Nearly-Linear Time and Massively Parallel Algorithms for k-Anonymity

Kevin Aydin *
Google Research
kaydin@google.com

Honghao Lin†
Carnegie Mellon University
honghaol@andrew.cmu.edu

David P. Woodruff
CMU & Google Research
dwoodruf@andrew.cmu.edu

Peilin Zhong
Google Research
pz2225@columbia.edu

Abstract

k-anonymity is a widely-used privacy-preserving concept that ensures each record in a dataset is indistinguishable from at least $k-1$ other records. We revisit k-anonymity by suppression and give an $O(k)$-approximation algorithm with a nearly-linear runtime of $\widetilde{O}(nd + n \cdot (n/k)^{1/C^2+o(1)})$ for any constant C, where n is the number of records and d is the number of attributes. Previous algorithms with provable guarantees either (1) achieve the same $O(k)$ approximation ratio but require at least $O(n^2k)$ runtime, or (2) provide a better $O(\log k)$ approximation ratio at the cost of an impractical $O(n^{2k})$ worst-case runtime for general d and k. Our algorithm extends to the Massively Parallel Computation (MPC) model, where it gives an MPC algorithm requiring $\widetilde{O}(\log^{1+\varepsilon} n)$ rounds and total space $O(n^{1+\gamma}(d+k))$. Empirically, we also demonstrate that our algorithmic ideas can be adapted to existing heuristic methods, leading to significant speed-ups while preserving comparable performance. On the hardness side, we study the related single-point k-anonymity problem, where the goal is to select $k-1$ additional records to make a given record indistinguishable. Assuming the dense vs random conjecture in complexity theory, we show that for $n = k^c$, no algorithm can achieve a $k^{1-O(1/c)}$ approximation in $\mathrm{poly}(n)$ time, providing evidence for the inherent hardness of the k-anonymity problem.

1 Introduction

As data becomes increasingly central to decision-making, research, and business intelligence, ensuring privacy while preserving data utility has become a critical challenge. Many datasets contain sensitive information, such as health records, financial transactions, or social behavior. However, even after removing direct identifiers (e.g., names, social security numbers, etc.), inadequate safeguards can still lead to re-identification. To mitigate these risks, privacy-preserving data publishing techniques have become essential for balancing data utility and privacy protection, with k-anonymity [Swe02] standing out as a foundational approach.

k-anonymity, introduced by [Swe02], ensures that each record in a dataset is indistinguishable from at least $k-1$ other records based on a set of quasi-identifiers. These quasi-identifiers, such as age, ZIP code, and gender, may not be uniquely identifying on their own but can enable re-identification when combined with external data sources. By applying generalization and suppression techniques,

*Equal Contribution.

†Part of the work was done while Honghao Lin was a student researcher in Google Research.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

k-anonymity reduces the risk of re-identification while preserving data utility for analysis. An example can be found in Table 1. In this paper, we will only consider the case of suppression where each entry of every attribute is either included in the output, or replaced with the '$\star$' character.

Most research on k-anonymity has focused on finding the optimal (or near-optimal) k-anonymous dataset. That is, the one that minimizes the number of hidden attributes and thereby best preserves the original data. The work of [MW04] demonstrated that finding the optimal solution is NP-hard but provided an $O(k \log k)$-approximation algorithm with a runtime exponential in k. Later, [AFK+05] improved this to an $O(k)$-approximation with a runtime of $O(n^2 k)$. Subsequently, [PS07, KT12] further enhanced the approximation to $O(\log k)$, though their algorithm has a worst-case runtime of $O(n^{2k})$. In addition to algorithms with provable guarantees, other studies have proposed heuristics for various anonymization approaches. For example [LDR06] introduced a heuristic algorithm for k-anonymization of quasi-identifiers, utilizing a construction similar to k-d trees, [DXTK15] by freeform generalization and [BKBL07, ZWL+18] proposed heuristics based on clustering.

Age	Marital status	Home country	Gender
20∼29	Single	USA	Male
30∼39	Divorce	China	Female
20∼29	Single	USA	Female
30∼39	Separation	Korea	Female

Age	Marital status	Home country	Gender
20∼29	Single	USA	$\star$
30∼39	$\star$	$\star$	Female
20∼29	Single	USA	$\star$
30∼39	$\star$	$\star$	Female

Table 1: An example of 2-anonymization [PS07]

Despite the extensive body of research on k-anonymity, many questions and challenges remain unresolved. First, the fastest algorithm for k-anonymity with a provable guarantee has a runtime of $O(n^2 k)$, where n is the number of data points. As the size of data continues to grow in many scenarios, even an $O(n^2)$ runtime may become impractical. This raises the question of whether an algorithm with linear runtime in n is possible.

> *Question* 1*: Is there an algorithm for k-anonymity that runs in linear time in n while providing a provable approximation guarantee?*

Second, to the best of our knowledge, no work has studied k-anonymity in the context of massively parallel computation. In fact, most existing algorithms for k-anonymity rely on sequential processing. Therefore, there is strong motivation to develop parallel algorithms that can leverage the power of distributed computing frameworks to achieve faster and more efficient solutions.

> *Question* 2*: Is it possible to design an algorithm for k-anonymity in the massively parallel computation model while minimizing the number of communication rounds?*

Also, although [PS07] introduced improvements to achieve more practical runtimes for their $O(\log k)$-approximation algorithm, these improvements may only be effective when the dimension d of each point is small. Moreover, their worst-case runtime still remains $O(n^{2k})$. A natural question arises:

> *Question* 3*: Is it possible to develop an algorithm with an $o(k)$ approximation ratio and a worst-case runtime polynomial in n, d, and k?*

1.1 Our Contributions

We present the first $O(k)$-approximation algorithm for k-anonymity with a nearly-linear runtime.

Theorem 1.1. *Given a table T with n records $r_i \in \Sigma^d$ $(i = 1, 2, \ldots, n)$, there is an algorithm that runs in time $\widetilde{O}\left(nd + n \cdot (n/k)^{1/C^2 + o(1)}\right)$ and with high probability outputs an $O(k)$-approximation to the k-anonymity problem on T, where the constant hidden in the approximation ratio depends on C.*

For C large, the running time approaches $\widetilde{O}(nd + n^{1+o(1)})$, which is nearly the time to read the input. Our algorithm can also be extended to the Massively Parallel Computation model (Section A.1) with a number of communication rounds that is logarithmic in n.

Theorem 1.2. *Given a table T with n records $r_i \in \Sigma^d$ $(i = 1, 2, \ldots, n)$, let $\gamma, \varepsilon \in (0, 1)$. There is a fully scalable MPC algorithm that outputs an $O\left(\frac{\log^2(1/\varepsilon)}{\gamma} \cdot k\right)$-approximation to the k-anonymity*

problem on T with high probability. The algorithm takes $O\left(\frac{\log 1/\varepsilon}{\gamma} \cdot \log^{1+\varepsilon}(n) \log\log(n)\right)$ parallel time and $\widetilde{O}\left(nd + n^{1+\gamma+o(1)} \cdot k\right)$ total space.

To obtain a better understanding of our third question, we propose and study the following single-point k-anonymity problem, where the goal is to select $k-1$ additional records to make a given record indistinguishable while minimizing the number of hidden attributes among these k points. Assuming the dense vs. random conjecture in complexity theory, we show that for $n = k^C$, no algorithm can achieve a $k^{1-O(1/c)}$ approximation in $\mathrm{poly}(k)$ time.

Theorem 1.3. *Assume Conjecture 4.3, and let $n = k^C$ and $d \geq k$. There is no algorithm which runs in polynomial n time that with high probability can output a $k^{1-O(1/C)}$-approximation to the single-point k-anonymity problem, even if each record is binary.*

Theorem 1.3 provides evidence of the inherent hardness of the k-anonymity problem. We remark that an open question here is whether we can extend this lower bound to the original k-anonymity problem and obtain a similar hardness.

1.2 Related Work

In addition to the work mentioned above, several studies have explored special cases of k-anonymity. For example, [AFK+05] proposed a 1.5-approximation algorithm for 2-anonymity and a 2-approximation algorithm for 3-anonymity. Similarly, [BDVDP13] presented a polynomial-time algorithm for the case when both d and $|\Sigma|$ are constant.

On the hardness side, [BDVDP13] showed that finding the optimal solution is $w[1]$-hard with respect to the value of the solution (and k). Furthermore, [BDVD11] demonstrated that c-approximation is hard for a fixed constant c in the following cases: (1) $|\Sigma| = 2$ and $k = 3$, or (2) $d \geq 8$ and $k = 4$.

Several studies have extended the definition of k-anonymity by introducing additional constraints, such as l-diversity [MKGV07] and t-closeness [LLV06], to enhance privacy protection for non-quasi-identifiers. Additionally, [CFL10] introduced the concept of k-isomorphism in social network graphs, which means that the graph can be decomposed into a union of k distinct isomorphic subgraphs. Recently, the work of [EEMM24] studied the smooth k-anonymity problem on a binary dataset, where they provide a detailed discussion comparing k-anonymity and differential privacy. In particular, in [EEMM24] the authors formally prove the following:

Theorem 1.4. *Let $\mathcal{M}$ be an arbitrary mechanism that satisfies ϵ-edge differential privacy. Then, in order to achieve $E[J(\mathcal{M}(G), G)] \geq \alpha$, it must hold that $\epsilon = \Omega(\log(\alpha^2 nm))$, where $J(\cdot,\cdot)$ denotes the Jaccard similarity.*

This result suggests that achieving high utility (i.e., preserving the structure of the original graph) under differential privacy requires a large value of ϵ. However, when ϵ is large, the algorithm likely maintains the graph G unmodified thus exposing users to re-identification risks. Recall that the guarantee provided by ϵ-DP is: $\Pr[\mathcal{M}(G) \in A] \leq e^{\epsilon} \Pr[\mathcal{M}(G') \in A]$, which becomes nearly vacuous for large ϵ.

In contrast, k-anonymity offers a different privacy-utility tradeoff. Prior work shows that if the optimal anonymized graph E_{opt} satisfies $J(E, E_{opt}) \geq 1 - O(1/\log k)$, then efficient algorithms can find solutions with comparable utility. Furthermore, in the case of smooth k-anonymity, where edge additions and deletions are allowed, the required assumption can be relaxed to $J(E, E_{opt}) \geq O(1)$.

2 Preliminaries

Notation. In the k-anonymity problem, we are given a table T having n records, each with d attributes. A record $r_i \in T$ is drawn from Σ^d, where Σ is a finite set of possible values for each attribute. Then $r_i[j]$ is the value of the j-th attribute in r_i. Let $\star$ be a symbol not in Σ. Given a record $r \in \Sigma^d$, let its binary expansion be $x \in \{0,1\}^{d|\Sigma|}$, where for $i \in [d], j \in [|\Sigma|]$, $p_{i,j} = 1$ if and only if the i-th attribute corresponds to the j-th value in Σ. Given two records $x, y \in \Sigma^m$, let their ℓ_0 distance be $\mathrm{dist}_{\ell_0}(x, y)$, which is the number of attributes for which x and y differ.

Given two vectors $x, y \in \mathbb{R}^m$, their ℓ_2 distance is $\mathrm{dist}_{\ell_2}(x, y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$. Given a finite point set $P \subseteq \mathbb{R}^m$, let $\rho_k(r)$ be the distance between r and its k-th nearest neighbor in P in ℓ_2 distance. Specifically, if $r \in P$ we let $\rho_1(r) = 0$ (i.e., r's 1-st nearest neighbor is r itself).

2.1 k-Anonymity

Definition 2.1 (k-Suppression Function)**.** A k-suppression function f maps each $r_i \in T$ to r'_i, by replacing some attributes of r_i by $\star$. Moreover, for every $r \in T$, there exist $k-1$ other $r_1, r_2, \ldots, r_{k-1} \in T$ such that $f(r) = f(r_1) = f(r_2) = \cdots = f(r_{k-1})$. Define $c(f)$ to be the cost of f on T, i.e., the number of attributes in T replaced by f, where note that if the same attribute is changed in multiple records, its contribution to the cost is the number of records it is changed in.

Definition 2.2 (k-anonymity via Suppression)**.** In the k-anonymity problem, we are given a table of n records and an anonymity parameter k. Our goal is to obtain a k-suppression function f so that $c(f)$ is minimized. Specifically, we say f is a C-approximation if $c(f) \leq C \cdot \min_{f'} c(f')$.

3 Nearly-Linear Time Algorithm for k-Anonymity

In this section, we present our nearly-linear time approximation algorithm for k-anonymity. At a high level, we will first show that achieving an $O(k)$ approximation for k-anonymity can be reduced to solving the minimum-size constrained clustering problem with an $O(1)$ pointwise guarantee under the squared ℓ_2 distance metric. Then, we will give an algorithm that solves this problem in nearly-linear time with high probability.

3.1 Reduction to Minimum Size Constrained Clustering

Recall that we have n records r_i $(i = 1, 2, \ldots, n)$ in table T. For each record $r_i \in \Sigma^d$, let $x_i \in \{0,1\}^{d|\Sigma|}$ be its binary expansion (i.e., a one-hot encoding of r_i, see definition in Section 2) and let S denote the set of binary expansions of all records. Then for each pair of records (r_i, r_j), the number of differing attributes between r_i and r_j is given by:

$$\mathrm{dist}_{\ell_0}(r_i.r_j) = \frac{1}{2}\mathrm{dist}_{\ell_0}(x_i, x_j) = \frac{1}{2}\mathrm{dist}_{\ell_2}(x_i, x_j)^2 \ .$$

For each record r_i, let r_j be its k-th nearest neighbor in T with respect to the ℓ_0 distance (i.e., the number of attributes on which the records differ, and recall that in our definition, the 1-st nearest neighbor of r_i is r_i itself). Consider an arbitrary partition of the k-anonymity problem on T. Since the group containing r_i has at least k records, the number of attributes that need to be suppressed for k-anonymity is at least $\mathrm{dist}_{\ell_0}(r_i, r_j) = \frac{1}{2}\rho_k(x_i)^2$. In the following lemma, we demonstrate that if there exists a partition $\mathcal{P} = \{P_1, P_2, \ldots, P_t\}$ with centers $c(P_1), c(P_2), \ldots, c(P_t)$ on S, such that for every point $p \in S$ in group P_j, the squared ℓ_2 distance $\mathrm{dist}_{\ell_2}(p, c(P_j))^2$ is at most $O(1) \cdot \rho_k(p)^2$, then this partition gives an $O(k)$-approximation to the k-anonymity problem on T. Formally, we have

Lemma 3.1. *Suppose that $\mathcal{P} = \{P_1, P_2, \ldots, P_t\}$ is a partition on S with centers $c(P_1), c(P_2), \ldots, c(P_t)$ such that for every P_i, $k \leq |P_i| \leq 2k-1$ and for every $p \in P_i$,*

$$\mathrm{dist}_{\ell_2}(p, c(P_i))^2 \leq C \cdot \rho_k(p)^2 \ ,$$

for some constant C. Then, the partition $\mathcal{P}$ is an $O(k)$-approximate solution for the k-anonymity problem on T.

Proof. Let $\mathcal{Q} = \{Q_1.Q_2, \cdots, Q_s\}$ with centers $c(Q_1), c(Q_2), \cdots, c(Q_s)$ be the optimal solution to the k-anonymity problem on T. For each point $p \in P_i$, let $N(p, P_i)$ denote the number of attributes we need to suppress for p with group P_i.

We first upper bound the total number of suppressed attributes over the partition $\mathcal{P}$. Given a point $p \in P_i$, recall that p is a one-hot encoding of some record in table T and let $r(p) \in T$ be the record that p corresponds to. We have

$$\begin{aligned} N(p, P_i) &\leq \sum_{q \in P_i, q \neq p} \mathrm{dist}_{\ell_0}(r(p), r(q)) \leq \sum_{q \in P_i, q \neq p} \frac{1}{2} \cdot \mathrm{dist}_{\ell_0}(p, q) = \sum_{q \in P_i, q \neq p} \frac{1}{2} \cdot \mathrm{dist}_{\ell_2}(p, q)^2 \\ &\leq \sum_{q \in P_i, q \neq p} \left(\mathrm{dist}_{\ell_2}(p, c(P_i))^2 + \mathrm{dist}_{\ell_2}(q, c(P_i))^2\right) \end{aligned}$$

The first inequality is due to the fact that if we need to hide the attribute j, then there must be at least one $r(q)$ in T such that the j-th attribute of $r(p)$ and $r(q)$ differ. Taking a sum over all P_i and $p \in P_i$,

and noting that we have $|P_i| \leq 2k-1$, we get that

$$\sum_{P_i \in \mathcal{P}} \sum_{p \in P_i} N(p, P_i) \leq (4k-4) \sum_{P_i \in \mathcal{P}} \sum_{p \in P_i} \mathrm{dist}_{\ell_2}(p, c(P_i))^2 \ .$$

This implies

$$\sum_{P_i \in \mathcal{P}} \sum_{p \in P_i} N(p, P_i) \leq (4k-4) \cdot C \sum_{p \in S} \rho_k(p)^2 \tag{1}$$

from the assumption of the clustering solution. We next turn to lower bound the total number of hidden attributes over the partition $\mathcal{Q}$. Give a $q \in Q_i$, let p denote its k-th nearest neighbor in S. Then we have

$$N(q, Q_i) \geq \max_{p' \in Q_i} \frac{1}{2} \cdot \mathrm{dist}_{\ell_0}(p', q) \ .$$

Since there are at least k points in Q_i, we have

$$N(q, Q_i) \geq \max_{p' \in Q_i} \frac{1}{2} \mathrm{dist}_{\ell_0}(p', q) \geq \frac{1}{2} \mathrm{dist}_{\ell_0}(p, q) = \frac{1}{2} \rho_k(q)^2$$

The last equation holds because both p and q are one-hot encodings of some records. Taking a sum over all Q_i and $q \in Q_i$, we get that

$$\frac{1}{2} \sum_{q \in S} \rho_k(q)^2 \leq \sum_{Q_i \in \mathcal{Q}} \sum_{q \in Q_i} N(q, Q_i) \tag{2}$$

Combining (1) and (2) we get that

$$\sum_{P_i \in \mathcal{P}} \sum_{p \in P_i} N(p, P_i) \leq (8k-8) \cdot C \sum_{Q_i \in \mathcal{Q}} \sum_{q \in Q_i} N(q, Q_i) \ ,$$

which is what we need. □

We next note that the condition $|P_i| \leq 2k-1$ can effectively be removed. If a group P_i contains more than $2k-1$ points, it can be divided into multiple smaller groups arbitrarily, each satisfying the condition.

Corollary 3.2. *Suppose that $\mathcal{P} = \{P_1, P_2, \ldots, P_t\}$ is a partition on S with centers $c(P_1), c(P_2), \ldots, c(P_t)$ such that for every P_i, $|P_i| \geq k$ and for every $p \in P_i$,*

$$\mathrm{dist}_{\ell_2}(p, c(P_i))^2 \leq C \cdot \rho_k(p)^2 \ ,$$

for some constant C. Then the partition $\mathcal{P}$ can be efficiently transferred to another partition $\mathcal{P}'$, which is an $O(k)$-approximate solution to the k-anonymity problem on T.

Finally, since the input to this clustering problem is the binary expansion of each record r_i (which is in $d|\Sigma|$ dimensions) but not the record itself, naïvely it yields a $|\Sigma|$ factor in time and space, which can be large in practice. However, note that in the entire proof of Corollary 3.2, all we care about are the pair-wise distances. Consequently, we can use the following Johnson-Lindenstrauss lemma to reduce the dimension of each point to $O(\log n)$. Recall that since each of the input binary expansions is d-sparse, we can compute each embedding Φx_i in $O(d \log n)$ time.

Lemma 3.3 (Johnson-Lindenstrauss lemma, [JLS86])**.** *Let $\Phi \in \mathbb{R}^{r \times d}$ be a matrix whose entries are i.i.d samples from $\mathcal{N}(0, 1/r)$. For every vector $u \in \mathbb{R}^d$ and $\varepsilon \in (0,1)$, we have* $\Pr[(1-\varepsilon)\|u\|_2 \leq \|\Phi u\|_2 \leq (1+\varepsilon)\|u\|_2] \geq 1 - \exp(\Omega(\varepsilon^2 r))$.

3.2 Solving the Minimum Size Constrained Clustering Problem

After establishing Corollary 3.2, our goal shifts to finding a partition $\mathcal{P} = \{P_1, P_2, \ldots, P_t\}$ on an $O(\log n)$-dimensional pointset S with centers $c(P_1), \ldots, c(P_t)$ such that $|P_i| \geq k$, and for every $p \in P_i$,

$$\mathrm{dist}_{\ell_2}(p, c(P_i))^2 \leq C \cdot \rho_k(p)^2 \ .$$

In the remainder of this section, we shall present an algorithm that solves this problem in time $\widetilde{O}(nd + n \cdot (n/k)^{1/C^2 + o(1)})$. Our algorithm is inspired by the work of [EMMZ22] that studies this problem in the MPC setting, and their algorithm is not hard to adapt into an $\widetilde{O}(nd + n^{1+1/C^2+o(1)} \cdot k)$-time algorithm. This is at least nk time, which can be as large as n^2 for $k = \Theta(n)$. We will significantly improve upon this runtime by using random sampling to reduce k-th nearest neighbor computations to 1-st nearest neighbor computations, described below.

We need the definition of locality sensitive hashing (LSH):

Lemma 3.4 ([AI08, And09]). *Let $S = \{p_1, p_2, \cdots, p_n\} \subset \mathbb{R}^d$. Given two parameters $R > 0$ and $C > 1$, there is a hash family $\mathcal{H}$ such that $\forall p, q \in S$:*

1. *If $\|p - q\|_2 \le R$, then $\Pr_{h\in\mathcal{H}}[h(p) = h(q)] \ge \mathcal{P}_1$ where $\mathcal{P}_1 \ge 1/n^{1/C^2+o(1)}$.*
2. *If $\|p - q\|_2 \ge c_u \cdot C \cdot R$, then $\Pr_{h\in\mathcal{H}}[h(p) = h(q)] \le \mathcal{P}_2$ where $\mathcal{P}_2 \le 1/n^4$ and $c_u > 1$ is a universal constant.*

Moreover, each hash function can be generated and evaluated in $n^{o(1)}d$ time.

We now present our algorithm. At a high level, our algorithm can be divided into the following steps:

1. Sample a random subset $J \subseteq S$ with $|J| = O((n \log n)/k)$. For each $p_i \in J$, compute an $O(1)$-approximation to $\rho_k(p_i)$. Denote this distance by d_i. Also compute a set N_i, where $|N_i| \ge k - 1$ and for every $q \in N_i$ we have $\mathrm{dist}_{\ell_2}(p_i, q) \le O(1) \cdot \rho_k(p_i)$ (Lemma 3.6).
2. For each point q not in J, find a $f(q)$ in J satisfying $\mathrm{dist}_{\ell_2}(q, f(q)) \le O(1) \cdot \min_{p\in J} \mathrm{dist}_{\ell_2}(q, p)$. For each $p \in J$, define $F(p) = \{q \in S \setminus J \mid f(q) = p\}$ (Lemma 3.7).
3. For $R_j = 3^j$, for $j = 1, 2, \ldots, O(\log d)$:
   (a) Let $A \subseteq J$ be the set of point p_i where $R_{j-1} < \widetilde{\rho}_k(p_i) \le R_j$ and p_i has not been assigned.
   (b) Let $B \subseteq A$ be the set of point p_i where none of the points in N_i has been assigned in the previous iteration.
   (c) Greedily find one maximal independent set C of B such that the points in C do not share the same point in their neighborhood set N_i. For each point $p_i \in C$, create a new cluster centered at p_i, assign points in N_i to p_i.
   (d) For each point p in $B \setminus C$, assume it shares the same neighbor with the point $s \in C$, assign p to center s.
   (e) For each point p_i in $A \setminus B$, assume one of its neighbor in N_i has been assigned to center s, assign p_i to s.
   (f) Furthermore, suppose that $p \in J$ has been assigned to center s, assign all unassigned points q in $F(p)$ to the same cluster if $\mathrm{dist}_{\ell_2}(q, p) \le R_j$ (check in each iteration).

It is clear that each cluster we create during this procedure has size at least k. Moreover, each point S during this procedure will be assigned to one cluster. To prove the correctness of the algorithm, we next present the following lemmas.

Lemma 3.5. *Given a set J' with size $|J'| = O(n/k)$ and a distance parameter R, we can preprocess the set $J \cup J'$ in time $\log n \cdot (n/k)^{1+1/C^2+o(1)}$ and then for every point $p \in J$, with probability $1 - 1/n^2$ we can compute a set I such that (1) I has size at least the number of points in J' that are within distance R from p, and (2) for every point q in I, we have $\mathrm{dist}_{\ell_2}(p, q) \le O(C) \cdot R$.*

Proof. We draw $s = \Theta\left(\frac{\log n}{\mathcal{P}_1}\right)$ independent LSH hash functions in Lemma 3.4 with $S = J \cup J'$ and parameters R and C. Then, for a fixed $i \in [s]$ and every $q \in J'$, we add q into I if $h_i(p) = h_i(q)$.

We next prove the correctness of the algorithm. Consider $p, q \in J \cup J'$ with $\|p - q\|_2 \ge 2c_u \cdot C \cdot R$. Then for a fixed $i \in [s]$, $\Pr[h_i(p) = h_i(q)] \le 1/(n/k)^4$ by Lemma 3.4. By taking a union bound over all such pairs of $\{p, q\}$ and all $i \in [s]$, with probability at least $1 - k/n$, we have for any $\{p, q\} \in S$ with $\|p - q\|_2 \ge 2c_u \cdot C \cdot R$, $h_i(p) \ne h_i(q)$ for all $i \in [s]$. Thus, if a point $q \in I$, we have $\|p - q\|_2 \le 2c_u \cdot C \cdot R = O(C) \cdot R$. Now consider two points $p, q \in S$ with $\|p - q\|_2 \le R$. By Lemma 3.4, with probability at least $1 - 1/(n/k)^3$, there exists an $i \in [s]$ such that $h_i(p) = h_i(q)$. By taking a union bound over all $\{p, q\}$ with $\|p - q\|_2 \le R$, with probability at least $1 - k/n$, we have that $q \in I$ for all such pairs $\{p, q\}$. Finally, note that the above procedure only has a success probability of at least $1 - k/n$, but we can run the same procedure $O(\log n)$ independent times to boost the success probability to $1 - 1/n^2$ (after obtaining I, we can check whether I satisfies the condition or not by computing the pairwise distances). □

Lemma 3.6. *We can preprocess the point set S and J in time $k \cdot (n/k)^{1+1/C^2+o(1)}$, and after that for every point $p \in J$, we can with probability at least $1 - 1/n^2$ compute an $O(C)$-approximation $\widetilde{\rho}_k(p)$ to $\rho_k(p)$ in time $k \cdot (n/k)^{1/C^2+o(1)}$ with set N_i such that $|N_i| \ge k - 1$ and for every $q \in N_i$, we have $\mathrm{dist}_{\ell_2}(p, q) \le \widetilde{\rho}_k(p)$.*

Proof. The procedure is defined as follows. We split S/J into $m = O(k)$ disjoint subsets $S/J = J_1 \cup J_2 \cup \dots J_m$ with each $|J_i| = n/k$. Let $R_i = 2^i$. For each R_i $(i = 0, 1, \cdots, O(\log d))$ and every $j \in [m]$, we run the procedure in Lemma 3.5. For a point $p \in J$, let N_i^j be the subset returned by Lemma 3.5 with distance parameter R_i, and set $J' = J_j$ and specifically, let N_i^0 be the subset returned by Lemma 3.5 with the set J itself. Let R_i be the smallest i for which $|\bigcup_{j=0}^m N_i^j| \geq k - 1$. We use R_i as an approximation to $\rho_k(p)$ and return the set $N_i = \bigcup_{j=0}^m N_i^j$.

We next prove the correctness of our algorithm. Let i be the integer for which $R_{i-1} < \rho_k(p) \leq R_i$. This means there are at least $k - 1$ points within distance R_i from p. From the guarantee of Lemma 3.5 we have that with probability $1 - 1/n^2$, $|\bigcup_{j=0}^m N_i^j| \geq k - 1$. On the other hand, for an $R_{i'} \leq \rho_k(p)/O(C)$, from the guarantee of Lemma 3.5 we have with probability $1 - 1/n^2$, we have $|\bigcup_{j=0}^m N_{i'}^j| < k - 1$. Moreover, similar to Lemma 3.5, we have that after taking a union bound, with probability at least $1 - 1/n^2$, for every point $q \in N_i$, $\mathrm{dist}_{\ell_2}(p, q) \leq O(C) \cdot \rho_k(p)$.

Finally, we consider the time complexity of the algorithm. Note that we do not need to explicitly compare the hash value of each pair $\{p, q\}$. Instead for the point p we care about, we can just look at the cell it falls in for each of the hash functions. Moreover, we can terminate the procedure and return N_i after the set N_i we maintain has size $k - 1$. Hence, the overall runtime for one point $p \in J$ is $k \cdot (n/k)^{1/C^2 + o(1)}$. □

Lemma 3.7. *Let Q be a subset of J with size $O(n \log n/k)$. We can pre-process Q and the point set S in time $k \cdot (n/k)^{1+1/C^2+o(1)}$, such that afterwards, given a point $p \in S$, with probability at least $1 - 1/n^2$ we can find a point $s_j \in Q$ in time $(n/k)^{1/C^2+o(1)}$ such that $\mathrm{dist}_{\ell_2}(p, s_j) \leq O(1) \cdot \min_{s_i \in Q} \mathrm{dist}_{\ell_2}(p, s_i)$.*

Proof. Note that we have $|Q| \leq O(n \log n/k)$. Similarly to what we do in Lemma 3.6, we split S into $m = O(k)$ disjoint subsets $S = J_1 \cup J_2 \cup \dots J_m$ with each $|J_i| = O(n/k)$. Let $R_i = 2^i$. For every R_i $(i = 0, 1, \cdots, O(\log d))$ and every $j \in [m]$, we run the procedure in Lemma 3.5 on $Q \cup J_j$. For a point $p \in S$. Let R_i be the smallest integer such that there exists an N_i^j such that $N_i^j \cap Q \neq \emptyset$, and the algorithm will return one arbitrary center s_ℓ in $N_i^j \cap Q$. Similar to the proof of Lemma 3.6, we have that this s_ℓ satisfies $\mathrm{dist}_{\ell_2}(p, s_\ell) \leq O(1) \cdot \min_i \mathrm{dist}_{\ell_2}(p, s_i)$, which is what we need. □

Lemma 3.8. *For every point $p_i \in J$, if $q \in N_i$, then we have $\rho_k(q) \leq O(C) \cdot \rho_k(p)$. Moreover, for every point $p \notin J$, with probability at least $1 - 1/n^2$, we have that $\rho_k(f(p)) \leq O(C) \cdot \rho_k(p)$.*

Proof. Let $N_i' = \{q_1', q_2', \dots, q_{k-1}'\}$ denote the set of p_i's true k-nearest neighbors. Note that from the property of N_i, we have that $\mathrm{dist}_{\ell_2}(p, q_j) \leq O(C) \cdot \rho_k(p)$. Consider an arbitrary $q_j \in N_i$ we have that for other $q_\ell' \in N_i'$, $\mathrm{dist}_{\ell_2}(q_j, q_\ell') \leq \mathrm{dist}_{\ell_2}(p, q_j) + \mathrm{dist}_{\ell_2}(p, q_{\ell'}) \leq O(C + 1) \cdot \rho_k(p)$. This implies $\rho_k(q_j) \leq O(C + 1) \cdot \rho_k(p)$. Moreover, for each $p \notin J$, since J has size $O(n \log n/k)$, we have that with probability at least $1 - 1/n^2$ there exists a $q_j' \neq p$ such that $q_j' \in J$. This implies that $\min_{q \in J} \mathrm{dist}_{\ell_2}(p, q) \leq \rho_k(p)$. Then we have $\mathrm{dist}_{\ell_2}(p, f(p)) \leq O(C) \cdot \min_{q \in J} \mathrm{dist}_{\ell_2}(p, q) \leq O(C) \cdot \rho_k(p)$. Then, from a similar argument we can get $\rho_k(f(p)) \leq O(C + 1) \cdot \rho_k(p)$. □

Lemma 3.9. *Suppose that in phase j the point p is assigned to s, then we have $\mathrm{dist}_{\ell_2}(p, s) \leq 3 \cdot R_j$.*

Proof. The proof is by induction. For $j = -1$, i.e., before the phase of $j = 0$, since there is no point assigned at this phase, the lemma statement automatically holds. Now consider the i-th phase. We suppose that every assignment before the i-th phase has radius at most $3 \cdot R_{j-1} = R_j$.

We first consider the assignments in step 3(c). In this case, for each point p_i in C, since $R_j \leq \widetilde{\rho}_k(p_i) \leq R_j$, the distance from p_i to the points in its neighbor set N_i will be at most R_j, which means this assignment has distance at most R_j. We next consider the assignment in step 3(d). In this case, since C is a maximal independent set, we have that for each point p in $B \setminus C$, it shares the same neighbor q with points $s \in C$. This implies $\mathrm{dist}_{\ell_2}(p, s) \leq \mathrm{dist}_{\ell_2}(p, q) + \mathrm{dist}_{\ell_2}(q, s) \leq R_j + R_j = 2 \cdot R_j$.

We next consider the assignment in step 3(e). For each point p_i in $A \setminus B$, we have that there exists one $q \in N_i$ has been assigned to center s in the previous iterations with distance at most $3 \cdot R_{j-1} = R_j$. We have $\mathrm{dist}_{\ell_2}(p_i, s) \leq \mathrm{dist}_{\ell_2}(p_i, q) + \mathrm{dist}_{\ell_2}(q, s) \leq R_j + R_j = 2 \cdot R_j$.

We finally consider the assignment in step 3(f). Suppose we assign $p \notin J$ to center s, then we have we also assign $f(p)$ to s. Then we have that $\mathrm{dist}_{\ell_2}(p, s) \leq \mathrm{dist}_{\ell_2}(p, f(p)) + \mathrm{dist}_{\ell_2}(f(p), s) \leq R_j + 2 \cdot R_j \leq 3 \cdot R_j$. □

Algorithm 1 k-Anonymity via Near Neighbors

1: **Input:** A table T that contain n records, parameters $k \geq 1$.
2: Let $\Phi \in \mathbb{R}^{r \times d|\Sigma|}$ be the JL matrix in Lemma 3.3. For each record $r_i \in T$, compute $p'_i = \Phi \cdot p_i$ where p_i is a binary expansion of r_i. Let $S = \{p'_1, p'_2, \ldots, p'_n\}$.
3: Use Lemma 3.11 obtain a partition $\mathcal{P} = \{P_1, P_2, \ldots, P_t\}$ on S.
4: For each $i \in [t]$, if $|P_i| \leq 2k-1$, let $\mathcal{P}_i = \{P_i\}$. Otherwise split $P_i = Q_{i,1} \cup Q_{i,2}, \cdots \cup Q_{i,\ell}$ where $k \leq |Q_i| \leq 2k-1$ and let $\mathcal{P}_i = \{Q_{i,1}, Q_{i,2}, \ldots, Q_{i,\ell}\}$.
5: Return the partition $\mathcal{Q} = \bigcup_{i=1}^{t} \mathcal{Q}_i$.

Lemma 3.10. *For every $p \in S$, with probability at least $1 - 1/n^2$, p is assigned with a distance to its center at most $O(C^3) \cdot \rho_k(p)$.*

Proof. We first consider the point $p \in J$. Let j be the number such that $R_{j-1} < \widetilde{\rho}_k(p) \leq R_j$. Then, from the procedure of the algorithm, we have that p must be assigned in or before the phase j. From Lemma 3.9 we have the distance from p to its center will be at most $3 \cdot R_j \leq 9 \cdot \widetilde{\rho}_k(p) \leq O(9C) \cdot \rho_k(p)$.

We next consider the points $p \notin J$. Let j be the number such that $R_{j-1} < O(C^2) \cdot \widetilde{\rho}_k(p) \leq R_j$. From Lemma 3.8 we have with probability at least $1 - 1/n^2$, we have $\rho_k(f(p)) \leq O(C) \cdot \rho_k(p)$ and $\mathrm{dist}_{\ell_2}(p, f(p)) \leq O(C) \cdot \rho_k(p)$. Hence, from the procedure of our algorithm, we have that p must be assigned in or before phase j, which implies the distance from p to its center will be at most $3 \cdot R_j \leq O(9C^3) \cdot \rho_k(p)$. □

By Lemma 3.6, Lemma 3.7, and Lemma 3.10, we get the correctness of the following lemma.

Lemma 3.11. *There is an algorithm, which outputs a partition $\mathcal{P} = \{P_1, P_2, \ldots, P_t\}$ on S such that with high probability, for every P_i, $|P_i| \geq k$ and for every $p \in P_i$,*

$$\mathrm{dist}_{\ell_2}(p, c(P_i))^2 \leq O(C^6) \cdot \rho_k(p)^2 \ ,$$

for some constant C. Moreover, the entire procedure runs in time $\widetilde{O}\left(n \cdot (n/k)^{1/C^2 + o(1)}\right)$.

Proof of Theorem 1.1. The entire algorithm is presented in Algorithm 1. We first prove the correctness of our algorithm. From Lemma 3.11 we have with high probability, the partition $\mathcal{P} = \{P_1, P_2, \ldots, P_t\}$ satisfies for every P_i, $|P_i| \geq k$ and for every $p \in P_i$. $\mathrm{dist}_{\ell_2}(p, c(P_i))^2 \leq O(C^6) \cdot \rho_k(p)^2$, Then, after splitting the subsets $|P_i| \geq 2k$, from Lemma 3.1 we have the partition $\mathcal{Q}$ is an $O(k)$-approximation solution to the k-anonymity on the table T. The overall time complexity is $\widetilde{O}\left(n \cdot (n/k)^{1/C^2 + o(1)}\right)$. □

4 Single-Point k-Anonymity

We study the following single-point k-anonymity problem.

Definition 4.1 (single-point k-anonymity)**.** In the single-point k-anonymity problem, we are given a table T that contains n records and a specific record $p \in T$. Then, we ask to choose $k-1$ other records $r_1, r_2, \ldots, r_{k-1}$ from T with the goal being to minimize the number of attributes the group $(p, r_1, r_2, \ldots, r_{k-1})$ has to be suppressed, i.e., the number of j such that

$$\exists a, b \in \{p, r_1, r_2, \ldots, r_{k-1}\}, a[j] \neq b[j].$$

We say a solution is a C-approximation if the number of hidden (suppressed) attributes in this solution is at most C times the number of hidden attributes in the optimal solution.

Upper Bound. We observe a straightforward method to achieve a $(k-1)$-approximation: select the i-th nearest neighbor of p in T (with respect to ℓ_0 distance, and excluding p itself) for $i = 1, 2, \ldots, k-1$. To see why this works, let r_i denote p's i-th nearest neighbor in T. On the one hand, we have that the number of hidden attributes in the optimal solution is at least $\mathrm{dist}_{\ell_0}(p, r_{k-1})$. On the other hand, we have that the number of hidden attributes in the solution $(r_1, r_2, \ldots, r_{k-1})$ is at most $\sum_{i=1}^{k-1} \mathrm{dist}_{\ell_0}(p, r_i) \leq (k-1) \cdot \mathrm{dist}_{\ell_0}(p, r_{k-1})$.

Lemma 4.2. *There is a deterministic algorithm that computes a $(k-1)$-approximation of the single-point k-anonymity problem in time $O(nd + n \log n)$.*

Lower Bound. We next consider lower bounds for the single-point k-anonymity problem. In [CDK12], the authors give the following conjecture about the time complexity of the following DENSE VS RANDOM problem: given a graph G, it is hard to distinguish between the following two cases: (1) $G = G(n, p)$ where $p = n^{\alpha-1}$ (and thus the graph has log-density concentrated around α), and (2) G is adversarially chosen so that the densest ℓ-subgraph has log density β where $\ell^\beta \gg p\ell$ (and thus the average degree inside this subgraph is approximately ℓ^β).

In [CDM17], the work studies the MkU problem and extends the conjecture to the hypergraph case: Given an r-uniform hypergraph G on n nodes, distinguish between the following two cases:(1) $G = G(n, p, r)$ where $p = n^{\alpha-(r-1)}$ (and thus the graph has log-density concentrated around α), and (2) G is adversarially chosen so that the densest ℓ-subhypergraph on ℓ vertices and has log density β where $\ell^\beta \gg p\ell$ (and thus the average degree inside this subhypergraph is approximately ℓ^β).

Conjecture 4.3. *For all constant r and $0 < \beta < r - 1$, for all sufficiently small $\varepsilon > 0$, and for all $\ell^{1+\beta} \le n^{(1+\alpha)/2}$, we cannot solve* HYPERGRAPH DENSE VS RANDOM *with log-density α and planted log-density β in polynomial time (w.h.p.) when $\beta \le \alpha - \varepsilon$.*

Assuming the above conjecture, we prove the following hardness result. Our construction is based on the lower bound for the minimum k-union problem studied in [CDM17].

Proof of Theorem 1.3. We shall show that, if we have an algorithm for single-point k-anonymity with approximation ratio $k^{1-O(1/C)}$, then it can be used to solve the HYPERGRAPH DENSE VS RANDOM with the specific parameters in Conjecture 4.3.

For sufficiently large constant r, let $\alpha = \sqrt{r} - 1$ and $\beta = \sqrt{r} - 1 - \varepsilon$, $n = d$, and $\ell = d^{1/\sqrt{r}}$. Given an instance of the input hypergraph in Conjecture 4.3, we construct the input instance to the single-point k-anonymity problem as follows. First, we set the specific record p to be a d-dimensional zero vector. Next, for the i-th edge in the hypergraph, we set the record r_i to be the binary vector where its j-th coordinate is 1 if and only if the j-vertex is included in this edge. Let the table T be the set that contains all r_i's and $k = \Theta(d^{1-\varepsilon/\sqrt{r}})$.

Then, consider the ℓ hypersubgraph in case two. With high probability it will have $\Theta(\ell^{1+\beta}) = \Theta(\ell^{\sqrt{r}-\varepsilon}) = \Theta(d^{1-\varepsilon/\sqrt{r}})$ edges. Hence, setting $k = \Theta(d^{1-\varepsilon/\sqrt{r}})$ and choosing the records that correspond to these edges in the subhypergraph, the number of attributes we need to hide is at most $\ell = d^{1/\sqrt{r}}$ nodes in G. We next consider G in case one. We claim that with high probability every k edges in G will cover at least $d^{1-1/\sqrt{r}+1/2r-\varepsilon/r^{3/2}}$ nodes in G. This means that every k records in T will need to have such a number of attributes hidden. To prove this, we only need to show that when $\widetilde{\ell} = d^{1-1/\sqrt{r}+1/2r-\varepsilon/r^{3/2}}$, with high probability for every $\widetilde{\ell}$ subhypergraph in G, this subhypergraph can only have at most $k - 1$ edges. To get this, note that the expectation of the number of edges in each of the subhypergraphs is on the order of $\widetilde{\ell}^r n^{\sqrt{r}-r} = d^{1/2-\varepsilon/\sqrt{r}}$. Then by Chernoff's bound we have with failure probability at most $2\exp(-d)$ that this subhypergraph has fewer than $d^{1-\varepsilon/\sqrt{r}}$ edge. Taking a union bound on the $\binom{d}{\widetilde{\ell}}$ subhypergraphs, we get the desired result.

The ratio of the two cases will be at least $\widetilde{\ell}/\ell = d^{1-2/\sqrt{r}+1/(2r)-\varepsilon/r^{3/2}}$, which rules out algorithms for single-point k anonymity with ratio $k^{1-O(1/\sqrt{r})}$. The constant r here can be sufficiently large and in our construction the total number of edges will be $\Theta(n^r \cdot n^{\sqrt{r}-r}) = \Theta(d^{\sqrt{r}}) \le k^{\sqrt{r}}$. □

5 Experiments

All of our experiments were conducted on a device with a 3.30GHz CPU and 16GB RAM. We will use the following dataset which has been widely used in the study of anonymized privacy protection:

- **Adult.**[3] The Adult data contains 48842 tuples from US Census data. The tuples with missing values are removed. In particular, we choose 8 attributes as quasi-identifier.

We observe that in the dataset we use, most points have several neighbors with a small distance. Hence, in our implementation we use MinHash as an instance of LSH for simplification.

As a baseline, we consider the clustering-based heuristic algorithms proposed in [ZWL+18], where the authors demonstrate that their approach outperforms other existing heuristic methods, such as Mondrian [LDR06]. Although this algorithm shows strong performance on real-world datasets, it also

[3]The Adult from the UCI Machine Learning Repository.

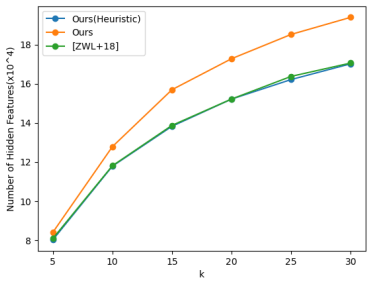

Figure 1: Test result for k-anonymity.

exhibits relatively high computational complexity. The core idea of the algorithm can be summarized as follows:

1. Iteratively selecting a new cluster center as the point with the highest average distance from existing centers, and
2. Assigning the $k-1$ nearest neighbors of that point to the same cluster. This strategy, while intuitive, results in at least $\Omega(n^2)$ time complexity, even when we disregard other parameters such as d and k.

Motivated by this bottleneck, we investigated whether the core heuristic could be accelerated using ideas from our own algorithm. We found that substantial speed-ups are indeed possible, with minimal impact on performance. In particular, we introduce the following modifications:

1. For the center selection step (Step 1), instead of computing the average distance for all points, we randomly sample a fixed number of candidate points.
2. Once a new center is chosen, we leverage Locality-Sensitive Hashing (LSH) to efficiently compute its approximate k-nearest neighbors. This process is similar to the procedures in Lemmas 3.5 and 3.6. [4]

In our experiments, these modifications lead to a substantial reduction in runtime while preserving performance comparable to the original heuristic. We refer to the improved version of our new algorithm as Ours (Heuristic).

To empirically assess runtime performance, we conducted experiments comparing the two algorithms:

- **Ours (Heuristic)** : Implemented in C++ for efficiency.
- [ZWL+18]: Since no official implementation was available, we implemented the algorithm ourselves in C++. We applied the same optimization strategies as in our own implementation.

Results Summary. The experimental results are presented in Figure 1. We vary the value of k from 5 to 30 and report the number of hidden attributes. As shown in the figure, our original approach performs similarly to [ZWL+18] when k is small but exhibits worse performance as k increases. In contrast, our second approach, which incorporates the heuristic, closely matches the performance of [ZWL+18] across the entire range of k.

We next evaluate the runtime of different algorithms on the Adult dataset for k values ranging from 5 to 30. As shown in Table 2, our algorithmic ideas can be adapted to existing heuristic methods, leading to significant speed-ups while preserving comparable performance.

Table 2: Runtime comparison on Adult dataset (s)

Dataset	ZWL+18	Ours (Heuristic)
Adult ($k=10$)	894.216	17.431
Adult ($k=15$)	394.386	5.914
Adult ($k=20$)	222.038	3.368

[4] In our experiments, we found that because the dataset is relatively small, this step does not significantly improve the runtime for this heuristic. Therefore, our current results only incorporate the first modification.

References

[AFK+05] Gagan Aggarwal, Tomas Feder, Krishnaram Kenthapadi, Rajeev Motwani, Rina Panigrahy, Dilys Thomas, and An Zhu. Approximation algorithms for k-anonymity. *Journal of Privacy Technology (JOPT)*, 2005. 2, 3

[AI08] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008. 6

[And09] Alexandr Andoni. *Nearest neighbor search: the old, the new, and the impossible*. PhD thesis, Massachusetts Institute of Technology, 2009. 6

[ANOY14] Alexandr Andoni, Aleksandar Nikolov, Krzysztof Onak, and Grigory Yaroslavtsev. Parallel algorithms for geometric graph problems. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 574–583, 2014. 20

[ASS+18] Alexandr Andoni, Zhao Song, Clifford Stein, Zhengyu Wang, and Peilin Zhong. Parallel graph connectivity in log diameter rounds. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 674–685. IEEE, 2018. 20, 21

[BDVD11] Paola Bonizzoni, Gianluca Della Vedova, and Riccardo Dondi. Anonymizing binary and small tables is hard to approximate. *Journal of combinatorial optimization*, 22:97–119, 2011. 3

[BDVDP13] Paola Bonizzoni, Gianluca Della Vedova, Riccardo Dondi, and Yuri Pirola. Parameterized complexity of k-anonymity: hardness and tractability. *Journal of combinatorial optimization*, 26:19–43, 2013. 3

[BKBL07] Ji-Won Byun, Ashish Kamra, Elisa Bertino, and Ninghui Li. Efficient k-anonymization using clustering techniques. In *International conference on database systems for advanced applications*, pages 188–200. Springer, 2007. 2

[BKS17] Paul Beame, Paraschos Koutris, and Dan Suciu. Communication steps for parallel query processing. *Journal of the ACM (JACM)*, 64(6):1–58, 2017. 20

[CDK12] Eden Chlamtac, Michael Dinitz, and Robert Krauthgamer. Everywhere-sparse spanners via dense subgraphs. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 758–767. IEEE Computer Society, 2012. 9

[CDM17] Eden Chlamtác, Michael Dinitz, and Yury Makarychev. Minimizing the union: Tight approximations for small set bipartite vertex expansion. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 881–899. SIAM, 2017. 9

[CFL10] James Cheng, Ada Wai-chee Fu, and Jia Liu. K-isomorphism: privacy preserving network publication against structural attacks. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 459–470, 2010. 3

[DG08] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008. 20

[DXTK15] Katerina Doka, Mingqiang Xue, Dimitrios Tsoumakos, and Panagiotis Karras. k-anonymization by freeform generalization. In Feng Bao, Steven Miller, Jianying Zhou, and Gail-Joon Ahn, editors, *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '15, Singapore, April 14-17, 2015*, pages 519–530. ACM, 2015. 2

[EEMM24] Alessandro Epasto, Hossein Esfandiari, Vahab Mirrokni, and Andrés Muñoz Medina. Smooth anonymity for sparse graphs. In Tat-Seng Chua, Chong-Wah Ngo, Roy Ka-Wei Lee, Ravi Kumar, and Hady W. Lauw, editors, *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, pages 621–624. ACM, 2024. 3

[EMMZ22] Alessandro Epasto, Mohammad Mahdian, Vahab S. Mirrokni, and Peilin Zhong. Massively parallel and dynamic algorithms for minimum size clustering. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 1613–1660. SIAM, 2022. 5, 20, 21, 22

[FMS+10] Jon Feldman, Shanmugavelayutham Muthukrishnan, Anastasios Sidiropoulos, Cliff Stein, and Zoya Svitkina. On distributing symmetric streaming computations. *ACM Transactions on Algorithms (TALG)*, 6(4):1–19, 2010. 20

[Goo96] Michael T Goodrich. Communication-efficient parallel sorting (preliminary version). In *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, pages 247–256, 1996. 21

[GSZ11] Michael T Goodrich, Nodari Sitchinava, and Qin Zhang. Sorting, searching, and simulation in the mapreduce framework. In *International Symposium on Algorithms and Computation*, pages 374–383. Springer, 2011. 20, 21

[IBY+07] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, and Dennis Fetterly. Dryad: distributed data-parallel programs from sequential building blocks. In *Proceedings of the 2nd ACM SIGOPS/EuroSys European conference on computer systems 2007*, pages 59–72, 2007. 20

[JLS86] William B Johnson, Joram Lindenstrauss, and Gideon Schechtman. Extensions of lipschitz maps into banach spaces. *Israel Journal of Mathematics*, 54(2):129–138, 1986. 5

[KSV10] Howard Karloff, Siddharth Suri, and Sergei Vassilvitskii. A model of computation for mapreduce. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 938–948. SIAM, 2010. 20

[KT12] Batya Kenig and Tamir Tassa. A practical approximation algorithm for optimal k-anonymity. *Data Mining and Knowledge Discovery*, 25:134–168, 2012. 2

[LDR06] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *22nd International conference on data engineering (ICDE'06)*, pages 25–25. IEEE, 2006. 2, 9

[LLV06] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd international conference on data engineering*, pages 106–115. IEEE, 2006. 3

[MKGV07] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *Acm transactions on knowledge discovery from data (tkdd)*, 1(1):3–es, 2007. 3

[MW04] Adam Meyerson and Ryan Williams. On the complexity of optimal k-anonymity. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 223–228, 2004. 2

[PS07] Hyoungmin Park and Kyuseok Shim. Approximate algorithms for k-anonymity. In Chee Yong Chan, Beng Chin Ooi, and Aoying Zhou, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China, June 12-14, 2007*, pages 67–78. ACM, 2007. 2

[Swe02] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002. 1

[Whi12] Tom White. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012. 20

[ZCF+10] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. In *2nd USENIX workshop on hot topics in cloud computing (HotCloud 10)*, 2010. 20

[ZWL+18] Wantong Zheng, Zhongyue Wang, Tongtong Lv, Yong Ma, and Chunfu Jia. K-anonymity algorithm based on improved clustering. In *Algorithms and Architectures for Parallel Processing: 18th International Conference, ICA3PP 2018, Guangzhou, China, November 15-17, 2018, Proceedings, Part II 18*, pages 462–476. Springer, 2018. 2, 9, 10

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, we list our main contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, we clearly state all theoretical assumption

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Yes, we clearly state all theoretical assumption.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we discuss the details about our implementation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Yes, we include the code in the supplementary materials.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we discuss the experiment details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report the errors by average but not report the standard deviation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we discuss the information about this.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Yes, we have confirmed.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, we discuss that our algorithm will be beneficial for the privacy preserving.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use public dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We only use LLMs to improve the writing.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Preliminaries

Given a graph $G(V, E)$, for any $v \in V$, let $\Gamma_G(v)$ be the set of the neighbors of v in G. A maximal independent set of G is a subset $I \subseteq V$ such that every vertex in the graph is at distance at most 1 from some vertex in I. A β-ruling set of a graph is an independent set I such that every vertex in the graph is at a distance of at most β from some vertex in I. In particular, a maximal independent is a 1-ruling set, which can be seen as a special case. Let G^2 to be the square graph of G that has the same set of vertices as G, but in which two vertices are connected when their distance in G is at most 2.

A.1 Massivaly Parallel Computing

The MPC model [FMS+10, KSV10, GSZ11, BKS17, ANOY14] is an abstract of modern massively parallel computing systems such as Map Reduce [DG08], Hadoop [Whi12], Dryad [IBY+07], Spark [ZCF+10] and others.

We follow the introduction in [EMMZ22]. In the MPC model, the input data has size N. The system consists of p machines, each with a local memory size of s. Hence, the total space available in the entire system is $p \cdot s$. The space here is measured by words, each of $O(\log(p \cdot s))$ bits. Specifically, if the total space $p \cdot s = O(N^{1+\gamma})$ for some $\gamma \geq 0$ and the local space $s = O(N^{\delta})$ for some $\delta \in (0, 1)$, then the model is referred to as the (γ, δ)-MPC model [ASS+18]. The computation proceeds in synchronized rounds. In each round, every machine processes the data stored in its local memory and sends messages to other machines at the end of the round. Although each machine can send messages to any other machine, the total size of the messages sent or received by a machine in a single round should be at most s.

Note that the space of each machine is sublinear in the input size. This means that we cannot collect all input data into one machine. An MPC algorithm is called *fully-scalable*, if it can work when the space per-machine is $O(N^{\delta})$ for any constant $\delta \in (0, 1)$. The goal in this work is to design fully-scalable MPC algorithms minimizing the number of rounds while using a small total space.

B Extending to MPC Model

In this section, we demonstrate that our algorithm can be adapted into a fully scalable MPC algorithm. At a high level, [EMMZ22] studied the minimum size constraint clustering problem in the MPC model and give an algorithm that can be efficiently implemented in the MPC model. However, since the input of this clustering problem is the binary expansion of each record r_i (which is in dimension $d|\Sigma|$) but not the record itself, naively using the algorithm in [EMMZ22] yields a $|\Sigma|$ factor in time and space, which can be large in practice. Instead, we open the procedure of this algorithm and show that since each input point is d-sparse, it is still achievable in the same order of time and space.

We will need the following concept of the C-approximate (R, r)-near neighbor graph.

Definition B.1 (C-approximate (R, k)-near neighbor graph, [EMMZ22])**.** Consider a point set P from a metric space $\mathcal{X}$. Let $C, R, k \geq 1$. If an undirected graph $G = (V, E)$ satisfies

(a) $V = P$,

(b) $\forall (p, p') \in E, \mathrm{dist}_{\mathcal{X}}(p, p') \leq C \cdot R$,

(c) For every $p \in P$, either $|\Gamma_G(p)| \geq k$ or $\{p' \in P \mid \mathrm{dist}_{\mathcal{X}}(p, p') \leq R\} \subseteq \Gamma_G(p)$,

then we say G is a C-approximate (R, k)-near neighbor graph of P.

We will show that given parameters C, R and a set S where each $p \in S$ represents a binary expansion of one record r_i in table T. We can efficiently build a C-approximate (R, k)-near neighbor graph G of S under ℓ_2 distance in $\widetilde{O}(nd + n^{1+1/C^2+o(1)}k)$ time in the static setting and in $O(1)$ rounds in the MPC model.

Lemma B.2. *Given parameters C, R, and a set S where each $p \in S$ is a binary expansion of a record r_i in T. Then, there is an algorithm that with high probability constructs a $O(C)$-approximate (R, k)-near neighbor graph of S in time $\widetilde{O}(nd + n^{1+1/C^2+o(1)}k)$. The size of G is at most $\widetilde{O}\left(n^{1+1/C^2+o(1)} \cdot k\right)$.*

Proof. Let $\Phi \in \mathbb{R}^{r \times d|\Sigma|}$ be a JL matrix in Lemma 3.3 with $r = O(\log n)$ and for each $p \in S$, we compute $p' = \Phi p$. Let $S' = \{p'_1, p'_2, \ldots, p'_n\}$. The construction of the graph G is as follows. We draw $s = \Theta\left(\frac{\log n}{\mathcal{P}_1}\right)$ independent LSH in Lemma 3.4 with parameters R and C. Then, for every $i \in [s]$ and every $p' \in S'$, we connect p' to k arbitrary points $q' \in S'$ in G with $h_i(p') = h_i(q')$. If there are less than k points with $h_i(q') = h_i(p')$, connect p' to all such q' in G.

We first prove the correctness of the algorithm. From Lemma 3.3 we have with probability at least $1 - 1/n$, we have for every pair of p' and q' in S', $0.9\|p - q\|_2 \leq \|p' - q'\|_2 \leq 1.1\|p-q\|_2$. Condition on this event occurs and consider $p, q \in S$ with $\|p-q\|_2 \geq 2c_u \cdot C \cdot R$. Then from the assumption we have $\|p' - q'\|_2 \geq 1.8c_u \cdot C \cdot R$, this means that for a fixed $i \in [s]$, $\Pr[h_i(p') = h_i(q')] \leq 1/n^4$ by Lemma 3.4. By taking union bound over all such pairs of (p, q) and all $i \in [s]$, with probability at least $1 - 1/n$, we have for any $(p, q) \in S$ with $\|p - q\|_2 \geq 2c_u \cdot C \cdot R$, $h_i(p') \neq h_i(q')$ for all $i \in [s]$. Thus, if an edge $(p, q) \in E$, we have $\|p - q\|_2 \leq 2c_u \cdot C \cdot R = O(C) \cdot R$. Now consider two points $p, q \in S$ with $\|p - q\|_2 \leq R$. From the assumption we have $\|p' - q'\| \leq 1.1 \cdot R$, By Lemma 3.4 and Chernoff bound, with probability at least $1 - 1/n^3$, there exists an $i \in [s]$ such that $h_i(p') = h_i(q')$. By taking union bound over all (p, q) with $\|p - q\|_2 \leq R$, with probability at least $1 - 1/n$, we have there is an edge $(p, q) \in E$ for all such pairs (p, q). Combining these two aspects, we have that for every $p \in P$, either $\{q \in P \mid \|p - q\|_2 \leq R\} \subseteq \Gamma_G(p)$ or $|\Gamma_G(p)| \geq k$.

We next consider the runtime complexity. First, for each $p \in S$, since p is d-sparse, we can compute $\Phi \cdot p$ is $O(d \log n)$ time, which implies we can form the set S' in $O(nd \log n)$ time (note that the algorithm does not need to write down p explicitly). Then from Lemma 3.4 we get that algorithm can evaluate $h_i(p')$ for every $i \in [s]$ and $p \in S'$ in time $O(n^{1+1/C^2+o(1)})$ as the dimension of each $p' \in S$ is $d' = O(\log n)$. Finally, to connect edges in graph G, we sort points in S' via their hash values and only consider to connect the points with the same hash values. Since for each hash function, we connect at most r edges from a point, we have this procedure can be done in time $\widetilde{O}\left(n^{1+1/C^2+o(1)} \cdot k\right)$ and the size of G is at most $\widetilde{O}\left(n^{1+1/C^2+o(1)} \cdot k\right)$. □

Lemma B.3. *Given parameters R, C, k and point set S. There is an MPC algorithm that builds an $O(C)$-approximate (R, r)-near neighbor graph of S with high probability in $O(1)$ rounds and $\widetilde{O}\left(nd + n^{1+1/C^2+o(1)}k\right)$ total space.*

Proof. We first note that, the Johnson-Lindenstrauss lemma can be implemented in $O(1)$ MPC round and $O(ndr)$ space where $r = O(\log n)$ in our case (See, e.g., Appendix A in [EMMZ22]). Next, we can handle LSH functions in parallel. According to Lemma 4.1, we use $O(1)$ rounds to compute LSH values for all points in S. To connect edges, we can sort points via their LSH values, make copies of some vertices and query indices in parallel. These operations can be done simultaneously in $O(1)$ rounds [Goo96, GSZ11, ASS+18]. Since we run s independent LSH functions and for each $i \in [s]$, every each connects to at most r vertices, we have the total space needed is $\widetilde{O}(nd + n^{1+1/C^2+o(1)}k)$. □

Given access to $O(C)$-approximate (R, k)-near neighbor graphs with different distance parameters $R_i = 2^i$ for $i = 0, 1, \cdots, O(\log d)$, the MPC algorithm presented in [EMMZ22] produces a partition $\mathcal{P}$ on S' that satisfies the condition outlined in Corollary 3.2.

Lemma B.4 (Essentially Theorem 4.31 in [EMMZ22])**.** *Let $\gamma, \varepsilon \in (0, 1)$. Given the C-approximate (R_i, k)-near neighbor graph G_i's, there is a fully scalable MPC algorithm that with high probability outputs a partition $\mathcal{P} = \{P_1, P_2, \ldots, P_t\}$ on S such that with high probability, for every P_i, $|P_i| \geq k$ and for every $p \in P_i$, $\mathrm{dist}_{\ell_2}(p, c(P_i))^2 \leq O\left(\frac{\log^2(1/\varepsilon)}{\gamma}\right) \cdot \rho_k(p)^2$. The algorithm takes $O\left(\frac{\log 1/\varepsilon}{\gamma} \cdot \log^{1+\varepsilon}(n) \log\log(n)\right)$ parallel time and $\widetilde{O}\left(n^{1+\gamma+o(1)} \cdot k\right)$ total space.*

The full algorithm of Lemma B.4 is presented in Algorithm 2 for completeness.

Algorithm 2 Clustering with Pointwise Guarantee

Input: A point set P, a parameter $k \geq 1$.
Let $C \geq 1$.
Let $t \leftarrow 0$. Initialize the family of clusters $\mathcal{P} \leftarrow \emptyset$.
Let Δ (δ) be an upper bound (a lower bound) of $\mathrm{dist}_{\ell_2}(p, q)$ for $p \neq q \in P$.
Let $L = \lceil \log(\Delta/\delta) \rceil$. For $i \in \{0, 1, 2, \cdots, L\}$, let $R_i \leftarrow 2^i \cdot \delta$.
for $i = 0 \to L$ **do**
  Compute a C-approximate (R_i, r)-near neighbor graph $G_i = (P, E_i)$ of P.
  Let $P_i' \subseteq P$ be the vertices with at least k neighbors in G_i, i.e., $P_i' = \{p \in P \mid |\Gamma_{G_i}(p)| \geq k\}$.
  Let $P_i'' = \left\{p \in P_i' \mid \mathrm{dist}_{G_i}\left(p, \bigcup_{Q \in \mathcal{P}} Q\right) > 1\right\}$.
  Compute a β-ruling set $S_i = \{s_{i,1}, s_{i,2}, \cdots, s_{i,t_i'}\}$ of $(G_i^2)[P_i'']$.
  Compute $P_i''' = \left\{p \in P \setminus \bigcup_{Q \in \mathcal{P}} Q \mid \mathrm{dist}_{G_i}(p, S_i) \leq 2 \cdot \beta\right\}$.
  Partition P_i''' into t_i' clusters $Q_{i,1}, Q_{i,2}, \cdots, Q_{i,t_i'}$ where the center $c(Q_{i,j})$ is $s_{i,j}$. For each point $p \in P_i''' \setminus S_i$, add p into an arbitrary cluster $Q_{i,j}$ such that $\mathrm{dist}_{G_i}(p, s_{i,j})$ is minimized.
  For each $p \in P_i' \setminus P_i'''$, if $p \notin \bigcup_{Q \in \mathcal{P}} Q$, find an arbitrary cluster $Q \in \mathcal{P}$ such that $\mathrm{dist}_{G_i}(p, Q) \leq 1$ and update Q by adding p into Q.
  Add $Q_{i,1}, Q_{i,2}, \cdots, Q_{i,t_i'}$ into $\mathcal{P}$. Let $t \leftarrow t + t_i'$.
end for
Output the partition $\mathcal{P} = \{P_1, P_2, \cdots, P_t\}$ and the centers $c : \mathcal{P} \to P$.

At a high level, the algorithm iteratively processes $R = 1, 2, 4, \ldots, \Delta$ (recall that in the case of our input, the distance of each pair of points is between 1 and $O(\sqrt{d})$). For each value of R, the algorithm needs to access a C-approximate (R, k)-near neighbor graph G, and then compute a β-ruling set of a subgraph of G^2. The algorithm maintains the following invariants at the end of the iteration with respect to the value of R:

(a) Every point p satisfies $\rho_k(p) \leq R$ must be assigned to some cluster.
(b) The radius of each cluster is at most $O(C \cdot R)$
(c) The size of each cluster is at least k.

For the correctness of Algorithm 2 and more details, we refer the readers to Section 3.3 in [EMMZ22].

Combining with Lemma B.3 and Lemma B.4, we can prove the correctness of our Theorem 1.2.