau > 0$, $\mathbb{E}[e^{-\tau \sum_t Y_t}] \le (1 - \rho + \rho e^{-\tau})^T$. The Chernoff bound gives $\mathbb{P}[\sum_t Y_t < \rho T/2] \le e^{-\rho T/8}$. By Equation (10), $\rho T \ge 2N_0 + 8\log(1/\beta_p)$, and hence $\mathbb{P}[\sum_t Y_t < N_0] \le \beta_p$.

Now intersect this event with $\mathcal{E}_E$ and the initialization event $F_0 \le M_0 B$. If $F_T > L$, then monotonicity on $\mathcal{E}_E$ implies $F_t > L$ in every round, so every round with $Y_t = 1$ gives the multiplicative decrease above. At least N_0 such rounds would imply

$$\begin{aligned} F_T &\le M_0 B(1 - 1/(16k))^{N_0} \\ &\le M_0 B e^{-N_0/(16k)} \le 16B \le L, \end{aligned}$$

which is a contradiction. If $N_0 = 0$, the initialization event already gives $F_0 \le 16B$; if $B = 0$, it gives $F_0 = 0$. Therefore $\Phi_u(C_T) \le \max\{16\,\mathrm{OPT}_k(u), 16kb_{\mathrm{sw}}\}$ except with probability at most $\beta_i + \beta_E + \beta_p$.

On $\mathcal{E}_g$, Equation (7) gives $\Phi_w(C_T) \le \Phi_u(C_T) + T_H(w)$ and $\mathrm{OPT}_k(u) \le \mathrm{OPT}_k(w) + T_H(w) + \Delta P_H(w)$. Writing $D = T_H(w)$ and $P = P_H(w)$,

$$\begin{aligned} \Phi_w(C_T) &\le \max\{16\,\mathrm{OPT}_k(u), 16kb_{\mathrm{sw}}\} + D \\ &\le \max\{16\,\mathrm{OPT}_k(w) + 16D + 16\Delta P, 16kb_{\mathrm{sw}}\} + D \\ &\le \max\{16\,\mathrm{OPT}_k(w), 16kb_{\mathrm{sw}}\} + 17D + 16\Delta P. \end{aligned}$$

The total failure probability is at most $\beta_g + \beta_i + \beta_E + \beta_p \le \beta$, which proves Equation (8). □

B.4 Separation Result

We first record the one-center bound used below. For nonzero nonnegative weights u, let Y be sampled proportionally to u_x. If c^* is an optimal public center, then

$$\begin{aligned} \mathbb{E}\Phi_u(Y) &= \sum_{x,y} \frac{u_x u_y}{\sum_z u_z} d(x, y) \\ &\le \sum_{x,y} \frac{u_x u_y}{\sum_z u_z} \big(d(x, c^*) + d(c^*, y)\big) = 2\,\mathrm{OPT}_1(u). \end{aligned}$$

Together with Equation (7), this implies that for every threshold outcome in $\mathcal{E}_g$,

$$\mathbb{E}[\Phi_w(Y) \mid H, \mathcal{E}_g] \le 2\,\mathrm{OPT}_1(w) + 3T_H(w) + 2\Delta P_H(w). \tag{12}$$

The expectation is only over the fresh sampling randomness. The total sampling weight is positive because every public location is either selected or belongs to a group represented in K_H.

Theorem B.1. *Let $A_0 \geq 1$, $N \geq 2$, $\Delta > 0$, and let $\epsilon_g, \epsilon_s > 0$ and $\beta_g \in (0,1)$. For the family in Section 5.2, use Equation* (5) *with $m = 1 + Nq$ and $R = 1$. Suppose $\lambda \geq 2$, $M \geq \max\{4A_0B, \kappa\}$, and $0 < \delta \leq \Delta/(4A_0)$. Then $L_{3\lambda}(w^{(j)}) = M$ for every j. Every algorithm that, after thresholding, uses only (H, w_H) and fresh randomness to return one public center has some input j such that*

$$\mathbb{P}\left[\Phi_{w^{(j)}}(\widehat{c}) \geq A_0 \operatorname{OPT}_1(w^{(j)}) + \frac{\Delta M}{2}\right] \geq (1-\beta_g)(1-1/N).$$

On the same family, form u from the public groups and sample one center proportionally to u_x. This algorithm is pure $(\epsilon_g + \epsilon_s)$-DP and, for every j,

$$\mathbb{E}\Phi_{w^{(j)}}(\widehat{c}) \leq 2\operatorname{OPT}_1(w^{(j)}) + 3M\delta + 2\Delta(N-1)\kappa + \beta_g n\Delta. \tag{13}$$

For fixed $N, \epsilon_g, \epsilon_s, \Delta$, take $\beta_g = q^{-2}$ and $\delta = \Delta/q^2$. As $q \to \infty$, the family can be chosen so that $\operatorname{OPT}_1(w^{(j)}) = o(\Delta M)$ and the right-hand side of Equation (13) *is $o(\Delta M)$, uniformly over j.*

Proof of Theorems 5.3 and B.1. The public metric consists of one location o and N disjoint groups $B_1, \ldots, B_N$, each with q locations. Distinct locations in the same group are at distance δ, and all other distinct pairs are at distance Δ. Under the stated assumption $\delta \leq \Delta/(4A_0) \leq \Delta$, the triangle inequality holds. Choose a representative $r_j \in B_j$ for each group and use the public partition $\{o\}, B_1, \ldots, B_N$, with o representing its singleton group.

Let $a = \lambda - 1$, $B = 3\lambda$, and $M = qa$. On input j, the weight at o is B, every location in B_j has weight a, and all other weights are zero. On $\mathcal{E}_g$, the selected set is exactly $H = \{o\}$: the location o passes the threshold, while every location of weight $\lambda - 1$ or zero remains below it. Hence all inputs give the same pair (H, w_H) and $L_{3\lambda}(w^{(j)}) = M$.

Consider any algorithm that uses only (H, w_H) after thresholding. Let p_j be the probability, given $\mathcal{E}_g$, that its output lies in B_j. This output distribution is the same for all input indices j, and the groups are disjoint, so $\sum_j p_j \leq 1$. Thus some j satisfies $p_j \leq 1/N$. On input j, every center outside B_j has cost at least $M\Delta$, while a center in B_j gives $\operatorname{OPT}_1(w^{(j)}) \leq B\Delta + M\delta$. The conditions $M \geq 4A_0B$ and $\delta \leq \Delta/(4A_0)$ imply $A_0 \operatorname{OPT}_1(w^{(j)}) \leq M\Delta/2$. Therefore the lower-bound event in the theorem has probability at least $1 - 1/N$ on $\mathcal{E}_g$, and hence at least $(1-\beta_g)(1-1/N)$ without conditioning on $\mathcal{E}_g$.

For grouped sampling, $R = 1$ and $\kappa = \lceil \max\{1, (e^{\epsilon_s} - 1)^{-1}\} \rceil$. On $\mathcal{E}_g$, the singleton group $\{o\}$ does not belong to I_H. The group B_j has total weight $M \geq \kappa$, while every other group has total weight zero. Thus $u_o = B$, $u_{r_j} = M$, $u_{r_i} = \kappa$ for $i \neq j$, $P_H(w) = (N-1)\kappa$, and $T_H(w) \leq M\delta$. Applying Equation (12) on $\mathcal{E}_g$ and using $\Phi_w(Y) \leq n\Delta$ on $\mathcal{E}_g^c$ gives Equation (13). Privacy follows from Theorem 4.3.

For the asymptotic statement, keep $N, \epsilon_g, \epsilon_s, \Delta$ fixed and set $m_q = 1 + Nq$, $\beta_g = q^{-2}$, $\delta_q = \Delta/q^2$, and $\lambda_q = \lceil \max\{\kappa, \epsilon_g^{-1} \log(m_q q^2)\} \rceil$. For sufficiently large q, the assumptions of Theorem B.1 hold. Moreover, $B/M \leq 6/q$, $\delta_q/\Delta = q^{-2}$, $(N-1)\kappa/M \to 0$, and $\beta_g n/M \to 0$. Therefore both $\operatorname{OPT}_1(w^{(j)})$ and the right-hand side of Equation (13) are $o(\Delta M)$ uniformly over j. Finally, on $\mathcal{E}_g$, $\sum_{x \notin H} w_x d(x, H) = M\Delta$ while $T_H(w) \leq M\delta$. This is the comparison stated in the main text. □

C Proofs for Row Subset Selection

We first state the row subset selection algorithm and then prove the sampling, privacy, and original-weight bounds used in Section 5.3.

C.1 Algorithm and Parameters

Set $\ell = 2r - 1$, $b = \lceil 16r/\eta \rceil$, and $s = \lceil \log_2(1/\beta_s) \rceil$. Each repetition uses one joint volume sample and at most b residual samples, so set $R = s(1+b)$ and let λ, κ be given by Equation (5). The tests $\operatorname{rank}(A_{K_H}) < \ell$ and $\operatorname{dist}(a_i, Q_j) = 0$ for all $i \in K_H$ depend only on H, the public rows, and previously sampled indices.

Algorithm 2 Private row subset selection with grouped weights

Input: Public matrix A; public groups $\mathcal{C}_1, \dots, \mathcal{C}_K$ of the row indices with representatives $r_1, \dots, r_K$; a fixed public order on the row indices; public parameters $r, \eta, \beta_s, \epsilon_g, \lambda, \kappa$; private weight vector w.

Output: A set S of row indices.

Set $\ell \leftarrow 2r-1$, $b \leftarrow \lceil 16r/\eta \rceil$, and $s \leftarrow \lceil \log_2(1/\beta_s) \rceil$.
Sample independent $\zeta_i \sim \mathrm{Lap}(1/\epsilon_g)$ and compute H and v by Equation (1).
Set $I_H = \{j : \mathcal{C}_j \setminus H \neq \varnothing\}$ and $t_j = \sum_{i \in \mathcal{C}_j \setminus H} w_i$ for $j \in I_H$.
Compute u and K_H by Equation (2).
if $\mathrm{rank}(A_{K_H}) < \ell$ **then**
    return the indices of a row basis of A_{K_H} chosen in the fixed public order.
end if
$S \leftarrow \varnothing$.
for $j = 1, \dots, s$ **do**
    Sample $V_j \subseteq K_H$, $|V_j| = \ell$, proportionally to $\det(A_{V_j} A_{V_j}^\top) \prod_{i \in V_j} u_i$.
    Set $Q_j = \mathrm{span}\{a_i : i \in V_j\}$ and $S \leftarrow S \cup V_j$.
    Set $e_i \leftarrow \mathrm{dist}(a_i, Q_j)^2$ for $i \in K_H$.
    if there exists $i \in K_H$ with $e_i > 0$ **then**
        for $h = 1, \dots, b$ **do**
            Sample $i_{j,h} \in K_H$ with probability $u_{i_{j,h}} e_{i_{j,h}} / \sum_{i \in K_H} u_i e_i$.
            $S \leftarrow S \cup \{i_{j,h}\}$.
        end for
    end if
end for
return S.

C.2 Sampling Bounds

Let $B = \mathrm{diag}((\sqrt{u_i})_{i \in K_H}) A_{K_H}$, and let $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$ be the eigenvalues of $B^\top B$. Then $\mathrm{OPT}_r(u) = \sum_{j>r} \sigma_j$. For $V(S) = \det(B_S B_S^\top)$ and $Z_\ell = \sum_{|S|=\ell} V(S)$, the expansion of $\det(I + tBB^\top)$ gives $Z_\ell = e_\ell(\sigma)$, where e_j is the sum of products over all j-element subsets of the eigenvalues.

If P_S denotes the orthogonal projector onto the span of the selected rows, the Gram determinant identity gives

$$V(S) \, \|B(I - P_S)\|_F^2 = \sum_{i \notin S} V(S \cup \{i\}).$$

When the selected rows are dependent, both sides are zero. Summing over all S with $|S| = \ell$ and counting each $(\ell+1)$-set $\ell+1$ times gives $\mathbb{E} \, \|B(I - P_S)\|_F^2 = (\ell+1) e_{\ell+1}(\sigma) / e_\ell(\sigma)$. Each product in $e_{\ell+1}(\sigma)$ contains at least $\ell+1-r$ eigenvalues with index larger than r. Expanding $(\sum_{j>r} \sigma_j) e_\ell(\sigma)$ and discarding the nonnegative terms that repeat an index therefore gives $(\ell + 1 - r) e_{\ell+1}(\sigma) \leq (\sum_{j>r} \sigma_j) e_\ell(\sigma)$. With $\ell = 2r - 1 \leq \mathrm{rank}(B)$, the expected projection error after the joint volume sample is at most $2 \, \mathrm{OPT}_r(u)$.

Now let Q_0 be the span after a joint volume sample, with projector P_0, and write $E = B(I - P_0)$ with rows $e_i^\top$ and $D = \|E\|_F^2$. If $D > 0$, the residual-sampling probabilities are $p_i = \|e_i\|_2^2 / D$. Let $h_1, \dots, h_{r'}$ be leading left singular vectors of B, where $r' = \min\{r, \mathrm{rank}(B)\}$. For the independent sampled indices $I_1, \dots, I_b$, define

$$z_j = P_0 B^\top h_j + \frac{1}{b} \sum_{t=1}^{b} \frac{h_j(I_t)}{p_{I_t}} e_{I_t}.$$

Each z_j belongs to the span of Q_0 and the sampled rows. Moreover, $\mathbb{E} z_j = B^\top h_j$ and $\mathbb{E} \left\| z_j - B^\top h_j \right\|_2^2 \leq D/b$. The variance bound follows from independence and $\sum_{i : p_i > 0} h_j(i)^2 \leq 1$.

Set $\widehat{B} = \sum_j h_j z_j^\top$. This matrix has rank at most r and its row space lies in the span of Q_0 and the sampled rows. Orthogonality between the leading left singular subspace and its complement gives

$$\left\|B - \widehat{B}\right\|_F^2 = \mathrm{OPT}_r(u) + \sum_j \left\|B^\top h_j - z_j\right\|_2^2.$$

Projecting B onto the row space of $\widehat{B}$ can only reduce the error. Hence, conditional on Q_0, the expected excess over $\mathrm{OPT}_r(u)$ after the b residual samples is at most $(r/b)D$. If $D = 0$, all rows already lie in Q_0 and no residual sampling is needed.

Combining this bound with the joint volume-sampling bound, the nonnegative excess X after one repetition satisfies $\mathbb{E}X \le (2r/b)\,\mathrm{OPT}_r(u) \le (\eta/8)\,\mathrm{OPT}_r(u)$. If $\mathrm{OPT}_r(u) > 0$, Markov's inequality gives $\mathbb{P}[X \le (\eta/4)\,\mathrm{OPT}_r(u)] \ge 1/2$. If $\mathrm{OPT}_r(u) = 0$, then $X = 0$ almost surely. The repetitions are independent for a given H and u, and the final union contains the span produced by every repetition. Therefore, with probability at least $1 - 2^{-s}$,

$$\mathrm{RSS}_r(u; S) \le (1 + \eta/4)\,\mathrm{OPT}_r(u). \tag{14}$$

If $\mathrm{rank}(A_{K_H}) < \ell$, the returned row basis spans all rows indexed by K_H. Since the positive weights are supported on K_H, this span contains an optimal rank-at-most-r subspace, so the same bound holds without sampling.

C.3 Privacy and Original-Weight Bound

Proof of Theorem 5.4. For privacy, the joint volume-sampling probability has numerator $\det(A_S A_S^\top) \prod_{i \in S} u_i$, so it satisfies Equation (3). Since $u_i > 0$ for every $i \in K_H$, scaling row a_i by $\sqrt{u_i}$ does not change the rank. Hence the volume-sampling normalizer is positive exactly when $\mathrm{rank}(A_{K_H}) \ge \ell$. The algorithm tests this condition using only H and the public rows. After the joint sample, each residual sample has probability proportional to $u_i\,\mathrm{dist}(a_i, Q_0)^2$, where Q_0 is determined by the sampled indices and the public rows. The test that all residuals are zero is therefore also determined by H, the sampled indices, and public information. Each repetition uses one joint volume sample and at most b residual samples, so there are at most $s(1 + b) = R$ sampling steps. By Theorem 4.3 and the calibration in Equation (5), the released union of sampled indices is pure ϵ-DP.

For utility, consider a threshold outcome in $\mathcal{E}_g$. For every linear subspace L and $\gamma > 0$, Young's inequality gives the pointwise bound stated in the main text. Summing over the locations outside H and using the grouped weights gives $\Phi_w(L) \le (1 + \gamma)\Phi_u(L) + (1 + 1/\gamma)T_{2,H}(w)$. In the other direction, moving the locations outside H to their representatives and then raising each group total to at least κ gives $\Phi_u(L) \le (1 + \gamma)\Phi_w(L) + (1 + 1/\gamma)T_{2,H}(w) + \mathcal{R}^2 P_H(w)$. The last term uses $\mathrm{dist}(a_{r_j}, L)^2 \le \|a_{r_j}\|_2^2 \le \mathcal{R}^2$. Taking the minimum over rank-at-most-r subspaces in the second inequality yields $\mathrm{OPT}_r(u) \le (1 + \gamma)\,\mathrm{OPT}_r(w) + (1 + 1/\gamma)T_{2,H}(w) + \mathcal{R}^2 P_H(w)$.

On the event in Equation (14), let $\alpha_0 = 1 + \eta/4$ and let L_S be a rank-at-most-r subspace in the span of S attaining $\mathrm{RSS}_r(u; S)$. Applying the first comparison above to L_S and then the bound on $\mathrm{OPT}_r(u)$ gives

$$\begin{aligned}\mathrm{RSS}_r(w; S) \le\ & \alpha_0(1 + \gamma)^2\,\mathrm{OPT}_r(w) \\ & + (1 + 1/\gamma)\big[\alpha_0(1 + \gamma) + 1\big]T_{2,H}(w) \\ & + \alpha_0(1 + \gamma)\mathcal{R}^2 P_H(w).\end{aligned}$$

Set $\gamma = \eta/4$. For $0 < \eta \le 1$, $(1 + \eta/4)^3 \le 1 + \eta$, $(1 + 4/\eta)[(1 + \eta/4)^2 + 1] \le 13/\eta$, and $(1 + \eta/4)^2 \le 2$. These inequalities give Equation (9). Combining the failure probabilities β_g and β_s gives the stated probability. The union contains at most $s(2r - 1 + b)$ distinct indices, while the case $\mathrm{rank}(A_{K_H}) < \ell$ returns at most $\ell - 1$ indices. □

D Experimental Details and Additional Results

D.1 Public data and count construction

Table 1 lists the benchmark dictionaries. Skin and Shuttle belong to benchmark families used in nonprivate clustering with outliers (Huang et al., 2024); Census appears in nonprivate k-median experiments (Huang et al., 2023). We download these metric inputs from UCI and use Euclidean metric k-median, without an outlier objective. Colon and Leukemia are repository versions of the expression-data families used for subset selection (Belhadji et al., 2020). USPS adds a larger public feature dimension. Its labels are unused.

Table 1: Public benchmark inputs. All counts in this table are synthetic integer multiplicities, independently assigned to the public rows. The expression matrices are transposed to row selection.

Task	Dictionary	m	d	Source
k-median	Skin	4,096–51,433	3	UCI
	Shuttle	4,096–32,768	9	UCI, training split
	Census1990	4,096–131,072	68	UCI
RSS	Colon	2,000	62	Feature Selection collection
	Leukemia (ALLAML)	7,129	72	Feature Selection collection
	USPS	9,298	256	Feature Selection collection

The metric dictionaries drop labels and identifiers, scale each feature using its public full-file minimum and maximum, and remove duplicate sites. Fixed public seeds define nested subsets: 72101 for Census and 72102 for Skin and Shuttle. Census uses a random-key reservoir over the original 2,458,285 records. We divide distances by the exact diameter of each selected public dictionary. This cached computation depends only on public features. Per-run distances to proposed centers are included in the algorithm timer.

The RSS inputs are downloaded from the Feature Selection collection. Colon has 2,000 genes and 62 samples; this repository version uses discretized expression values. Leukemia uses `ALLAML.mat`, with 7,129 genes and 72 samples, rather than the separate 7,070-gene file. USPS uses its original 256 pixel features. We divide each matrix by a single maximum public row norm and leave it uncentered. A global scale preserves the optimization problem up to objective scaling. No feature transformation is fitted using private counts.

Synthetic counts allocate exactly n records to probabilities proportional to $i^{-\alpha}$ for $\alpha \in \{0, 1, 2\}$, using largest remainders, then independently permute frequency ranks across the dictionary. Thus the public vectors are real benchmark data while the multiplicities are controlled inputs. The count total is used to construct and evaluate an instance; it does not select algorithm parameters during a release. Raw-source and transformed-array hashes are stored in the dataset registries.

D.2 Algorithms, budgets, and timing

Benchmark parameters. We use the calibration in Equation (5). For k-median, we set $\epsilon_g = 0.2\epsilon$, $\epsilon_s = 0.2\epsilon$, and $\epsilon_{\mathrm{sw}} = 0.6\epsilon$; for RSS, we set $\epsilon_g = \epsilon_s = \epsilon/2$. We use $\beta_g = \beta_E = 0.01$. The practical k-median schedule has $k = 8$ and $T = 64$. RSS uses $r = 2$, $\eta = 1$, and $b = \lceil 16r/\eta \rceil = 32$. Colon and Leukemia use one repetition, giving row cap 35; USPS uses four, giving row cap 140. These caps keep the outputs below the rank of the public row space. Additional recalibrated parameter sweeps are given in Appendix D.8. Every comparator within a setting has the same row cap.

The primary implementation follows Equation (5); we refer to this grouped-weight method as GW. With the public bound R on the number of sampling steps, it sets $\kappa = \lceil \max\{1, 1/(e^{\epsilon_s/R} - 1)\} \rceil$ and $\lambda = \lceil \max\{\kappa, \log(m/\beta_g)/\epsilon_g\} \rceil$. Every group in I_H is represented, including groups with $t_j = 0$. When r_j itself belongs to H, the selected-weight contribution and the group contribution are added at the same coordinate, as in Equation (2). GW uses the grouped weights u inside the algorithm and does not release them. Sampling and private swap decisions use fresh randomness after thresholding.

Farthest-point grouping starts with the public point nearest the public mean, repeatedly takes the point farthest from the current representatives, and assigns every point to its nearest representative. Each

representative belongs to its own group. The random-group control fixes K original representatives and a balanced random assignment using public seed 93017. GROUP-ALL moves every count to its group representative, including large counts; it retains the same privacy allocation and solver. We use this variant to assess the effect of keeping selected weights at their original locations.

HIST adds independent $\mathrm{Lap}(1/\epsilon)$ noise to each original count and keeps noisy counts at least $2\log(m/\beta_g)/\epsilon$. HIST-CLIP clips the same noisy histogram at zero. HIST-GROUP first releases the same threshold outcome H as GW. It then aggregates the counts at locations in $\mathcal{X} \setminus H$ at their public representatives, retains the counts at locations in H, adds independent $\mathrm{Lap}(1/(\epsilon - \epsilon_g))$ noise on the support determined by H and the public partition, and clips at zero. For any threshold outcome H, a unit count change affects at most one aggregate by one, so this histogram has ℓ_1 sensitivity one after the thresholding step. Its subsequent optimizer is post-processing. HIST-GROUP does not apply the lower bound κ in Equation (2), so P_H is not reported for this baseline.

TG denotes the threshold-only variant that uses the selected weights after thresholding, and NP denotes the nonprivate reference. All k-median methods use the same distance implementation, initialization rule, candidate-sampling rule, and public round cap. GW and TG retain the private no-op utility and exponential-mechanism decisions. Histogram methods and NP use greedy decisions on their available weights. Thus a comparison with histograms includes the cost of private decisions under GW; it does not force the histogram to spend privacy twice. NP is a heuristic quality reference, not a certified metric optimum. The measured $T = 64$ schedule satisfies the stated privacy accounting but is shorter than the schedule required by the high-probability approximation theorem. We do not claim that theorem's utility guarantee for these short runs.

Each timer starts before thresholding or histogram construction and ends after the output indices are produced. It includes group summation, sampling, optimization, and private decisions where applicable. It excludes shared public preprocessing, warm-up, synthetic-count generation, and offline objective evaluation. Methods execute in randomized order within each paired seed, serially with one BLAS thread, on Windows 11 with an Intel Core i5-12600KF and Python 3.12.7. Public preprocessing is timed separately and is not included in the per-call algorithm times.

D.3 Partition geometry and group granularity

The farthest-point partition uses a greedy covering-radius construction. The alternative fits public MiniBatch k-means with $K = 128$, seed 93031, three initializations, batch size 2,048, and at most 100 iterations. Each cluster center is replaced by the nearest original row in that cluster, and all rows are reassigned to their nearest selected row. Empty clusters, if any, are filled by farthest public points. Neither partition uses counts or measured private cost. This comparison is reported separately from the primary grid, using fresh seeds 78000–78029 on all six dictionaries under both uniform and Zipf-1 counts, for 1,440 executions. Figure 18 reports both public partitions for all six datasets. The supplementary data also record T_H/n for k-median and $T_{2,H}/n$ for RSS under each partition, with pointwise confidence intervals for the cost difference.

Figure 3 varies $K \in \{8, 32, 128, 512\}$. The quantity T_H/n for k-median, or $T_{2,H}/n$ for RSS, measures the contribution from replacing locations by their representatives. The quantity P_H/n measures the weight added when group totals below κ are raised to κ. These are the terms appearing in the bounds that compare the grouped and original objectives; they are not measured excess loss. The random partition and GROUP-ALL controls distinguish the roles of public geometry and retaining selected weights at their original locations. The displayed dictionaries are Skin and USPS; every dataset's sweep is retained in the source tables. Random groups were evaluated at $K = 8, 32, 512$; no $K = 128$ random-group point is interpolated.

D.4 RSS accounting and output sizes

RSS uses $\ell = 2r - 1 = 3$ volume rows and $b = \lceil 16r/\eta \rceil = 32$ residual draws per repetition. Each residual batch uses the span Q_0 from that repetition's joint volume sample, as in Section 5.3. GW counts the joint volume sample as one sampling step, giving $R = s(1 + b)$. GW-SEQ uses the same joint sampling rule but counts each volume row separately, giving $R = s(\ell + b)$, and recalibrates λ and κ. This change in calibration can alter H, u, and therefore the sampling distribution. The output cap remains $s(\ell + b)$ for both. One repetition on Colon and Leukemia corresponds to $\beta_s = 1/2$; four

Table 2: Paired public-partition comparisons. Cost change is $100(\bar{c}_{\text{k-means}}/\bar{c}_{\text{farthest}}-1)$ with a pointwise 95% interval; negative values favor public k-means groups. The last two columns report T_H/n for the metric data and $T_{2,H}/n$ for RSS. The different task-specific units are not compared across rows.

Dictionary	Counts	Cost change (%)	Farthest-point groups	Public k-means groups
Skin	Zipf-1	-2.18 [-3.06, -1.24]	0.0188	0.0105
Skin	Uniform	-6.39 [-6.93, -5.84]	0.0471	0.0264
Shuttle	Zipf-1	-2.87 [-4.53, -1.36]	0.0060	0.0025
Shuttle	Uniform	-8.70 [-9.58, -7.77]	0.0150	0.0063
Census	Zipf-1	-6.74 [-7.84, -5.61]	0.1356	0.0757
Census	Uniform	-22.66 [-22.82, -22.49]	0.3406	0.1903
Colon	Zipf-1	-0.40 [-0.79, -0.10]	0.1282	0.0916
Colon	Uniform	-1.62 [-1.88, -1.39]	0.4402	0.3152
Leukemia	Zipf-1	-0.35 [-0.56, -0.15]	0.3080	0.2285
Leukemia	Uniform	-4.05 [-4.38, -3.69]	1.1541	0.8559
USPS	Zipf-1	-0.02 [-0.03, -0.02]	0.0997	0.0485
USPS	Uniform	-1.21 [-1.37, -1.07]	0.2460	0.1196

Table 3: Cached public setup times in seconds, $K = 128$. Each partition time is one actual construction, not a median over algorithm seeds. The diameter computation is shared by all metric methods. These setup costs are separate from every per-call timing figure.

Dictionary	Metric diameter	Farthest-point groups	Public k-means groups
Skin	1.713	0.103	0.178
Shuttle	3.471	0.221	0.115
Census	13.518	1.205	0.344
Colon	—	0.027	0.122
Leukemia	—	0.281	0.290
USPS	—	1.207	0.812

on USPS correspond to $\beta_s = 1/16$. The different repetition counts prevent the smaller expression matrices from being saturated by the output budget.

Public QR returns a fixed public set for a fixed dictionary; its recorded time measures factorization and selection, and that output can be reused across count vectors. Public QR selects pivot columns of $A^\top$ and public uniform samples distinct original rows without replacement. The weighted samplers can repeat indices and therefore use fewer distinct rows. All released sets are evaluated using the best rank-two reconstruction in their span, computed offline with the original weights. The fitted reconstruction is not part of the algorithm's release. For each run, we record the output set, its span rank, its cardinality, and the unrestricted rank-two optimum. The comparison fixes maximum output size; it does not assume all samplers realize identical cardinalities.

With uniform count 100, the threshold selects almost no rows into H, but GW continues to sample using the group totals. Its mean costs on Colon, Leukemia, and USPS are 0.39692, 0.79586, 0.22725, compared with 0.75465, 1.00000, 0.71435 for TG. Figure 5 also shows the other private representations and public uniform selection. The latter performs well when the true counts themselves are uniform. The test therefore identifies information lost when sampling uses only the selected weights, rather than a universal advantage over all public row selectors.

D.5 Complete comparisons and uncertainty

The primary grid contains 36 k-median and 30 RSS configurations, totaling 14,130 method executions with seeds 74000–74029. Its public default is $\epsilon = 1$, $n/m = 100$, Zipf-1, and $K = 128$. For each large metric dictionary, we vary $\epsilon \in \{0.25, 1, 4\}$, density in $\{10, 100, 1000, 10000\}$, profile in {uniform, Zipf-1, Zipf-2}, and $K \in \{8, 32, 128, 512\}$, and include a nested $m = 8{,}192$ dictionary.

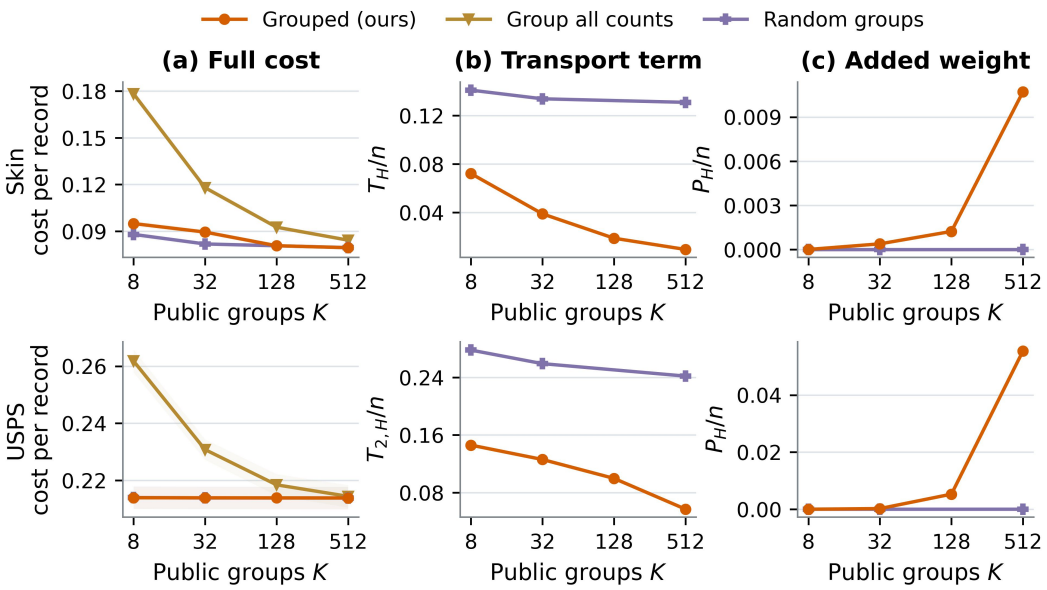


Figure 3: Group granularity at $\epsilon = 1$, Zipf-1, and $n/m = 100$. Columns show original-weight cost per record, T_H/n for k-median or $T_{2,H}/n$ for RSS, and P_H/n. Rows show Skin (k-median) and USPS (RSS). GROUP-ALL also moves the large counts to representatives; random groups keep the same number of groups but ignore geometric proximity. Bands show pointwise 95% intervals. Each plotted point comes from the corresponding measured group count.

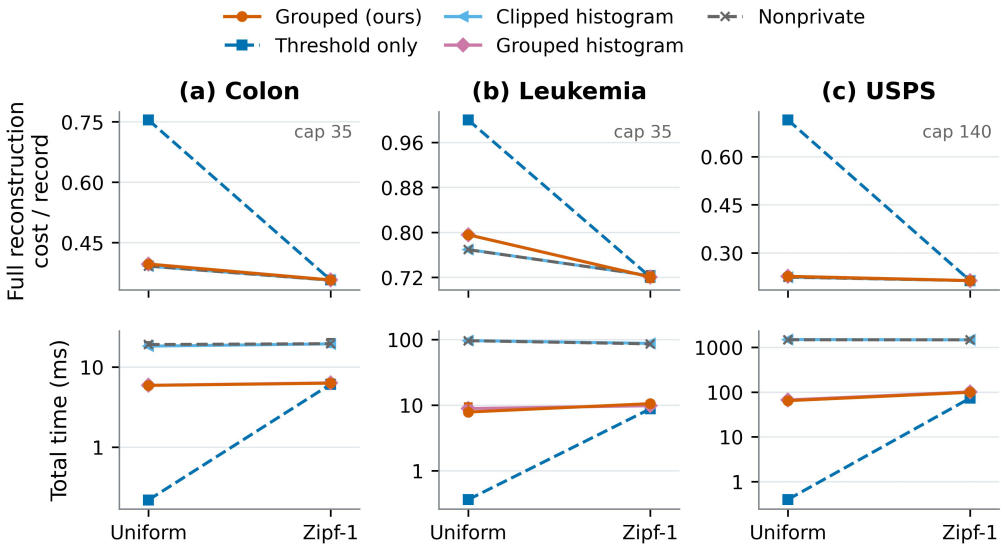


Figure 4: Full RSS cost and time comparison, extending the threshold-only ablation in Figure 16b. RSS at $n/m = 100$, $K = 128$, $\epsilon = 1$, and $r = 2$, with uniform and Zipf-1 counts. Top: mean original-weight rank-two reconstruction cost per record. Bottom: median total algorithm time in milliseconds. Output caps are 35, 35, and 140, shared by every method. Pointwise 95% bootstrap intervals use 30 paired seeds.

RSS varies the same privacy budgets, profiles, and group counts, with densities $10, 100, 1000$. These are one-factor changes around the default. Group-count sweeps use GW, HIST-GROUP, GROUP-ALL, and random-group GW; other primary settings include the full comparator set. The exact combinations are provided in the accompanying artifacts.

Table 4: Realized RSS output sizes at $\epsilon = 1$, Zipf-1, and $n/m = 100$. Each cell reports mean distinct rows / mean span rank over 30 seeds. The public row ranks are checked with the same numerical tolerance used by the solver.

Method	Colon (cap 35)	Leukemia (cap 35)	USPS (cap 140)
GW	28.5 / 28.5	29.9 / 29.9	75.8 / 75.8
TG	25.1 / 25.1	27.9 / 27.9	65.0 / 65.0
HIST	29.3 / 29.3	30.3 / 30.3	97.3 / 97.3
HIST-CLIP	28.6 / 28.6	30.8 / 30.8	102.8 / 102.8
HIST-GROUP	28.5 / 28.5	29.9 / 29.9	75.2 / 75.2
GW-SEQ	28.3 / 28.3	28.4 / 28.4	74.7 / 74.7
Public QR	35.0 / 35.0	35.0 / 35.0	140.0 / 140.0
Public UNIFORM	35.0 / 35.0	35.0 / 35.0	140.0 / 140.0
NP	28.6 / 28.6	30.6 / 30.6	102.8 / 102.8

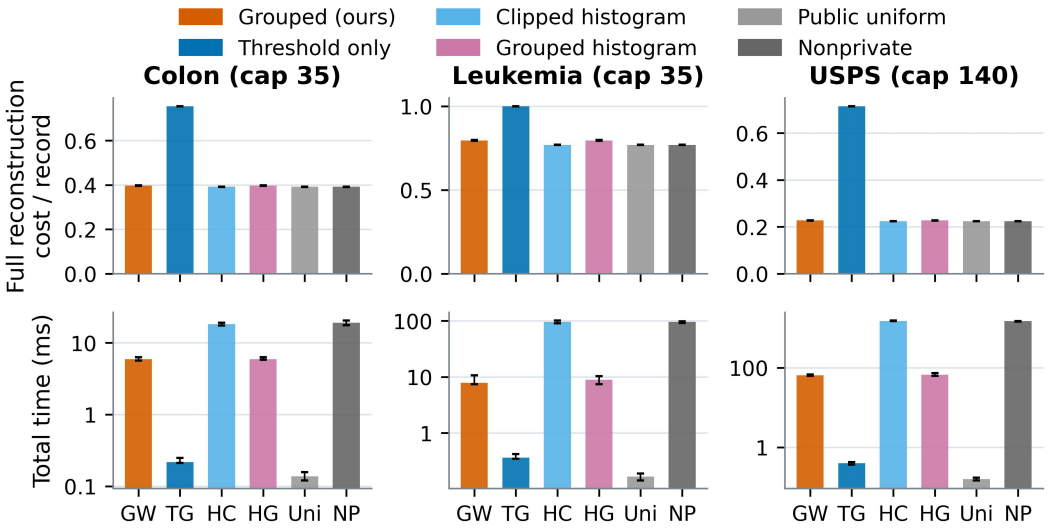


Figure 5: RSS with uniform count 100 per public row and $\epsilon = 1$. Output caps are 35 on Colon and Leukemia and 140 on USPS. Upper panels show full rank-two reconstruction cost per record; bars start at zero. Lower panels show total milliseconds. GW: grouped weights; TG: threshold only; HC: clipped histogram; HG: grouped histogram; Uni: public uniform rows; NP: nonprivate. Error bars are pointwise 95% intervals over 30 seeds. Uniform selection uses the full cap; weighted sampling can repeat rows.

An initial 369-execution pilot used disjoint seeds 73000–73002. The public-partition extension, observed-count experiments, and finite-family study use separate seed ranges. The supplementary tables report every comparator at the default density for all three count profiles; the accompanying artifacts contain the remaining configurations, including smaller dictionaries. Figure 6 shows the privacy-budget sweep.

We use 2,000 bootstrap resamples of seeds for pointwise 95% intervals. Relative cost changes are ratios of the two sample means, with paired resampling. For observed counts only algorithm randomness varies; synthetic-frequency assignments also vary across paired seeds. These intervals describe the stated settings, without a simultaneous superiority or noninferiority claim over the grid.

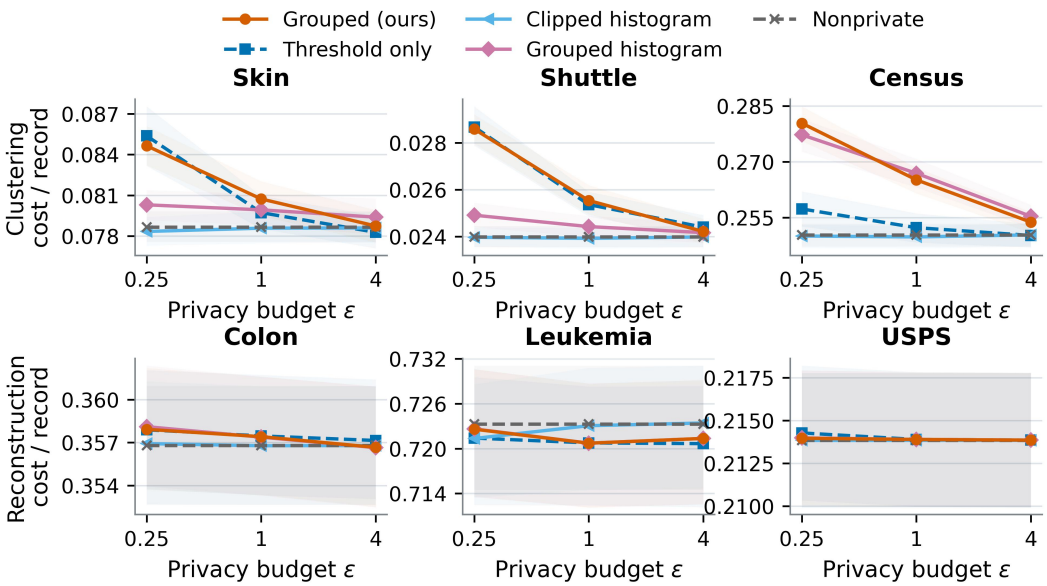


Figure 6: Privacy-budget sweep on all six dictionaries, with Zipf-1 counts, $n/m = 100$, and $K = 128$. The row caps and k-median round cap are fixed across privacy budgets. Curves show mean original-weight cost per record and pointwise 95% seed-bootstrap intervals. NP is a nonprivate reference and its parameters do not use ϵ.

Table 5: Metric k-median across count profiles, $\epsilon = 1$, $n/m = 100$, and $K = 128$. Cost is mean original-weight cost per record; time is median total algorithm milliseconds over 30 paired seeds. All comparators are included. Pointwise intervals and output diagnostics are in the source tables.

	Skin		Shuttle		Census	
Method	Cost	Time (ms)	Cost	Time (ms)	Cost	Time (ms)
Zipf-1 counts						
GW	0.08074	9.30	0.02554	10.30	0.26518	12.44
TG	0.07973	8.04	0.02536	8.59	0.25231	11.57
HIST	0.07849	54.74	0.02399	66.04	0.24938	190.98
HIST-CLIP	0.07859	175.79	0.02393	243.54	0.24983	582.31
HIST-GROUP	0.07994	8.04	0.02443	8.82	0.26693	10.49
GROUP-ALL	0.09263	7.06	0.02913	7.28	0.36618	7.43
NP	0.07866	178.49	0.02399	239.87	0.25038	585.70
Uniform counts						
GW	0.09932	7.34	0.03111	9.25	0.36973	7.80
TG	0.10850	0.72	0.04630	0.82	0.34000	0.77
HIST	0.09291	204.05	0.02735	247.17	0.29831	579.53
HIST-CLIP	0.09291	205.58	0.02735	247.59	0.29831	580.15
HIST-GROUP	0.09927	5.43	0.03083	5.93	0.36972	6.02
GROUP-ALL	0.09932	7.31	0.03111	7.25	0.36973	7.72
NP	0.09304	202.06	0.02741	244.67	0.29838	578.95
Zipf-2 counts						
GW	0.00825	7.35	0.00282	7.84	0.02519	9.53
TG	0.00830	6.45	0.00284	6.71	0.02517	6.71
HIST	0.00800	5.85	0.00247	5.98	0.02511	7.12
HIST-CLIP	0.00801	96.81	0.00247	124.11	0.02509	309.73
HIST-GROUP	0.00801	6.98	0.00247	5.92	0.02509	6.66
GROUP-ALL	0.05150	7.30	0.01672	9.32	0.34535	8.03
NP	0.00800	19.23	0.00246	20.77	0.02510	49.74

Table 6: Row subset selection across count profiles, $\epsilon = 1$, $n/m = 100$, and $K = 128$. Cost is mean original-weight cost per record; time is median total algorithm milliseconds over 30 paired seeds. All comparators are included. Pointwise intervals and output diagnostics are in the source tables.

	Colon		Leukemia		USPS	
Method	Cost	Time (ms)	Cost	Time (ms)	Cost	Time (ms)
Zipf-1 counts						
GW	0.35741	6.29	0.72072	10.53	0.21390	98.75
TG	0.35746	6.09	0.72076	8.78	0.21390	73.09
HIST	0.35674	9.56	0.72066	32.23	0.21385	547.21
HIST-CLIP	0.35678	19.49	0.72306	87.30	0.21385	1469.76
HIST-GROUP	0.35738	6.35	0.72073	9.78	0.21390	101.13
GROUP-ALL	0.37733	6.25	0.78296	7.82	0.21848	63.65
NP	0.35679	19.62	0.72327	85.76	0.21385	1471.44
GW-SEQ	0.35745	6.53	0.72434	10.14	0.21390	96.57
Public QR	0.37016	3.90	0.77968	18.80	0.21390	169.75
Public UNIFORM	0.36827	0.13	0.75012	0.16	0.21387	0.16

Row subset selection (continued)

Method	Colon Cost	Colon Time (ms)	Leukemia Cost	Leukemia Time (ms)	USPS Cost	USPS Time (ms)
Uniform counts						
GW	0.39692	5.90	0.79586	7.83	0.22725	64.18
TG	0.75465	0.22	1.00000	0.36	0.71435	0.40
HIST	0.39173	17.77	0.76959	95.90	0.22449	1472.62
HIST-CLIP	0.39173	18.26	0.76959	96.29	0.22449	1492.78
HIST-GROUP	0.39694	5.92	0.79589	8.87	0.22734	66.75
GROUP-ALL	0.39692	5.87	0.79586	8.25	0.22725	72.98
NP	0.39175	19.13	0.76935	95.84	0.22449	1474.17
GW-SEQ	0.39692	6.10	0.79586	8.74	0.22723	67.38
Public QR	0.39505	3.83	0.80190	19.35	0.22453	175.92
Public UNIFORM	0.39177	0.14	0.76934	0.17	0.22449	0.16
Zipf-2 counts						
GW	0.12209	6.11	0.21400	8.96	0.07438	69.58
TG	0.12211	4.71	0.21394	5.33	0.07438	53.10
HIST	0.12212	4.22	0.21397	8.32	0.07438	66.05
HIST-CLIP	0.12212	12.98	0.21397	51.21	0.07438	797.23
HIST-GROUP	0.12211	6.16	0.21412	7.28	0.07438	65.47
GROUP-ALL	0.27574	6.49	0.66717	7.74	0.10408	67.60
NP	0.12217	8.74	0.21427	13.42	0.07438	253.75
GW-SEQ	0.12209	6.14	0.21402	7.10	0.07438	68.25
Public QR	0.20206	3.77	0.49280	19.60	0.07486	176.51
Public UNIFORM	0.20466	0.13	0.46032	0.15	0.07483	0.17

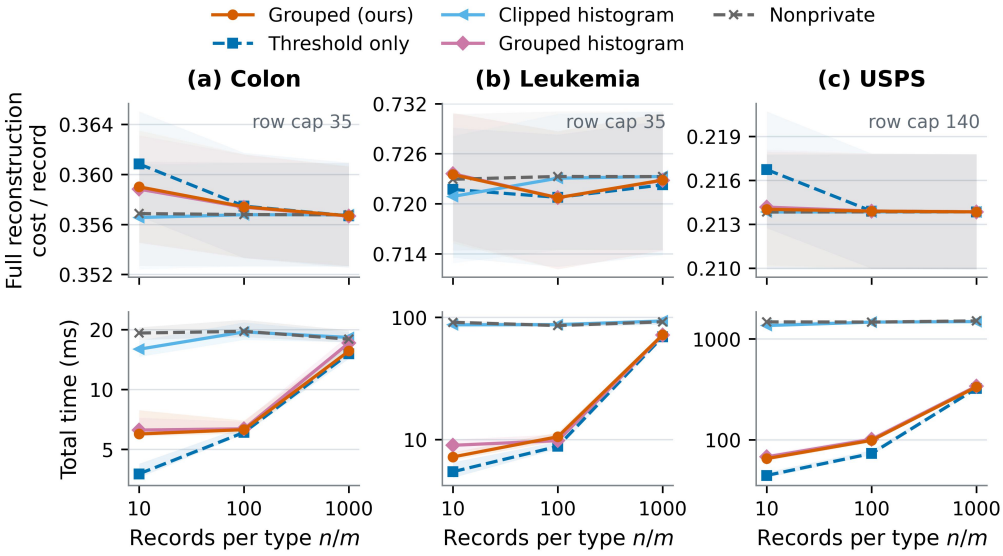


Figure 7: Full RSS density sweep under Zipf-1 counts, $K = 128$, and $\epsilon = 1$. Top: mean full rank-two reconstruction cost. Bottom: median total milliseconds. Row caps are 35, 35, and 140. Pointwise 95% intervals are computed over 30 paired seeds.

D.6 Observed counts

We also evaluate observed UCI Bike Sharing counts and NYC yellow taxi counts. This is a separate observed-count study with six task–period settings, 30 seeds (75000–75029), and 1,080 executions. Every setting uses $K = 128$ and $\epsilon = 1$.

Bike 2011 and 2012 have 8,760 and 8,784 public hourly positions, with total counts 1,243,103 and 2,049,576. Their five features encode normalized calendar position, sine/cosine of hour, and half-scaled sine/cosine of weekday. Taxi January and February 2024 use the same 6,312 public zone–hour combinations. Zone centroids are translated and divided by the larger coordinate range; hour sine/cosine coordinates have amplitude 0.2. We retain trips with valid public zone and hour identifiers, giving count totals 2,952,588 and 2,996,418. Metric distances are normalized by the exact public diameter.

For RSS on the taxi counts, a fixed public Fourier map gives 256 coordinates, $\sqrt{2/256}\cos(x^\top\omega_j + b_j)$, with $\omega_j \sim N(0, 12^2 I)$ and $b_j \sim \mathrm{Unif}[0, 2\pi]$. Its public seed is 93021; a global maximum-row-norm scale is then applied. Both months use the same public map. These RSS inputs are constructed features of the taxi dictionary, rather than original sensor measurements. RSS uses $r = 2$, $b = 32$, and four repetitions, with row cap 140.

Table 7: RSS with observed taxi counts and 256 public Fourier features. Values are mean full reconstruction cost per record / median milliseconds. Every method has row cap 140.

Method	January 2024	February 2024
GW	0.62373 / 223.87	0.62358 / 207.49
TG	0.62373 / 200.42	0.62358 / 186.33
HIST	0.62373 / 354.99	0.62358 / 342.06
HIST-CLIP	0.62373 / 883.74	0.62358 / 840.84
HIST-GROUP	0.62373 / 217.84	0.62358 / 213.53
NP	0.62373 / 798.26	0.62358 / 759.44

D.7 Published private clustering baselines

The public-universe formulations of Gupta et al. (2010) and Fan et al. (2023) match our metric objective and add/remove-one count adjacency. We implement their complete swap enumeration on nested dictionaries with $m \in \{128, 256, 512\}$, $k = 4$, and $n/m = 100$. These instances keep all $k(m-k)$ candidate swaps available at every round. All methods in this comparison share one dense public distance matrix. We use $K = 16$ for the grouped methods, 30 paired seeds 82000–82029, and $\epsilon = 1$ across sizes, with additional $\epsilon \in \{0.25, 4\}$ runs at $m = 256$ on every dataset. Large-dictionary experiments use the shared on-demand distance backend instead.

Gupta's algorithm draws each swap with probability proportional to $\exp[-\epsilon\Phi_w(C')/(2\Delta(T+1))]$, then privately selects among the first T iterates using the same coefficient. It executes sampled swaps even when their realized gain is negative. We test both $T = \lceil 6k \log m \rceil$ and a 64-round practical schedule. Replacing demand sums by integer-weighted sums preserves sensitivity Δ and the privacy allocation. The original set-demand utility bound is not used to certify these multiplicity experiments.

DP-HST uses Algorithms 3–7 of Fan et al. (2023): a public ball-cover hierarchy, private subtree counts, disjoint-subtree selection, greedy descent, and complete private swap search. We allocate $\epsilon/2$ to initialization and $\epsilon/2$ to 64 swaps and private selection among all 65 iterates. The main DP-HST comparison uses their experimental depth $L = 8$; a terminal cluster returns its root, chosen uniformly from its public sites during tree construction. We also test DP-HST-FULL with $L = \lceil \log_2(\Delta/\delta) \rceil + 1$, where public minimum separation δ ensures singleton leaves. The public covering radius at depth t is $\Delta/2^t$. At depth $t \geq 1$, noise scale is $2^t/(\epsilon/2)$; the level budgets sum to less than $\epsilon/2$. The root is excluded from count queries and from selecting $k \geq 2$ disjoint subtrees. The public trees use seed 93041 and are timed separately. These are our implementations of the cited procedures with the stated public tree constructions and practical round cap.

At $m = 256$ and $\epsilon = 1$, DP-HST attains lower mean cost than GW on all three dictionaries. On Census their costs are 0.30489 and 0.35330, with times of 38.39 and 13.48 ms. HIST-CLIP gives 0.29437 in 10.88 ms at this size. These small-instance results complement the larger-dictionary curves: complete support can be inexpensive on a few hundred sites, while grouping reduces the work that grows with the dictionary.

The histogram baselines answer a complementary question: how a full-budget input perturbation compares with keeping weights inside the optimizer. HIST-GROUP is especially relevant because it

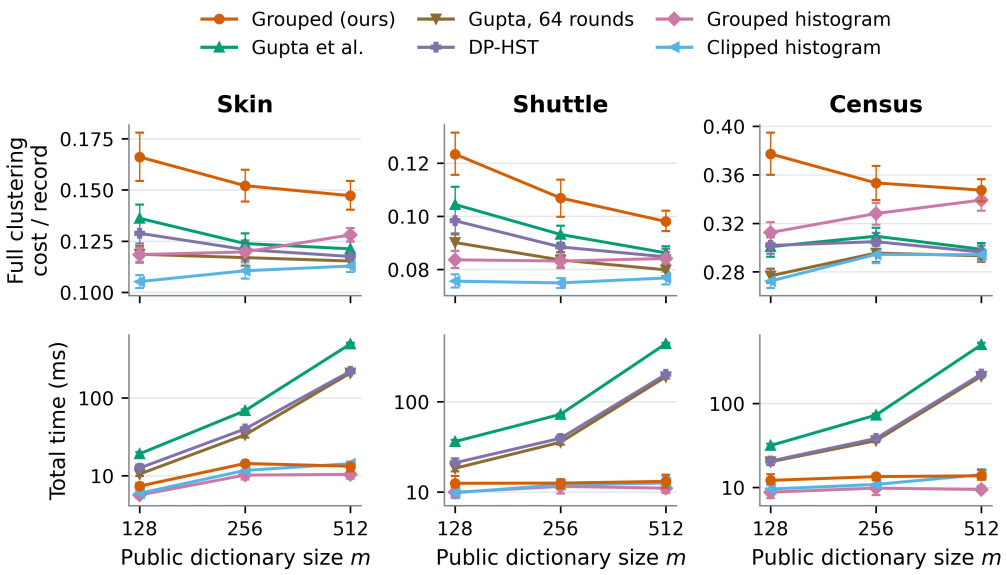


Figure 8: Published private clustering baselines at $\epsilon = 1$, $k = 4$, and Zipf-1 density 100. Gupta denotes the $\lceil 6k \log m \rceil$ schedule; Gupta, 64 rounds uses the shorter schedule. DP-HST spends half its budget on private initialization. Top: mean original-weight cost. Bottom: median total algorithm milliseconds, including private initialization but excluding public distance, tree, and partition setup. All methods use the same complete public candidate universe and 30 paired seeds.

uses the same support K_H as GW. Its greedy swap decisions are post-processing, whereas GW must pay for private decisions. TG, GROUP-ALL, and random groups test the roles of selected weights, group totals, and public geometry. NP, public QR, and public uniform are quality references. For RSS we compare these row-selection rules at a common row cap and include histogram privatizations of the same weighted solver.

Table 8: Published-baseline comparison on $m = 256$. Each cell gives mean original-weight cost on the first line and median time in milliseconds on the second. All three tested privacy budgets are included. Per-seed outputs and public setup times are provided in the accompanying artifacts.

Dictionary, ϵ	GW	Gupta	Gupta-64	DP-HST	H-CLIP
Skin, 1	0.15207 14.4	0.12381 68.3	0.11696 33.5	0.12088 39.9	0.11053 11.7
Skin, 0.25	0.19058 14.5	0.15707 69.1	0.13953 34.2	0.15388 40.2	0.11039 11.4
Skin, 4	0.12471 14.2	0.11174 69.6	0.10970 32.6	0.11150 39.7	0.11039 10.6
Shuttle, 1	0.10689 12.5	0.09316 72.4	0.08356 35.6	0.08846 39.3	0.07493 12.0
Shuttle, 0.25	0.13097 13.6	0.11260 70.3	0.10537 33.7	0.10926 40.3	0.07495 11.8
Shuttle, 4	0.08942 13.9	0.07774 70.1	0.07542 34.5	0.07721 38.7	0.07497 11.4
Census, 1	0.35330 13.5	0.30937 72.9	0.29549 36.1	0.30489 38.4	0.29437 10.9
Census, 0.25	0.45991 13.7	0.37682 73.2	0.32560 35.4	0.36186 39.0	0.29396 11.7
Census, 4	0.31717 15.4	0.29208 72.9	0.29118 36.1	0.29244 40.1	0.29439 11.7

D.8 Recalibrated parameter sweeps

The additional grid contains 53 configurations and uses 30 fresh paired seeds, 81000–81029. The clustering studies vary $T \in \{16, 32, 64, 128, 256\}$, $k \in \{4, 8, 16\}$, and public dictionary size, one factor at a time. Every change recalculates $R = k + T$, λ, κ, and the per-round decision coefficient. A larger round cap consequently changes both optimization effort and its privacy calibration. The mean costs in Figure 9 need not be monotone in T; the default remains the previously specified 64 rounds.

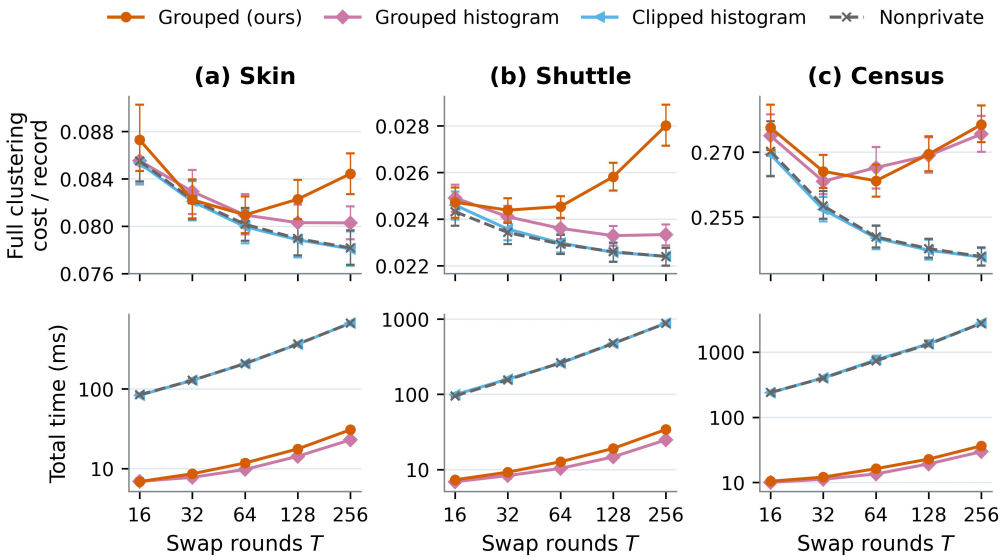


Figure 9: Swap-round sensitivity, $k = 8$, $m = 32{,}768$, Zipf-1 density 100, and $\epsilon = 1$. The values of λ and κ and the private decision coefficient are recalibrated at every T. Histogram decisions remain post-processing. Top: original-weight cost. Bottom: total milliseconds. Error bars are pointwise 95% intervals over 30 fresh paired seeds.

Table 9: GW calibration and executed swaps in the round sweep. The calibration gives $\lambda = \kappa$ with the listed value on these dictionaries. Executed swaps are averaged over the 30 seeds. The round cap counts proposals; private selection can instead choose the no-op.

T	$\lambda = \kappa$	Skin swaps	Shuttle swaps	Census swaps
16	120	6.30	5.80	7.23
32	200	7.70	6.27	8.80
64	360	7.73	5.20	10.50
128	680	6.73	3.50	8.13
256	1320	4.67	2.07	6.30

For RSS we vary rank $r \in \{1, 2, 3\}$ on all three dictionaries with one repetition and $\eta = 1$. On USPS we also vary $\eta \in \{1, 1/2, 1/4\}$ and $s \in \{1, 2, 4\}$ at $r = 2$, changing one at a time. The residual count is always $b = \lceil 16r/\eta \rceil$; all comparators share the resulting cap $s(2r - 1 + b)$. These settings keep every released span below the public rank, including the largest residual batch. We record realized cardinalities and span ranks rather than interpreting the cap as the number of distinct sampled rows.

Figure 12 separates the accounting change from the measured objective. For a common rank and output cap, GW has $R = s(1 + b)$ and GW-SEQ has $R = s(2r - 1 + b)$. At $r = 1$ the two charges coincide. At larger ranks the joint charge reduces κ, although residual draws still dominate the total. The final panel uses paired differences of original-weight costs; its intervals resample the same seed in both methods. In this rank sweep, every nonzero paired difference has a 95% interval containing

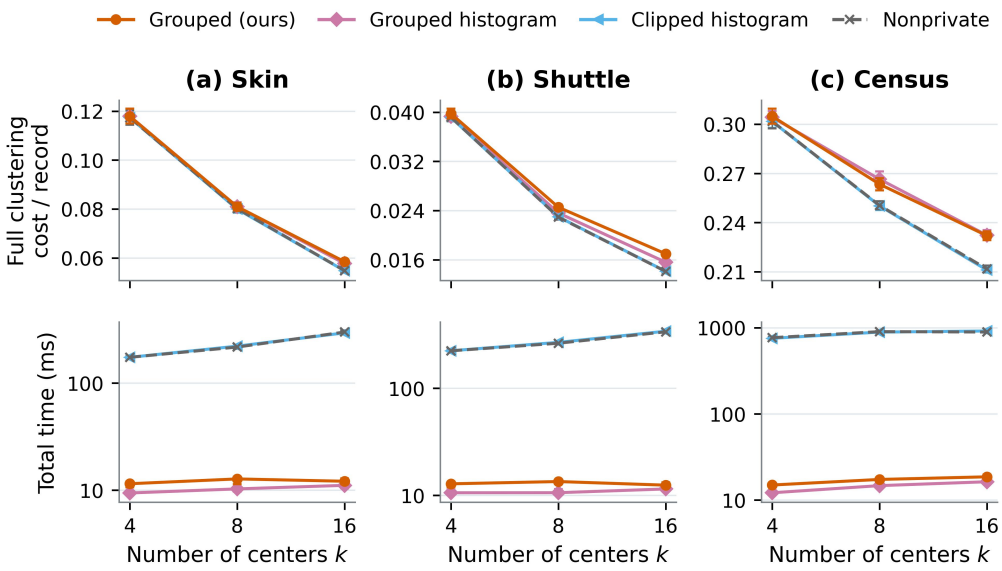


Figure 10: Center-count sensitivity with $T = 64$ and otherwise the same settings as Figure 9. Every method releases exactly k centers. Top: original-weight cost. Bottom: total milliseconds. Error bars are pointwise 95% intervals.

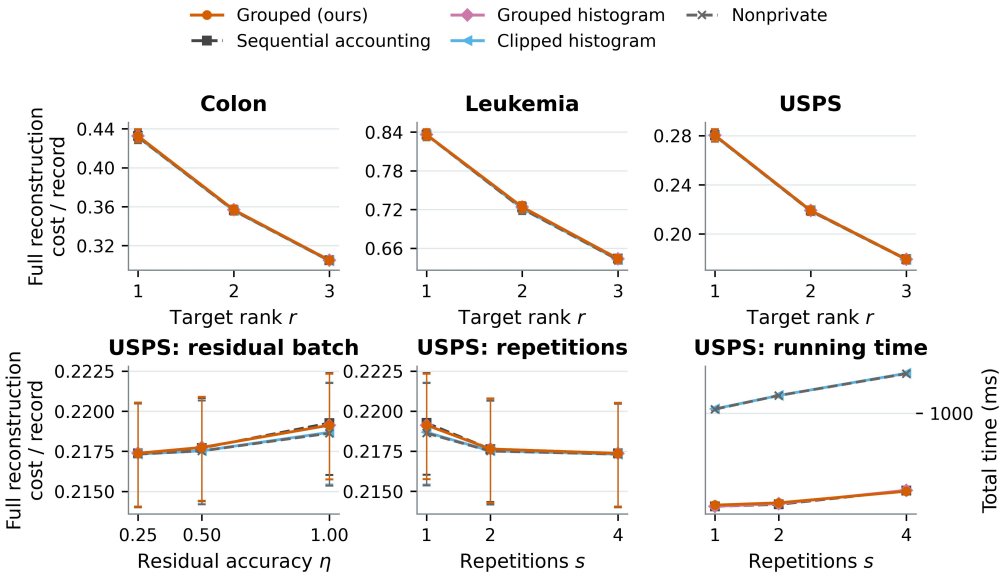


Figure 11: RSS parameter sensitivity with Zipf-1 density 100, $K = 128$, and $\epsilon = 1$. The rank panels use $s = 1$, $\eta = 1$; the residual-accuracy panel uses $r = 2$, $s = 1$; the repetition panels use $r = 2$, $\eta = 1$. All settings use $b = \lceil 16r/\eta \rceil$ and recalibrated λ and κ. Costs are original-weight rank-r reconstruction losses per record; the final panel reports total milliseconds.

zero. The accounting reduction is exact, while the observed reconstruction differences are small and vary in direction. The accompanying artifacts contain the paired results for every setting.

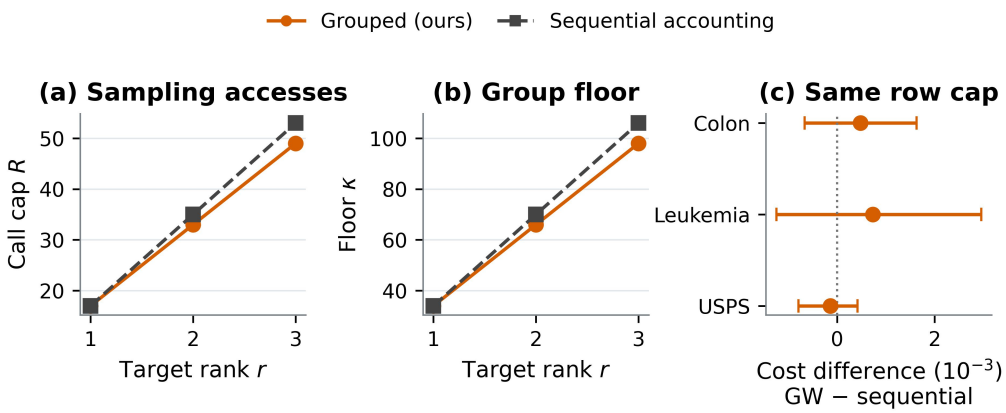


Figure 12: Joint versus sequential volume accounting. (a,b) The sampling-step bound R and calibrated value of κ as rank varies with $s = 1$ and $\eta = 1$. (c) Paired original-weight-cost difference at $r = 2$ on all three dictionaries, also with $s = 1$ and row cap 35. Negative values favor joint accounting; the dotted line denotes zero difference. Intervals are pointwise 95% paired-bootstrap intervals. Unlike the default USPS benchmark, this panel uses one repetition.

D.9 Public dictionary scaling and setup

The nested dictionary study uses $m = 4096, 8192, 16384, 32768$ for all three metric datasets, then all 51,433 distinct Skin sites and 65,536 Census sites. Census uses a random-key reservoir of 100,000 candidates before deduplication, preserving the nested subsets and full-file normalization. The count density remains 100, so this experiment changes the number of actual public vectors rather than only their multiplicities.

Figure 13 relates the primary density sweep to the number of locations actually passed to the solver. The grouped methods approach full-support work as $|H|$ grows. Residual differences in time reflect aggregation, sampling, and solver operations as well as support size. Figure 14 adds the measured public setup to one or more calls at the largest dictionary sizes. Diameter computation is shared by all methods; partition construction is added for GW and HIST-GROUP. The displayed totals are setup time plus the number of calls times the measured median call time. They are a decomposition of measured costs, not a new batch benchmark. Reusing a public partition does not reuse the privacy budget: independent releases compose their privacy charges.

D.10 Perturbations of the grouped sampling weights

We hold the realized threshold outcome H, the public partition, and the sampling coefficients fixed to distinguish two changes to the sampling distribution. The first adds independent histogram-scale Laplace noise to the grouped weights u and clips at zero; its total variation distance is measured against the distribution that uses u directly. The second compares the GW distribution with the distribution formed from the same selected locations and group totals before applying the lower bounds λ and κ. For clustering, the coefficients are distances to the first public representative; for RSS they are squared residuals to r public pivoted-QR rows. These coefficients are chosen without reading counts.

We evaluate this comparison on all six dictionaries, at densities 10 and 100 and privacy budgets 0.25, 1, and 4, using 30 fresh seeds 83000–83029. All 36 configurations are retained in the source data; the following values use $\epsilon = 1$. At density 100, adding noise to the grouped weights u changes the distribution by 0.00011–0.00162 in total variation. The difference caused by applying the lower bounds λ and κ ranges from 0.00007 to 0.02035 and is larger on five of the six dictionaries. This comparison separates the effects of the lower bounds and public geometry from the effect of adding noise. Using u directly avoids the additional perturbation. Because this comparison conditions on realized private quantities, we use it only to describe the sampling distribution; the optimizer experiments use the privacy allocations stated above.

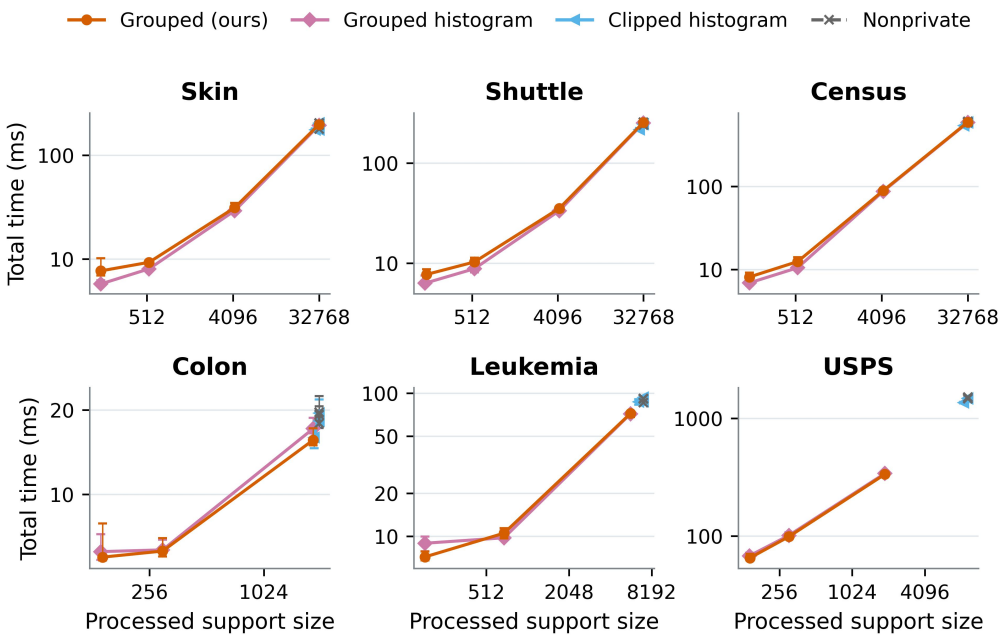


Figure 13: Processed support and total algorithm time for the primary Zipf-1 density sweep at $K = 128$ and $\epsilon = 1$. Each point is an existing measured setting; clustering dictionaries have $m = 32{,}768$. Both axes are logarithmic. Connecting points shows the density sweep within a method; it is not a fitted complexity law.

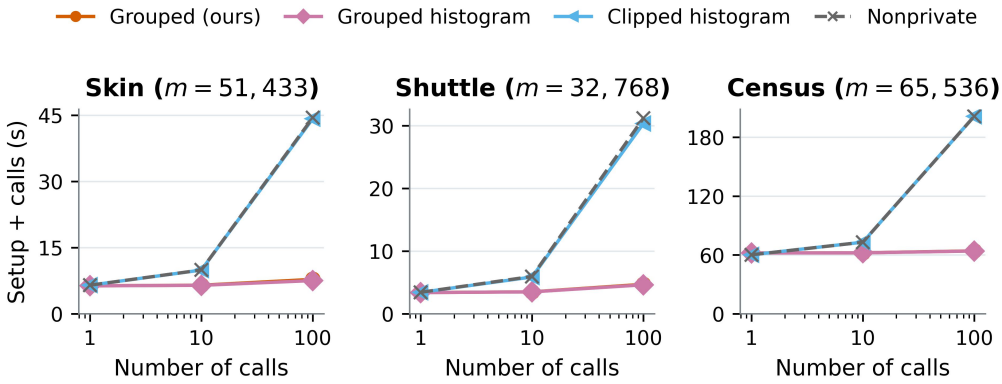


Figure 14: Public setup plus repeated calls at the largest tested dictionary sizes. The vertical axis is seconds. Totals equal one measured public setup plus q times the median algorithm time for $q = 1, 10, 100$. Shared exact-diameter computation is included for all methods; public partition construction is included for grouped methods. Each setup was measured once, separately from the paired algorithm runs.

D.11 Additional parameter results

Tables 10 and 11 include every additional parameter configuration, including repeated defaults belonging to separate sweeps. RSS rank sweeps use one repetition on each dictionary; residual and repetition sweeps use USPS at rank two. The accompanying artifacts contain the pointwise intervals, output ranks, support sizes, calibrated privacy charges, and per-seed outputs for all reported configurations. They also contain all eight methods at all 15 published-baseline settings, including the full-depth HST variant, grouped histogram, and NP omitted from the compact printed table.

Table 10: Additional clustering runs. Each cell gives mean original-weight cost on the first line and median time in milliseconds on the second, over 30 paired seeds. All settings have Zipf-1 counts, $n/m = 100$, $\epsilon = 1$, and $K = 128$. The changed parameter is shown; remaining parameters follow the study defaults.

Dictionary	Setting	GW	H-GROUP	H-CLIP	NP
Skin	$T = 16$	0.08730 6.9	0.08554 6.9	0.08532 83.2	0.08552 84.5
Skin	$T = 32$	0.08223 8.6	0.08293 7.7	0.08210 128.3	0.08223 128.3
Skin	$T = 64$	0.08096 11.7	0.08094 9.7	0.07996 207.4	0.08016 208.6
Skin	$T = 128$	0.08228 17.6	0.08030 14.3	0.07885 364.5	0.07896 366.4
Skin	$T = 256$	0.08444 30.7	0.08028 22.9	0.07809 673.3	0.07817 669.5
Skin	$k = 4$	0.11788 11.5	0.11792 9.4	0.11734 173.9	0.11731 173.9
Skin	$k = 8$	0.08096 12.7	0.08094 10.3	0.07996 221.1	0.08016 216.8
Skin	$k = 16$	0.05848 12.1	0.05774 11.1	0.05478 294.1	0.05483 299.2
Skin	$m = 4096$	0.08138 7.9	0.07426 6.9	0.07288 29.8	0.07297 31.1
Skin	$m = 8192$	0.08108 9.6	0.07761 7.4	0.07637 54.5	0.07633 55.2
Skin	$m = 16384$	0.08086 9.5	0.07923 8.2	0.07795 104.1	0.07789 105.6
Skin	$m = 32768$	0.08096 11.2	0.08094 9.5	0.07996 206.9	0.08016 204.0
Skin	$m = 51433$	0.08224 14.7	0.08093 11.9	0.08021 380.9	0.08023 383.2
Shuttle	$T = 16$	0.02473 7.3	0.02492 6.9	0.02461 98.9	0.02432 95.0
Shuttle	$T = 32$	0.02439 9.2	0.02410 8.3	0.02358 159.3	0.02343 155.7
Shuttle	$T = 64$	0.02454 12.7	0.02361 10.3	0.02298 259.5	0.02293 262.6
Shuttle	$T = 128$	0.02582 19.1	0.02330 14.7	0.02258 472.4	0.02261 476.2
Shuttle	$T = 256$	0.02801 34.1	0.02334 24.8	0.02241 885.5	0.02240 877.8
Shuttle	$k = 4$	0.03976 12.8	0.03933 10.6	0.03914 223.9	0.03918 223.6
Shuttle	$k = 8$	0.02454 13.5	0.02361 10.6	0.02298 268.3	0.02293 263.3
Shuttle	$k = 16$	0.01697 12.5	0.01560 11.5	0.01412 340.8	0.01410 336.4
Shuttle	$m = 4096$	0.04931 8.8	0.04269 6.6	0.04281 30.7	0.04269 30.5
Shuttle	$m = 8192$	0.03994 9.6	0.03657 8.3	0.03632 59.8	0.03635 59.4
Shuttle	$m = 16384$	0.02534 11.2	0.02324 9.1	0.02301 144.8	0.02296 140.9

Table 10 (continued)

Dictionary	Setting	GW	H-GROUP	H-CLIP	NP
Shuttle	$m = 32768$	0.02454 13.3	0.02361 12.5	0.02298 271.9	0.02293 280.5
Census	$T = 16$	0.27565 10.4	0.27377 10.1	0.26948 239.0	0.27021 240.3
Census	$T = 32$	0.26558 12.1	0.26327 11.2	0.25677 402.8	0.25763 399.4
Census	$T = 64$	0.26339 16.3	0.26644 13.5	0.25019 760.5	0.25045 737.3
Census	$T = 128$	0.26954 22.8	0.26924 19.3	0.24737 1338.8	0.24776 1320.2
Census	$T = 256$	0.27638 36.5	0.27418 29.7	0.24580 2782.1	0.24591 2769.0
Census	$k = 4$	0.30483 14.9	0.30420 12.1	0.30169 755.1	0.30174 767.6
Census	$k = 8$	0.26339 17.3	0.26644 14.7	0.25019 891.0	0.25045 902.7
Census	$k = 16$	0.23199 18.6	0.23227 16.3	0.21103 917.6	0.21171 892.9
Census	$m = 4096$	0.25990 10.2	0.25758 8.0	0.23561 98.7	0.23526 99.6
Census	$m = 8192$	0.25102 9.9	0.24889 8.3	0.23600 195.5	0.23558 196.8
Census	$m = 16384$	0.26932 11.4	0.26713 9.6	0.24347 406.2	0.24418 403.5
Census	$m = 32768$	0.26339 15.1	0.26644 13.0	0.25019 819.8	0.25045 811.3
Census	$m = 65536$	0.25826 21.6	0.25883 20.1	0.24819 1424.3	0.24888 1423.8

Table 11: Additional RSS runs. Each cell gives mean original-weight cost on the first line and median time in milliseconds on the second, over 30 paired seeds. All settings have Zipf-1 counts, $n/m = 100$, $\epsilon = 1$, and $K = 128$. The changed parameter is shown; remaining parameters follow the study defaults.

Dictionary	Setting	GW	H-GROUP	H-CLIP	NP	GW-SEQ
Colon	$r = 1$	0.43263 6.1	0.43259 6.1	0.43144 14.0	0.43147 13.9	0.43263 6.1
Colon	$r = 2$	0.35684 9.3	0.35677 8.8	0.35596 24.0	0.35595 25.0	0.35636 9.0
Colon	$r = 3$	0.30489 12.0	0.30490 12.3	0.30430 33.9	0.30440 33.7	0.30510 11.9
Leukemia	$r = 1$	0.83608 12.1	0.83608 12.4	0.83557 75.9	0.83584 74.6	0.83608 12.5
Leukemia	$r = 2$	0.72391 16.5	0.72388 17.2	0.72161 143.5	0.72102 142.5	0.72317 17.2
Leukemia	$r = 3$	0.64368 21.0	0.64372 21.5	0.64224 191.8	0.64203 187.4	0.64409 21.8
USPS	$r = 1$	0.28024 134.3	0.28024 137.0	0.27969 857.5	0.27990 856.3	0.28024 129.9
USPS	$r = 2$	0.21913 128.8	0.21913 135.0	0.21868 1182.2	0.21863 1156.8	0.21926 131.2

Table 11 (continued)

Dictionary	Setting	GW	H-GROUP	H-CLIP	NP	GW-SEQ
USPS	$r = 3$	0.17946 147.8	0.17947 152.8	0.17919 1471.3	0.17921 1495.0	0.17960 148.1
USPS	$\eta = 1$	0.21913 125.1	0.21913 127.2	0.21868 1091.0	0.21863 1109.9	0.21926 124.8
USPS	$\eta = 0.5$	0.21774 103.6	0.21775 106.2	0.21753 1108.6	0.21752 1106.2	0.21770 102.0
USPS	$\eta = 0.25$	0.21738 78.7	0.21738 81.2	0.21732 1126.9	0.21732 1114.5	0.21738 81.0
USPS	$s = 1$	0.21913 130.9	0.21913 127.2	0.21868 1101.0	0.21863 1091.6	0.21926 127.2
USPS	$s = 2$	0.21764 137.6	0.21764 135.1	0.21752 1480.9	0.21750 1479.1	0.21763 132.8
USPS	$s = 4$	0.21736 177.6	0.21736 181.7	0.21731 2420.0	0.21731 2400.1	0.21737 180.8

D.12 Scaling beyond 100,000 public locations

We extend the Census study to $m \in \{8,192, 32,768, 65,536, 131,072\}$, using 30 new paired seeds 85000–85029 and all four methods in Figure 2 (480 executions). A public random-key reservoir of 200,000 candidates from the same original file is deduplicated and truncated. The 32,768- and 65,536-location prefixes match the previous dictionaries exactly. Feature scaling, exact-diameter normalization, farthest-point groups, privacy calibration, and solver logic are unchanged. the largest setting has $n = 13,107,200$ through multiplicities and $m = 131,072$ distinct vectors in 68 dimensions.

Across the four sizes, GW's mean cost exceeds HIST-CLIP's by 3.7–5.8% in the paired relative-cost comparison. Thus the timing benefit accompanies a measured objective tradeoff. At $m = 131,072$, HIST-GROUP has cost 0.2643 and median time 39.3 ms; this control uses the same support K_H as GW. The comparison supports computational savings from grouping, while the threshold-only ablations test the contribution of the retained group totals.

At the largest size, exact public-diameter computation takes 231.69 s and public partition construction takes 5.31 s. Figure 15 adds these measured public costs to median algorithm time. The diameter is shared by every method; the partition is used by GW and HIST-GROUP. First-use totals are dominated by this public setup. The main timing curves describe calls that reuse the public geometry; each private release still consumes its own privacy budget. Offline full-data scoring and input-file loading are outside the algorithm timer for every method. No dense $m \times m$ distance matrix is allocated.

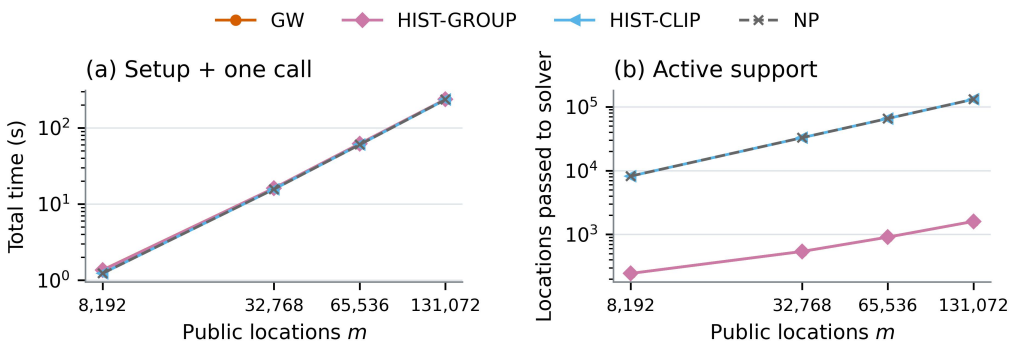


Figure 15: Public setup and support size for the Census extension. (a) Measured public setup plus median time for one call; these totals are a decomposition, not a separately timed batch. (b) Mean number of locations with positive weight passed to the solver. The results use all 30 seeds and all four methods.

The configuration, raw measurements, and plotting code for this experiment are provided under `experiments/analysis/scaling_20260924`.

D.13 Expanded published-baseline comparison

We extend the Fan et al. comparison to nested $m \in \{1,024, 2,048\}$ dictionaries on all three metric datasets, with $k = 8$, $K = 128$, $T = 64$, $\epsilon = 1$, and Zipf-1 density 100. Thirty new paired seeds 86000–86029 give 720 executions. Figure 1a–b shows all three datasets at the preselected largest size; Table 12 retains both sizes.

DP-HST uses our implementation of Fan et al. (2023), Algorithms 3–7, as detailed in the earlier published-baseline comparison. It uses depth eight, half the privacy budget for initialization, and half for 64 complete-swap rounds plus private selection among all 65 iterates. No restricted proposal subset replaces the complete enumeration. All methods share the same dense public distance matrix, and their timers include private count processing and optimization but exclude offline scoring. Tree construction for DP-HST and partition construction for GW/HIST-GROUP are separately recorded. We report this dense-distance comparison separately from the scaling experiments, which compute distances on demand. Methods run serially with one BLAS thread and randomized execution order.

Table 12: Expanded private clustering comparison. Each cell gives mean original-weight cost on the first line and median time in milliseconds on the second, over 30 paired seeds.

Dictionary, m	GW	Fan et al.	HIST-CLIP	HIST-GROUP
Skin, 1,024	0.0969 6.4	0.0741 721.8	0.0692 9.3	0.0709 5.5
Skin, 2,048	0.0883 11.1	0.0725 3109.8	0.0711 20.3	0.0729 8.8
Shuttle, 1,024	0.0633 11.7	0.0484 772.0	0.0428 13.6	0.0432 8.2
Shuttle, 2,048	0.0584 13.5	0.0465 3116.9	0.0445 23.7	0.0447 12.0
Census, 1,024	0.2759 11.4	0.2360 779.7	0.2336 14.9	0.2554 8.4
Census, 2,048	0.2625 11.5	0.2325 3104.4	0.2364 25.2	0.2527 9.6

HIST-GROUP can improve on GW in both objective and runtime: after releasing noisy grouped weights, its downstream optimization is post-processing, whereas GW pays for private swap decisions. This comparison therefore reflects the cost of GW's private swap-selection step rather than a general advantage over grouped histograms. The RSS main panels similarly retain public uniform selection on USPS, where it attains lower error than GW. Public QR selects a reusable public set; its factorization time must not be interpreted as a cost incurred anew for each private count vector. Additional comparisons with uniform counts and observed counts are reported below.

The accompanying artifacts contain the per-seed results, setup measurements, and plotting code for these comparisons. The main RSS bars use all 30 seeds of each primary Zipf-1 setting. Main cost plots use total objectives, obtained by multiplying each per-record cost by its count total before aggregation; appendix tables retain their stated per-record units. The main time bars include public QR factorization, whose selected public set can be reused across count vectors.

D.14 Additional main-comparison figures

The following figures report additional results for the experimental settings described above.

D.15 Finite separation experiment and implementation checks

Figure 21 shows the input counts, grouped weights u, and the corresponding sampling-probability and cost curves.

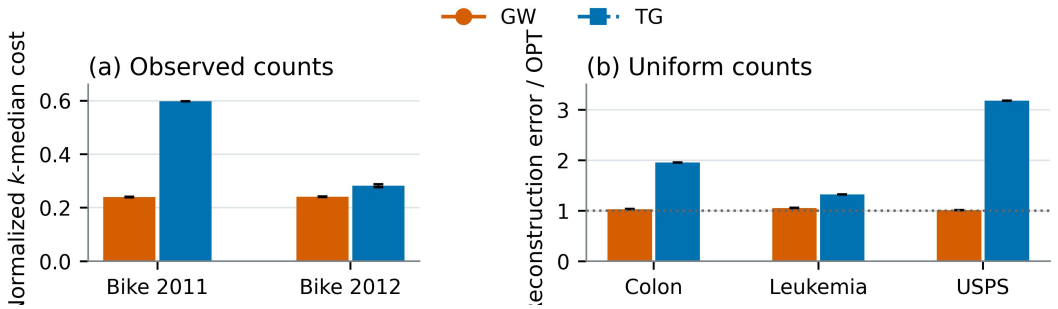


Figure 16: Grouping versus thresholding alone, at $\epsilon = 1$ and $K = 128$. (a) Normalized k-median cost on both Bike years. (b) RSS error relative to the optimal rank-two error, with uniform count 100 per row; the dotted line is optimal. Bars average 30 paired seeds with 95% intervals. Full baseline, count-profile, and timing comparisons are in Appendices D.4 and D.6.

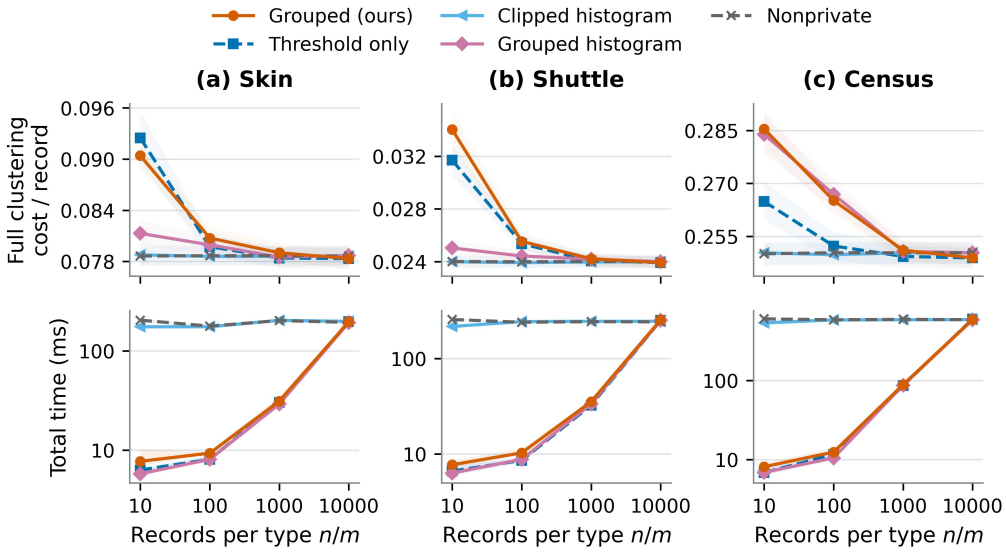


Figure 17: Metric k-median with Zipf-1 counts, $m = 32{,}768$, $K = 128$, $k = 8$, $T = 64$, and $\epsilon = 1$. Top: full clustering cost per record. Bottom: total algorithm milliseconds. Increasing n/m changes multiplicities rather than the number of public vectors. Bands show pointwise 95% intervals.

The finite separation family uses $N = 8$, $q \in \{8, 16, 32, 64, 128, 256, 512, 1024\}$, $\beta_g = q^{-2}$, $\epsilon_g = \epsilon_s = 1/2$, $\kappa = 2$, $B = 3\lambda$, and $M = q(\lambda - 1)$. Within-group distance is q^{-2}; all distances between different groups and all distances to o are one. The input index j is sampled uniformly from $\{1, \ldots, N\}$. We simulate the thresholding step followed by one center sampled proportionally to u, using 1,000 seeds per setting, for 32,000 method executions. The histogram uses one full-budget noisy count vector followed by the same one-center sampling rule. The uniform-group control chooses a public group uniformly. The cost curve includes all threshold outcomes; the sampling-probability curve uses the runs with $H = \{o\}$, with their number recorded in the accompanying data. Thus the measured probability is compared with the corresponding exact formula.

The sampling-rule tests enumerate 5,184 neighboring pairs under a common threshold outcome H, including zero-total groups and representatives that also belong to H. They verify the common sampling set K_H, one-coordinate changes in u, and the unit bound in Lemma 4.2. Twelve joint-volume probability checks compare neighboring-input likelihoods on small enumerated instances. Ten checks compare the original and grouped objectives; four RSS checks verify the residual sample count and sampling-step accounting. The implementation checks also verify the configured privacy

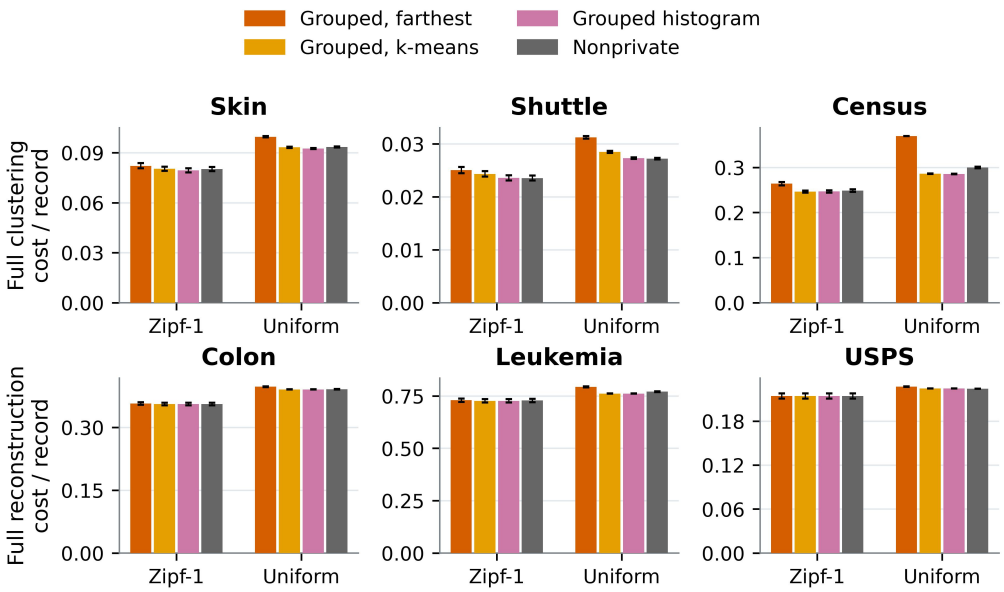


Figure 18: Public partition comparison at $K = 128$, $n/m = 100$, and $\epsilon = 1$. GW uses farthest-point or public k-means groups; the grouped histogram uses the latter partition. All six dictionaries and both count profiles are included. Bars show mean original-weight cost from 30 fresh paired seeds, with pointwise 95% intervals; axes begin at zero. Construction costs and the T_H/n and $T_{2,H}/n$ measurements are in Appendix D.3.

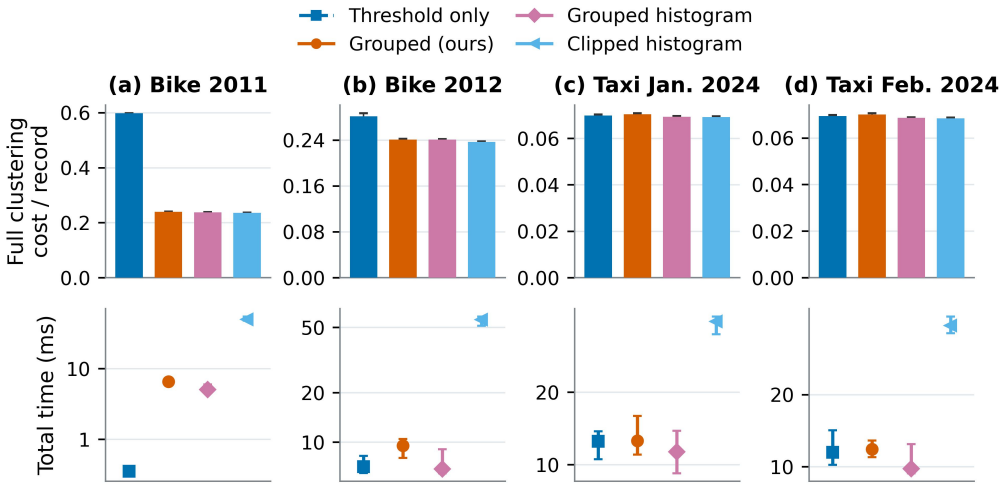


Figure 19: Observed-count k-median for all four periods, with $k = 8$, $T = 64$, $K = 128$, and $\epsilon = 1$. Top: original-weight cost per record, with bars starting at zero. Bottom: total milliseconds on a logarithmic scale. Error bars are pointwise 95% intervals over 30 algorithm seeds. The method order is shared across panels.

charge and row cap. These checks confirm that the implementation matches the formulas used in the analysis; the privacy guarantees are established by the proofs above.

The implementation uses Float64 arithmetic. Original weights, losses, group totals, and fitted reconstruction matrices are used only for offline evaluation and are not part of the analyzed output. The accompanying artifacts contain the raw runs, configu-

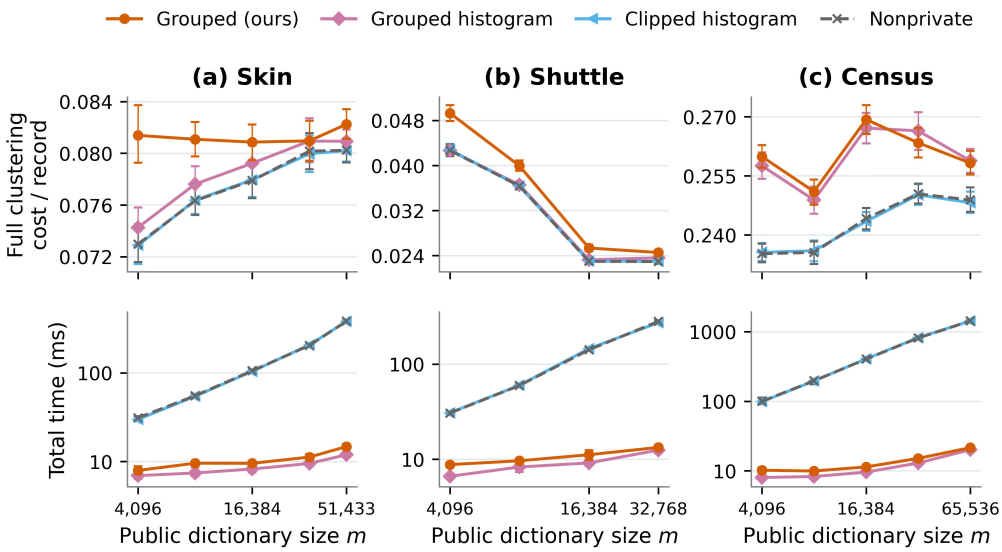


Figure 20: Public dictionary size at fixed Zipf-1 density $n/m = 100$, $K = 128$, $k = 8$, $T = 64$, and $\epsilon = 1$. Top: original-weight cost per record. Bottom: total algorithm milliseconds. Each point uses 30 fresh paired seeds. Skin ends at 51,433 distinct sites; Census extends to 65,536. Public preprocessing is shown in Figure 14.

rations, and plotting code under `experiments/analysis/grouped_20260922` and `experiments/analysis/revision_20260923`.

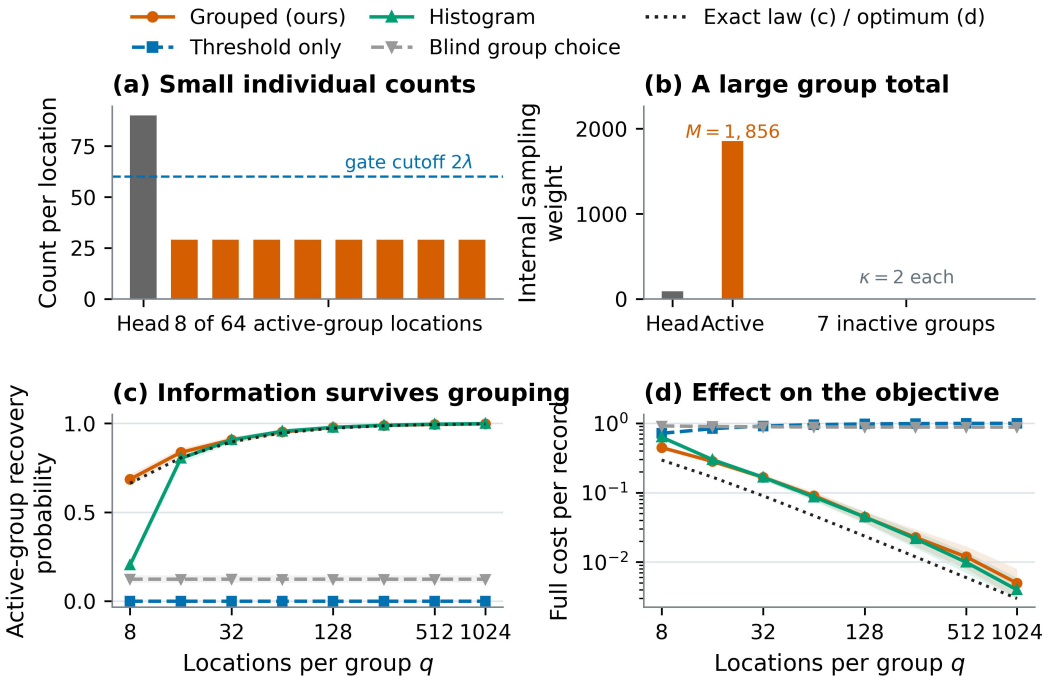


Figure 21: Information retained by grouping for the family in Section 5.2. (a) At $q = 64$, eight displayed locations in B_j each have count $\lambda - 1$. The dashed line marks 2λ, the threshold applied to the noisy counts. (b) The grouped weights u on the event $H = \{o\}$: the representative r_j receives the group total M, while each of the other seven group representatives receives weight κ. (c) Empirical probability of sampling r_j on the event $H = \{o\}$, with 95% Wilson intervals and the exact probability from the grouped weights. (d) Mean original-weight cost over all threshold outcomes, with 95% seed-bootstrap intervals; the dotted curve is the full optimum. The metric has within-group distance q^{-2} and between-group distance one. Each point uses 1,000 draws. This is a one-center experiment, separate from the $k \geq 2$ local-search benchmark.