Differentially Private Distance-Sensitive Set Representations for General Metrics

Full version

Paul Cesaretti
The Graduate Center, City University of New York, New York, USA
paul.cesaretti@gmail.com

Abstract

A distance-sensitive set representation stores a set S from a metric space (U, d) and answers queries of the form "is some element of S within distance r of q?". Existing differentially private constructions are confined to Hamming space, because they route the query through a signature map built for that metric. We give a framework that removes this restriction. The released object records only the occupancy of a fixed public partition of U, so privacy transfers unchanged from an off-the-shelf private membership structure, and the entire metric dependence collapses into one packing quantity, the *straddle number* κ: the largest number of cells a ball of radius r can meet. Any metric admitting a partition of width at most $(c-1)r$ and straddle number κ then has an ε-differentially private (c, r)-representation with $\varepsilon = \ln(2\kappa/\alpha)$ in $O(k \log(\kappa/\alpha))$ bits.

In ℓ_∞ we give partition-based constructions and a lower bound for all partition-based constructions whose dimension-dependent terms agree up to a factor $O(\log c)$, and agree exactly at $c = 3$. If the partition may be drawn at random publicly, a single probe suffices and $\varepsilon = \ln(2/\alpha)$ independently of d, with the dimension and the error moved into the approximation factor and a weaker, non-oblivious guarantee. For metrics available only through a distance oracle, we build the partition from a public net and a public navigable graph, and show that greedy search on that graph both locates and enumerates the cells a query must probe; on a λ-doubling metric this gives $\varepsilon \le 5 \ln \lambda + \ln(2/\alpha)$, with a public structure of size polynomial in the aspect ratio and exponential only in the doubling dimension. We further show that the per-query form of the guarantee is forced rather than chosen: any representation correct on all queries simultaneously requires $\varepsilon \ge \ln M + \ln(1/\alpha) - O(1)$ for M the packing number of the whole space, and on doubling metrics this is achievable up to an additive $O(\dim_{\mathrm{D}} \log \frac{c}{c-1})$. Finally, the space bound is optimal in k and α.

Keywords: differential privacy, distance-sensitive filters, approximate membership, secluded partitions, navigable graphs, near-neighbor search.

Contents

1 Introduction **3**
1.1 Why this is not routine 3
1.2 Our approach: separating the private and the geometric layer 3
1.3 What is new 4

1.4 Relation to the Hamming-space construction . . . 5
1.5 Results . . . 5
1.6 Technical overview of the distance-oracle construction . . . 7
1.7 Related work . . . 7
1.8 Organization . . . 8

2 Preliminaries **9**
2.1 Metric notation . . . 9
2.2 Differential privacy . . . 9
2.3 The target: DPDS-representations . . . 9
2.4 The private building block . . . 10

3 The Partition Framework **11**
3.1 Cell decompositions and the straddle number . . . 11
3.2 The construction . . . 11
3.3 Analysis . . . 12

4 Instantiations in Normed Spaces **14**
4.1 Secluded partitions in ℓ_∞ . . . 15
4.2 Euclidean distance, obliviously . . . 16
4.3 Randomized partitions and one-probe constructions . . . 17

5 Metrics Given by a Distance Oracle **19**
5.1 The access model, and a running example . . . 19
5.2 Overview . . . 19
5.3 Step 1: a public net . . . 20
5.4 Step 2: a public navigable graph . . . 21
5.5 Step 3: cells as fibers of greedy descent . . . 23
5.6 Step 4: encoding, the only step that sees the data . . . 24
5.7 Step 5: queries . . . 25
5.8 The guarantee . . . 26
5.9 Limitations and variants . . . 27

6 For-All Guarantees **28**
6.1 Three levels of guarantee . . . 28
6.2 Intuition . . . 29
6.3 The lower bound . . . 29
6.4 A sharper bound . . . 31
6.5 A matching upper bound: expensive, not impossible . . . 31
6.6 Why the per-query constructions escape . . . 32

7 Optimality and Open Problems **32**
7.1 Space optimality . . . 32
7.2 Is the straddle number inherent? . . . 33
7.3 Further directions . . . 33

8 Conclusion **34**

A Deferred Proofs **37**

1 Introduction

A *set representation* stores a set S of at most k elements from a large universe U so that membership queries can be answered from far fewer bits than an explicit list would need. The Bloom filter [5] is the canonical example: it accepts every element of S, accepts each non-element with a small probability, and uses about $k \log(1/p)$ bits for false-positive rate p [7]. A *distance-sensitive* set representation answers a geometric version of the same question. The set lives in a metric space (U, d), and a query q asks whether some element of S is within distance r of q. Because exact answers to this question are as expensive as storing S, an approximation factor $c > 1$ is allowed: queries within r of S must be accepted, queries farther than cr from all of S must be rejected, and queries in between may be answered either way [23, 15]. Bloom filters are the case $r = 0$.

Such structures are natural objects to publish. A plagiarism or near-duplicate detection service may wish to let outside users test a document against a confidential archive; a genomic consortium may wish to let laboratories test whether a sample is close to any sample in a cohort; a health department may wish to let residents test whether they were near a confirmed case. In each case the released structure itself can leak the stored set, and the natural requirement is that the release be *differentially private* [13] with respect to adding or removing one element of S.

1.1 Why this is not routine

Existing distance-sensitive filters are not private for structural reasons. The filters of Goswami, Pagh, Silvestri and Sivertsen [15] and of Kirsch and Mitzenmacher [23] draw all their randomness from public hash functions that are independent of the input. Once those functions are fixed, the filter is a deterministic function of S, so two neighbouring sets produce distinguishable filters with certainty, regardless of how well the filter answers queries. They can be made private by perturbation, but at a cost that grows with their sensitivity: one element sets up to h bits of a Bloom filter with h hash functions, and randomized response on every bit (as in BLIP [1]) must be calibrated to that; a distance-sensitive filter built from ℓ locality-sensitive hash tables places one element in ℓ buckets, and group privacy multiplies the privacy parameter by ℓ.

A second obstruction concerns the shape of the error. A randomized-response argument on a single membership bit (Fact 2.9) shows that any differentially private representation errs on *both* sides with probability $\Omega(e^{-\varepsilon})$. A private structure therefore cannot inherit the one-sided (no false negatives) guarantee of Bloom filters, and its error cannot be driven to zero by spending more space.

Patel, Persiano, Seo and Yeo [27] resolved the exact-membership case ($r = 0$) with a random linear system whose error $1/(e^{\varepsilon} + 1)$ and space $O(k\varepsilon)$ are both optimal. Combining their mechanism with the vector signatures of Goswami et al. gave the first private distance-sensitive representation, in Hamming space [9]. That construction, however, is tied to the Hamming geometry: the signature map, the ℓ_1 expansion of the query ball, and the candidate enumeration at decode time are all specific to $\{0, 1\}^d$ (see Section 1.4).

1.2 Our approach: separating the private and the geometric layer

The observation behind this paper is that the private layer and the geometric layer of the problem can be separated completely (Figure 1).

The geometric layer is a partition $\mathcal{C}$ of U into cells of small diameter, fixed in advance and

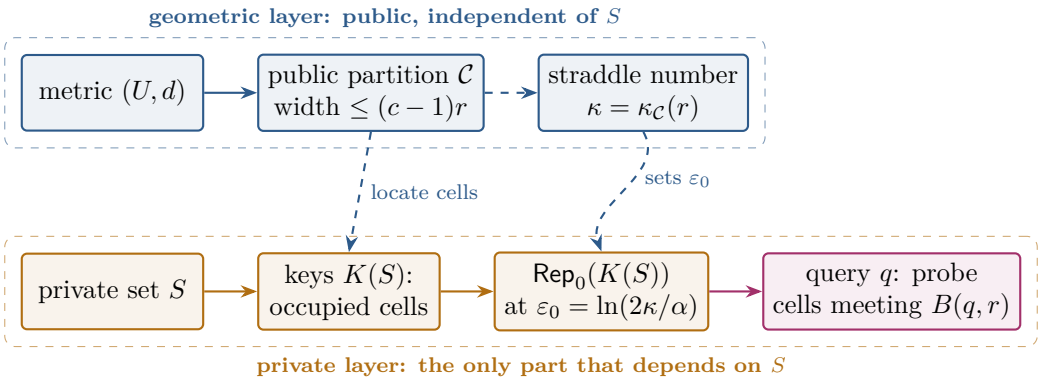


Figure 1: The two layers of the framework. The geometry enters only through a public partition and its straddle number κ; the privacy comes entirely from a black-box private membership structure storing cell identifiers.

independently of the data. *The private layer* is any differentially private membership structure, used as a black box. To encode S we record only which cells S occupies, and store those cell identifiers in the private structure. To answer q we probe every cell that meets the ball $B(q, r)$.

Two simple facts make this work. Because $\mathcal{C}$ is a partition, adding one point to S changes the set of occupied cells by at most one identifier, so the privacy analysis of the membership structure applies verbatim. Because each cell has diameter at most $(c-1)r$, every cell that meets $B(q, r)$ lies inside $B(q, cr)$, so a stored element found in a probed cell is always a correct c-approximate answer: no false positive is caused by the geometry, and every error comes from the membership structure. The number of probes per query is at most the *straddle number*

$$\kappa_{\mathcal{C}}(r) = \sup_{q \in U} |\{C \in \mathcal{C} : C \cap B(q, r) \neq \emptyset\}|,$$

and a union bound over the probes gives the main theorem.

1.3 What is new

The reduction itself is deliberately simple, and putting public locality structures in front of private payloads is natural (the near-neighbour counting structures of [2, 3] do so). Our contributions are what turns it into a usable and well-understood tool.

(1) We identify the straddle number as the exact price of partition-based constructions: the upper bound $\varepsilon = \ln(2\kappa/\alpha)$ holds in every metric, and in ℓ_∞ it is tight for partitions (Section 4).

(2) For metrics given only by a distance oracle, the reduction leaves two problems that do not arise in $\mathbb{R}^d$: encoder and decoder must agree on the cell of a point without exact nearest-neighbour search, and the decoder must list the cells near a query without coordinates. Section 5 solves both with public nets and navigable graphs; this is our main technical contribution.

(3) We pin down which utility guarantee is achievable: we separate per-query guarantees by when the query may be chosen (Definition 2.8) and show that a guarantee for all queries at once costs a global packing number, with matching upper and lower bounds on doubling metrics (Section 6).

1.4 Relation to the Hamming-space construction

The construction of [9] (by the present author) runs the structure of [27] on the vector signatures of [15] and recovers distance sensitivity by enumerating, at decode time, the N_Q candidate signatures near the query's; its privacy parameter carries $\ln N_Q$, which depends on r, c, k and α but not on d. In Hamming space it remains the better choice when $d \gg r$: the oracle construction of Section 5 pays $\ln \kappa'$, which by packing is at most $O(cr \log(d/r))$ there. It does not, however, transfer to other metrics. An embedding into Hamming space with distortion D turns the approximation factor c into cD, and even isometric embeddings are costly: ℓ_1 on a bounded grid embeds into Hamming space by unary encoding, but controlling the rounding error forces a grid spacing of about $(c-1)r/d$, so the Hamming radius becomes about $d/(c-1)$ and the candidate count of [9] grows with d. General n-point metrics need distortion $\Theta(\log n)$ into ℓ_1 [6, 25], and some doubling spaces admit no bounded-distortion embedding into ℓ_1 at all [10]. All results of this paper are for settings that the Hamming-space construction does not cover. The two constructions are also structurally different: the Hamming-space construction is not partition-based in the sense of Definition 3.1, and its decoder enumerates candidates whose number depends on k, while the number of probes here is a property of the partition alone.

1.5 Results

Our results are summarized in Table 1.

The reduction (Section 3). Any metric admitting a partition of width at most $(c-1)r$ and straddle number κ has an ε_0-differentially private (c, r, α)-representation with $\varepsilon_0 = \ln(2\kappa/\alpha)$, using $O(k \log(\kappa/\alpha))$ bits beyond the public partition and at most κ probes per query (Theorem 3.7). Neither r nor $|U|$ appears in the privacy parameter or in the size of the private release; the approximation factor c enters only through κ. The public partition must still be described, which is free in $\mathbb{R}^d$ but not in a distance-oracle metric. We also state a slightly more general version in which the query may probe any superset of the necessary cells, provided every probed cell stays inside $B(q, cr)$ (Theorem 3.8); this is what the distance-oracle construction needs.

Normed spaces (Section 4). In $\mathbb{R}^d$ under ℓ_∞, the secluded partitions of Vander Woude, Dixon, Pavan, Radcliffe and Vinodchandran [29, 30], developed for deterministic rounding, give an explicit family trading approximation factor for straddle number. Transferring it yields $\varepsilon_0 = \lceil d/f \rceil \ln(f+1) + \ln(2/\alpha)$ at $c = 2f+1$ (Theorem 4.4), in particular $\ln(d+1) + \ln(2/\alpha)$ at $c = 2d+1$ (Corollary 4.5), and their lower bound transfers to every partition-based construction: $\varepsilon_0 \geq d \ln \frac{c+1}{c-1}$ (Theorem 4.6). The d-dependent terms agree up to a factor $O(\log c)$, exactly at $c = 3$, so $\varepsilon_0 = \Theta(d)$ at constant c; at $c = \Theta(d)$ the lower bound is only $\Omega(1)$. Table 2 gives concrete numbers. Under ℓ_2 the same partitions give $c = \Theta(d^{3/2})$ (Theorem 4.8). If the partition may be drawn at random from a public distribution, a single probe suffices and $\varepsilon_0 = \ln(2/\alpha)$ independently of d, with the dimension and the error moved into the approximation factor, $c = 1 + 4d^{1+1/p}/\alpha$ for shifted grids in ℓ_p (Theorem 4.11 and Corollary 4.13). The randomized construction is *non-oblivious* (Definition 2.8): its guarantee holds only for queries chosen without knowledge of the drawn partition. Whether obliviousness itself costs privacy is open.

Metrics given by a distance oracle (Section 5). When the metric is available only through distance evaluations, cells cannot be located by arithmetic, and enumerating the cells that

Space	Guarantee	Approximation c	Privacy ε_0	Reference
any, partition $\mathcal{C}$	oblivious	admissible $\mathcal{C}$	$\ln(2\kappa_{\mathcal{C}}(r)/\alpha)$	Theorem 3.7
$(\mathbb{R}^d, \ell_\infty)$	oblivious	$2f+1$	$\lceil d/f \rceil \ln(f+1) + \ln(2/\alpha)$	Theorem 4.4
$(\mathbb{R}^d, \ell_\infty)$	oblivious	$2d+1$	$\ln(d+1) + \ln(2/\alpha)$	Corollary 4.5
$(\mathbb{R}^d, \ell_\infty)$, any partition	oblivious	c	$\geq d \ln \frac{c+1}{c-1}$	Theorem 4.6
$(\mathbb{R}^d, \ell_2)$	oblivious	$2f\sqrt{d}+1$	$\lceil d/f \rceil \ln(f+1) + \ln(2/\alpha)$	Theorem 4.8
$(\mathbb{R}^d, \ell_p)$	non-oblivious	$1 + 4d^{1+1/p}/\alpha$	$\ln(2/\alpha)$, one probe	Corollary 4.13
λ-doubling, oracle	oblivious	2	$\leq 5 \ln \lambda + \ln(2/\alpha)$	Theorem 5.14
any	for-all	c	$\geq \ln M(U, cr) + \ln \frac{1}{\alpha} - O(1)$	Proposition 6.5
bounded, n cells	for-all	admissible $\mathcal{C}$	$\leq \ln(n/\alpha)$	Proposition 6.7

Table 1: Summary of results. All upper bounds use $O(k\varepsilon_0)$ bits beyond public data, which is optimal in k and α (Corollary 7.2). The lower bound in the fourth row applies only to partition-based constructions; "oblivious" is defined in Definition 2.8.

meet a query ball becomes a search problem. We build the partition from a public ρ-net and a public navigable graph on it, and define the cells as the fibers of greedy descent on the graph. A cell-radius lemma (Lemma 5.7) shows the cells are small, and an enumeration lemma (Lemma 5.11) shows that breadth-first search restricted to a ball of radius 3τ finds every net point in the ball of radius τ that a query must probe. Since the net and the graph depend only on the metric, publishing them costs no privacy. On a metric with doubling constant λ this gives $\varepsilon_0 \leq 5 \ln \lambda + \ln(2/\alpha)$ at $c = 2$, with polylogarithmic query time in the aspect ratio (Theorem 5.14). The public net and graph have $\Delta_{\text{asp}}^{O(\dim_{\text{D}})}$ points and edges; they are built once and shared by all releases, and the space must be bounded.

The shape of the guarantee (Section 6). All of the above are *per-query* guarantees: each query chosen without looking at the release is answered correctly with probability $1 - \alpha$. Once a release is published its false positives are fixed, so correctness against queries chosen after seeing it requires the stronger *for-all* guarantee, in which one draw of the release is correct on all queries simultaneously. Non-private filters provide it only in part: Bloom filters have no false negatives for every draw, but bound false positives per query, as do structures based on locality-sensitive hashing [19]. We show that it costs a *global* rather than a local quantity: $e^{\varepsilon_0} \geq M(1 - \alpha - \delta_0)$ for M the packing number of U at scale cr (Theorem 6.4), which we sharpen to $e^{\varepsilon_0} \geq M(U, cr)(1 - \alpha - \delta_0)/\alpha$ (Proposition 6.5) and, on doubling metrics, match up to an additive $O(\dim_{\text{D}} \log \frac{c}{c-1})$ with a per-cell randomized-response construction (Proposition 6.7 and Corollary 6.8). On a bounded space the for-all guarantee is therefore expensive rather than impossible; on an unbounded one it is impossible.

Optimality (Section 7). Restricting any representation to a cr-separated subset yields an ordinary private set representation, so the $O(k\varepsilon_0)$ space bound is tight in k and α (Corollary 7.2); this argument cannot detect κ. Whether the straddle number must appear in ε_0 for representations that do not route membership through a partition is left as a conjecture (Conjecture 7.3).

1.6 Technical overview of the distance-oracle construction

The reduction of Section 3 leaves one question per metric: find a public partition of small width and small straddle number, and give the decoder a way to list the cells that meet $B(q, r)$. In $\mathbb{R}^d$ the second part is free, because cells are located by coordinatewise rounding. In a metric presented only by a distance oracle there are no coordinates to round, and both parts have to be built from distance evaluations alone. Our construction does this in five steps (Figure 5); only the fourth sees the data.

Step 1. *A public net.* A greedy scan in a fixed public order produces a ρ-net $\mathcal{N}$: every point is within ρ of a net point, and net points are more than ρ apart. Separation is what will bound the privacy cost, through a packing argument.

Step 2. *A public navigable graph.* A directed graph G on $\mathcal{N}$ is γ-*navigable* if for every vertex s and every target t, some out-neighbour of s is a factor γ closer to t than s is. The standard pruning construction achieves this by design, with out-degree $(O(\gamma))^{\dim_{\mathrm{D}}} \log \Delta_{\mathrm{asp}}$.

Step 3. *Cells as fibers of greedy descent.* We define the cell of a point u as the net point at which greedy descent toward u stops. Greedy is deterministic, so its fibers partition U, and a contraction argument shows every point lands within $\theta(\gamma)\rho$ of its cell's net point, $\theta(\gamma) = (4\gamma - 2)/(\gamma - 1)$. Exact Voronoi cells would not do: greedy on a γ-navigable graph with $\gamma > 1$ finds only an approximate nearest net point, so encoder and decoder would disagree about where points live.

Step 4. *Encoding.* The data set S is mapped to the set of net points $\mathsf{Greedy}(u)$, $u \in S$, and stored in the private membership structure. Since $\mathcal{N}$, G and the cells are the same for every S, this step is data-independent preprocessing and the privacy proof never mentions the graph.

Step 5. *Querying.* A point of S within r of q has its net point within $\tau = r + \theta\rho$ of q, so it suffices to probe every net point in $B(q, \tau)$. The enumeration lemma shows that navigability makes this a local search: a navigable path between two net points of $B(q, \tau)$ never leaves $B(q, 3\tau)$, so breadth-first search inside $B(q, 3\tau)$, started at $\mathsf{Greedy}(q)$, lists them all. The search may look at more points than it probes; that costs time but never privacy, which depends only on the number of probes.

On a λ-doubling metric, packing bounds the number of probes by $\lambda^{\lceil \log_2(2\tau/\rho) \rceil}$, which is λ^5 for $c = 2$, $\gamma = 3$, $\rho = r/10$. Two features of the construction deserve emphasis. First, the navigational layer contributes only computability: it never enters the privacy analysis, and enlarging the search radius never increases ε_0. Second, the construction is the same for every data set and is built once from the metric, so in applications its cost is amortized over all releases, while each release costs only $O(k\varepsilon_0)$ bits.

1.7 Related work

Distance-sensitive filters. Kirsch and Mitzenmacher [23] introduced distance-sensitive Bloom filters, replacing the hash functions of a Bloom filter by locality-sensitive hash functions; the resulting structure has two-sided error. Goswami, Pagh, Silvestri and Sivertsen [15] recovered the absence of false negatives in Hamming space via vector signatures and proved matching space bounds. Neither construction is private.

Private set representations. Unmodified Bloom filters provide only weak deniability. Patel, Persiano, Seo and Yeo [27] gave differentially private set representations from random linear systems, with error $1/(e^{\varepsilon}+1)$ matching a randomized-response lower bound and space $O(k\varepsilon)$ matching a compression lower bound. We use their structure as the black box of Definition 2.10. Earlier, BLIP [1] applied randomized response to the bits of a Bloom filter to compute private similarity estimates. The first private distance-sensitive representation [9] runs their mechanism on Goswami et al.'s signatures and recovers distance sensitivity by enumerating an ℓ_1 ball of candidate signatures at decode time; it is specific to Hamming space (Section 1.4).

Packing lower bounds. Our for-all lower bound (Section 6) is an instance of the packing argument of Hardt and Talwar [17], applied to accept regions.

Private near-neighbor statistics. Differentially private approximate near-neighbor counting has been studied by Andoni, Indyk, Mahabadi and Narayanan [2] and by Aumüller, Boninsegna and Silvestri [3], both using locality-sensitive filters as public structures with private payloads, an architecture our framework shares. Neither addresses the decision problem or space optimality.

Rounding and secluded partitions. Secluded partitions arose in the study of deterministic rounding schemes and replicability [29, 30], and were used by Canonne, Su and Vadhan [8] to determine the randomness complexity of differentially private summation. The spherical cubes of Kindler, O'Donnell, Rao and Wigderson [22] give an optimal noise-resistant randomized discretization of $\mathbb{R}^d$.

Navigable graphs and nets. Graph-based nearest-neighbor indexes such as HNSW [26] and DiskANN [28] are widely used in practice. Indyk and Xu [20] gave worst-case guarantees for such indexes on low-doubling-dimension data, and Diwan et al. [12], Khanna, Padaki and Waingarten [21], and Avi and Musco [4] studied constructions and limits of navigable graphs. Our use of nets and greedy navigation is close to the navigating nets of Krauthgamer and Lee [24]; the difference is that our structure is public and independent of the data, and is used to locate cells of a partition rather than to find a nearest stored point. Nets in doubling metrics can be built in near-linear time with net-trees [18]. MESS [11] combines locality-sensitive hashing, randomized response and a multi-graph HNSW index for private semantic search; it protects queries against an untrusted server, a different threat model, and gives no space or optimality guarantees.

1.8 Organization

Section 2 fixes notation and the private building block. Section 3 gives the partition framework and the main reduction. Section 4 instantiates it in normed spaces, oblivious and randomized. Section 5 develops the distance-oracle construction step by step. Section 6 studies the for-all guarantee. Section 7 proves space optimality and discusses open problems, and Section 8 concludes. Routine proofs are in Appendix A.

2 Preliminaries

2.1 Metric notation

Throughout, (U, d) is a metric space, $k \geq 1$ bounds the size of the represented set, and $r > 0$, $c > 1$ and $\alpha \in (0, 1/2)$ are the radius, the approximation factor and the error budget. We write $B(q, \rho) = \{u \in U : d(u, q) \leq \rho\}$ for the closed ball and $\operatorname{diam}(A) = \sup_{u,v \in A} d(u, v)$ for the diameter of a set. For $A \subseteq U$ we write $d(q, A) = \inf_{u \in A} d(q, u)$.

Definition 2.1 (Separated sets, nets, packing number)**.** Let $\rho > 0$. A set $X \subseteq U$ is *ρ-separated* if $d(x, x') > \rho$ for all distinct $x, x' \in X$, and *ρ-covering* if every $u \in U$ has some $x \in X$ with $d(u, x) \leq \rho$. A *ρ-net* is a set that is both. The *packing number* $M(U, \rho)$ is the supremum of $|X|$ over ρ-separated $X \subseteq U$.

Definition 2.2 (Doubling)**.** (U, d) has *doubling constant* λ if every ball of radius $2R$ can be covered by λ balls of radius R, for every $R > 0$. Its *doubling dimension* is $\dim_{\mathrm{D}} = \log_2 \lambda$.

For example $(\mathbb{R}^d, \ell_\infty)$ has doubling constant 2^d, and every finite metric has a finite doubling constant, although possibly a large one (a star with many leaves has a large λ).

Lemma 2.3 (Packing in doubling spaces [16])**.** *In a metric with doubling constant λ, every ρ-separated set has at most $\lambda^{\lceil \log_2(2\tau/\rho) \rceil}$ points in any ball of radius $\tau \geq \rho/2$.*

Proof. Let $m = \lceil \log_2(2\tau/\rho) \rceil$. Applying the doubling property m times covers $B(x, \tau)$ by λ^m balls of radius $\tau/2^m \leq \rho/2$. Two points in one such ball are at distance at most ρ, so by separation each small ball contains at most one point of the set. □

2.2 Differential privacy

Definition 2.4 (Neighbouring sets)**.** Sets $S, S' \subseteq U$ of size at most k are *neighbouring*, written $S \sim S'$, if $S' = S \cup \{x\}$ for some $x \in U \setminus S$, or symmetrically.

Definition 2.5 (Differential privacy [13])**.** A randomized algorithm $\mathcal{M}$ on sets is (ε, δ)-*differentially private* if for all $S \sim S'$ and all measurable sets T of outputs, $\Pr[\mathcal{M}(S) \in T] \leq e^\varepsilon \Pr[\mathcal{M}(S') \in T] + \delta$. It is *pure* if $\delta = 0$.

We use two standard closure properties [14]. *Post-processing*: if $\mathcal{M}$ is (ε, δ)-DP then so is $g \circ \mathcal{M}$ for any (randomized) map g independent of the input. *Preprocessing*, in the form we need, is the following.

Lemma 2.6 (Neighbour-preserving preprocessing)**.** *Let f map sets of size at most k in U to sets of size at most k in a key universe $\mathcal{I}$, such that $S \sim S'$ implies $f(S) = f(S')$ or $f(S) \sim f(S')$. If $\mathcal{M}$ is (ε, δ)-DP on key sets, then $\mathcal{M} \circ f$ is (ε, δ)-DP on sets in U.*

Proof. Fix $S \sim S'$. If $f(S) = f(S')$ the output distributions coincide; otherwise $f(S) \sim f(S')$ and the privacy of $\mathcal{M}$ applies directly. □

2.3 The target: DPDS-representations

Definition 2.7 ((c, r, α)-DPDS-representation)**.** A pair $(\mathsf{Rep}, \mathsf{Query})$, with Rep a randomized map from sets of size at most k to strings and Query a map from a string and a point $q \in U$ to $\{0, 1\}$, is an (ε, δ)-*differentially private* (c, r, α)-*distance-sensitive set representation* if:

(i) *Completeness.* If $d(u,q) \le r$ for some $u \in S$, then $\Pr[\mathsf{Query}(\mathsf{Rep}(S),q)=1] \ge 1-\alpha$.
(ii) *Soundness.* If $d(u,q) > cr$ for every $u \in S$, then $\Pr[\mathsf{Query}(\mathsf{Rep}(S),q)=1] \le \alpha$.
(iii) *Privacy.* Rep is (ε,δ)-differentially private.
No guarantee is made for q in the annulus $r < d(q,S) \le cr$, and the probabilities in (i) and (ii) are taken *per query*: the quantifier over q is outside the probability.

Since Query only reads the release, any number of queries is post-processing and costs no additional privacy. A per-query guarantee is meaningful only for queries chosen without knowledge of the randomness it averages over, so we make explicit what a query may depend on. Besides the private release Π, a construction may use public data that Query reads, such as a partition, and this public data may itself be random.

Definition 2.8 (Oblivious and non-oblivious representations)**.** A DPDS-representation is *oblivious* if (i) and (ii) of Definition 2.7 hold for every query q chosen as an arbitrary function of the public data, the probability being over Rep alone; this holds automatically when the public data is deterministic. It is *non-oblivious* if (i) and (ii) hold for each q chosen independently of the public data, the probability being over the public randomness and Rep.

No per-query construction is correct against queries chosen after seeing Π itself: once Π is published its false positives are fixed, and a user who reads it can find queries that it answers wrongly. Correctness against such queries is the *for-all* guarantee of Section 6, which moves the quantifier over q inside the probability and changes the cost of privacy from a local to a global quantity. The three levels are thus: queries independent of all randomness (non-oblivious), queries that may depend on the public data but not on Π (oblivious), and queries that may depend on Π (for-all); Table 4 compares their costs.

The next fact explains why false negatives must be admitted into the error budget. It is a randomized-response argument on one membership bit and appears in several forms in the literature [27, 9].

Fact 2.9 (Two-sided error is unavoidable)**.** *Let* $(\mathsf{Rep},\mathsf{Query})$ *be such that* Rep *is* (ε,δ)*-DP. Fix* S *and* $S' = S \cup \{x\}$ *with* $x \notin S$*, and let* $a = \Pr[\mathsf{Query}(\mathsf{Rep}(S'),x)=1]$ *and* $b = \Pr[\mathsf{Query}(\mathsf{Rep}(S),x)=1]$*. Then*

$$b \;\ge\; e^{-\varepsilon}(a-\delta) \qquad \text{and} \qquad 1-a \;\ge\; e^{-\varepsilon}\big((1-b)-\delta\big).$$

In particular, if both error rates are at most α*, then* $\alpha \ge (1-\delta)/(e^{\varepsilon}+1)$*.*

Proof. The event $\{\mathsf{Query}(\cdot,x)=1\}$ is a post-processing of Rep. Applying DP to it and to its complement in both directions gives $a \le e^{\varepsilon}b+\delta$ and $1-b \le e^{\varepsilon}(1-a)+\delta$, which rearrange to the two bounds. If $1-a \le \alpha$ and $b \le \alpha$ then $1-\alpha \le a \le e^{\varepsilon}\alpha+\delta$. □

2.4 The private building block

We use a private membership structure as a black box and isolate the properties the analysis needs.

Definition 2.10 (Private membership structure)**.** A pair $(\mathsf{Rep}_0,\mathsf{Query}_0)$ over a key universe $\mathcal{I}$ is an $(\varepsilon_0,\delta_0,\beta,p)$*-private membership structure for* k *keys* if for every $K \subseteq \mathcal{I}$ with $|K| \le k$:
1. *recall error:* $\Pr[\mathsf{Query}_0(\mathsf{Rep}_0(K),\iota)=0] \le \beta$ for every $\iota \in K$;
2. *false-positive rate:* $\Pr[\mathsf{Query}_0(\mathsf{Rep}_0(K),\iota)=1] \le p$ for every $\iota \notin K$;

3. *privacy:* Rep_0 is $(\varepsilon_0, \delta_0)$-DP under adding or removing one key.

The false-positive guarantee is a *marginal* statement: the events for distinct keys are determined by the same random object and need not be independent. Every bound in this paper uses only union bounds, which require no independence.

Fact 2.11 (Patel, Persiano, Seo and Yeo [27])**.** *In the model with fully random hash functions, for every $\varepsilon_0 > 0$ and $\delta_0 \in (0,1)$ there is an $(\varepsilon_0, \delta_0, \beta, p)$-private membership structure with $\beta = p = 1/(e^{\varepsilon_0}+1)$, using $1.05\, k\varepsilon_0 \log_2 e$ bits plus the description of three hash functions, with encoding time $O(k \log(1/\delta_0))$ and decoding time $O(\log(1/\delta_0))$. A pure variant ($\delta_0 = 0$) uses $k\varepsilon_0 \log_2 e$ bits with decoding time $O(k)$.*

The structure embeds the key set in a random linear system over a finite field: each retained key contributes one equation, keys are retained independently with a fixed exclusion probability, and free variables are set uniformly at random. Exclusion produces the false negatives, chance satisfaction of a non-key's equation produces the false positives, and the two together give privacy. In the approximate variant, δ_0 is the probability that the random band system fails to be solvable, in which case the input is released unmodified. By Fact 2.9 the relation $\beta = p = 1/(e^{\varepsilon_0}+1)$ is the best possible.

3 The Partition Framework

The reduction has two moving parts: a public partition carrying the geometry, and the private membership structure of Definition 2.10 carrying the privacy. We fix the partition first, then show that combining the two costs nothing beyond what each already guarantees on its own.

3.1 Cell decompositions and the straddle number

Definition 3.1 (Cell decomposition, width, straddle number)**.** A *cell decomposition* is a partition $\mathcal{C} = \{C_\iota\}_{\iota \in \mathcal{I}}$ of U. We write $\mathcal{C}(u)$ for the cell containing u and $\mathrm{id}(C) \in \mathcal{I}$ for the index of a cell C. The *width* of $\mathcal{C}$ is $\Delta(\mathcal{C}) = \sup_{C \in \mathcal{C}} \mathrm{diam}(C)$, and its *$\rho$-straddle number* is

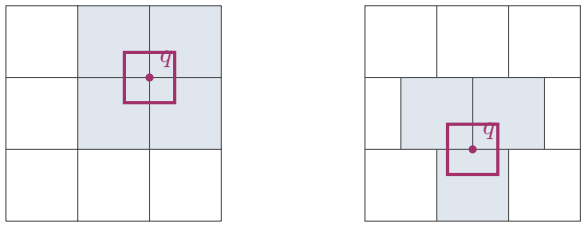
$$\kappa_{\mathcal{C}}(\rho) = \sup_{q \in U} \big|\{C \in \mathcal{C} : \; C \cap B(q,\rho) \neq \emptyset\}\big|.$$

$\mathcal{C}$ is (c,r)-*admissible* if $\Delta(\mathcal{C}) \le (c-1)r$.

The two parameters pull against each other. Small width is needed for the approximation guarantee and small straddle number for the privacy–utility trade-off, and refining cells to reduce the former increases the latter. Figure 2 shows that the arrangement of the cells, not only their size, determines κ.

3.2 The construction

Fix a (c,r)-admissible $\mathcal{C}$, public and independent of S, and put $K(S) = \{\mathrm{id}(\mathcal{C}(u)) : u \in S\}$, so that $|K(S)| \le |S| \le k$.



aligned grid: $\kappa = 4 = 2^d$ offset rows: $\kappa = 3 = d + 1$

Figure 2: Why the straddle number, not the width, is the quantity to optimize ($d = 2$, ℓ_∞ balls). In an aligned grid a ball centred at a lattice corner meets 2^d cells. Offsetting alternate rows destroys that corner, and a ball of the same radius meets only $d + 1$. As $\varepsilon_0 = \ln(2\kappa/\alpha)$ (Theorem 3.7), this is the difference between a privacy parameter linear in d and one logarithmic in it. The picture is schematic: the d-dimensional partitions attaining $\kappa = d + 1$ are the secluded partitions of [30], not naive row offsets.

Algorithm 1 $\mathsf{Rep}(S)$

Require: $S \subseteq U$ with $|S| \le k$; public (c, r)-admissible $\mathcal{C}$; privacy parameter ε_0

$K \leftarrow \{\mathrm{id}(\mathcal{C}(u)) : u \in S\}$

return $\Pi \leftarrow \mathsf{Rep}_0(K, \varepsilon_0)$

3.3 Analysis

Lemma 3.2 (Sensitivity)**.** *If $S \sim S'$ then $K(S)$ and $K(S')$ are equal or differ in exactly one element.*

Proof. Write $S' = S \cup \{x\}$. As $\mathcal{C}$ is a partition, x lies in exactly one cell $\mathcal{C}(x)$. If $\mathcal{C}(x) \cap S \neq \emptyset$ then $\mathrm{id}(\mathcal{C}(x)) \in K(S)$ already and the key sets coincide; otherwise exactly $\mathrm{id}(\mathcal{C}(x))$ is added. □

Corollary 3.3 (Privacy)**.** *If $(\mathsf{Rep}_0, \mathsf{Query}_0)$ is $(\varepsilon_0, \delta_0)$-DP, then Rep is $(\varepsilon_0, \delta_0)$-DP.*

Proof. $\mathcal{C}$ is fixed independently of S, so $S \mapsto K(S)$ is a data-independent map that carries neighbours to neighbours or to identical inputs (Lemma 3.2). Apply Lemma 2.6. □

The proof uses that $\mathcal{C}$ is a *partition* in an essential way. If the cells formed a cover, as the buckets of several locality-sensitive hash functions do, one point could occupy many cells, the map $S \mapsto K(S)$ would have sensitivity equal to the maximum number of cells containing a point, and group privacy would multiply ε_0 by that number.

Lemma 3.4 (Width bound)**.** *If $\mathcal{C}$ is (c, r)-admissible and $C \cap B(q, r) \neq \emptyset$, then $C \subseteq B(q, cr)$.*

Proof. Let $v \in C \cap B(q, r)$ and $u \in C$. Then $d(u, q) \le d(u, v) + d(v, q) \le \mathrm{diam}(C) + r \le (c - 1)r + r = cr$. □

Lemma 3.4 is the structural payoff: under admissibility, every false positive of Query comes from the membership structure and none from the geometry.

Theorem 3.5 (Soundness)**.** *If $\mathcal{C}$ is (c, r)-admissible and $d(u, q) > cr$ for all $u \in S$, then* $\Pr[\mathsf{Query}(\mathsf{Rep}(S), q) = 1] \le \kappa_{\mathcal{C}}(r) \cdot p$.

Algorithm 2 $\mathsf{Query}(\Pi, q)$

Require: release Π; query $q \in U$; public $\mathcal{C}$

```
P(q) ← {C ∈ C : C ∩ B(q, r) ≠ ∅}
for each C ∈ P(q) do
    if Query_0(Π, id(C)) = 1 then return 1
    end if
end for
return 0
```

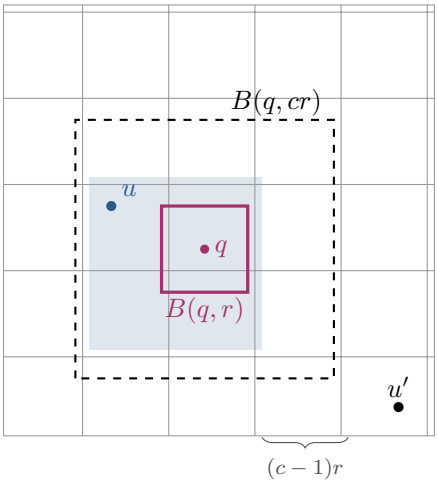


Figure 3: The query of Algorithm 2 in $\mathbb{R}^2$ under ℓ_∞ with $c = 3$. It probes exactly the cells met by $B(q, r)$ (shaded), here four. Since each cell has width at most $(c-1)r$, Lemma 3.4 places every probed cell inside $B(q, cr)$ (dashed): a stored element in a probed cell, such as u, is a legitimate c-approximate answer, while a distant u' lies in no probed cell. No false positive is therefore geometric.

Proof. Let $C \in \mathcal{P}(q)$. By Lemma 3.4, $C \subseteq B(q, cr)$, so C contains no element of S and $\mathrm{id}(C) \notin K(S)$. The query outputs 1 only if some probed key is a false positive, each of probability at most p by Definition 2.10. There are $|\mathcal{P}(q)| \le \kappa_{\mathcal{C}}(r)$ probes, and a union bound applies. □

Theorem 3.6 (Completeness)**.** *If $d(u, q) \le r$ for some $u \in S$, then* $\Pr[\mathsf{Query}(\mathsf{Rep}(S), q) = 0] \le \beta$.

Proof. $u \in \mathcal{C}(u) \cap B(q, r)$, so $\mathcal{C}(u) \in \mathcal{P}(q)$ is probed, and $\mathrm{id}(\mathcal{C}(u)) \in K(S)$. The query returns 0 only if Query_0 fails to recall this stored key. □

Theorem 3.7 (Main reduction)**.** *Let (U, d) admit a (c, r)-admissible cell decomposition $\mathcal{C}$ with $\kappa = \kappa_{\mathcal{C}}(r)$, and instantiate Algorithms 1 and 2 with an $(\varepsilon_0, \delta_0, \beta, p)$-private membership structure (Definition 2.10) with*

$$\beta \le \frac{\alpha}{2}, \qquad p \le \frac{\alpha}{2\kappa}.$$

Then $(\mathsf{Rep}, \mathsf{Query})$ is an oblivious $(\varepsilon_0, \delta_0)$-differentially private (c, r, α)-DPDS-representation making at most κ probes per query. In particular, the structure of Fact 2.11 at $\varepsilon_0 = \ln(2\kappa/\alpha)$ satisfies these conditions and uses $O(k \log(\kappa/\alpha))$ bits beyond the public $\mathcal{C}$.

Proof. Privacy is Corollary 3.3. Soundness: Theorem 3.5 bounds the error by $\kappa p \le \alpha/2$. Completeness: Theorem 3.6 bounds the error by $\beta \le \alpha/2$. Both bounds hold for every q, whatever public data it depends on, since $\mathcal{C}$ is deterministic, so the representation is oblivious. For the structure of Fact 2.11, $\beta = p = 1/(e^{\varepsilon_0}+1) = 1/(2\kappa/\alpha+1) \le \alpha/(2\kappa)$, and its space is $O(k\varepsilon_0)$. □

For the distance-oracle construction of Section 5 it is convenient to let the decoder probe more cells than strictly necessary. The proof above goes through unchanged as long as every probed cell stays inside $B(q, cr)$.

Theorem 3.8 (Probing a superset)**.** *Let $\mathcal{C}$ be a cell decomposition (not necessarily admissible), and let $q \mapsto \mathcal{P}'(q) \subseteq \mathcal{C}$ be a map computed from q and public data only, such that for every $q \in U$:*
(a) every cell meeting $B(q, r)$ belongs to $\mathcal{P}'(q)$, and
(b) every $C \in \mathcal{P}'(q)$ satisfies $C \subseteq B(q, cr)$.
Let $\kappa' = \sup_q |\mathcal{P}'(q)|$. If Query *probes $\mathcal{P}'(q)$ instead of $\mathcal{P}(q)$, and the private membership structure satisfies $\beta \le \alpha/2$ and $p \le \alpha/(2\kappa')$ (for instance that of Fact 2.11 at $\varepsilon_0 = \ln(2\kappa'/\alpha)$), then* (Rep, Query) *is an $(\varepsilon_0, \delta_0)$-differentially private (c, r, α)-DPDS-representation with at most κ' probes per query, oblivious if $\mathcal{C}$ and the map $q \mapsto \mathcal{P}'(q)$ are deterministic.*

Proof. Privacy does not depend on the query. For soundness, (b) shows that no probed cell contains a point of S, so every acceptance is a false positive of one of at most κ' probes. For completeness, (a) shows that the cell of a witness $u \in S \cap B(q, r)$ is probed. The calculation is that of Theorem 3.7 with κ' in place of κ. □

Theorem 3.7 is the special case $\mathcal{P}'(q) = \mathcal{P}(q)$, where (b) is Lemma 3.4. Note that (a) and (b) together force $\mathcal{C}$ to be "admissible near every query": if a cell C meets $B(q, r)$ then $C \subseteq B(q, cr)$.

Remark 3.9 (What the bounds depend on). The construction uses r only as a unit: rescaling the metric and r together leaves it unchanged. Neither $|U|$ nor the number of cells $|\mathcal{C}|$ appears in the privacy parameter or in the size of the private release; the public partition must be described separately, which is free in $\mathbb{R}^d$ and costs $\Delta_{\mathrm{asp}}^{O(\dim_{\mathrm{D}})}$ in the distance-oracle construction (Theorem 5.14). The approximation factor is not absent but hidden inside κ, and it is the dominant lever (Theorem 4.6). All of this follows from one observation: the underlying structure stores k keys at false-positive rate p in about $k \log_2(1/p)$ bits, the classical Bloom-filter bound, which carries no $\log |U|$ term. Privacy costs space only by imposing the floor $p \ge e^{-\varepsilon_0}$ (Fact 2.9); the metric costs space only by imposing the ceiling $p \le \alpha/(2\kappa)$ through Theorem 3.5. Everything acts by moving the single scalar p.

Remark 3.10 (Comparison with the Hamming-space construction). The construction of [9] routes membership through a vector signature and enumerates candidate signatures at decode time, so its privacy parameter carries the logarithm of the candidate count. Here the number of probes is the straddle number of a partition of the query space, a property of the partition alone. The two constructions are not directly comparable; Section 1.4 discusses when each applies.

4 Instantiations in Normed Spaces

Theorem 3.7 turns every question about privacy into a question about κ. We instantiate it in three settings of increasing generality: explicit ℓ_∞ partitions nearly matching a lower bound for partitions (Section 4.1), their ℓ_2 reading at an extra $\sqrt{d}$ factor (Section 4.2), and a randomly

drawn partition giving a privacy parameter independent of d in every ℓ_p at once, with the dimension moved into the approximation factor and a weaker guarantee (Section 4.3).

4.1 Secluded partitions in ℓ_∞

Definition 4.1 (Secluded partition [30])**.** A partition $\mathcal{P}$ of $\mathbb{R}^d$ is (k, ε)-*secluded* if for every $x \in \mathbb{R}^d$ the ball $B_\infty(x, \varepsilon)$ meets at most k members of $\mathcal{P}$.

A (k, ε)-secluded partition is exactly a cell decomposition with straddle number at most k at radius ε; only rescaling is required.

Lemma 4.2 (Rescaling)**.** *Let $\mathcal{P}$ be (k, ε)-secluded with cells of ℓ_∞-diameter at most 1. Then for $s > 0$ the dilation $s\mathcal{P}$ has width at most s and $\kappa_{s\mathcal{P}}(s\varepsilon) \le k$. Choosing $s = r/\varepsilon$ gives $\kappa_{s\mathcal{P}}(r) \le k$, and $s\mathcal{P}$ is (c, r)-admissible whenever $c \ge 1 + 1/\varepsilon$.*

Proof. Dilation scales all ℓ_∞ distances by s, so diameters scale by s and a ball of radius $s\varepsilon$ meets exactly the dilations of the cells that a ball of radius ε met before. With $s = r/\varepsilon$, admissibility requires $r/\varepsilon \le (c-1)r$. □

Fact 4.3 (Vander Woude et al. [29, 30])**.** *For every $d \ge 1$ and every $f = f(d) \in \{1, \dots, d\}$ there is a partition of $\mathbb{R}^d$ into axis-parallel unit hypercubes that is $((f+1)^{\lceil d/f \rceil}, 1/(2f))$-secluded. For $f = d$ it is explicit and the cell of a point is computable in $\mathrm{poly}(d)$ time, with $k = d + 1$ and $\varepsilon = 1/(2d)$. Conversely, every partition of $\mathbb{R}^d$ into Lebesgue-measurable members of ℓ_∞-diameter at most 1 that is (k, ε)-secluded satisfies $k \ge (1 + 2\varepsilon)^d$.*

Theorem 4.4 (ℓ_∞ upper bound)**.** *Fix $d \ge 1$ and $f = f(d)$. For $U = \mathbb{R}^d$ under ℓ_∞ there is an $(\varepsilon_0, \delta_0)$-differentially private (c, r, α)-DPDS-representation with*

$$c = 2f + 1, \qquad \varepsilon_0 = \left\lceil \frac{d}{f} \right\rceil \ln(f+1) + \ln \frac{2}{\alpha},$$

whose cell decomposition is explicit, with at most $(f+1)^{\lceil d/f \rceil}$ probes per query.

The proof applies Lemma 4.2 with $\varepsilon = 1/(2f)$ and substitutes into Theorem 3.7; it is in Appendix A.

Corollary 4.5. *Taking $f = d$: $c = 2d + 1$ and $\varepsilon_0 = \ln(d+1) + \ln(2/\alpha) = O(\log d) + \ln(1/\alpha)$, with at most $d + 1$ probes per query.*

Listing the cells. The decoder must list the cells meeting $B_\infty(q, r)$, not only the cell of q. After rescaling, cells are axis-parallel cubes of side 1 and the ball is a cube of side $2\varepsilon = 1/f \le 1$. In each coordinate a cell's interval is at least as long as the ball's, so it cannot lie strictly inside it, and the intersection contains an endpoint of the ball's interval; hence every cell meeting the ball contains a *vertex* of the ball. Locating the 2^d vertices therefore lists the probed cells, in 2^d cell computations. This is fine for moderate d, but we do not know an enumeration in time polynomial in d for the partitions of [30]; the randomized construction of Section 4.3 needs only the cell of q.

For comparison, routing $\mathbb{R}^d$ through its doubling constant (Theorem 5.14 below) gives only $\varepsilon_0 = \Theta(d)$, since ℓ_∞ on $\mathbb{R}^d$ has doubling constant $2^{\Theta(d)}$. The improvement is purchased by letting c grow; taking $f = 1$ recovers constant $c = 3$ at $\varepsilon_0 = d \ln 2 + \ln(2/\alpha)$. That this trade is unavoidable for partitions follows from the lower bound of [30].

Theorem 4.6 (Lower bound for partitions)**.** *Let $\mathcal{C}$ be any (c, r)-admissible cell decomposition of $(\mathbb{R}^d, \ell_\infty)$ into Lebesgue-measurable cells. Then*

$$\kappa_{\mathcal{C}}(r) \;\geq\; \Big(1 + \frac{2}{c-1}\Big)^d = \Big(\frac{c+1}{c-1}\Big)^d,$$

hence any instantiation of Theorem 3.7 over $\mathcal{C}$ with $\delta_0 \leq 1/4$ has $\varepsilon_0 \geq d \ln \frac{c+1}{c-1} \geq \frac{2d}{c+1}$. In particular $\varepsilon_0 = \Omega(d)$ for constant c, and $\varepsilon_0 = O(\log d)$ requires $c = \Omega(d/\log d)$.

Proof. Admissibility gives $\Delta(\mathcal{C}) \leq (c-1)r$. Rescale by $1/\Delta(\mathcal{C})$, so cells have ℓ_∞-diameter at most 1 (hence bounded measure) and the query radius becomes $\varepsilon = r/\Delta(\mathcal{C}) \geq 1/(c-1)$; the straddle number at radius r becomes the seclusion number at radius ε. By Fact 4.3, $\kappa_{\mathcal{C}}(r) \geq (1+2\varepsilon)^d \geq (1+2/(c-1))^d$. An instantiation of Theorem 3.7 has $\beta \leq \alpha/2$ and $p \leq \alpha/(2\kappa)$, while Fact 2.9 applied to the membership structure gives $1 - \beta \leq e^{\varepsilon_0} p + \delta_0$; for $\alpha \leq 1/2$ and $\delta_0 \leq 1/4$ this yields $e^{\varepsilon_0} \geq 2\kappa(1 - \beta - \delta_0)/\alpha \geq \kappa$. For the last claim, $\ln(1+x) \geq x/(1+x)$ with $x = 2/(c-1)$ gives $d \ln \frac{c+1}{c-1} \geq \frac{2d}{c+1}$, which is $O(\log d)$ only if $c = \Omega(d/\log d)$. □

Remark 4.7 (Scope of the lower bound). Theorem 4.6 constrains decoders of the form of Algorithms 1 and 2. A construction that does not route membership through a partition of the ambient space—for instance one based on locality-sensitive hashing, where the cells form a cover—is not constrained by it. Leaving the partition setting is not free, however: Lemma 3.2 requires a partition, and for a cover the sensitivity of the occupancy map grows with the number of cells containing a point. Whether the bound holds for every representation is Conjecture 7.3.

How tight is this? Writing $c = 2f + 1$, the d-dependent terms of Theorems 4.4 and 4.6 are $\lceil d/f \rceil \ln(f+1) \leq (d/f + 1)\ln(f+1)$ and $d \ln(1 + 1/f) \geq d/(f+1)$, whose ratio is at most $(f+1)(1/f + 1/d)\ln(f+1) \leq 4\ln(f+1) = O(\log c)$ for $1 \leq f \leq d$; at $c = 3$ both equal $d \ln 2$. At $c = \Theta(d)$, however, the lower bound is only $\Omega(1)$, and we do not know whether the $\ln(d+1)$ of Corollary 4.5 is necessary. Table 2 gives concrete numbers.

4.2 Euclidean distance, obliviously

The same partitions may be read under ℓ_2. Since $\|x\|_\infty \leq \|x\|_2$ we have $B_2(q, r) \subseteq B_\infty(q, r)$, so the straddle number transfers unchanged; but a hypercube of side s has ℓ_2-diameter $s\sqrt{d}$, so admissibility costs a factor $\sqrt{d}$.

Theorem 4.8 (ℓ_2, oblivious)**.** *For $U = \mathbb{R}^d$ under ℓ_2 and any $f = f(d)$ there is an oblivious $(\varepsilon_0, \delta_0)$-differentially private (c, r, α)-DPDS-representation with $c = 2f\sqrt{d} + 1$ and $\varepsilon_0 = \lceil d/f \rceil \ln(f + 1) + \ln(2/\alpha)$, over the decomposition of Theorem 4.4. At $f = d$ this is $c = \Theta(d^{3/2})$ with $\varepsilon_0 = O(\log d) + \ln(1/\alpha)$.*

Remark 4.9 (The $\sqrt{d}$, and the Euclidean lower bound). The loss is exactly the gap between the diameter and the inradius of a hypercube in ℓ_2: the straddle number is governed by the inradius and admissibility by the diameter. A secluded partition family designed for ℓ_2 could improve the factor to $\Theta(d)$; none appears to be known, those of [30] being stated for ℓ_∞. Theorem 4.6 likewise does not transfer as stated, its rescaling being norm-dependent; what survives directly is the volume bound $\kappa \geq (2/(c-1))^d$, exponential only for $c < 3$. Two questions are therefore open: whether an oblivious partition of $(\mathbb{R}^d, \ell_2)$ achieves $c = \Theta(d)$ with $\varepsilon_0 = O(\log d)$, and whether partitions of $(\mathbb{R}^d, \ell_2)$ need $\varepsilon_0 = \Omega(d)$ for constant $c \geq 3$.

d	construction	c	ε_0	bits/key	probes	LB on ε_0
10	aligned grid	3	9.9	15.0	2^{10}	6.9
	secluded, $f = d$ (Corollary 4.5)	21	5.4	8.2	11	1.0
	shifted grid (Corollary 4.13), non-obliv.	401	3.0	4.5	1	–
100	aligned grid	3	72.3	109.5	2^{100}	69.3
	secluded, $f = d$	201	7.6	11.5	101	1.0
	shifted grid, non-obliv.	4001	3.0	4.5	1	–
1000	aligned grid	3	696.1	1054.5	2^{1000}	693.1
	secluded, $f = d$	2001	9.9	15.0	1001	1.0
	shifted grid, non-obliv.	40001	3.0	4.5	1	–

Table 2: Concrete parameters in $(\mathbb{R}^d, \ell_\infty)$ at $\alpha = 0.1$, $\delta_0 > 0$, using Fact 2.11 ($1.05\,\varepsilon_0 \log_2 e$ bits per stored cell). The aligned grid of side $2r$ is the naive partition; the secluded partition at $f = 1$ has the same parameters, which are optimal for partitions at $c = 3$ (last column, Theorem 4.6). An aligned grid at $c = 2d + 1$ still has $\kappa = 2^d$. Without a partition, the structure of [27] over the raw points answers only exact-membership queries ($r = 0$).

4.3 Randomized partitions and one-probe constructions

Definition 2.7 quantifies over queries outside the probability, so the decomposition need not be fixed: it may be drawn from a public distribution, and correctness required only for each q on average over that draw. This relaxation buys a great deal: the probe count falls to one and ε_0 becomes independent of the dimension. The price is that such constructions are non-oblivious (Definition 2.8). A user who knows the realized partition, which is public because the decoder needs it, can choose q so that $B(q, r)$ is split by a cell boundary; if the only nearby point of S lies across that boundary, the query misses it with probability close to 1. For an oblivious construction the error at such a q is still at most α. All results of Sections 3, 4.1 and 4.2 are oblivious; Table 4 compares the levels of guarantee.

Definition 4.10 (Separating distribution)**.** A distribution $\mathcal{D}$ over cell decompositions of U is (c, r, η)-*separating* if every $\mathcal{C}$ in its support has width at most $(c-1)r$, and for every $q \in U$,

$$\Pr_{\mathcal{C}\sim\mathcal{D}}\big[\exists\, C \in \mathcal{C} :\ B(q, r) \subseteq C\big] \ \geq\ 1 - \eta.$$

Theorem 4.11 (One-probe reduction)**.** *Let $\mathcal{D}$ be (c, r, η)-separating. Draw $\mathcal{C} \sim \mathcal{D}$ publicly, run Algorithm 1 on $\mathcal{C}$, and let* Query *probe only the single cell containing q. With the private membership structure of Fact 2.11 at $\varepsilon_0 = \ln(2/\alpha)$ and $\eta \leq \alpha/2$, the result is a non-oblivious $(\varepsilon_0, \delta_0)$-differentially private (c, r, α)-DPDS-representation using $O(k \log(1/\alpha))$ bits, with one probe per query.*

Proof. Privacy is Corollary 3.3, unaffected because $\mathcal{C}$ is drawn independently of S. Soundness: the probed cell contains q and has diameter at most $(c-1)r$, so any $u \in S$ lying in it satisfies $d(u, q) \leq (c-1)r \leq cr$; hence no false positive is geometric and the single probe errs with probability at most $p = 1/(e^{\varepsilon_0} + 1) = \alpha/(2 + \alpha) \leq \alpha/2$. Completeness: on the separating event, which fails with probability at most η, every u with $d(u, q) \leq r$ lies in the probed cell, and the probe then errs with probability at most $\beta = p \leq \alpha/2$; a union bound gives error at most $\eta + \beta \leq \alpha$. □

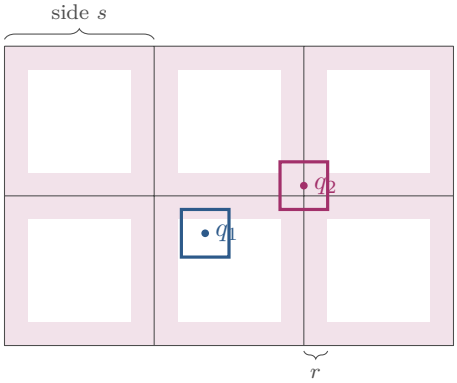


Figure 4: A randomly shifted grid of side s (Corollary 4.13), $d = 2$. The ball $B(q, r)$ lies inside a single cell unless q falls in the shaded strip of width r along the cell boundaries. Under a uniform random shift this happens with probability at most $2dr/s$ for every fixed q. The query q_1 is answered with one probe; q_2 is a bad query for this draw, which is why the guarantee is non-oblivious.

The metric now enters only through the relation between c and η, which for a tiling is governed by two shape parameters (Figure 4).

Lemma 4.12 (Shape bound)**.** *Let T be a* convex *body that tiles $\mathbb{R}^d$ under a lattice, with volume* 1*, surface area A and diameter Δ in the norm considered. Randomly translating the tiling and scaling it by s gives a (c, r, η)-separating distribution with $\eta \le Ar/s$ and $c = 1 + s\Delta/r$; taking $s = Ar/\eta$,*

$$c = 1 + \frac{A\Delta}{\eta}.$$

Corollary 4.13 (Randomly shifted grids in ℓ_p)**.** *For $U = \mathbb{R}^d$ under ℓ_p, $1 \le p \le \infty$, the randomly shifted axis-aligned grid gives a non-oblivious representation with*

$$c = 1 + \frac{4d^{1+1/p}}{\alpha}, \qquad \varepsilon_0 = \ln \frac{2}{\alpha}, \qquad \textit{space } O(k \log(1/\alpha)),$$

with one probe and $O(d)$ time per query. In particular $c = O(d^{3/2}/\alpha)$ for ℓ_2 and $O(d/\alpha)$ for ℓ_∞. The privacy parameter does not depend on d; the dimension and the error budget have moved into the approximation factor.

Remark 4.14 (Other tiles, and the limit of the method). By Lemma 4.12 the quantity to minimize is the product $A\Delta$ of a convex unit-volume tile. For the cube $A = \Theta(d)$ and $\Delta = \Theta(d^{1/p})$. The isoperimetric inequality gives $A = \Omega(\sqrt{d})$ for every unit-volume body, and any unit-volume body has Euclidean diameter $\Omega(\sqrt{d})$; hence $A\Delta = \Omega(d)$ in ℓ_2, and no convex tiling can push the method below $c = \Omega(d/\alpha)$ there. The spherical cubes of Kindler, O'Donnell, Rao and Wigderson [22] tile $\mathbb{R}^d$ under $\mathbb{Z}^d$ with surface area $O(\sqrt{d})$ and lie in a cube of side 2, so $A\Delta = O(d)$. They are not convex, however, and Lemma 4.12 does not apply to them: for a non-convex tile the region within r of the boundary can have volume much larger than r times the surface area (many small holes add little surface but a large neighbourhood). Their analysis bounds the probability that two nearby points are separated, not that a whole ball is. Whether they, or any tiling, achieve $c = O(d/\alpha)$ in ℓ_2 with one probe is open.

Remark 4.15 (Oblivious versus non-oblivious). Under Definition 2.7 alone, Corollary 4.13 reaches $\varepsilon_0 = \ln(2/\alpha)$, below the $\ln(d+1)+\ln(2/\alpha)$ of Corollary 4.5. The two are not directly comparable. They operate at different approximation factors: $c = 2d+1$ for the secluded partition against $c = 1 + 4d/\alpha$ for the shifted grid in ℓ_∞. They also give different guarantees: the secluded construction is oblivious, so its error bound holds even for queries chosen with knowledge of the partition, while the shifted grid's does not. At $c = \Theta(d)$ Theorem 4.6 gives only $\varepsilon_0 = \Omega(1)$ for partitions, so we do not know whether obliviousness itself costs privacy; this is open.

5 Metrics Given by a Distance Oracle

In $\mathbb{R}^d$ the cell containing a point, and the cells meeting a query ball, are found by arithmetic. In a general metric presented only by a distance oracle, both become search problems. This section shows that a public net together with a public navigable graph on it supplies exactly the missing computability, and that because both depend only on the metric, they cost nothing in privacy. We develop the construction one step at a time, prove each ingredient, and assemble the guarantee in Theorems 5.13 and 5.14.

5.1 The access model, and a running example

We assume U is finite (or, more generally, that the query domain of interest is a finite set given in advance), listed in a fixed public order $u_1, u_2, \ldots$, and that the only operation on points is the evaluation of $d(x, y)$ for points we hold. There are no coordinates, no lattice and no way to "round" a point. Write $\operatorname{diam}(U)$ for the diameter and, for a scale $\rho > 0$, $\Delta_{\text{asp}} = \operatorname{diam}(U)/\rho$ for the *aspect ratio* at that scale. Finiteness is used in two places: to build the net by a scan, and to bound the length of greedy searches. Section 5.9 discusses what survives without it.

Example 5.1 (Exposure notification on a road network)**.** A health department wishes to publish, every day, where confirmed cases have been, so that residents can check whether they were near one, without exposing the cases. The universe U is the set of intersections and short road segments of a city's road graph, obtainable from a public map. The metric is shortest-path walking distance, computed by Dijkstra's algorithm; this is the distance oracle. The data S are the locations visited by cases on one day, with k around 1000. A query asks "was a case within $r = 500\,\text{m}$ of a place I visited?", and with $c = 2$ the answer must be negative, with probability $1 - \alpha$, when every case was more than 1 km away. A grid on the map would be a poor partition here: two points 300 m apart across a river can be 5 km apart by road, so cells that are small in the plane can be huge in road distance. The numbers in this example are illustrative.

5.2 Overview

Figure 5 shows the construction. Steps 1–3 form the public layer and never look at the data; they are a fixed function of the metric and a public seed. Step 4 is the only place where S enters. Step 5 answers a query.

Notation. We write $\gamma > 1$ for the navigability parameter of Definition 5.3 and

$$\theta(\gamma) = \frac{4\gamma - 2}{\gamma - 1}$$

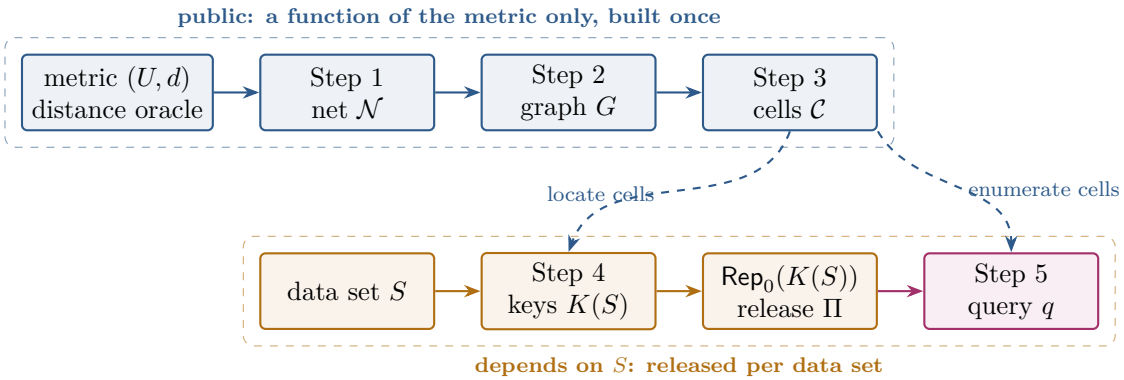


Figure 5: The distance-oracle construction. Everything in the upper box is independent of the data and can be published. Step 4 maps each data point to the identifier of its cell by greedy descent on G; Step 5 uses G to list the cells a query must probe.

for the resulting cell-radius factor (Lemma 5.7). We write γ rather than α for navigability to avoid a clash with the error budget.

5.3 Step 1: a public net

Algorithm 3 Greedy net

Require: points of U in a fixed public order $u_1, u_2, \ldots$; scale $\rho > 0$

$\mathcal{N} \leftarrow \emptyset$
for $i = 1, 2, \ldots$ **do**
  if $d(u_i, z) > \rho$ for every $z \in \mathcal{N}$ **then**
    $\mathcal{N} \leftarrow \mathcal{N} \cup \{u_i\}$
  end if
end for
return $\mathcal{N}$

Lemma 5.2. *Algorithm 3 returns a ρ-net of U. If U has doubling constant λ, then $|\mathcal{N}| \le \lambda^{\lceil \log_2(2\Delta_{\mathrm{asp}}) \rceil} = \Delta_{\mathrm{asp}}^{O(\dim_{\mathrm{D}})}$.*

Proof. Every kept point was more than ρ from all earlier kept points, so $\mathcal{N}$ is ρ-separated. Every rejected point was within ρ of some kept point, so $\mathcal{N}$ is ρ-covering. The size bound is Lemma 2.3 applied to a ball of radius $\mathrm{diam}(U)$ containing U. □

Covering is what will bound the size of cells (Step 3); separation is what will bound how many net points fit in a query ball, and hence the privacy cost (Step 5). The greedy scan uses $O(|U| \cdot |\mathcal{N}|)$ distance evaluations; net-trees [18] reduce this to near-linear in $|U|$ in doubling metrics. The scan order and ρ are fixed in advance and S is never consulted; a net built around the data points would itself reveal where the data are. Figure 6 shows a net on a sample.

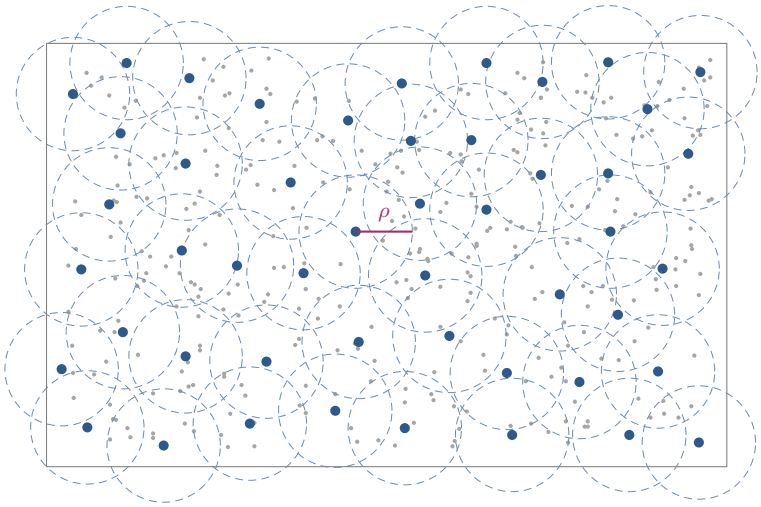


Figure 6: A greedy ρ-net (Algorithm 3) on a sample of points (grey), computed in the Euclidean plane as a stand-in for a general metric. Every sample point lies in one of the dashed ρ-balls around the net points (covering), and net points are more than ρ apart (separation).

5.4 Step 2: a public navigable graph

Definition 5.3 (Navigability)**.** A directed graph $G = (\mathcal{N}, E)$ is γ-*navigable* ($\gamma \geq 1$) if for all distinct $s, t \in \mathcal{N}$ there is an edge $(s, w) \in E$ with

$$d(w, t) < \frac{d(s, t)}{\gamma}.$$

The guarantee is existential and per target: for every vertex s and every target t, *some* out-neighbour of s is a factor γ closer to t (Figure 7). It says nothing about the other out-neighbours, and nothing about the degree, which must be controlled separately. For $\gamma = 1$ it only asks for strict progress; for $\gamma > 1$ the distance to the target shrinks geometrically along greedy paths.

Algorithm 4 Pruning construction of G

```
for each s ∈ N do
    sort N \ {s} by increasing d(s, ·); out(s) ← ∅
    for each t in this order do
        if no w ∈ out(s) has d(w, t) < d(s, t)/γ then
            out(s) ← out(s) ∪ {t}
        end if
    end for
end for
```

Algorithm 4 is the standard "slow preprocessing" of DiskANN [28]; it is navigable by design.

Lemma 5.4 (Pruning)**.** *The graph produced by Algorithm 4 is γ-navigable. If (U, d) has doubling constant λ, every out-degree is at most $\lceil \log_2 \Delta_{\text{asp}} \rceil \cdot \lambda^{\lceil \log_2 4\gamma \rceil + 1} = (O(\gamma))^{\dim_{\text{D}}} \log \Delta_{\text{asp}}$.*

Proof. Navigability. Fix $s \neq t$. Either t was added to $\text{out}(s)$, and $w = t$ gives $d(w, t) = 0 < d(s, t)/\gamma$; or t was skipped, which happens only because some $w \in \text{out}(s)$ already has $d(w, t) < d(s, t)/\gamma$.

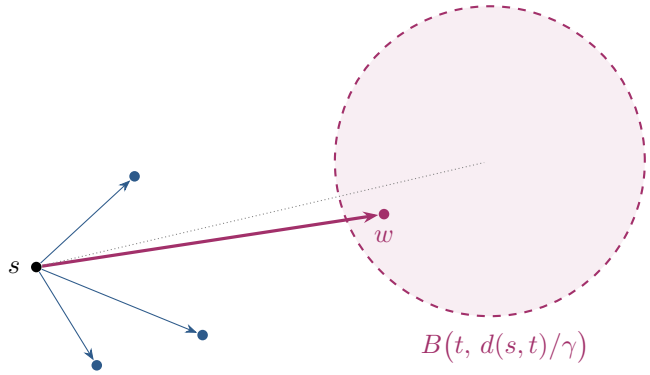


Figure 7: The navigability condition. For every target t, some edge (s, w) lands inside the dashed ball $B(t, d(s,t)/\gamma)$. The pruning construction (Algorithm 4) keeps an edge $s \to t$ only when no existing edge already lands in this ball.

Degree. Let $w, w' \in \mathrm{out}(s)$ with w processed first. When w' was examined, w was already an out-neighbour and w' was still added, so $d(w, w') \geq d(s, w')/\gamma$. Group the out-neighbours into annuli $A_i = \{x : 2^i\rho \leq d(s,x) < 2^{i+1}\rho\}$. Within A_i the out-neighbours are pairwise at distance at least $2^i\rho/\gamma$, hence $(2^{i-1}\rho/\gamma)$-separated, and lie in $B(s, 2^{i+1}\rho)$, so Lemma 2.3 bounds their number by $\lambda^{\lceil \log_2(8\gamma)\rceil} \leq \lambda^{\lceil \log_2 4\gamma\rceil+1}$. Since net points are more than ρ apart and at most $\mathrm{diam}(U)$ apart, only $\lceil \log_2 \Delta_{\mathrm{asp}} \rceil$ annuli are non-empty. □

The pruning construction takes $O(|\mathcal{N}|^2(\log|\mathcal{N}| + \deg))$ time, and the quadratic dependence on $|\mathcal{N}|$ is its bottleneck. A hierarchy of nets gives the same degree in near-linear time.

Lemma 5.5 (Hierarchical construction)**.** *For $i = 0, 1, \ldots, \lceil \log_2 \Delta_{\mathrm{asp}} \rceil$ let $Y_i \subseteq \mathcal{N}$ be a $2^i\rho$-net of $\mathcal{N}$, and set*

$$\mathrm{out}(s) = \big(\mathcal{N} \cap B(s, 2\gamma\rho)\big) \;\cup\; \bigcup_i \big(Y_i \cap B(s, (2\gamma+1)2^i\rho)\big).$$

The resulting graph is γ-navigable, with out-degree $(O(\gamma))^{\dim_{\mathrm{D}}} \log \Delta_{\mathrm{asp}}$ in a λ-doubling metric.

Proof. Fix $s \neq t$ and let $D = d(s,t)$. If $D \leq 2\gamma\rho$ then t itself is an out-neighbour. Otherwise $D/\gamma > 2\rho$, so there is a largest $i \geq 0$ with $2^i\rho < D/\gamma$, and then $2^{i+1}\rho \geq D/\gamma$. Let $w \in Y_i$ be a point with $d(w,t) \leq 2^i\rho$ (covering). Then $d(w,t) < D/\gamma$, and $d(s,w) \leq D + 2^i\rho \leq 2\gamma \cdot 2^i\rho + 2^i\rho = (2\gamma+1)2^i\rho$, so (s,w) is an edge. For the degree, each term is a separated set inside a ball whose radius is $O(\gamma)$ times the separation, so Lemma 2.3 bounds it by $(O(\gamma))^{\dim_{\mathrm{D}}}$, and there are $O(\log \Delta_{\mathrm{asp}})$ terms. □

Any γ-navigable graph can be used in what follows; one may also take the graph of Indyk and Xu [20] as a black box, with degree $2^{O(\dim_{\mathrm{D}})} \log \Delta_{\mathrm{asp}}$. The point that matters for privacy is that G is computed from $\mathcal{N}$ and the metric alone.

Example 5.1, continued. With $\rho = r/10 = 50\,\mathrm{m}$ on a city with about 3000 km of streets, the net has at most on the order of 60,000 points. With $\gamma = 3$ the pruning construction gives out-degrees of a few tens, and with 4-byte identifiers the graph occupies on the order of tens of megabytes. It is computed once from the map and downloaded once by every client.

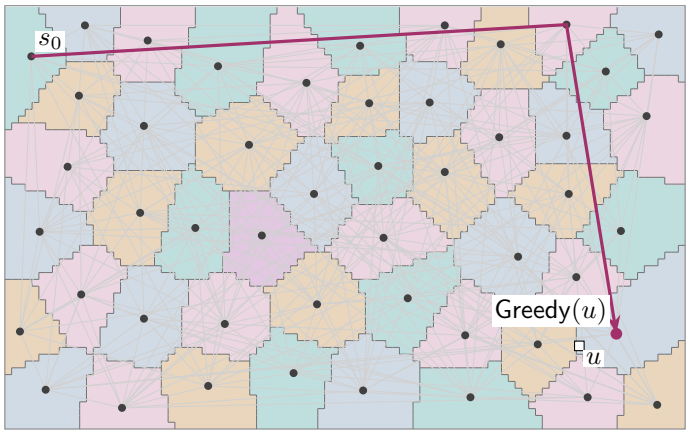


Figure 8: Greedy cells, computed on the net of Figure 6 with the pruned graph of Algorithm 4 ($\gamma = 1.6$, edges in light grey). Every point of the plane is shaded by the net point at which greedy descent toward it stops; each shaded region is a cell C_z, with boundaries drawn in grey (shades are reused only by non-adjacent cells). Cells are not Voronoi cells and can be irregular, yet their radius is bounded by Lemma 5.7. The arrow shows the greedy path from s_0 toward the query point u (square), which stops at $\mathsf{Greedy}(u)$, the identifier of u's cell.

5.5 Step 3: cells as fibers of greedy descent

Fix a start vertex $s_0 \in \mathcal{N}$ and let $T = \lceil \log_\gamma \Delta_{\text{asp}} \rceil$, where here $\Delta_{\text{asp}} = \operatorname{diam}(\mathcal{N})/\rho$.

Algorithm 5 $\mathsf{Greedy}(u)$

$v \leftarrow s_0$
for $j = 1, \dots, T$ **do**
  $b \leftarrow \arg\min_{w \in \text{out}(v)} d(w, u)$ ▷ ties broken by the public order
  if $d(b, u) < d(v, u)$ **then**
    $v \leftarrow b$
  else break
  end if
end for
return v

Definition 5.6 (Greedy cells)**.** For $z \in \mathcal{N}$ let $C_z = \{u \in U : \mathsf{Greedy}(u) = z\}$, and let $\mathcal{C} = \{C_z : z \in \mathcal{N}\}$ (discarding empty sets), with $\text{id}(C_z) = z$.

Because Greedy is a deterministic function on U, its fibers are disjoint and cover U, so $\mathcal{C}$ is a partition; this is all that Lemma 3.2 needs. Defining cells by the procedure itself is essential. Greedy descent on a γ-navigable graph with $\gamma > 1$ returns an *approximate* nearest net point; were the cells the exact Voronoi cells of $\mathcal{N}$, the encoder and the decoder would need exact nearest-neighbour search to agree on where points live. With greedy cells, both sides run the same deterministic procedure. Figure 8 shows greedy cells on the net of Figure 6.

Lemma 5.7 (Cell radius)**.** *Let G be γ-navigable with $\gamma > 1$. For every $u \in U$,*

$$d(u, \mathsf{Greedy}(u)) \;\le\; \theta(\gamma)\,\rho, \qquad \theta(\gamma) = \frac{4\gamma - 2}{\gamma - 1}.$$

Consequently $\Delta(\mathcal{C}) \le 2\theta(\gamma)\rho$.

Proof. By covering, fix $z^* \in \mathcal{N}$ with $d(u, z^*) \le \rho$. Let $v_0 = s_0, v_1, \dots$ be the vertices visited by $\mathsf{Greedy}(u)$ and $D_j = d(v_j, u)$. The sequence D_j is strictly decreasing, so it suffices to show that some visited vertex, or the final one, is within $\theta\rho$ of u.

Suppose the walk is at $v_j \neq z^*$. Navigability gives an out-neighbour w of v_j with $d(w, z^*) < d(v_j, z^*)/\gamma \le (D_j + \rho)/\gamma$, and the greedy choice b satisfies

$$d(b, u) \;\le\; d(w, u) \;\le\; d(w, z^*) + \rho \;<\; \frac{D_j + \rho}{\gamma} + \rho \;\le\; \frac{D_j}{\gamma} + 2\rho. \tag{1}$$

There are three ways the walk can end.

- It visits z^*. Then the final distance is at most $d(z^*, u) \le \rho \le \theta\rho$.
- It stops at some $v_j \neq z^*$ because $d(b, u) \ge D_j$. Then (1) gives $D_j < D_j/\gamma + 2\rho$, i.e. $D_j < 2\rho\gamma/(\gamma - 1) \le \theta\rho$.
- It moves at every one of the T steps. Then $D_{j+1} < D_j/\gamma + 2\rho$ for all $j < T$, and unrolling,

$$D_T \;<\; \frac{D_0}{\gamma^T} + 2\rho \sum_{i \ge 0} \gamma^{-i} \;=\; \frac{D_0}{\gamma^T} + \frac{2\rho\gamma}{\gamma - 1}.$$

  Since $D_0 \le d(s_0, z^*) + \rho \le \operatorname{diam}(\mathcal{N}) + \rho$ and $\gamma^T \ge \Delta_{\text{asp}} = \operatorname{diam}(\mathcal{N})/\rho$, the first term is at most $\rho + \rho/\gamma^T \le 2\rho$, so $D_T < 2\rho + 2\rho\gamma/(\gamma - 1) = \theta(\gamma)\rho$.

Finally, any two points of one cell C_z are within $\theta\rho$ of z, so $\operatorname{diam}(C_z) \le 2\theta\rho$. □

Remark 5.8. Keeping the term $(D_j + \rho)/\gamma$ instead of bounding it by $D_j/\gamma + \rho$ improves the constant to $(3\gamma - 1)/(\gamma - 1)$, e.g. 4 instead of 5 at $\gamma = 3$. We keep $\theta(\gamma)$ for simplicity. In practice the bound is loose: on a planar instance the largest observed $d(u, \mathsf{Greedy}(u))$ was well below $\theta\rho$ (Section 5.9).

Corollary 5.9 (Admissibility)**.** *The greedy cells are* (c, r)*-admissible as soon as* $2\theta(\gamma)\rho \le (c-1)r$*. With* $\gamma = 3$ *we have* $\theta = 5$*, so* $c = 2$ *is achieved with* $\rho = r/10$*.*

The cells are never stored. Any party holding $\mathcal{N}$ and G computes the cell of a point by running Algorithm 5, in $T \cdot \max \deg$ distance evaluations.

Example 5.1, continued. With $\rho = 50\,\text{m}$ and $\gamma = 3$, every location is assigned to a net point at most 250 m away by road, so every cell has road diameter at most $500\,\text{m} = (c-1)r$. A city of diameter 20 km needs $T = \lceil \log_3(20\,\text{km}/50\,\text{m}) \rceil = 6$ greedy steps.

5.6 Step 4: encoding, the only step that sees the data

Encoding is Algorithm 1 over the greedy cells:

$$K(S) = \{\mathrm{id}(\mathcal{C}(u)) : u \in S\} = \{\mathsf{Greedy}(u) : u \in S\}, \qquad \Pi = \mathsf{Rep}_0(K(S), \varepsilon_0).$$

By Corollary 3.3, Rep is $(\varepsilon_0, \delta_0)$-DP. The proof uses only that $\mathcal{C}$ is a partition fixed independently of S; it never mentions the net or the graph.

Remark 5.10 (Why the public layer costs nothing in privacy). Releasing $\mathcal{N}$, G and the description of $\mathcal{C}$ reveals nothing about S, because they are the same for every S. This is where our use of navigable graphs differs from their use as nearest-neighbour indexes, which are built on the data points: had $\mathcal{N}$ or G been built from S, the argument of Corollary 3.3 would fail at its first line.

Example 5.1, continued. The department maps each of the day's $k = 1000$ case locations to its cell by running Greedy, obtaining at most 1000 keys (two cases 30 m apart usually share one), and publishes $\Pi = \mathsf{Rep}_0(K(S))$. By Remark 3.9 this takes about $k \log_2(2\kappa/\alpha)$ bits: with $\kappa = 20$ and $\alpha = 0.1$, about 8.6 bits per key, roughly 1 KB in total.

5.7 Step 5: queries

A point $u \in S$ with $d(u, q) \le r$ has its net point $\mathsf{Greedy}(u)$ within $r + \theta\rho$ of q, by Lemma 5.7. It therefore suffices to probe the cells of all net points in $B(q, \tau)$, where

$$\tau = r + \theta(\gamma)\rho.$$

Listing these net points is a search problem in the metric, and navigability makes it a local one.

Lemma 5.11 (Enumeration via navigability)**.** *Let G be γ-navigable for some $\gamma \ge 1$, let $q \in U$, $\tau > 0$ and $s \in \mathcal{N} \cap B(q, \tau)$. Then breadth-first search from s in the subgraph of G induced by $\mathcal{N} \cap B(q, 3\tau)$ visits every point of $\mathcal{N} \cap B(q, \tau)$.*

Proof. Fix $t \in \mathcal{N} \cap B(q, \tau)$, $t \ne s$. Starting from $v_0 = s$, navigability gives an edge (v_j, v_{j+1}) with $d(v_{j+1}, t) < d(v_j, t)/\gamma \le d(v_j, t)$ as long as $v_j \ne t$. Distances to t strictly decrease and $\mathcal{N}$ is finite, so the path reaches t. Every vertex on it satisfies

$$d(v_j, t) \le d(s, t) \le d(s, q) + d(q, t) \le 2\tau, \qquad \text{hence} \qquad d(v_j, q) \le d(v_j, t) + d(t, q) \le 3\tau.$$

So the path lies in the induced subgraph on $\mathcal{N} \cap B(q, 3\tau)$, and breadth-first search from s in that subgraph reaches t. □

Algorithm 6 $\mathsf{Query}(\Pi, q)$ in a distance-oracle metric

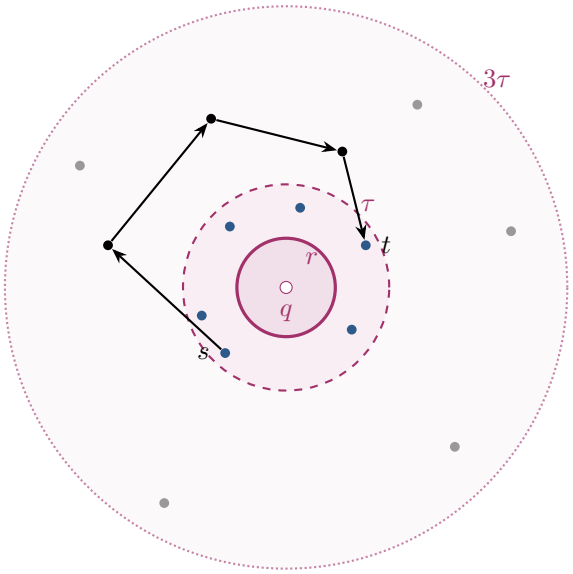
```
s ← Greedy(q)                                   ▷ d(q, s) ≤ θρ ≤ τ by Lemma 5.7
P ← the points of N ∩ B(q, τ) found by breadth-first search from s inside N ∩ B(q, 3τ)
for each z ∈ P do
    if Query_0(Π, z) = 1 then return 1
    end if
end for
return 0
```

Lemma 5.12 (The probe set is correct)**.** *Assume $2\theta(\gamma)\rho \le (c-1)r$ and let $\mathcal{P}'(q) = \{C_z : z \in \mathcal{N} \cap B(q, \tau)\}$. Then $\mathcal{P}'(q)$ satisfies conditions (a) and (b) of Theorem 3.8, and $|\mathcal{P}'(q)| \le |\mathcal{N} \cap B(q, \tau)|$.*

Proof. (a) If C_z meets $B(q, r)$ at a point v, then $d(z, q) \le d(z, v) + d(v, q) \le \theta\rho + r = \tau$ by Lemma 5.7, so $z \in B(q, \tau)$. (b) If $d(z, q) \le \tau$ and $u \in C_z$, then $d(u, q) \le d(u, z) + d(z, q) \le \theta\rho + \tau = r + 2\theta\rho \le cr$. The size bound is immediate. □

Note that the probe set may be strictly larger than the set of cells meeting $B(q, r)$: it is the price of locating cells by a radius test on their net points rather than by inspecting the cells themselves.

Example 5.1, continued. A resident's phone checks each place it visited. For a location q it runs $\mathsf{Greedy}(q)$, explores the graph within $3\tau = 2.25$ km, collects the net points within $\tau = 750$ m, and looks each one up in the downloaded 1 KB release. The department never learns who checked or where, because the release is public and queries run locally.



a navigable path from s to t may leave $B(q,\tau)$ but never leaves $B(q,3\tau)$ (Lemma 5.11)

probed: net points in $B(q,\tau)$ (blue); privacy depends only on how many are probed, not on the search

Figure 9: The three balls used by a query. Only cells whose net points lie in the dashed ball $B(q,\tau)$, $\tau = r + \theta\rho$, are probed; the search may explore out to the dotted ball $B(q,3\tau)$. Grey points are net points outside $B(q,\tau)$.

5.8 The guarantee

Theorem 5.13 (Distance-oracle construction)**.** *Let (U,d) be a finite metric given by a distance oracle, let $c > 1$ and $\gamma > 1$, and set*

$$\theta = \theta(\gamma), \qquad \rho = \frac{(c-1)r}{2\theta}, \qquad \tau = r + \theta\rho = \frac{(c+1)r}{2}, \qquad \kappa' = \sup_{q \in U} |\mathcal{N} \cap B(q,\tau)|.$$

Build $\mathcal{N}$ by Algorithm 3, a γ-navigable G on $\mathcal{N}$, and the greedy cells, and instantiate Rep_0 with the structure of Fact 2.11 at $\varepsilon_0 = \ln(2\kappa'/\alpha)$. Then $(\mathsf{Rep}, \mathsf{Query})$, with Query given by Algorithm 6, is an oblivious $(\varepsilon_0, \delta_0)$-differentially private (c, r, α)-DPDS-representation using $O(k \log(\kappa'/\alpha))$ bits beyond the public $(\mathcal{N}, G)$, with at most κ' probes per query.

Proof. By Lemma 5.7 and the choice of ρ, $2\theta\rho = (c-1)r$. By Lemma 5.11 applied with $s = \mathsf{Greedy}(q) \in B(q,\tau)$, line 2 of Algorithm 6 computes exactly $\mathcal{N} \cap B(q,\tau)$, so the algorithm probes $\mathcal{P}'(q)$ of Lemma 5.12. That lemma verifies the hypotheses of Theorem 3.8, which gives the result. Condition (b) is what rules out a wrong "yes" caused by the geometry: although the decoder probes the cell of every net point within τ of q, rather than only the cells meeting $B(q,r)$, every probed cell lies inside $B(q,cr)$, so a stored point in a probed cell is a legitimate answer. The net, the graph and the probe map are deterministic, so the representation is oblivious. □

Theorem 5.14 (Doubling metrics)**.** *If in addition (U,d) has doubling constant λ, then*

$$\kappa' \;\le\; \lambda^{\lceil \log_2(2\theta(c+1)/(c-1)) \rceil}, \qquad \varepsilon_0 \;\le\; \left\lceil \log_2 \frac{2\theta(c+1)}{c-1} \right\rceil \ln\lambda + \ln\frac{2}{\alpha}.$$

In particular, for $c = 2$ and $\gamma = 3$ (so $\theta = 5$, $\rho = r/10$, $\tau = 3r/2$), $\kappa' \le \lambda^5$ and $\varepsilon_0 \le 5\ln\lambda + \ln(2/\alpha) = O(\dim_{\mathrm{D}}) + \ln(1/\alpha)$, using $O(k(\log\lambda + \log(1/\alpha)))$ bits. The public structure

Component	Cost	Depends on S?
Net $\mathcal{N}$ (Algorithm 3)	$\lvert\mathcal{N}\rvert \le \Delta_{\text{asp}}^{O(\dim_{\text{D}})}$ points	no
Graph G, pruning (Algorithm 4)	$O(\lvert\mathcal{N}\rvert^2(\log\lvert\mathcal{N}\rvert + \Lambda))$ time, degree Λ	no
Graph G, net hierarchy (Lemma 5.5)	near-linear time in $\lvert\mathcal{N}\rvert$, same degree	no
Locating a cell (Algorithm 5)	$T = \lceil \log_\gamma \Delta_{\text{asp}} \rceil$ greedy steps	no
Release Π	$O(k\varepsilon_0) = O(k \log(\kappa'/\alpha))$ bits	yes
Query (Algorithm 6)	Greedy + search of $\mathcal{N} \cap B(q, 3\tau)$ + $\le \kappa'$ probes	reads Π

Table 3: Where the costs of the distance-oracle construction go; $\Lambda = (O(\gamma))^{\dim_{\text{D}}} \log \Delta_{\text{asp}}$ is the degree bound of Lemma 5.4. Only the release depends on the data, and it is small; the public structure is shared by all releases.

has $\lvert\mathcal{N}\rvert \le \lambda^{\lceil \log_2(2\Delta_{\text{asp}})\rceil} = \Delta_{\text{asp}}^{O(\dim_{\text{D}})}$ *points (Lemma 5.2) and at most* $\lvert\mathcal{N}\rvert\Lambda$ *edges; it is built once, independently of the data, with* $O(\lvert U\rvert \cdot \lvert\mathcal{N}\rvert)$ *distance evaluations for the net and* $O(\lvert\mathcal{N}\rvert^2(\log\lvert\mathcal{N}\rvert + \Lambda))$ *time for the pruned graph, or near-linear time in* $\lvert\mathcal{N}\rvert$ *with Lemma 5.5. With a graph of out-degree* $\Lambda = (O(\gamma))^{\dim_{\text{D}}} \log \Delta_{\text{asp}}$ *(Lemmas 5.4 and 5.5), encoding uses* $O(kT\Lambda)$ *distance evaluations plus the cost of* Rep_0*, and a query uses* $O((T + \lambda^{\lceil \log_2(6\tau/\rho)\rceil})\Lambda)$ *distance evaluations plus* κ' *probes of* Query_0*, where* $T = \lceil \log_\gamma \Delta_{\text{asp}} \rceil$*. Both are polylogarithmic in* Δ_{asp} *and exponential only in* $\dim_{\text{D}}$*.*

Proof. $\mathcal{N}$ is ρ-separated, so Lemma 2.3 gives $\kappa' \le \lambda^{\lceil \log_2(2\tau/\rho)\rceil}$, and $2\tau/\rho = (c+1)r \cdot 2\theta/((c-1)r) = 2\theta(c+1)/(c-1)$. For $c = 2$, $\theta = 5$ this is 30, and $\lceil \log_2 30 \rceil = 5$. Substitute into Theorem 5.13. For the running time, Greedy makes at most T steps of Λ distance evaluations; breadth-first search visits at most $\lvert\mathcal{N} \cap B(q, 3\tau)\rvert \le \lambda^{\lceil \log_2(6\tau/\rho)\rceil}$ vertices and evaluates the distance from q to each of their out-neighbours. □

The breadth-first search explores a ball of radius 3τ while only cells surviving the radius-τ filter are probed: enlarging the search radius costs time but never privacy, since ε_0 depends on the probe count alone. For $(\mathbb{R}^d, \ell_\infty)$, whose doubling constant is $2^{\Theta(d)}$, Theorem 5.14 gives only $\varepsilon_0 = \Theta(d)$ at $c = 2$; the secluded partitions of Section 4.1 do better there because they exploit coordinates.

Example 5.1, continued. With $r = 500\,\text{m}$ and $c = 2$: $\rho = 50\,\text{m}$, $\gamma = 3$, cell radius at most 250 m, and $\tau = 750\,\text{m}$. If the measured number of net points within 750 m of any location is about $\kappa' = 20$, then $\varepsilon_0 = \ln(2 \cdot 20/0.1) = \ln 400 \approx 6.0$ per location, the daily release is about 1 KB for $k = 1000$ locations, and the public map structure is tens of megabytes, built and downloaded once.

5.9 Limitations and variants

Bounded universe. The construction assumes a finite aspect ratio. On an unbounded space no finite-degree γ-navigable graph exists for $\gamma > 1$: if all out-neighbours of s are within distance R of s, a target t with $d(s,t) > \gamma R/(\gamma - 1)$ has no witness, since then $d(w,t) \ge d(s,t) - R > d(s,t)/\gamma$ for every out-neighbour w. The enumeration lemma, however, only uses pairs at distance at most 2τ, so for queries a *locally* navigable graph—navigable for pairs within distance 2τ—suffices, with degree independent of Δ_{asp}. The remaining global ingredient is locating a starting cell.

The worst case is loose. The bounds $\theta\rho$ on the cell radius and $\lambda^{\lceil \log_2(2\tau/\rho)\rceil}$ on the probe count are worst-case. In a small planar experiment the largest observed $d(u, \mathsf{Greedy}(u))$ was about a fifth of $\theta\rho$, and a query probed several times more cells than actually met $B(q, r)$. Since $\varepsilon_0 = \ln(2\kappa'/\alpha)$, a tighter cell-radius bound, or a probe set closer to $\mathcal{P}(q)$, lowers ε_0 directly. In applications one should report the measured κ' on the actual net rather than the doubling bound; a single hub makes λ large everywhere.

The privacy unit. Neighbouring data sets differ in one location. A person contributing m locations is protected at level $m\varepsilon_0$ by group privacy; capping the contribution per person is the usual remedy.

Public storage. The statement that neither $|U|$ nor $|\mathcal{C}|$ enters the space bound refers to the private release. In a distance-oracle metric the public structure must be stored somewhere; it is shared across all data sets, but its size grows as $\Delta_{\mathrm{asp}}^{O(\dim_{\mathrm{D}})}$.

High-dimensional metrics. The construction is efficient exactly when the doubling dimension is small. Worst-case string sets under edit distance, like the Hamming cube, do not have bounded doubling dimension, so this route gives no advantage there without further structure; we return to this in Section 7.

6 For-All Guarantees

Definition 2.7 states utility per query, and so do most non-private filters for part of their guarantee: a Bloom filter, and the filter of [15], accepts every query it must accept for every draw (no false negatives), but bounds false positives only per query, as do structures based on locality-sensitive hashing [19]. By the discussion after Definition 2.8, a guarantee for all queries at once is exactly what correctness against queries chosen after seeing the release requires. This section asks whether the per-query choice is a convenience or a necessity, and answers that it is forced: a guarantee that holds simultaneously for all queries costs a *global* quantity, the packing number of the whole space, where the per-query constructions pay only for the neighbourhood of a single query.

6.1 Three levels of guarantee

Table 4 collects the three levels of Definition 2.8 and the discussion after it, ordered by what a query may depend on.

Definition 6.1 (Accept region)**.** For a release Π, the *accept region* is $R(\Pi) = \{q \in U : \mathsf{Query}(\Pi, q) = 1\}$.

The accept region is a function of Π when Query is deterministic given Π, so that all randomness lives in Rep. Every construction in this paper has this form: the membership structure of Fact 2.11 answers queries deterministically from the released linear system. We assume it throughout this section.

Level	What is promised	Privacy cost
For-all (Definition 6.2)	With probability $\geq 1-\alpha$ over one draw of Π, the answer is correct at *every* q, including queries chosen after seeing Π.	$\ln M(U, cr) + \ln \frac{1}{\alpha} \pm O(\cdot)$: global (Propositions 6.5 and 6.7)
Oblivious per-query (Definition 2.8)	For each q, even one chosen with knowledge of all public data (but not of Π), correct with probability $\geq 1-\alpha$ over the membership structure's coins.	$\ln(2\kappa/\alpha)$: local (Theorem 3.7)
Non-oblivious per-query (Definition 2.8)	Partition drawn publicly; for each q chosen independently of the draw, correct with probability $\geq 1-\alpha$ over the draw and the coins.	$\ln(2/\alpha)$, one probe (Theorem 4.11)

Table 4: Three levels of utility guarantee. The strongest has a global cost; the two per-query guarantees have local costs.

Definition 6.2 (For-all guarantee)**.** $(\mathsf{Rep}, \mathsf{Query})$ satisfies the *for-all guarantee with error* α if for every S with $|S| \leq k$, with probability at least $1-\alpha$ over $\Pi \leftarrow \mathsf{Rep}(S)$,

$$\bigcup_{u \in S} B(u, r) \;\subseteq\; R(\Pi) \;\subseteq\; \bigcup_{u \in S} B(u, cr).$$

The left inclusion is completeness at every query, and the right inclusion is soundness at every query. The only formal difference from Definition 2.7 is where the probability sits:

$$\text{per-query:} \quad \forall q \quad \Pr_{\Pi}[\text{correct at } q] \geq 1-\alpha, \qquad\qquad \text{for-all:} \quad \Pr_{\Pi}[\forall q \text{ correct at } q] \geq 1-\alpha.$$

6.2 Intuition

Take M points $x_1, \ldots, x_M$ that are far apart, and consider the $M+1$ data sets $\emptyset, \{x_1\}, \ldots, \{x_M\}$. Every $\{x_i\}$ is a neighbour of $\emptyset$.

- Under $S = \{x_i\}$, a correct accept region must contain the small ball $B(x_i, r)$ and stay inside the larger ball $B(x_i, cr)$: it is a "blob" around x_i and nowhere else.
- Because the points are far apart, a blob around x_i cannot also be a blob around x_j, so the accept region reveals which x_i is in the data.
- Differential privacy says the release on $\{x_i\}$ looks almost the same as the release on $\emptyset$, so the release on $\emptyset$ must produce a blob around x_i with probability about $e^{-\varepsilon_0}$, for *each* i.
- These M events are disjoint, so their probabilities sum to at most 1, and $Me^{-\varepsilon_0} \lesssim 1$.

This is the standard packing argument for differential privacy [17], applied to accept regions; it is M-ary randomized response in disguise.

6.3 The lower bound

Fix $X = \{x_1, \ldots, x_M\} \subseteq U$ and, for $x \in X$, let

$$\mathcal{R}_x = \{R \subseteq U : \; B(x, r) \subseteq R \subseteq B(x, cr)\}$$

be the family of accept regions that are correct for the singleton $\{x\}$.

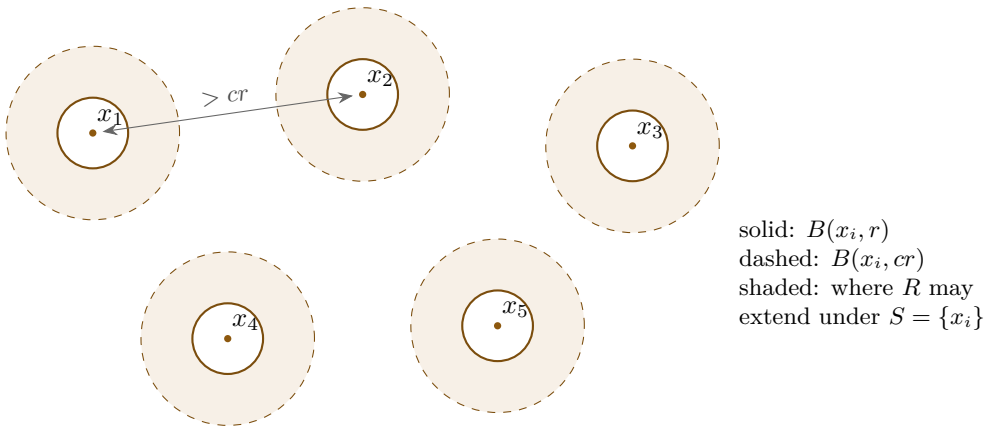


Figure 10: The packing argument. Under $S = \{x_i\}$ a correct accept region contains the solid disc and lies inside the dashed disc. If the points are pairwise more than cr apart, no single region is correct for two different singletons (Lemma 6.3), so a for-all release identifies the data point among M candidates.

Lemma 6.3 (Disjointness)**.** *If X is cr-separated, the families $\mathcal{R}_x$, $x \in X$, are pairwise disjoint, and every region in them is non-empty.*

Proof. Suppose $R \in \mathcal{R}_x \cap \mathcal{R}_{x'}$ with $x \neq x'$. Then $x' \in B(x', r) \subseteq R \subseteq B(x, cr)$, so $d(x, x') \leq cr$, contradicting separation. Every $R \in \mathcal{R}_x$ contains x. □

Using x' itself, rather than an arbitrary point $v \in B(x', r)$, is what lets cr-separation suffice; an arbitrary v would only give disjointness for $2cr$-separated sets.

Theorem 6.4 (For-all lower bound)**.** *If U contains a cr-separated set of size M and* (Rep, Query) *is $(\varepsilon_0, \delta_0)$-differentially private and satisfies the for-all guarantee with error α, then $e^{\varepsilon_0} \geq M(1 - \alpha - \delta_0)$.*

Proof. Step 1 (disjointness). By Lemma 6.3, the events $\{R(\Pi) \in \mathcal{R}_x\}$, $x \in X$, are pairwise disjoint for any distribution of Π.

Step 2 (utility on singletons). Apply the for-all guarantee to $S = \{x\}$: its two inclusions say exactly that $R(\Pi) \in \mathcal{R}_x$, so $\Pr[R(\mathsf{Rep}(\{x\})) \in \mathcal{R}_x] \geq 1 - \alpha$.

Step 3 (transfer to the empty set). The event $\{R(\Pi) \in \mathcal{R}_x\}$ is a function of Π alone, since Query is deterministic given Π. As $\emptyset \sim \{x\}$, privacy gives

$$1 - \alpha \;\leq\; \Pr\big[R(\mathsf{Rep}(\{x\})) \in \mathcal{R}_x\big] \;\leq\; e^{\varepsilon_0} \Pr\big[R(\mathsf{Rep}(\emptyset)) \in \mathcal{R}_x\big] + \delta_0,$$

that is, $\Pr[R(\mathsf{Rep}(\emptyset)) \in \mathcal{R}_x] \geq e^{-\varepsilon_0}(1 - \alpha - \delta_0)$ for every $x \in X$.

Step 4 (summation). The M events concern the same random variable $\mathsf{Rep}(\emptyset)$ and are disjoint, so

$$M\, e^{-\varepsilon_0}(1 - \alpha - \delta_0) \;\leq\; \sum_{x \in X} \Pr\big[R(\mathsf{Rep}(\emptyset)) \in \mathcal{R}_x\big] \;\leq\; 1. \qquad \square$$

Separation gives disjointness (Step 1); the for-all guarantee gives a region-level event, rather than a per-query one, that holds with probability $1 - \alpha$ (Step 2); privacy moves that event to the common reference data set $\emptyset$ (Step 3); and disjointness turns M lower bounds into one contradiction (Step 4).

6.4 A sharper bound

Step 4 bounds the total probability of the events by 1. The for-all guarantee also applies to the empty set itself, and there it is much stronger.

Proposition 6.5 (Sharpened for-all lower bound)**.** *Under the hypotheses of Theorem 6.4,*

$$e^{\varepsilon_0} \;\geq\; \frac{M(1-\alpha-\delta_0)}{\alpha}, \qquad \textit{i.e.} \qquad \varepsilon_0 \;\geq\; \ln M + \ln\frac{1}{\alpha} + \ln(1-\alpha-\delta_0).$$

Proof. For $S = \emptyset$ the right inclusion of Definition 6.2 reads $R(\Pi) \subseteq \emptyset$, so $\Pr[R(\mathsf{Rep}(\emptyset)) \neq \emptyset] \leq \alpha$. By Lemma 6.3 every region in $\bigcup_x \mathcal{R}_x$ is non-empty, so the sum in Step 4 is at most $\Pr[R(\mathsf{Rep}(\emptyset)) \neq \emptyset] \leq \alpha$ rather than 1. Combine with Step 3. □

The extra $\ln(1/\alpha)$ gives the lower bound the same shape, $\ln(\text{count}/\alpha)$, as the upper bound $\ln(2\kappa/\alpha)$ of Theorem 3.7; only the count changes, from a local to a global one.

Corollary 6.6 (The cube)**.** *Let $U = [0,m]^d$ under ℓ_∞ with $m \geq cr$. Any (ε_0,δ_0)-differentially private for-all representation has*

$$\varepsilon_0 \;\geq\; d\ln\frac{m}{cr} + \ln\frac{1}{\alpha} + \ln(1-\alpha-\delta_0) = \Omega\big(d\log(m/cr)\big) + \ln\frac{1}{\alpha} - O(1).$$

Proof. A grid in $[0,m]^d$ with spacing slightly larger than cr along each axis is cr-separated in ℓ_∞ and has at least $m/(cr)$ points per axis, so $M(U,cr) \geq (m/(cr))^d$. Apply Proposition 6.5. □

Compare the per-query constructions on the same space: $\varepsilon_0 = \ln(d+1) + \ln(2/\alpha)$ at $c = 2d+1$ (Corollary 4.5), or $\varepsilon_0 = \ln(2/\alpha)$, independent of d and m, for the non-oblivious grid (Corollary 4.13). The gap is linear in d and grows with the side length m.

6.5 A matching upper bound: expensive, not impossible

On a bounded space the for-all guarantee can be achieved, by publishing a noisy occupancy vector of a partition at resolution cr.

Proposition 6.7 (Per-cell randomized response)**.** *Let $\mathcal{C}$ be a (c,r)-admissible cell decomposition of U with $n < \infty$ cells, fixed publicly. Release, independently for each cell C, the bit*

$$b_C = \mathbf{1}[C \cap S \neq \emptyset] \;\oplus\; \xi_C, \qquad \xi_C \sim \text{Bernoulli}(p), \qquad p = \alpha/n,$$

and let $\mathsf{Query}(\Pi, q) = 1$ if and only if some cell meeting $B(q,r)$ has $b_C = 1$. Then the release is ε_0-differentially private with $\varepsilon_0 = \ln\frac{1-p}{p} \leq \ln\frac{n}{\alpha}$, and it satisfies the for-all guarantee with error α. Listing only the cells with $b_C = 1$, it uses $O((k+\alpha)\log n)$ bits in expectation.

Proof. Privacy. By Lemma 3.2, adding or removing a point changes the occupancy of at most one cell. The bits are independent, so the two output distributions differ in at most one coordinate, where they are randomized response with flip probability p, whose likelihood ratio is at most $(1-p)/p$.

For-all utility. With probability at least $1 - np = 1 - \alpha$, no bit is flipped. On that event, every $q \in B(u,r)$ with $u \in S$ is accepted, since $\mathcal{C}(u)$ meets $B(q,r)$ and is reported. Conversely an accepted q has a truly occupied cell C meeting $B(q,r)$, and its point $v \in S \cap C$ satisfies $d(v,q) \leq \operatorname{diam}(C) + r \leq cr$, so $q \in \bigcup_{u\in S} B(u,cr)$.

Space. The expected number of cells with $b_C = 1$ is at most $k + p(n-k) \leq k + \alpha$, each named by $\lceil \log_2 n \rceil$ bits; the list is a post-processing of the bit vector. □

How close is n to M? In a doubling metric they are close.

Corollary 6.8 (For-all cost in doubling metrics)**.** *Let (U, d) be bounded with doubling constant λ. The optimal privacy parameter ε^* of a pure for-all representation with error $\alpha \le 1/4$ satisfies*

$$\ln M(U, cr) + \ln \frac{1}{\alpha} - O(1) \;\le\; \varepsilon^* \;\le\; \ln M(U, cr) + \ln \frac{1}{\alpha} + \left\lceil \log_2 \frac{4c}{c-1} \right\rceil \ln \lambda.$$

Proof. The lower bound is Proposition 6.5 with $\delta_0 = 0$. For the upper bound, let $\mathcal{N}$ be a ρ-net with $\rho = (c-1)r/2$ and let $\mathcal{C}$ be its Voronoi cells, ties broken by a public order. Each point is within ρ of its cell's net point, so $\mathcal{C}$ has width at most $2\rho = (c-1)r$ and is admissible. Let $X \subseteq \mathcal{N}$ be a maximal cr-separated subset of $\mathcal{N}$; then $|X| \le M(U, cr)$, and every net point lies in $B(x, cr)$ for some $x \in X$. Each such ball contains at most $\lambda^{\lceil \log_2(2cr/\rho) \rceil}$ points of the ρ-separated set $\mathcal{N}$ (Lemma 2.3), and $2cr/\rho = 4c/(c-1)$. Hence $n = |\mathcal{N}| \le M(U, cr)\, \lambda^{\lceil \log_2(4c/(c-1)) \rceil}$; apply Proposition 6.7. □

Up to an additive $O(\dim_{\mathrm{D}} \log \frac{c}{c-1})$, the for-all guarantee therefore costs exactly $\ln(M/\alpha)$, against $\ln(\kappa/\alpha)$ per query. It is truly impossible only when $M = \infty$, that is, when U is unbounded at scale cr. The Voronoi cells in the proof are used only for existence; in a distance-oracle metric, the greedy cells of Section 5 at a slightly smaller scale serve the same purpose.

6.6 Why the per-query constructions escape

The construction of Theorem 3.7 does not satisfy the for-all guarantee, and it should not. Its release accepts each non-stored key with probability $p = 1/(e^{\varepsilon_0} + 1)$. With n cells in total, about np of them are accepted spuriously in a single draw, which for large n is almost surely at least one; so $R(\Pi)$ contains a spurious region somewhere and the right inclusion of Definition 6.2 fails. What Theorem 3.7 guarantees is that a *fixed* query probes at most κ cells, and so meets a spurious one with probability at most $\kappa p \le \alpha/2$. Per-query utility pays for the cells one query looks at; for-all utility pays for every far-apart location in the space. The first is independent of the size of U; the second grows with it.

Example 5.1, continued. With road distance, $r = 500\,\mathrm{m}$, $c = 2$ and $\alpha = 0.1$, so $cr = 1\,\mathrm{km}$: a city of about $300\,\mathrm{km}^2$ has on the order of a few hundred points pairwise more than 1 km apart by road, say $M \approx 400$, and Proposition 6.5 gives $\varepsilon_0 \ge \ln(400/0.1) + \ln 0.9 \approx 8.2$ for a for-all release, against the per-query $\varepsilon_0 \approx 6$ of Section 5. A country of about $10^6\,\mathrm{km}^2$ has $M \approx 10^6$, and the for-all bound rises to about 16, while the per-query cost is still about 6, because κ depends only on the neighbourhood of a query. The numbers are illustrative; the point is structural.

7 Optimality and Open Problems

7.1 Space optimality

Lemma 7.1 (Separated restriction)**.** *Let $W \subseteq U$ be cr-separated. The restriction of any $(\varepsilon_0, \delta_0)$-differentially private (c, r, α)-DPDS-representation to inputs $S \subseteq W$ and queries $q \in W$ is an $(\varepsilon_0, \delta_0)$-differentially private set representation of subsets of W with two-sided error α.*

Proof. Privacy is inherited. For $q \in W$: if $q \in S$ then $d(q, q) = 0 \le r$ and completeness applies; if $q \notin S$ then every $u \in S$ is a different point of W, so $d(u, q) > cr$, and soundness applies. □

Corollary 7.2 (Space optimality)**.** *If U contains a cr-separated set of size M, then every $(\varepsilon_0, \delta_0)$-differentially private (c, r, α)-DPDS-representation uses, in expectation, $\Omega(\min\{k \log(1/\alpha - 1),\ \log_2 \binom{M}{k}\})$ bits. In particular the $O(k \log(\kappa/\alpha))$ bits of Theorem 3.7 are optimal in k and in the dependence on α; the bound says nothing about the dependence on κ.*

Proof. By Lemma 7.1 the representation induces a private set representation of k-subsets of a universe of size M with error α, and the space lower bound of Patel et al. [27, Theorem 1.4] applies. That bound is proved by a compression argument and holds even when false negatives are allowed. □

Lemma 7.1 operates where every query ball meets one cell, that is, at $\kappa = 1$, and so it is structurally incapable of producing a $\log \kappa$ term: it removes precisely the geometry that makes κ large. Whether κ must appear in ε_0 (and in the space) for representations of arbitrary internal structure is the main question left open by this paper.

7.2 Is the straddle number inherent?

Conjecture 7.3. *Let $U = [0, m]^d$ under ℓ_∞, with $c = O(1)$ and $\alpha = \Omega(1)$. Then every $(\varepsilon_0, \delta_0)$-differentially private (c, r, α)-DPDS-representation satisfies $\varepsilon_0 = \Omega(d)$.*

Theorem 4.6 proves this for representations of the form of Algorithms 1 and 2. Two heuristic observations suggest that leaving the partition setting does not help.

Remark 7.4 (Heuristic evidence). To discuss cardinalities, discretize to the lattice $U_\delta = [0, m]^d \cap \delta\mathbb{Z}^d$ with $\delta < r$.

Phantom points. A "blanket" mechanism releases S together with phantom points drawn from a public distribution, so that a real point is hidden among decoys. For the presence of one real point at an arbitrary lattice location to be masked at level e^{ε_0}, phantoms must appear at each lattice point with probability about $e^{-\varepsilon_0}$; the expected number of phantoms within cr of a query is then about $e^{-\varepsilon_0} |B(q, cr) \cap U_\delta| \approx e^{-\varepsilon_0} (2cr/\delta)^d$, and keeping it below α forces $\varepsilon_0 \gtrsim d \ln(2cr/\delta)$, which is $\Omega(d)$. This is worse than a partition by the ratio of the volume of a ball to that of a cell.

Multi-table hashing. With ℓ independent locality-sensitive hash tables, one element occupies ℓ buckets, so the occupancy map has sensitivity ℓ and group privacy multiplies ε_0 by ℓ.

Proving Conjecture 7.3 seems to need a fingerprinting or reconstruction argument that couples many queries against a single draw, without assuming the global consistency that a fixed partition provides for free.

Remark 7.5 (Hamming space). The construction of [9] achieves a privacy parameter independent of the dimension at constant c in $\{0, 1\}^d$, by a decoder that is not partition-based (Section 1.4). This is not a counterexample to Conjecture 7.3, which concerns $\mathbb{R}^d$ under ℓ_∞, and the lower bound of Theorem 4.6 has not been shown to apply to the Hamming cube. Instantiating Theorem 3.7 on $(\{0, 1\}^d, \text{Hamming})$ and computing the best straddle number there is an open, well-defined calculation, and understanding what the signature-based decoder does differently would clarify which feature of partitions the lower bound really uses.

7.3 Further directions

Euclidean space. The oblivious ℓ_2 construction (Theorem 4.8) inherits a $\sqrt{d}$ loss from the diameter–inradius gap of the cube, giving $c = \Theta(d^{3/2})$ where a partition family designed natively

for ℓ_2 might achieve $\Theta(d)$ (Remark 4.9). For one-probe randomized constructions, it is open whether $c = O(d/\alpha)$ is achievable in ℓ_2, the best possible for convex lattice tilings; the spherical cubes of [22] would give it if their ball-splitting probability were bounded as for convex tiles (Remark 4.14).

Listing cells in ℓ_∞. Is there a poly(d)-time enumeration of the cells of the secluded partitions of [30] that meet a query ball? Only the 2^d vertex method of Section 4.1 is known to us.

Multiple radii. The fixed radius r makes a single scale suffice. A variant supporting a range of radii would recover the hierarchy of navigating nets [24], with one public partition per scale; the question is how to share the privacy budget across scales.

Metrics without a locality-sensitive hash family. Edit distance has no known worst-case locality-sensitive hash family, but it is exactly computable, so the distance-oracle construction of Section 5 applies to it directly, with one private structure and no composition across repetitions. Worst-case string sets do not have bounded doubling dimension, so the question is whether the support of a realistic data distribution does, and whether it admits a navigable graph with guarantees comparable to those of [20].

Randomness complexity. Canonne, Su and Vadhan [8] determined the number of random bits needed for differentially private summation, using the same rounding-scheme literature that supplies our secluded partitions. In our constructions the data-dependent randomness is only the subsampling coins of the membership structure, $\Theta(k\varepsilon_0 e^{-\varepsilon_0})$ bits, exponentially below the space of the structure; the open question concerns the public randomness. Vandermonde rows derandomize the row function and give $\delta_0 = 0$, but privacy forces the accept set A of the membership structure to satisfy $\mathbb{E}|A| \geq e^{-\varepsilon_0}(1 - \beta - \delta_0)(|\mathcal{I}| - k)$, while any low-degree hash function gives $|A| = O(k)$. Whether an information-theoretic hash function with seed length $o(|\mathcal{I}|)$ exists is open.

8 Conclusion

The privacy of a distance-sensitive set representation and its geometry can be separated. A public partition of small width carries the geometry; an off-the-shelf private membership structure carries the privacy; and the only interface between them is the straddle number, which sets the privacy parameter to $\ln(2\kappa/\alpha)$. In normed spaces this reduces the problem to the geometry of rounding, where both upper and lower bounds for partitions are available. In metrics known only through a distance oracle, a public net and a public navigable graph make the partition computable, and because both are independent of the data they cost nothing in privacy. Finally, the per-query form of the guarantee is not a convenience: a guarantee for all queries at once costs the logarithm of the packing number of the whole space, which can be paid on a bounded space and cannot on an unbounded one.

Acknowledgements

The author used Claude to perform grammar revision, spelling checks, paraphrasing and rewording. Additionally, Claude was consulted to run numerical checks on arguments for missing cases and

sign errors, and the author took suggestions from Claude for revising proofs, proof strategies, and drafting LaTeX figures. After using this tool, the author reviewed and edited the content as needed and assumes full responsibility for the content of the publication.

References

[1] M. Alaggan, S. Gambs, and A.-M. Kermarrec. BLIP: Non-interactive differentially-private similarity computation on Bloom filters. In *Stabilization, Safety, and Security of Distributed Systems (SSS 2012)*, LNCS vol. 7596, pages 202–216, 2012.

[2] A. Andoni, P. Indyk, S. Mahabadi, and S. Narayanan. Differentially private approximate near neighbor counting in high dimensions. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023.

[3] M. Aumüller, F. Boninsegna, and F. Silvestri. Differentially private high-dimensional approximate range counting, revisited. In *6th Symposium on Foundations of Responsible Computing (FORC)*, LIPIcs vol. 329, pages 15:1–15:24, 2025.

[4] P. Avi and C. Musco. Almost navigable graphs. arXiv:2607.14564, 2026.

[5] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[6] J. Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52(1–2):46–52, 1985.

[7] A. Broder and M. Mitzenmacher. Network applications of Bloom filters: A survey. *Internet Mathematics*, 1(4):485–509, 2004.

[8] C. L. Canonne, F. E. Su, and S. P. Vadhan. The randomness complexity of differential privacy. In *16th Innovations in Theoretical Computer Science Conference (ITCS)*, LIPIcs vol. 325, pages 27:1–27:21, 2025.

[9] P. Cesaretti. Differentially private distance-sensitive set representations. In *Proceedings of the Italian Conference on Theoretical Computer Science (ICTCS 2026)*, CEUR Workshop Proceedings vol. 4269, pages 222–235, 2026.

[10] J. Cheeger and B. Kleiner. Differentiating maps into L^1, and the geometry of BV functions. *Annals of Mathematics*, 171(2):1347–1385, 2010.

[11] H. Cui, Z. Li, T. T. A. Dinh, and M. Wang. MESS: Fast and private semantic search on multi-graph HNSW. arXiv:2607.28999, 2026.

[12] H. Diwan, J. Gou, C. Musco, C. Musco, and T. Suel. Navigable graphs for high-dimensional nearest neighbor search: Constructions and limits. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, LNCS vol. 3876, pages 265–284, 2006.

[14] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[15] M. Goswami, R. Pagh, F. Silvestri, and J. Sivertsen. Distance sensitive Bloom filters without false negatives. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 257–269, 2017.

[16] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 534–543, 2003.

[17] M. Hardt and K. Talwar. On the geometry of differential privacy. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 705–714, 2010.

[18] S. Har-Peled and M. Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM Journal on Computing*, 35(5):1148–1184, 2006.

[19] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

[20] P. Indyk and H. Xu. Worst-case performance of popular approximate nearest neighbor search implementations: Guarantees and limitations. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023.

[21] S. Khanna, A. Padaki, and E. Waingarten. Sparse navigable graphs for nearest neighbor search: Algorithms and hardness. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1606–1630, 2026.

[22] G. Kindler, R. O'Donnell, A. Rao, and A. Wigderson. Spherical cubes and rounding in high dimensions. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 189–198, 2008.

[23] A. Kirsch and M. Mitzenmacher. Distance-sensitive Bloom filters. In *Proceedings of the 8th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 41–50, 2006.

[24] R. Krauthgamer and J. R. Lee. Navigating nets: Simple algorithms for proximity search. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 798–807, 2004.

[25] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[26] Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020.

[27] S. Patel, G. Persiano, J. Y. Seo, and K. Yeo. Differentially private set representations. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024. arXiv:2501.16680.

[28] S. J. Subramanya, F. Devvrit, R. Kadekodi, R. Krishnaswamy, and H. V. Simhadri. DiskANN: Fast accurate billion-point nearest neighbor search on a single node. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019.

[29] J. Vander Woude, P. Dixon, A. Pavan, J. Radcliffe, and N. V. Vinodchandran. Geometry of rounding. arXiv:2211.02694, 2022.

[30] J. Vander Woude, P. Dixon, A. Pavan, J. Radcliffe, and N. V. Vinodchandran. Geometry of rounding: Near optimal bounds and a new neighborhood Sperner's lemma. arXiv:2304.04837, 2023.

A Deferred Proofs

Proof of Theorem 4.4. By Fact 4.3 there is a partition of $\mathbb{R}^d$ into unit hypercubes that is (k, ε)-secluded with $k = (f+1)^{\lceil d/f \rceil}$ and $\varepsilon = 1/(2f)$; unit hypercubes have ℓ_∞-diameter 1. By Lemma 4.2 with $s = r/\varepsilon = 2fr$, the dilated partition has $\kappa(r) \le k$ and is (c, r)-admissible for $c \ge 1 + 1/\varepsilon = 2f + 1$. Theorem 3.7 gives $\varepsilon_0 = \ln(2k/\alpha) = \lceil d/f \rceil \ln(f+1) + \ln(2/\alpha)$. □

Proof of Corollary 4.5. Take $f = d$ in Theorem 4.4. A query probes at most $d+1$ cells, listed as described in Section 4.1, and each probe of the structure of Fact 2.11 takes $O(\log(1/\delta_0))$ time. □

Proof of Theorem 4.8. Dilate the (k, ε)-secluded partition of Fact 4.3 by $s = r/\varepsilon = 2fr$, so that $B_\infty(q, r)$, and hence $B_2(q, r) \subseteq B_\infty(q, r)$, meets at most $k = (f+1)^{\lceil d/f \rceil}$ cells. Each cell is a cube of side s, whose ℓ_2-diameter is $s\sqrt{d}$; admissibility under ℓ_2 requires $s\sqrt{d} \le (c-1)r$, that is $c \ge 2f\sqrt{d} + 1$. Apply Theorem 3.7. □

Proof of Lemma 4.12. Translate the tiling by a uniformly random vector of a fundamental domain of the lattice and scale it by s. For a fixed q, the ball $B(q, r)$ fails to lie in a single tile only if q lies within distance r of the boundary of its tile. By translation invariance, the probability of this event is the fraction of the volume of the tile sT that lies within distance r of its boundary. Since sT is convex, this inner r-neighbourhood has volume at most r times the surface area As^{d-1}, so the fraction is at most Ar/s. Admissibility requires $s\Delta \le (c-1)r$. Choosing $s = Ar/\eta$ makes the failure probability at most η and gives $c = 1 + s\Delta/r = 1 + A\Delta/\eta$. □

Proof of Corollary 4.13. An ℓ_p ball of radius r has ℓ_∞-extent r: its projection on each coordinate axis is an interval of length $2r$. For a grid of side s under a uniform random shift, this interval contains a grid point of axis i with probability at most $2r/s$, so the probability that some coordinate hyperplane of the grid separates $B(q, r)$ is at most $2dr/s$. Take $s = 2dr/\eta$, so the distribution is separating with parameter η. A cube of side s has ℓ_p-diameter $sd^{1/p}$, so admissibility holds for $c \ge 1 + 2d^{1+1/p}/\eta = 1 + 4d^{1+1/p}/\alpha$. Apply Theorem 4.11 with $\eta = \alpha/2$ and the structure of Fact 2.11, for which $\beta = p \le \alpha/2$. Locating the cell of q is coordinatewise rounding, in $O(d)$ time. □