Supplementary Material

A. Missing Proofs

A.1. Proof of Lemma 8and Lemma 9

To prove Lemma 9, we will use the following fact, which is simple to verify.

Fact 23. *For any $\xi > 0$ and any $a, b \in \mathbb{R}$, $(a+b)^2 \leq (1+\xi)a^2 + (1+1/\xi)\cdot b^2$.*

Proof of Lemma 9. We have

$$\begin{aligned}
\mathrm{cost}_{\mathbf{S}}(\phi \circ \Psi, \mathbf{C}) &= \sum_{y \in \mathbb{B}^d} w_{\mathbf{S}}(y) \cdot \|y - c_{\phi(\Psi(y))}\|^2 \\
&= \sum_{x \in \mathbb{B}^d} \sum_{y \in \Psi^{-1}(x)} w_{\mathbf{S}}(y) \cdot \|y - c_{\phi(x)}\|^2 \\
\text{(Triangle Inequality)} &\leq \sum_{x \in \mathbb{B}^d} \sum_{y \in \Psi^{-1}(x)} w_{\mathbf{S}}(y) \cdot \left(\|x - c_{\phi(x)}\| + \|x - y\|\right)^2 \\
\text{(Fact 23)} &\leq \sum_{x \in \mathbb{B}^d} \sum_{y \in \Psi^{-1}(x)} w_{\mathbf{S}}(y) \cdot \left((1+\xi)\cdot\|x - c_{\phi(x)}\|^2 + (1+1/\xi)\cdot\|x - y\|^2\right) \\
&= (1+\xi)\cdot\left(\sum_{x \in \mathbb{B}^d} w_{\mathbf{S}}(\Psi^{-1}(x)) \cdot \|x - c_{\phi(x)}\|^2\right) \\
&\qquad + (1+1/\xi)\cdot\left(\sum_{y \in \mathbb{B}^d} w_{\mathbf{S}}(y)\cdot\|\Psi(y) - y\|^2\right) \\
&\leq (1+\xi)\cdot\left(\sum_{x \in \mathbb{B}^d} w_{\mathbf{S}'}(x)\cdot\|x - c_{\phi(x)}\|^2\right) \\
&\qquad + 4(1+\xi)\cdot\left(\sum_{x \in \mathbb{B}^d} |w_{\mathbf{S}}(\Psi^{-1}(x)) - w_{\mathbf{S}'}(x)|\right) \\
&\qquad + (1+1/\xi)\cdot\left(\sum_{y \in \mathbb{B}^d} w_{\mathbf{S}}(y)\cdot\|\Psi(y) - y\|^2\right) \\
&\leq (1+\xi)\cdot\mathrm{cost}_{\mathbf{S}'}(\phi, \mathbf{C}) + 4(1+1/\xi)\cdot\mathrm{mt}(\Psi, \mathbf{S}, \mathbf{S}'),
\end{aligned}$$

yielding the first inequality.

To prove the second inequality, let $\phi_* : \mathbb{B}^d \to [k]$ denote the map of each point to its closest center in $\mathbf{C}$. We have

$$\begin{aligned}
\mathrm{cost}_{\mathbf{S}'}(\mathbf{C}) &= \sum_{x \in \mathbb{B}^d} w_{\mathbf{S}'}(x)\cdot\|x - c_{\phi_*(x)}\|^2 \\
&\leq 4\left(\sum_{x \in \mathbb{B}^d} |w_{\mathbf{S}}(\Psi^{-1}(x)) - w_{\mathbf{S}'}(x)|\right) + \sum_{x \in \mathbb{B}^d} w_{\mathbf{S}}(\Psi^{-1}(x))\cdot\|x - c_{\phi_*(x)}\|^2 \\
&= 4\left(\sum_{x \in \mathbb{B}^d} |w_{\mathbf{S}}(\Psi^{-1}(x)) - w_{\mathbf{S}'}(x)|\right) + \sum_{x \in \mathbb{B}^d} \sum_{y \in \Psi^{-1}(x)} w_{\mathbf{S}}(y)\cdot\|x - c_{\phi_*(x)}\|^2 \\
&\leq 4\left(\sum_{x \in \mathbb{B}^d} |w_{\mathbf{S}}(\Psi^{-1}(x)) - w_{\mathbf{S}'}(x)|\right) + \sum_{x \in \mathbb{B}^d} \sum_{y \in \Psi^{-1}(x)} w_{\mathbf{S}}(y)\cdot\|x - c_{\phi_*(y)}\|^2
\end{aligned}$$

$$\begin{aligned}
\text{(Triangle Inequality)} &\leq 4\left(\sum_{x \in \mathbb{B}^d} |w_{\mathbf{S}}(\Psi^{-1}(x)) - w_{\mathbf{S}'}(x)|\right) \\
&\quad + \sum_{x \in \mathbb{B}^d} \sum_{y \in \Psi^{-1}(x)} w_{\mathbf{S}}(y) \cdot (\|y - c_{\phi_*(y)}\| + \|x - y\|)^2 \\
&\leq 4\left(\sum_{x \in \mathbb{B}^d} |w_{\mathbf{S}}(\Psi^{-1}(x)) - w_{\mathbf{S}'}(x)|\right) \\
&\quad + \sum_{x \in \mathbb{B}^d} \sum_{y \in \Psi^{-1}(x)} w_{\mathbf{S}}(y) \cdot \left((1+\xi) \cdot \|y - c_{\phi_*(y)}\|^2 + (1 + 1/\xi) \cdot \|x - y\|^2\right) \\
&= (1+\xi) \cdot \left(\sum_{y \in \mathbb{B}^d} w_{\mathbf{S}}(y) \cdot \|y - c_{\phi_*(y)}\|^2\right) \\
&\quad + 4\left(\sum_{x \in \mathbb{B}^d} |w_{\mathbf{S}}(\Psi^{-1}(x)) - w_{\mathbf{S}'}(x)|\right) \\
&\quad + (1 + 1/\xi) \cdot \left(\sum_{y \in \mathbb{B}^d} w_{\mathbf{S}}(y) \cdot \|\Psi(y) - y\|^2\right) \\
&\leq (1+\xi) \cdot \mathrm{cost}_{\mathbf{S}}(\mathbf{C}) + 4(1 + 1/\xi) \cdot \mathrm{mt}(\Psi, \mathbf{S}, \mathbf{S}'),
\end{aligned}$$

as desired. □

With Lemma 9 ready, we turn our attention back to the proof of Lemma 8.

Proof of Lemma 8. Consider any (ordered) set $\mathbf{C}$ of centers. Let ϕ_* be the mapping that maps every point to its closest center in $\mathbf{C}$. On the one hand, applying the second inequality of Lemma 9, we get

$$\begin{aligned}
\mathrm{cost}_{\mathbf{S}'}(\mathbf{C}) &\leq (1 + 0.5\xi) \cdot \mathrm{cost}_{\mathbf{S}}(\mathbf{C}) + 4(1 + 2/\xi) \cdot \mathrm{mt}(\Psi, \mathbf{S}, \mathbf{S}') \\
&\leq (1 + \xi) \cdot \mathrm{cost}_{\mathbf{S}}(\mathbf{C}) + 4(1 + 2/\xi)t,
\end{aligned}$$

where the second inequality follows from the assumed upper bound on $\mathrm{mt}(\mathbf{S}, \mathbf{S}')$.

Let $\Psi : \mathbb{B}^d \to \mathbb{B}^d$ be such that $\mathrm{mt}(\Psi, \mathbf{S}, \mathbf{S}') = \mathrm{mt}(\mathbf{S}, \mathbf{S}')$. We can use the first inequality of Lemma 9 to derive the following inequalities.

$$\begin{aligned}
\mathrm{cost}_{\mathbf{S}'}(\mathbf{C}) &= \mathrm{cost}_{\mathbf{S}'}(\phi_*, \mathbf{C}) \\
\text{(Lemma 9)} &\geq \frac{1}{1 + 0.5\xi} \cdot \mathrm{cost}_{\mathbf{S}}(\phi_* \circ \Psi) - \frac{4(1 + 2/\xi)}{1 + 0.5\xi} \cdot \mathrm{mt}(\Psi, \mathbf{S}, \mathbf{S}') \\
&\geq (1 - 0.5\xi) \cdot \mathrm{cost}_{\mathbf{S}}(\phi_* \circ \Psi) - 4(1 + 2/\xi) \cdot \mathrm{mt}(\mathbf{S}, \mathbf{S}').
\end{aligned}$$

As a result, we can conclude that $\mathbf{S}'$ is an $(\xi, 4(1 + 2/\xi)t)$-coreset of $\mathbf{S}$ as desired. □

A.2. Proof of Theorem 20

Proof.

- First, from the operation of Algorithm 2, we have $\tau_i \leq \Gamma k a \leq 2^{O_\xi(d)} \cdot k \cdot (\log n)$.

  By how $\mathcal{T}$ is constructed, the number of internal nodes is $\tau_1 + \cdots + \tau_T$, which is at most $T\Gamma k a \leq 2^{O_\xi(d)} \cdot k \cdot (\log n)^2$. Finally, since by Lemma 15 the branching factor B is also just $2^{O(d)}$, the total number of nodes is indeed $2^{O_\xi(d)} \cdot k \cdot (\log^2 n)$.

- Using Lemma 16, we have

$$\begin{aligned}\operatorname{mt}(\Psi_{\mathcal{T}}, \mathbf{S}, \mathbf{S}_{\mathcal{T}}) &\leq \left(\sum_{z \in \text{leaves}(\mathcal{T})} |f_z - \tilde{f}_z|\right) + \sum_{z \in \text{leaves}(\mathcal{T})} f_z \cdot (4\rho^2_{\text{level}(z)}) \\ &\leq 4\left(\sum_{z \in \text{leaves}(\mathcal{T})} \tilde{f}_z \cdot (\rho^2_{\text{level}(z)})\right) + 2^{O_\xi(d)} \cdot k \cdot (\log^2 n) \cdot \eta, \qquad (4)\end{aligned}$$

  where the second inequality follows from the bound on the number of nodes in the first item and the η-accuracy guarantee of the frequency oracle.

To bound the summation term on the right hand side of (4), we may rearrange it as

$$\sum_{z \in \text{leaves}(\mathcal{T})} \tilde{f}_z \cdot (\rho^2_{\text{level}(z)}) = \left(\sum_{i \in [T-1]} 2^{-2i} \cdot \left(\sum_{z \in \text{leaves}(\mathcal{T}_i)} \tilde{f}_z\right)\right) + 2^{-2T} \cdot \sum_{z \in \text{leaves}(\mathcal{T}_T)} \tilde{f}_z. \qquad (5)$$

Using Lemma 19, we may bound the first term above by

$$\begin{aligned}&\left(\sum_{i \in [T-1]} 2^{-2i} \cdot \left(\sum_{z \in \text{leaves}(\mathcal{T}_i)} \tilde{f}_z\right)\right) \\ &\leq \sum_{i \in [T-1]} 2^{-2i} \cdot \left(2 \operatorname{bottom}_{m_i - \tau_i - ka}\left((\tilde{f}_z)_{z \in \text{leaves}(\mathcal{T}_i)}\right) + (n + |\mathcal{T}_i|\eta)/2^{\Gamma}\right) \\ &\leq \left(\sum_{i \in [T-1]} 2^{1-2i} \cdot \operatorname{bottom}_{m_i - \tau_i - ka}\left((f_z)_{z \in \text{leaves}(\mathcal{T}_i)}\right)\right) + O\left(1 + 2^{O_\xi(d)} \cdot k \cdot (\log^2 n)\right),\end{aligned}$$

where the second inequality follows from our choice of Γ and the fact that $\eta \leq n$ which may be assumed without loss of generality (otherwise, we might just let the frequency oracle be zero everywhere) and the bound on the number of nodes in $\mathcal{T}$ from the first item. Next, to bound the first summation term above, let $r_i := \theta \cdot 2^{-i}$. We have

$$\begin{aligned}&\left(\sum_{i \in [T-1]} 2^{1-2i} \cdot \operatorname{bottom}_{m_i - \tau_i - ka}\left((f_z)_{z \in \text{leaves}(\mathcal{T}_i)}\right)\right) \\ \text{(Corollary 18)} &\leq \left(\sum_{i \in [T-1]} 2^{1-2i}/r_i^2 \cdot \mathrm{OPT}^k_{\mathbf{S} \cap \Psi_{\mathcal{T}}^{-1}(\text{leaves}(\mathcal{T}_i))}\right) \\ \text{(Our choice of } \theta) &= \frac{\xi}{32(1 + 2/\xi)} \cdot \left(\sum_{i \in [T-1]} \mathrm{OPT}^k_{\mathbf{S} \cap \Psi_{\mathcal{T}}^{-1}(\text{leaves}(\mathcal{T}_i))}\right) \\ &\leq \frac{\xi}{32(1 + 2/\xi)} \cdot \left(\mathrm{OPT}^k_{\bigcup_{i \in [T-1]}\left(\mathbf{S} \cap \Psi_{\mathcal{T}}^{-1}(\text{leaves}(\mathcal{T}_i))\right)}\right) \\ &\leq \frac{\xi}{32(1 + 2/\xi)} \cdot \mathrm{OPT}^k_{\mathbf{S}}.\end{aligned}$$

Finally, we may bound the second term in (5) by

$$2^{-2T} \cdot \sum_{z \in \text{leaves}(\mathcal{T}_T)} \tilde{f}_z \leq (n + |\mathcal{T}_T| \cdot \eta)/n \leq 2^{O_\xi(d)} \cdot k \cdot (\log^2 n),$$

where we used the bound $2^{-2T} \leq 1/n$ which follows from our choice of T.

Combining the above four inequalities together, we get

$$\mathrm{mt}(\Psi_{\mathcal{T}}, \mathbf{S}, \mathbf{S}_{\mathcal{T}}) \leq \frac{\xi}{8(1+2/\xi)} \cdot \mathrm{OPT}^k_{\mathbf{S}} + 2^{O_\xi(d)} \cdot k \cdot (\log^2 n) \cdot \eta.$$

- Applying Lemma 8 to the above inequality implies that $S_{\mathcal{T}}$ is a $(\xi, 2^{O_\xi(d)} \cdot k \cdot (\log^2 n) \cdot \eta)$-coreset of S as desired.

In terms of the running time, it is obvious that apart from Line 7 in Algorithm 1, all other steps run in time $\mathrm{poly}(|\mathcal{T}|)$, which is at most $\mathrm{poly}(N_T)$ times the running time of a frequency oracle call. As for Line 7 in Algorithm 1, we may compute the set $\mathrm{children}(z)$ for some node $z \in \mathcal{L}_j$ as follows. First, we use Lemma 10 to compute the set $\mathrm{cand}(z)$ of all nodes $z' \in \mathcal{L}_{j+1}$ such that $|z - z'| \leq \rho_j$; this takes time $2^{O(d)}$. Next, for each $z' \in \mathrm{cand}(z)$, we check whether z is its closest point in $\mathcal{L}_j$, which once again can be done via Lemma 10 in time $2^{O(d)}$. Thus, each execution of Line 7 in Algorithm 1 takes only $2^{O(d)}$ time; hence, in total this step only takes $2^{O(d)} \cdot |\mathcal{T}| \leq \mathrm{poly}(N_T)$ time. □

A.3. Proof of Theorem 22

To prove this theorem, we will also use the following simple well-known fact (see e.g., Aggarwal et al., 2009, Proposition 1), which tell us an excess in k-means objective for each cluster in terms of the distance between the true center and the noised center.

Fact 24. *For any weighted point set* $\mathbf{S}$ *and* $c \in \mathbb{R}^d$*,* $\mathrm{cost}_{\mathbf{S}}(c) - \mathrm{OPT}^1_{\mathbf{S}} = |\mathbf{S}| \cdot \|\mu(\mathbf{S}) - c\|^2$.

Proof of Theorem 22. Since each of $\mathrm{Enc}^{\mathrm{hist}}_{(\epsilon/2,\delta/2)}, \mathrm{Enc}^{\mathrm{vec}}_{(\epsilon/2,\delta/2)}$ is $(\epsilon/2, \delta/2)$-DP, basic composition theorem immediately implies that $\mathrm{CLUSTERINGENCODER}_{\epsilon,\delta}$ is (ϵ, δ)-DP.

Next, notice that we only call the oracles $\tilde{f}$ (resp. $\tilde{v}$) on the nodes of the tree $\mathcal{T}$. Since the number of nodes is at most $N_{\mathcal{T}}$, a union bound ensures that all of these queries provide η-accurate (resp. $\tilde{\eta}$-accurate) answers with probability at least $1 - 0.1\beta$. Henceforth, we may assume that such an accuracy guarantee holds for all queries.

For notational convenience, let $\mathbf{S} = \{x_1, \dots, x_n\}, \mathbf{S}' = \{x'_1, \dots, x'_n\}, \tilde{\mathbf{S}} = \{\tilde{x}_1, \dots, \tilde{x}_n\}, \mathbf{C} = \{c_1, \dots, c_k\}$, and $\mathbf{C}' = \{c'_1, \dots, c'_k\}$.

From Theorem 20, we have $\mathbf{S}_{\mathcal{T}}$ is a $\left(0.1\alpha, 2^{O_\alpha(d')} \cdot k \cdot (\log^2 n) \cdot \eta\right)$-coreset of $\mathbf{S}'$. From this and from the κ-approximation guarantee of algorithm $\mathcal{A}$, we have

$$\mathrm{cost}_{\mathbf{S}_{\mathcal{T}}}(\mathbf{C}') \leq \kappa \cdot \mathrm{OPT}^k_{\mathbf{S}_{\mathcal{T}}} \leq \kappa(1 + 0.1\alpha)\,\mathrm{OPT}^k_{\mathbf{S}} + \kappa \cdot 2^{O_\alpha(d')} \cdot k \cdot (\log^2 n) \cdot \eta.$$

Let $\phi_* : \mathrm{leaves}(\mathcal{T}) \to [k]$ denote the mapping from each leaf to its closest center, and let $\phi' : \mathbf{S}' \to [k]$ be $\phi' := \phi_* \circ \Psi_{\mathcal{T}}$. With this notation, we have that the following holds with probability at least $1 - 0.1\beta$:

$$\begin{aligned}
\mathrm{cost}_{\mathbf{S}'}(\phi') &= \mathrm{cost}_{\mathbf{S}'}(\phi_* \circ \Psi_{\mathcal{T}}) \\
&\leq \mathrm{cost}_{\mathbf{S}'}(\phi_* \circ \Psi_{\mathcal{T}}, \mathbf{C}') \\
\text{(Lemma 9)} &\leq (1 + 0.1\alpha) \cdot \mathrm{cost}_{\mathbf{S}_{\mathcal{T}}}(\phi_*, \mathbf{C}') + 4(1 + 1/\xi) \cdot \mathrm{mt}(\Psi_{\mathcal{T}}, \mathbf{S}', \mathbf{S}_{\mathcal{T}}) \\
\text{(Theorem 20)} &\leq (1 + 0.1\alpha) \cdot \mathrm{cost}_{\mathbf{S}_{\mathcal{T}}}(\phi_*, \mathbf{C}') \\
&\quad + 4(1 + 1/\xi) \cdot \left(\frac{\xi}{8(1+2/\xi)} \cdot \mathrm{OPT}^k_{\mathbf{S}} + 2^{O_\xi(d)} \cdot k \cdot (\log^2 n) \cdot \eta\right) \\
\text{(From } \xi = 0.1\alpha\text{)} &\leq (1 + 0.15\alpha) \cdot \mathrm{cost}_{\mathbf{S}_{\mathcal{T}}}(\phi_*, \mathbf{C}') + 2^{O_\alpha(d')} \cdot k \cdot (\log^2 n) \cdot \eta \\
\text{(guarantee of } \mathcal{A}\text{)} &\leq \kappa(1 + 0.15\alpha) \cdot \mathrm{OPT}^k_{\mathbf{S}_{\mathcal{T}}} + 2^{O_\alpha(d')} \cdot k \cdot (\log^2 n) \cdot \eta \\
\text{(Theorem 20)} &\leq \kappa(1 + 0.15\alpha)(1 + 0.1\alpha) \cdot \mathrm{OPT}^k_{\mathbf{S}'} + 2^{O_\alpha(d')} \cdot k \cdot (\log^2 n) \cdot \eta \\
&\leq \kappa(1 + 0.3\alpha) \cdot \mathrm{OPT}^k_{\mathbf{S}'} + 2^{O_\alpha(d')} \cdot k \cdot (\log^2 n) \cdot \eta.
\end{aligned}$$

For notational convenience, we let $\tilde{\phi}$ (respectively ϕ) denote the canonical mapping from $\tilde{\mathbf{S}}$ (respectively $\mathbf{S}$) to $[k]$ corresponding to ϕ'. Standard concentration inequalities imply that with probability $1 - 0.1\beta$ we have $\|\tilde{x}_i\| \leq 1/\Lambda$, meaning that $x'_i = \Lambda \tilde{x}_i$ for all $i \in [n]$. When this holds, we simply have

$$\mathrm{cost}_{\mathbf{S}'}(\phi') = \Lambda^2 \cdot \mathrm{cost}_{\tilde{\mathbf{S}}}(\tilde{\phi}).$$

Plugging this into the previous inequality, we have

$$\mathrm{cost}_{\tilde{\mathbf{S}}}(\tilde{\phi}) = \kappa(1 + 0.3\alpha) \cdot \mathrm{OPT}^k_{\tilde{\mathbf{S}}} + (1/\Lambda^2) \cdot 2^{O_\alpha(d')} \cdot k \cdot (\log^2 n) \cdot \eta. \tag{6}$$

Next, applying Theorem 21, the following holds with probability $1 - 0.1\beta$:

$$\mathrm{cost}_{\tilde{\mathbf{S}}}(\tilde{\phi}) \geq (d/d') \cdot \mathrm{cost}_{\mathbf{S}}(\phi)/(1 + 0.1\alpha),$$

and

$$\mathrm{OPT}^k_{\tilde{\mathbf{S}}} \leq (1 + 0.1\alpha)(d/d') \cdot \mathrm{OPT}^k_{\mathbf{S}}.$$

Plugging the above two inequalities into (6), we arrive at

$$\begin{aligned}\mathrm{cost}_{\mathbf{S}}(\phi) &\leq \kappa(1 + 0.3\alpha)(1 + 0.1\alpha)^2 \, \mathrm{OPT}^k_{\mathbf{S}} + (d'/d) \cdot (1/\Lambda^2) \cdot 2^{O_\alpha(d')} \cdot k \cdot (\log^2 n) \cdot \eta \\ &\leq \kappa(1 + \alpha) \, \mathrm{OPT}^k_{\mathbf{S}} + 2^{O_\alpha(d')} \cdot k \cdot (\log^2 n) \cdot \log(n/\beta) \cdot \eta. \end{aligned} \tag{7}$$

Next, we will bound $\mathrm{cost}_{\mathbf{S}}(\mathbf{C})$ in comparison to $\mathrm{cost}_{\mathbf{S}}(\phi)$ that we had already bounded above. To do this, first notice that

$$\begin{aligned}\mathrm{cost}_{\mathbf{S}}(\mathbf{C}) \leq \mathrm{cost}_{\mathbf{S}}(\phi, \mathbf{C}) &= \sum_{j \in [k]} \mathrm{cost}_{\phi^{-1}(j)}(c_j) \\ \text{(Fact 24)} &= \sum_{j \in [k]} \left(\mathrm{OPT}^1_{\phi^{-1}(j)} + |\phi^{-1}(j)| \cdot \|\mu(\phi^{-1}(j)) - c_j\|^2 \right) \\ &= \mathrm{cost}_{\mathbf{S}}(\phi) + \left(\sum_{j \in [k]} |\phi^{-1}(j)| \cdot \|\mu(\phi^{-1}(j)) - c_j\|^2 \right). \end{aligned} \tag{8}$$

Furthermore, since we assume that the vector summation oracle is $\tilde{\eta}$-accurate, using the triangle inequality we get

$$\|v^j - \tilde{v}^j\| \leq \tilde{\eta} \cdot |\mathcal{T}|. \tag{9}$$

Similarly, let $n^j = \phi^{-1}(j)$. From the fact that the frequency oracle is η-accurate, we have

$$|\tilde{n}^j - n^j| \leq \eta \cdot |\mathcal{T}|. \tag{10}$$

We next consider two cases, based on how large n^j is.

- Case I: $n^j \leq 2(\eta + \tilde{\eta}) \cdot |\mathcal{T}|$. In this case, we simply use the straightforward fact that $\|\mu(\phi^{-1}(j)) - c_j\| \leq 2$. This gives

$$|\phi^{-1}(j)| \cdot \|\mu(\phi^{-1}(j)) - c_j\|^2 \leq 8(\eta + \tilde{\eta}) \cdot |\mathcal{T}|.$$

- Case II: $n^j > 2(\eta + \tilde{\eta}) \cdot |\mathcal{T}|$. In this case, first notice that

$$\begin{aligned}\|\mu(\phi^{-1}(j)) - c_j\| &\leq \|\mu(\phi^{-1}(j)) - \tilde{c}_j\| \\ &= \left\| \frac{v^j}{n^j} - \frac{\tilde{v}^j}{\tilde{n}^j} \right\| \\ &= \frac{1}{n^j \tilde{n}^j} \cdot \|v^j \tilde{n}^j - \tilde{v}^j n^j\| \\ \text{(From the triangle inequality, (10), } n^j > 2\eta \cdot |\mathcal{T}|) &\leq \frac{2}{(n^j)^2} \cdot \left(\|v^j(\tilde{n}^j - n^j)\| + \|(\tilde{v}^j - v^j)n^j\| \right)\end{aligned}$$

$$\text{(From (9), (10), and } \|v^j\| \leq n^j) \leq \frac{2}{(n^j)^2} \cdot \left(n^j \cdot \eta \cdot |\mathcal{T}| + \tilde{\eta} \cdot |\mathcal{T}| \cdot n^j\right) = \frac{2(\eta + \tilde{\eta})|\mathcal{T}|}{n^j}.$$

From this, we have

$$|\phi^{-1}(j)| \cdot \|\mu(\phi^{-1}(j)) - c_j\|^2 \leq n^j \cdot \frac{4(\eta + \tilde{\eta})^2|\mathcal{T}|^2}{(n^j)^2} \leq 2(\eta + \tilde{\eta})|\mathcal{T}|.$$

Thus, in both cases, we have $|\phi^{-1}(j)| \cdot \|\mu(\phi^{-1}(j)) - c_j\|^2 \leq 8(\eta + \tilde{\eta})|\mathcal{T}|$. Plugging this back to (8), we get

$$\text{cost}_{\mathbf{S}}(\mathbf{C}) \leq \text{cost}_{\mathbf{S}}(\phi) + 8k(\eta + \tilde{\eta})|\mathcal{T}| \leq \text{cost}_{\mathbf{S}}(\phi) + 2^{O_\alpha(d')} \cdot k^2 \cdot (\log^2 n) \cdot (\eta + \tilde{\eta}),$$

where the second inequality follows from the first item of Theorem 20. Finally, plugging this into (7), we can conclude that

$$\begin{aligned}\text{cost}_{\mathbf{S}}(\mathbf{C}) &\leq \kappa(1+\alpha)\,\text{OPT}^k_{\mathbf{S}} + 2^{O_\alpha(d')} \cdot k \cdot (\log^2 n) \cdot \log(n/\beta) \cdot \eta \\ &\quad + 2^{O_\alpha(d')} \cdot k^2 \cdot (\log^2 n) \cdot (\eta + \tilde{\eta}) \\ \text{(From } d = O_\alpha(\log k)) &\leq \kappa(1+\alpha)\,\text{OPT}^k_{\mathbf{S}} + k^{O_\alpha(1)}(\log^2 n)\left(\log(n/\beta) \cdot \eta + \tilde{\eta}\right),\end{aligned}$$

as desired.

The running time claim for the decoder (Algorithm 4) follows immediately from the running time of BUILDTREE from Theorem 20 and from $2^{O(d')} = k^{O_\alpha(1)}$ due to our choice of d'. As for the encoder (Algorithm 3), it is clear that every step runs in $\text{poly}(ndk, \mathtt{t}(\text{Enc}^{\text{hist}}), \mathtt{t}(\text{Enc}^{\text{vec}}))$ time, except Lines 7 and 9 where we need to find a closest point in $\mathcal{L}_j$ from some given point. However, Lemma 10 ensures that this can be computed in time $2^{O(d')}$ which is equal to $k^{O_\alpha(1)}$ due to our choice of parameters. This completes our proof. □

B. Frequency and Vector Summation Oracles in Local Model

In this section, we explain the derivations of the bounds in Section 2.4 in more detail. To do so, we first note that given an algorithm for histogram (resp., bucketized vector summation), we can easily derive an algorithm for generalized histogram (resp., generalized bucketized vector summation) with a small overhead in the error. This is formalized below. (Note that, for brevity, we say that an algorithm runs in time $\mathtt{t}(\cdot)$ if both the encoder and the decoder run in time at most $\mathtt{t}(\cdot)$.) We remark that, while we focus on the local model in this section, Lemma 25 works for any model; indeed we will use it for the shuffle model in the next section.

Lemma 25. *Suppose that there is a $\mathtt{t}(n, Y, \epsilon, \delta)$-time (ϵ, δ)-DP $(\eta(n, \epsilon, \delta), \beta(n, \epsilon, \delta))$-accurate algorithm for histogram. Then, there is an $O(T \cdot \mathtt{t}(nT, Y, \epsilon/T, \delta/T))$-time (ϵ, δ)-DP $(\eta(nT, \epsilon/T, \delta/T), \beta(nT, \epsilon/T, \delta/T))$-accurate algorithm for generalized histogram.*

Similarly, suppose that there is a $\mathtt{t}(n, Y, d, \epsilon, \delta)$-time (ϵ, δ)-DP $(\eta(n, d, \epsilon, \delta), \beta(n, d, \epsilon, \delta))$-accurate algorithm for bucketized vector summation. Then, there is an $O(T \cdot \mathtt{t}(nT, Y, d, \epsilon/T, \delta/T))$-time (ϵ, δ)-DP $(\eta(nT, d, \epsilon/T, \delta/T), \beta(nT, d, \epsilon/T, \delta/T))$-accurate algorithm for generalized bucketized vector summation.

Proof. Suppose there is an (ϵ, δ)-DP $(\eta(n, \epsilon, \delta), \beta(n, \epsilon, \delta))$-accurate algorithm for histogram. To solve generalized histogram, each user runs the T encoders in parallel, each on an element $y \in Y_i$ and with $(\epsilon/T, \delta/T)$-DP. By basic composition, this algorithm is (ϵ, δ)-DP. On the decoder side, it views the randomized input as inputs from nT users and then runs the standard decoder for histogram. As a result, this yields an $(\eta(nT, \epsilon/T, \delta/T), \beta(nT, \epsilon/T, \delta/T))$-accurate algorithm for generalized histogram. The running time claim also follows trivially.

The argument for bucketized vector summation is analogous to the above. □

The above lemma allows us to henceforth only focus on histogram and bucketized vector summation.

B.1. Histogram Frequency Oracle

In this subsection, we briefly recall a frequency oracle of Bassily et al. (2020) (called ExplicitHist in their paper), which we will extend in the next section to handle vectors. We assume that the users and the analyzer have access to public randomness in the form of a uniformly random $Z \in \{\pm 1\}^{|Y| \times n}$. Note that while this requires many bits to specify, as noted in Bassily et al. (2020), for the purpose of the bounds below, it suffices to take Z that is pairwise independent in each column (but completely independent in each row) and thus it can be compactly represented in $O(n \log |Y|)$ bits.

The randomizer and analyzer from Bassily et al. (2020) can be stated as follows.

Algorithm 5 ExplicitHist Encoder

```
procedure $\textsc{ExplicitHistEncoder}_\epsilon(x_i; Z)$
    $\tilde{x}_i \leftarrow Z_{x_i,i}$
    $y_i = \begin{cases} \tilde{x}_i & \text{with probability } \frac{e^\epsilon}{e^\epsilon+1} \\ -\tilde{x}_i & \text{with probability } \frac{1}{e^\epsilon+1} \end{cases}$
    return $y_i$
```

Algorithm 6 ExplicitHist Decoder.

```
procedure $\textsc{ExplicitHistDecoder}_\epsilon(v; y_1, \dots, y_n; Z)$
    return $\frac{e^\epsilon+1}{e^\epsilon-1} \cdot \sum_{i \in [n]} y_i \cdot Z_{v,i}$
```

Bassily et al. (2020) proved the following guarantee on the above frequency oracle:

Theorem 26. ExplicitHist *is an* $(O(\sqrt{n \log(|Y|/\beta)}/\epsilon), \beta)$*-accurate* ϵ*-DP algorithm for histogram in the local model. Moreover, it can be made to run in time* $\mathrm{poly}(n, \log |Y|)$.

The above theorem in conjunction with the first part of Lemma 25 implies Theorem 12.

B.2. Vector Summation Oracle

Duchi et al. (2013) proved the following lemma[10], which can be used to aggregate d-dimensional vectors of bounded Euclidean norm.

Lemma 27 ((Duchi et al., 2013))**.** *For every* $d \in \mathbb{N}$ *and* $\epsilon \in (0, O(1))$*, there exists* $B = \Theta(\sqrt{d}/\epsilon)$ *such that there is a polynomial-time* ϵ*-DP algorithm* $\mathcal{R}^{\mathrm{vec}}_\epsilon$ *in the local model that, given an input vector* x*, produces another vector* z *such that* $\|z\| = B$ *and* $\mathbb{E}[z] = x$.

Notice that this algorithm allows us to compute an estimate of a sum of vectors by simply adding up the randomized vectors; this gives an error of $O(\sqrt{dn}/\epsilon)$. Below we combine this with the techniques of Bassily et al. (2020) to get a desired oracle for vector summation. Specifically, the algorithm, which we call ExplicitHistVector, are presented below (where y_i is the bucket and x_i is the vector input).

Algorithm 7 ExplicitHistVector Encoder

```
procedure $\textsc{ExplicitHistVectorEncoder}_\epsilon(y_i, x_i; Z)$
    $z_i \leftarrow \mathcal{R}_{\mathrm{vec}}(Z_{y_i,i} \cdot x_i)$
    return $z_i$
```

Lemma 28. ExplicitHistVector *is an* $(O(\sqrt{nd \log(d|Y|/\beta)}/\epsilon), \beta)$*-accurate* ϵ*-DP algorithm for bucketized vector summation in the local model. Moreover, it can be made to run in time* $\mathrm{poly}(nd, \log |Y|)$.

To prove Lemma 28, we require a concentration inequality for sum of independent vectors, as stated below. It can be derived using standard techniques (see e.g., Jin et al., 2019, Corollary 7, for an even more general form of the inequality).

[10] See expression (19) and Lemma 1 of Duchi et al. (2013). Note that we use $L = 1$; their choice of $B = \frac{e^\epsilon+1}{e^\epsilon-1} \cdot \frac{\pi\sqrt{d}\Gamma\left(\frac{d-1}{2}+1\right)}{\Gamma\left(\frac{d}{2}+1\right)}$ indeed satisfies $B = O(\sqrt{d})$.

Algorithm 8 ExplicitHistVector Decoder.

```
procedure $\textsc{ExplicitHistVectorDecoder}_\epsilon(v; z_1, \dots, z_n; Z)$
    return $\sum_{i \in [n]} z_i \cdot Z_{v,i}$
```

Lemma 29. *Suppose that $u_1, \dots, u_n \in \mathbb{R}^d$ are random vectors such that $\mathbb{E}[u_i] = \mathbf{0}$ for all $i \in [n]$ and that $\|u_i\| \leq \sigma$. Then, with probability $1 - \beta$, we have $\left\|\sum_{i \in [n]} u_i\right\| \leq O\left(\sigma\sqrt{n \log(d/\beta)}\right)$.*

Proof of Lemma 28. Since we only use the input (x_i, y_i) once as the input to $\mathcal{R}^{\text{vec}}_\epsilon$ and we know that $\mathcal{R}^{\text{vec}}_\epsilon$ is ϵ-DP, we can conclude that $\textsc{ExplicitHistVectorDecoder}$ is also ϵ-DP.

To analyze its accuracy, consider any $v \in Y$. For $i \in [n]$, let

$$u_i = \begin{cases} z_i \cdot Z_{v,i} - x_i & \text{if } v \in V, \\ z_i \cdot Z_{v,i} & \text{if } v \notin Y. \end{cases}$$

Notice that the error of our protocol at v is exactly $\sum_{i \in [n]} u_i$. Furthermore, from the guarantee of $\mathcal{R}^{\text{vec}}_\epsilon$, it is not hard to see that $\mathbb{E}[u_i] = \mathbf{0}$ and that $\|u_i\| \leq \|z_i - x_i\| \leq O(\sqrt{d}/\epsilon)$. As a result, applying Lemma 29, we can conclude that with probability $1 - \beta$, $\left\|\sum_{i \in [n]} u_i\right\| \leq O(\sqrt{nd \log(d/\beta)}/\epsilon)$, as desired.

Similar to $\textsc{ExplicitHist}$, it suffices to take Z that is pairwise independent in each column, which can be specified in $O(n \log |Y|)$ bits. As a result, the total running time of the algorithm is $\text{poly}(nd, \log |Y|)$ as desired. □

Lemma 28 and the second part of Lemma 25 imply Lemma 13.

C. Shuffle Model

In this section, we derive our bounds for the shuffle DP model (Theorem 2).

The shuffle DP model (Bittau et al., 2017; Erlingsson et al., 2019; Cheu et al., 2019) has gained traction due to it being a middle-ground between the central and local DP models. In the shuffle DP model, a shuffler sits between the encoder and the decoder; this shuffler randomly permutes the messages from the encoders before sending it to the decoder (aka *analyst*). Two variants of this model have been studied in the literature: in the single-message model, each encoder can send one message to the shuffler and in the multi-message model, each encoder can send multiple messages to the shuffler. As in the local DP model, a one-round (i.e., non-interactive) version of the shuffle model can be defined as follows. (Here we use $\mathcal{X}$ to denote the set of possible inputs and $\mathcal{Y}$ to denote the set of possible messages.)

Definition 30. *For an n-party protocol $\mathcal{P}$ with encoding function Enc that produces m messages per user, and for an input sequence $\mathbf{x} \in \mathcal{X}^n$, we let $S^{\text{Enc}}_{\mathbf{x}}$ denote the distribution on $\mathcal{Y}^{nm}$ obtained by applying Enc on each element of x and then randomly shuffling the outputs.*

Definition 31 (Shuffle DP)**.** *A protocol $\mathcal{P}$ with encoder $\text{Enc} : \mathcal{X} \to \mathcal{Y}^m$ is (ϵ, δ)-DP in the shuffle model if the algorithm whose output distribution is $S^{\text{Enc}}_{\mathbf{x}}$ is (ϵ, δ)-DP.*

Recent research on the shuffle DP model includes work on aggregation (Balle et al., 2019; Ghazi et al., 2020d; Balle et al., 2020; Ghazi et al., 2021b), histograms and heavy hitters (Ghazi et al., 2021a; Balcer and Cheu, 2020; Ghazi et al., 2020a;c), and counting distinct elements (Balcer et al., 2021; Chen et al., 2021).

C.1. Frequency and Vector Summation Oracles

We start by providing algorithms for frequency and vector summation oracles in the shuffle DP model. These are summarized below.

Theorem 32 ((Ghazi et al., 2021a))**.** *There is an $\left(O(\text{poly}\log\left(\frac{|Y|T}{\delta\beta}\right)/\epsilon), \beta\right)$-accurate (ϵ, δ)-DP algorithm for generalized histogram in the shuffle model. The encoder and the decoder run in time $\text{poly}\left(nT/\epsilon, \log\left(\frac{|Y|}{\delta\beta}\right)\right)$.*

Theorem 33. *There is an* $(O(\frac{T\sqrt{d}}{\epsilon} \cdot \mathrm{poly}\log(dT/(\delta\beta))), \beta)$*-accurate* (ϵ, δ)*-DP algorithm for generalized bucketized vector summation in the shuffle model. The encoder and the decoder run in time* $\mathrm{poly}(ndT/\beta, \log\left(\frac{|Y|}{\epsilon\delta}\right))$.

We will prove these two theorems in the next two subsections.

C.1.1. Frequency Oracle

(Ghazi et al., 2021a) gave a frequency oracle for histogram with the following guarantee:

Theorem 34 ((Ghazi et al., 2021a))**.** *There is an* $(O(\mathrm{poly}\log\left(\frac{|Y|}{\delta\beta}\right)/\epsilon), \beta)$*-accurate* (ϵ, δ)*-DP algorithm for histogram in the shuffle model. The encoder and the decoder run in time*
$\mathrm{poly}\left(n/\epsilon, \log\left(\frac{|Y|}{\delta\beta}\right)\right)$.

We note that as stated in Ghazi et al. (2021a), the protocol underlying Theorem 34 uses $\mathrm{poly}(|Y| \cdot n \cdot \log(1/\beta))$ bits of public randomness. This can be exponentially reduced using the well-known fact that pairwise independence is sufficient for the Count Sketch data structure (which is the basis of the proof of Theorem 34).

Combining Theorem 34 and Lemma 25 yields Theorem 32.

C.1.2. Vector Summation Oracle

For any two probability distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, we denote by $\mathrm{SD}(\mathcal{D}_1, \mathcal{D}_2)$ the statistical (aka total variation) distance between $\mathcal{D}_1$ and $\mathcal{D}_2$.

We next present a bucketized vector summation oracle in the shuffle model that exhibits almost central accuracy.

Theorem 35. *There is an* $(O(\frac{\sqrt{d}}{\epsilon} \cdot \mathrm{poly}\log(d/(\delta\beta))), \beta)$*-accurate* (ϵ, δ)*-DP algorithm for bucketized vector summation in the shuffle model. The encoder and the decoder run in time* $\mathrm{poly}(nd/\beta, \log\left(\frac{|Y|}{\epsilon\delta}\right))$.

The rest of this subsection is decidated to the proof of Theorem 35; note that the theorem and the second part of Lemma 25 immediately imply Theorem 33.

In order to prove Theorem 35, we recall the following theorem about the analysis of a variant of the split-and-mix protocol of Ishai et al. (2006). For more context, see e.g., (Ghazi et al., 2020d; Balle et al., 2020) and the references therein.

We start by defining the notion of secure protocols; roughly, their transcripts are statistically indistinguishable when run on any two inputs that have the same function value.

Definition 36 (σ-secure shuffle protocols)**.** *Let* ν *be a positive real number. A one-round shuffle model protocol* $\mathcal{P} = (\mathrm{Enc}, \mathcal{A})$ *is said to be* ν-secure *for computing a function* $f : \mathcal{X}^n \to \mathbb{Z}$ *if for any* $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^n$ *such that* $f(\mathbf{x}) = f(\mathbf{x}')$*, we have* $\mathrm{SD}(S_{\mathbf{x}}^{\mathrm{Enc}}, S_{\mathbf{x}'}^{\mathrm{Enc}}) \le 2^{-\nu}$.

Theorem 37 ((Ghazi et al., 2020d; Balle et al., 2020))**.** *Let* n *and* q *be positive integers, and* $\nu > 0$ *be a real number. The split-and-mix protocol of Ishai et al. (2006) (Algorithm 9) with* n *parties and inputs in* $\mathbb{F}_q$ *is* ν*-secure for* $f(\mathbf{x}) = \sum_{i=1}^n x_i$ *when* $m \ge O(1 + \frac{\nu + \log q}{\log n})$.

We will also use the discrete Gaussian distribution from Canonne et al. (2020).

Definition 38 ((Canonne et al., 2020))**.** *Let* μ *and* $\sigma > 0$ *be real numbers. The* discrete Gaussian distribution *with location* μ *and scale* σ*, denoted by* $\mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2)$*, is the discrete probability distribution supported on the integers and defined by*

$$\Pr_{X \sim \mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2)}[X = x] = \frac{e^{-(x-\mu)^2/(2\sigma^2)}}{\sum_{y \in \mathbb{Z}} e^{-(y-\mu)^2/(2\sigma^2)}},$$

for all $x \in \mathbb{Z}$.

We will use the following well-known concentration inequality.

Definition 39 (Bernstein inequality)**.** *Let* $X_1, \dots, X_n$ *be independent zero-mean random variables. Suppose that* $|X_i| \le$

M for all $i \in \{1, \ldots, n\}$, where M is a non-negative real number. Then, for any positive real number t, it holds that

$$\Pr\left[\sum_{i=1}^{n} X_i \geq t\right] \leq \exp\left(-\frac{0.5 \cdot t^2}{\sum_{i=1}^{n} \mathbb{E}[X_i^2] + \frac{1}{3}Mt}\right).$$

We will also need the following tail bounds for Gaussian random variables.

Proposition 40 (Tail bound for continuous Gaussians)**.** *For any positive real number x, it holds that*

$$\Pr_{X \sim \mathcal{N}(0,1)}[X \geq x] \leq \frac{\exp(-x^2/2)}{x\sqrt{2\pi}}.$$

Proposition 41 (Tail bound for discrete Gaussians; Proposition 25 of (Canonne et al., 2020))**.** *For any positive integer m and any positive real number σ, it holds that*

$$\Pr_{X \sim \mathcal{N}_{\mathbb{Z}}(0,\sigma^2)}[X \geq m] \leq \Pr_{X \sim \mathcal{N}(0,\sigma^2)}[X \geq m-1].$$

We will moreover need the following upper bound on the variance of discrete Gaussians.

Proposition 42 (Proposition 21 of (Canonne et al., 2020))**.** *For any positive real number σ, it holds that*

$$\mathrm{Var}_{X \sim \mathcal{N}_{\mathbb{Z}}(0,\sigma^2)}[X] < \sigma^2.$$

Finally, we will use the following theorem quantifying the differential privacy property for the discrete Gaussian mechanism.

Theorem 43 (Theorem 15 of (Canonne et al., 2020))**.** *Let $\sigma_1, \ldots, \sigma_d$ be positive real numbers. Let $Y_1, \ldots, Y_d$ be i.i.d. random variables each drawn from $\mathcal{N}_{\mathbb{Z}}(0, \sigma_j^2)$. Let $M : \mathcal{X}^n \to \mathbb{Z}^d$ be a randomized algorithm given by $M(x) = q(x) + Y$, where $Y = (Y_1, \ldots, Y_d)$. Then, M is (ϵ, δ)-DP if and only if for all neighboring $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^n$, it holds that*

$$\delta \geq \Pr[Z > \epsilon] - e^{\epsilon} \cdot \Pr[Z < -\epsilon],$$

where

$$Z := \sum_{j=1}^{d} \frac{(q(x)_j - q(x')_j)^2 + 2 \cdot (q(x)_j - q(x')_j) \cdot Y_j}{2\sigma_j^2}.$$

Using Theorem 43, we obtain the following:

Corollary 44. *Let $\epsilon > 0$ and $\delta \in (0, 1)$ be given real numbers. Let $\mathcal{X}$ denote the set of all vectors in $\mathbb{Z}^d$ with ℓ_2-norm at most C. Define the randomized algorithm $M : \mathcal{X}^d \to \mathbb{Z}^d$ as $M(x) = q(x) + Y$, where $Y = (Y_1, \ldots, Y_d)$ with $Y_1, \ldots, Y_d$ being i.i.d. random variables each drawn from $\mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$, and where $q(x) = \sum_{i=1}^{n} x_i$ is the vector sum. Then, there exists $\sigma = \frac{10C\log(d/\delta)}{\epsilon}$ for which M is (ϵ, δ)-DP.*

Proof of Corollary 44. Let Z be defined as in Theorem 43. We will show that $\delta \geq \Pr[Z > \epsilon]$, which by Theorem 43 means that the algorithm is (ϵ, δ)-DP. First, note that

$$\mathbb{E}[Z] = \frac{1}{2\sigma^2}\sum_{j=1}^{d}(q(x)_j - q(x')_j)^2 = \frac{\|q(x) - q(x')\|^2}{2\sigma^2}.$$

Moreover, using Proposition 42, we have that

$$\mathrm{Var}[Z] \leq \frac{1}{\sigma^2}\sum_{j=1}^{d}(q(x)_j - q(x')_j)^2 = \frac{\|q(x) - q(x')\|^2}{\sigma^2}.$$

For each $i \in [d]$, define the event $\mathcal{E}_i$ that $|Y_i| \leq M$, where

$$M = 2 \cdot \sigma \cdot \sqrt{2\ln(2d/\delta)}. \tag{11}$$

Moreover, let $\mathcal{E} = \cap_{i=1}^{d} \mathcal{E}_i$. Using the fact that $Y_i \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ with $\sigma \geq 1$, and applying Propositions 40 and 41, we get that $\Pr[\mathcal{E}_i] \geq 1 - \delta/(2d)$. By a union bound, we obtain

$$\Pr[\mathcal{E}] \geq 1 - \delta/2. \tag{12}$$

Henceforth, we condition on the event $\mathcal{E}$. We next argue that conditioning on $\mathcal{E}$ leaves the expectation of Z unchanged, and can only decrease its variance. Namely, since both the random variable Y_j and the event $\mathcal{E}_j$ are symmetric around 0, we have that $\mathbb{E}[Y_j|\mathcal{E}_j] = 0$, and hence,

$$\mathbb{E}[Z|\mathcal{E}] = \sum_{j=1}^{d} \frac{(q(x)_j - q(x')_j)^2 + 2 \cdot (q(x)_j - q(x')_j) \cdot \mathbb{E}[Y_j|\mathcal{E}_j]}{2\sigma_j^2} = \frac{\|q(x) - q(x')\|^2}{2\sigma^2} = \mathbb{E}[Z]. \tag{13}$$

Moreover, we have that

$$\begin{aligned}
\mathrm{Var}[Z|\mathcal{E}] &= \sum_{j=1}^{d} \frac{(q(x)_j - q(x')_j)^2}{\sigma_j^4} \cdot \mathrm{Var}[Y_j|\mathcal{E}_j] \\
&= \sum_{j=1}^{d} \frac{(q(x)_j - q(x')_j)^2}{\sigma_j^4} \cdot \mathbb{E}[Y_j^2|\mathcal{E}_j] \\
&\leq \sum_{j=1}^{d} \frac{(q(x)_j - q(x')_j)^2}{\sigma_j^4} \cdot \mathbb{E}[Y_j^2] \qquad (14) \\
&\leq \frac{\|q(x) - q(x')\|^2}{\sigma^2}, \qquad (15)
\end{aligned}$$

where inequality (14) follows from the definition of the event $\mathcal{E}_j$, and inequality (15) follows from Proposition 42. Applying the Bernstein inequality (Theorem 38), we get that:

$$\Pr[Z > \epsilon|\mathcal{E}] \leq \exp\left(- \frac{0.5 \cdot (\epsilon - \mathbb{E}[Z|\mathcal{E}])^2}{\mathrm{Var}[Z|\mathcal{E}] + C \cdot M \cdot (\epsilon - \mathbb{E}[Z|\mathcal{E}])/(3\sigma^2)} \right) \leq \frac{\delta}{2}, \tag{16}$$

where the last inequality follows from plugging in (11), (13), and (15), and using a sufficiently large $\sigma = \frac{10C \log(d/\delta)}{\epsilon}$. Finally, combining (16) and (12), we deduce that $\Pr[Z > \epsilon] \leq \delta$. □

We are now ready to prove Theorem 35.

Proof of Theorem 35. The pseudocode for the encoder and decoder of the protocol is given in Algorithms 10 and 11 respectively. Also, note that the number t of incoming messages in Algorithm 11 is equal to $s \cdot m$. We point out that the sum in Algorithm 10 is in $\mathbb{F}_q$. We set:

- $\eta \leftarrow \frac{1}{n}$.
- $s \leftarrow \frac{2 \cdot n}{\beta}$.
- $\sigma \leftarrow \frac{20 \log(sd/\delta)}{\epsilon}$.
- $p \leftarrow$ smallest prime larger than $\frac{2n}{\eta} + 20\sigma \log(sd/\beta)$.
- $m \leftarrow O(1 + \frac{\log(2dp/\delta)}{\log n})$.

Privacy Analysis. The analyst observes the output of the shuffler, which is the multiset $\cup_{i=1}^{n} \mathcal{M}_i$ of messages sent by all the users. This is equivalent to observing a vector $a \in \mathbb{Z}^s$, where s is the number of buckets set in Algorithm 10. Let $\nu = \log(2d/\delta)$ and $m = O(1 + \frac{\nu + \log p}{\log n})$. Applying Theorem 37 along with a union bound, we get that the distribution of a is $(d \cdot 2^{-\nu})$-close in statistical distance to the output of the central model protocol that computes the true vector sum and then adds a $\mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ noise random variable to each of the d coordinates. The latter protocol is $(\epsilon, \delta/(2s))$-DP for $\sigma = \frac{20\log(sd/\delta)}{\epsilon}$ by Corollary 44.[11] By a union bound, we thus conclude that the output of the shuffler is (ϵ, δ)-DP.

Utility Analysis. First, since we pick $s = 2n/\beta$, the probability that a collision occurs between the hash value of a given $y \in Y$ and the hash value of another bucket held by one of the users is at most $n/s \leq \beta/2$. In other words, with probability at least $1 - \beta/2$ there is no collision with the given $y \in Y$.

Secondly, by Propositions 41 and 40, we have that with probability at least $1 - \beta/(2s)$, it holds that for a discrete Gaussian added to each coordinate of each bucket during an execution of Algortihm 10, its absolute value is at most $10\sigma \cdot \log(sd/\beta)$. Thus, by a union bound, with probability $1 - \beta/2$, it holds that the noise to each coordinate of each bucket is at most $10\sigma \cdot \log(sd/\beta)$ in absolute value.

When the above two events hold, the error is the sum of four independent components: one due to quantization of the input vectors, and the other due to the privacy noise.

To upper-bound the quantization error, note that the error in each coordinate of each input is at most η. Thus, the error per coordinate of the sum is at most $\eta \cdot n$. Thus, the ℓ_2-error due to quantization in the vector sum of any bucket is at most $\eta \cdot n \cdot \sqrt{d}$. This is at most $O(\sqrt{d})$ for any $\eta = O(1/n)$.

From the second event, we immediately have that the ℓ_2-error due to the privacy noise is at most $10\sigma \cdot \log(sd/\beta) \cdot \sqrt{d}$.

Putting things together, a union bound implies that with probability at least $1 - \beta$, the total ℓ_2-error is at most $O(\sigma \cdot \sqrt{d} \cdot \log(sd/\beta)) = O(\frac{\sqrt{d}}{\epsilon} \cdot \operatorname{poly}\log(d/(\delta\beta)))$.

Communication and Running Time. Each of the n users sends in total $O\left(\frac{n}{\beta} \cdot \log(\frac{nd}{\beta\epsilon\delta})\right)$ messages each consisting of at most $\log(\frac{nd}{\beta\epsilon\delta})$ bits. Note that pairwise independence can be used to reduce the number of random bits of public randomness, resulting in a total running time for the analyst which is $\operatorname{poly}(nd/\beta, \log\left(\frac{|Y|}{\epsilon\delta}\right))$. □

Algorithm 9 Encoder in Split-and-Mix Protocol

```
procedure SPLITANDMIXENCODER($x_i, p, m$)
    Sample $z_1, \ldots, z_{m-1}$ i.i.d. uniformly at random from $\mathbb{Z}_p$.
    $z_m \leftarrow x_i - \sum_{j=1}^{m-1} z_j$.
    return the multiset $\{z_1, \ldots, z_m\}$.
```

Algorithm 10 Encoder in Shuffle DP Protocol for Bucketized Vector Summation

```
1:  procedure SHUFFLEBVSENCODER$_{\epsilon,\delta}$($x; Z, i, y, n, d, \sigma, \eta, s, p, m$)
2:      $\overline{x} \leftarrow \lfloor x/\eta \rfloor$ (i.e., $\overline{x}_i$ is the quantization of $x_i$ up to precision $\eta$).
3:      $\mathcal{M} \leftarrow \{\}$.
4:      for $(\ell, k) \in \{1, \ldots, s\} \times \{1, \ldots, d\}$ :
5:          if $\ell = Z_{1,y}$:
6:              $u_\ell \leftarrow x_k$.
7:          else
8:              $u_\ell \leftarrow 0$.
9:          if $i = 1$:
10:             $u_\ell \leftarrow u_\ell + \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$.
11:         $\mathcal{M} \leftarrow \mathcal{M} \cup (\{\ell\} \times \{k\} \times$ SplitAndMixEncoder($u_\ell, p, m$)).
12:     return $\mathcal{M}$
```

We end by remarking that in Algorithm 10 we let only the first user add the discrete Gaussian noise (Line 10). This is only for simplicity of analysis; it is possible to split the noise between the users and achieve the same asymptotic guarantees

[11] A similar argument with Discrete Laplace, instead of Discrete Gaussian, noise was previously used in the proof of Lemma 4.1 of Balle et al. (2020).

Algorithm 11 Decoder in Shuffle DP Protocol for Bucketized Vector Summation.

```
procedure ShuffleBVSDecoder($((\ell_1, k_1, z_1), \dots, (\ell_t, k_t, z_t); Z, p, y)$)
    for $j = 1, \dots, d$
        $v_j \leftarrow \sum_{i \in \{1,\dots,t\}: \ell_i = Z_{1,y}, k_i = j} z_i$.
        if $v_j > p/2$ : then $\overline{v}_j \leftarrow (v_j - p)\eta$.
        else $\overline{v}_j \leftarrow v_j \eta$.
    return $(v_1, \dots, v_d)$.
```

(see Kairouz et al. (2021)).

C.2. Proof of Theorem 2

Proof of Theorem 2. Let $\beta = 0.1$. From Theorem 12, there is an $(\eta, 0.1\beta/N_T)$-accurate 0.5ϵ-DP algorithm for generalized histogram in the shuffle model with

$$\eta = O\left(\text{poly}\log\left(\frac{|\mathcal{L}_1 \cup \cdots \cup \mathcal{L}_T| \cdot T \cdot N_T}{\delta\beta}\right)/\epsilon\right).$$

Since we set $T = O(\log n)$ (in Theorem 20), $N_T = k^{O_\alpha(1)} \cdot \text{poly}\log n$ by our choice of parameters, and since $|\mathcal{L}_1 \cup \cdots \cup \mathcal{L}_T| \leq \exp(O(Td'))$ by a volume argument, we get $\eta = O(\text{poly}\log(nd/\delta)/\epsilon)$.

Since we set $T = O(\log n)$ (in Theorem 20) and $N_T = k^{O_\alpha(1)} \cdot \text{poly}\log n$ from our choice of parameters, we get $\eta = O(\sqrt{d} \cdot \text{poly}\log(nd/\delta)/\epsilon)$.

Similarly, from Lemma 13, there is an $(\tilde{\eta}, 0.1\beta/N_T)$-accurate 0.5ϵ-DP algorithm for generalized histogram with

$$\tilde{\eta} = O\left(\frac{T\sqrt{d}}{\epsilon} \cdot \text{poly}\log(dT/(\delta\beta))\right),$$

which as before yields $\tilde{\eta} = O(\sqrt{d} \cdot \text{poly}\log(nd/\delta)/\epsilon)$. Plugging this into Theorem 22, we indeed arrive at a one-round (ϵ, δ)-shuffle DP $(\kappa(1+\alpha), k^{O_\alpha(1)} \cdot \sqrt{d} \cdot \text{polylog}(nd/\delta)/\epsilon)$-approximation algorithm for k-means (with failure probability 0.1). It is easy to verify that the encoder and the decoder run in time $\text{poly}(n, d, k^{O_\alpha(1)}, \log(1/\delta))$. □

D. Additional Experiment Details

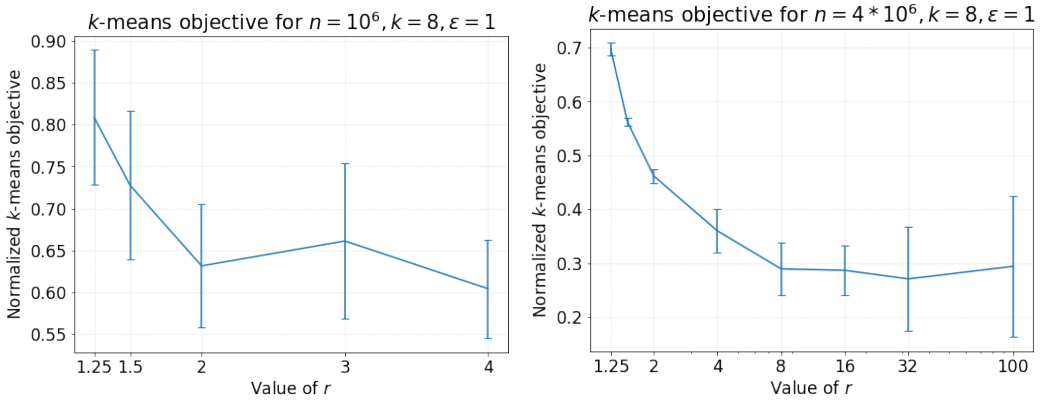

Figure 2. Normalized k-means objective of the output clusters for varying r, the separation parameter of the synthetic dataset. Each set of parameters is run 10 times; the average and the standard deviation of the normalized k-means objectives are included.

Parameter Settings. Our experiments show that bucketized vector summation oracles often contribute to the final objective more than that of the histogram oracle; thus, we allocate more privacy budget to the former compared to the latter (0.9ϵ and 0.1ϵ respectively). We view the number of levels T and thresholds $\tau_1, \dots, \tau_T$ to be hyperparameters and roughly tune them. In the end, we find that $T = \lceil \log_2(k) \rceil + 3$ and only branching when the approximate frequency is at least $1.5\lfloor n/k \rfloor$ give reasonably competitive objectives with little amount of tuning, and the results reported below are for these parameters. Another heuristic we find to be helpful is to split the dataset, instead of the privacy budget ϵ, over the number of levels of the tree, i.e., randomly split the data into T partitions with only the users in the ith partition contributing to the frequency oracle at level-i of the tree.

Effects of Separation of Gaussians. Recall that we use r to denote the ratio between the separation of a of each pair of centers divided by its expected cluster size. While our plots in the main body use $r = 100$, we remark that such a large ratio is often unnecessary. Specifically, when $k = 8$, we often observe that $r \geq 8$ already gives essentially as good a result as $r = 100$; this is presented in Figure 2.