

Differentially Private High-Dimensional Approximate Range Counting, Revisited

Martin Aumüller 

IT University of Copenhagen, Denmark

Fabrizio Boninsegna¹ 

Department of Information Engineering, University of Padova, Italy

Francesco Silvestri 

Department of Information Engineering, University of Padova, Italy

Abstract

Locality Sensitive Filters are known for offering a quasi-linear space data structure with rigorous guarantees for the Approximate Near Neighbor search (ANN) problem. Building on Locality Sensitive Filters, we derive a simple data structure for the Approximate Near Neighbor Counting (ANNC) problem under differential privacy (DP). Moreover, we provide a simple analysis leveraging a connection with concomitant statistics and extreme value theory. Our approach produces a simple data structure with a tunable parameter that regulates a trade-off between space-time and utility. Through this trade-off, our data structure achieves the same performance as the recent findings of Andoni et al. (NeurIPS 2023) while offering better utility at the cost of higher space and query time. In addition, we provide a more efficient algorithm under pure ϵ -DP and elucidate the connection between ANN and differentially private ANNC. As a side result, the paper provides a more compact description and analysis of Locality Sensitive Filters for Fair Near Neighbor Search, improving a previous result in Aumüller et al. (TODS 2022).

2012 ACM Subject Classification Theory of computation → Sorting and searching; Mathematics of computing → Probabilistic algorithms; Security and privacy

Keywords and phrases Differential Privacy, Locality Sensitive Filters, Approximate Range Counting, Concomitant Statistics

Digital Object Identifier 10.4230/LIPIcs.FORC.2025.15

Funding This work was supported in part by the MUR PRIN 20174LF3T8 AHeAD project, by MUR PNRR CN00000013 National Center for HPC, Big Data and Quantum Computing, and by Marsden Fund (MFP-UOA2226).

Acknowledgements The authors would like to thank Ninh Pham and Rasmus Pagh for useful discussions.

1 Introduction

Since the emergence of deep learning-based text and image embeddings, such as CLIP [28], the management of collections of high-dimensional vectors has become a critical challenge. Efficiently handling these collections and supporting complex query operations is essential for various applications, including social networks [8] and recommendation systems [29]. The required query operations in applications are often similarity search primitives, which have been widely studied in the literature [30, 33]. In particular, the *r-Near Neighbor Search* (*r*-NNS) or *Count* (*r*-NNC) problems are fundamental primitives: given a set $\mathcal{S} \subset \mathbb{R}^d$ of n vectors of d dimensions and a radius $r > 0$, construct a data structure that, for any query $\mathbf{q} \in \mathbb{R}^d$, returns a point in $\mathcal{S}$ with distance at most r from $\mathbf{q}$ if such a point exists, or counts the

¹ Corresponding author, work partially done while visiting IT University of Copenhagen.



© Martin Aumüller, Fabrizio Boninsegna, and Francesco Silvestri;
licensed under Creative Commons License CC-BY 4.0

6th Symposium on Foundations of Responsible Computing (FORC 2025).

Editor: Mark Bun; Article No. 15; pp. 15:1–15:24



Leibniz International Proceedings in Informatics

LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

number of such points. Unfortunately, these problems suffer from the *curse of dimensionality*, which refers to the phenomenon that any exact data structure with polynomial size requires a query time exponential in the dimension of the input space. This is supported by popular algorithmic hardness conjectures [2, 34]. To address this issue, approximate approaches have been proposed: for a given approximation factor $c > 1$, we consider the (c, r) -Approximate Near Neighbor Search (c, r) -ANNS problem and the (c, r) -Approximate Near Neighbor Count (c, r) -ANNC problem. These relax the original problem constraints in such a way that the data structure may use points at distance at most cr to answer a query. For a search operation, this means that a point at distance at most cr from $\mathbf{q}$ can be returned; for a count operation, points at distance between r and cr may be counted as near neighbors. Locality Sensitive Hashing (LSH) [22] and Locality Sensitive Filters (LSF) [5, 9] are the most common approaches for solving approximate near neighbor problems with theoretical guarantees.

In another line of research, there is an increasing demand for developing solutions for data analysis tasks that preserve the privacy of sensitive personal information in the input data. This was, for example, highlighted in an invited talk at PODS in 2019 by Dwork [12], who discussed the application of *differential privacy* (DP) [13] in the US census. Differential privacy is a widely adopted notion that can provide rigorous guarantees of privacy: intuitively, DP guarantees that the removal or addition of a single input entry cannot significantly affect the final result of an analysis task. The main idea of the DP approach is to inject well-calibrated noise into the data (structure) that protects privacy without significantly affecting the accuracy of the analysis task. Counting queries, which can be seen as a more general form of the near neighbor counting problem studied in this paper, are a well-studied area in DP (see [19, 32]). However, the error in the final count must be polynomial in n and the dimension of the input space to guarantee privacy [20, 24]. To reduce the error, the requirements should be relaxed to provide any count within a fuzzy range around the query [21]. Although this is different from the curse of dimensionality above, achieving an efficient solution similarly necessitates the use of approximate counts. Thus, it is *natural* to study approximate counting problems for the high-dimensional ANN problem when considering them in a DP setting.

In this paper, we focus on ANN problems for the inner product on the d -dimensional unit sphere $\mathbb{S}^{d-1} := \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 = 1\}$, which is often called *cosine* or *angular similarity*. In these settings, the goal is to *find* or *count* points with large inner products. More specifically, we study the (α, β) -ANN count and search problems with $0 \leq \beta < \alpha < 1$. Let $B(\mathbf{q}, \alpha) := \{\mathbf{x} \in \mathbb{S}^{d-1} \mid \langle \mathbf{x}, \mathbf{q} \rangle \geq \alpha\}$ be the set of unit vectors that have an inner product of at least α with $\mathbf{q} \in \mathbb{S}^{d-1}$. The counting variant asks for a query $\mathbf{q}$ to count all points in a dataset $\mathcal{S}$ with inner product at least α but tolerates points with inner product at least β . That means that the resulting estimate E should satisfy $|\mathcal{S} \cap B(\mathbf{q}, \alpha)| \leq E \leq |\mathcal{S} \cap B(\mathbf{q}, \beta)|$. This is a common notation in inner product search, and intuitively α and β are equivalent to r and r/c in (c, r) -ANNC. The first result for differentially private (α, β) -ANNC in high dimensions has been recently provided by Andoni et al. [4], where the authors use a linear space tree-based data structure based on the concept of LSF.²

In this work, we explore the design space of locality sensitive filtering-based solutions to ANN and ANNC problems. We show that by revisiting the LSF framework for ANN it is possible to derive a simpler and more compact solution for ANN. Building on this result,

² As observed by [4] as well, a solution on the unit sphere leads to a solution for the whole Euclidean space thanks to embedding methods. We will defer all discussion of this embedding and its applicability to Appendix C.

■ **Algorithm 1** DPTop-1 Data Structure for DP-ANNC.

<pre> 1: procedure CONSTRUCTION($\mathcal{S} \subseteq \mathbb{S}^{d-1}, \alpha, \beta, \varepsilon, \delta$) 2: $\rho \leftarrow \frac{(1-\alpha^2)(1-\beta^2)}{(1-\alpha\beta)^2}; m \leftarrow \lceil n \frac{\rho}{1-\alpha^2} \rceil$ 3: $\mathcal{A}^m = (\mathbf{a}_1, \dots, \mathbf{a}_m)$ with $\mathbf{a}_i \sim \mathcal{N}(0, 1)^d$ 4: $T[1, \dots, m] \leftarrow (0, \dots, 0)$ 5: for $\mathbf{x} \in \mathcal{S}$ do 6: $i \leftarrow \arg \max_{i \in [m]} \langle \mathbf{a}_i, \mathbf{x} \rangle$ 7: $T[i] \leftarrow T[i] + 1$ 8: $\eta \leftarrow \alpha \sqrt{2 \log m} - \sqrt{2(1-\alpha^2) \log \log m}$ 9: $\tilde{T} \leftarrow \text{make_private}(T, \varepsilon, \delta)$ 10: return $\mathcal{D} = (T, \mathcal{A}^m, m, \eta)$ </pre>	<pre> 1: procedure QUERY($\mathbf{q}$) 2: $B \leftarrow \{i \in [m] : \langle \mathbf{a}_i, \mathbf{q} \rangle \geq \eta\}$ 3: $\widetilde{\text{ans}} \leftarrow 0$ 4: for $i \in B$ do 5: $\widetilde{\text{ans}} \leftarrow \widetilde{\text{ans}} + \tilde{T}[i]$ 6: return $\widetilde{\text{ans}}$ </pre>
---	--

we derive a novel solution for ANNC under differential privacy that extends the range of applicability of the state of the art [4] by removing some limitations on parameter ranges and differential privacy assumptions. In particular, we provide strong guarantees in the regime of *pure DP* (in contrast to approximate DP), and show that balancing the noise term of DP with the approximation error of LSF is not the only design choice: in fact, spending more space and query time results in more accurate solutions. The following section will provide more details on the technical contribution.

1.1 Our Contribution

Revisiting LSF for ANN

Our work is based on a construction for (α, β) -ANN first proposed by Aumüller et al. [6] in the context of algorithmic fairness. We provide a more compact description and analysis of LSF for (α, β) -ANN, and obtain a data structure with a lower pre-processing time of $O(dn^{1+o(1)})$ (from $O(dn^{1+\rho+o(1)})$ where $0 < \rho := \rho(\alpha, \beta) \leq 1$ is the *strength of the filter*). Moreover, assuming that some random variables follow a limiting distribution (Theorem 5), we get a more compact and simpler proof than [6] leveraging an elegant connection with concomitant statistics and extreme value theory, and the parameters used in our solution naturally follow from this theory. In Section 5.1, we demonstrate that an alternative construction procedure, **CloseTop-1**, remains effective without assuming any limiting distributions, matching the convergent properties.

From ANN to DP-ANNC

We then present a solution for (α, β) -ANNC under differential privacy. More specifically, we provide a general methodology that allows to translate a variant of a list-of-points data structure [5] for ANN into a data structure for DP-ANNC. Intuitively, a list-of-points is a data structure where input points are organized in a collection of lists and a query consists of a scan of some of these lists: this is the case, for instance, of methods based on LSH or LSF. This approach offers a way to develop the data structure in two steps by describing a data structure satisfying certain characteristics for ANN, and then applying a suitable DP mechanism on top of it.

When the data structure is built on top of the previous result, we get the **DPTop-1** data structure presented in Algorithm 1, which we will now describe in words. Given a dataset $\mathcal{S} \subseteq \mathbb{S}^{d-1}$ consisting of n points, two similarity thresholds $0 \leq \beta < \alpha < 1$, and privacy parameters $\varepsilon > 0$ and $\delta \in [0, 1)$, the data structure samples $m = n^{O(1)}$ Gaussian vectors. We associate with each such vector a counter, initialized with 0; each point in $\mathcal{S}$ increments the counter of the vector that maximizes the inner product. Then, the counts

are made differentially private by a suitable DP mechanism `make_private`, for example the Truncated Laplace mechanism [17] or the ALP mechanism [7]. Depending on the mechanism used, different privacy guarantees can be provided. Since each point increments exactly one counter, the absence or presence of a data point affects only a small part of the data structure. As a result, the sensitivity of the data structure is low and only a small amount of noise has to be added. The m vectors with their noisy counts form the data structure that can be released publicly. For any query $\mathbf{q} \in \mathbb{S}^{d-1}$, the estimate of the number of near neighbors with inner product at least α is the sum of counters associated to vectors with inner product similarity to $\mathbf{q}$ greater than $\eta(\alpha) = \alpha\sqrt{2\log m} - \sqrt{2(1-\alpha^2)\log\log m}$. This choice is guided by the theory of concomitant statistics and extreme value theory of Gaussian random variables [10], which we will formally introduce in Section 2.3. As we will detail in Section 3, in the asymptotic regime – hence for $n \rightarrow \infty$ – `DPTop-1` offers a simple and elegant solution for the (α, β) -ANNC problem under differential privacy. The following theorem provides the guarantees of `DPTop-1` when using the Truncated Laplace mechanism [17] as a privacy mechanism (we refer to Theorem 9 for the exact statements regarding the ANN data structure, and Theorem 13 for the exact statements of the DP-ANNC implementation).

► **Theorem 1.** *Consider the asymptotic regime, $\varepsilon > 0$, $\delta \in (0, \frac{1}{2})$, $0 \leq \beta < \alpha < 1$, and $\alpha - \beta = \Omega(\sqrt{\frac{\log\log n}{\log n}})$. Let $\mathcal{S} = \{x_i\}_{i=1,\dots,n} \subseteq \mathbb{S}^{d-1}$ and let $\mathbf{q} \in \mathbb{S}^{d-1}$. Then `DPTop-1` (Algorithm 1) with Truncated Laplace mechanism satisfies (ε, δ) -DP, and with probability at least $2/3$, the query returns $\widehat{\text{ans}}$ such that*

$$(1 - o(1))|\mathcal{S} \cap B(\mathbf{q}, \alpha)| - O\left(\frac{\log(1/\delta)}{\varepsilon} n^{\rho+o(1)}\right) \leq \widehat{\text{ans}} \leq |\mathcal{S} \cap B(\mathbf{q}, \beta)| + O\left(\frac{\log(1/\delta)}{\varepsilon} n^{\rho+o(1)}\right).$$

The data structure has pre-processing time $O(d \cdot n^{1+\frac{\rho}{1-\alpha^2}})$, expected query time and space $O(d \cdot n^{\frac{\rho}{1-\alpha^2}})$.

This simple algorithm matches the accuracy of the solution found by Andoni et al. [4] and results in a straightforward space partitioning of $\mathbb{S}^{d-1}$, one of the main goals of [4]. Furthermore, our approach provides a solution that works for almost all similarity thresholds on the unit sphere, while [4] supports a single distance threshold and relies on embedding techniques and scaling for all other distance thresholds.

While Algorithm 1 is potentially already practical, both the space and running time requirements are worse than the solution presented in [4] due to the large number of filters m . We suggest two improvements to the algorithm that do not affect the compactness of the algorithm and the proof. We first observe that the theoretical result in Theorem 1 works only for $n \rightarrow \infty$: we drop this limitation in Section 5.1 thanks to a small but novel variation in the construction procedure. We then observe in Section 5.2 how to achieve almost linear pre-processing time, linear space, and $d \cdot n^{\rho+o(1)}$ expected query time, which is optimal due to a space-time tradeoffs lower bound [5]³, by concatenating $\text{polylog}(n)$ data structures, using a technique called *tensorization* [9, 6]. We call this final version `TensorCloseTop-1`. Table 1 summarizes the guarantees of all algorithms described in this paragraph and compares them with the state of the art approach [4].

³ It is sufficient to set $\rho_u = 0$ in Theorem 3.3 in the reference paper to get the lower bound for the running time

■ **Table 1 Results for DP-ANNC**, $\sigma = \frac{(1-\alpha^2)(1-\beta^2)}{(1-\alpha\beta)^2 + (\alpha-\beta)^2}$, $\rho = \frac{(1-\alpha^2)(1-\beta^2)}{(1-\alpha\beta)^2}$, and $\sigma < \rho < 2\sigma$ for any $0 \leq \beta < \alpha < 1$. The bound (*) holds only in expectation. Time and space bounds omit factor d .

Mechanism	Privacy	Additive Error	Preprocessing Time	Expected Query Time	Space
DPTop-1 w/ Truncated Laplace	(ε, δ)	$O\left(\frac{\log(1/\delta)}{\varepsilon} n^{\rho+o(1)}\right)$	$O\left(n^{1+\frac{\rho}{1-\alpha^2}}\right)$	$O\left(n^{\frac{\rho}{1-\alpha^2}}\right)$	$O\left(n^{\frac{\rho}{1-\alpha^2}}\right)$
Andoni et al. [4], and TensorCloseTop-1 w/ Truncated Laplace	(ε, δ)	$O\left(\frac{\log(1/\delta)}{\varepsilon} n^{\rho+o(1)}\right)$	$n^{1+o(1)}$	$n^{\rho+o(1)}$	$O(n)$
Andoni et al. [4]	$(\varepsilon, 0)$	$O\left(\frac{1}{\varepsilon} n^{\rho+o(1)}\right)$	$n^{1+o(1)}(*)$	$n^{\rho+o(1)}$	$O(n)^{(*)}$
TensorCloseTop-1 w/ Max Projection	$(\varepsilon, 0)$	$O\left(\frac{1}{\varepsilon} n^{\rho+o(1)}\right)$	$n^{1+o(1)}$	$n^{\rho+o(1)}$	$O(n)$
Unbalanced TensorCloseTop-1 with Laplace	$(\varepsilon, 0)$	$O\left(\frac{1}{\varepsilon} n^{\sigma+o(1)}\right)$	$O\left(n^{\frac{\sigma}{1-\alpha^2}}\right)$	$n^{2\sigma+o(1)}$	$O\left(n^{\frac{\sigma}{1-\alpha^2}}\right)$

Balanced and Unbalanced Data Structures

As can be seen from the theorem statement, the provided estimate comes with an additive error of $O\left(\log(1/\delta)n^{\rho+o(1)}/\varepsilon\right)$. This term includes two fundamentally different error sources. First, we might include “far away points”, i.e., points with inner product below β , in the count through the ANN data structure. This error does not depend on the choice of the privacy mechanism. The second source of error, which is due to privacy, arises from summing over $n^{\rho+o(1)}$ noisy counters, as a query is expected to search through $n^{\rho+o(1)}$ buckets on average. DPTop-1 and the data structure of Andoni et al. [4] *balance* these two errors so that both are upper bounded by $n^{\rho+o(1)}$; however, as we will show in Sections 3 and 4 this is not necessarily an optimal trade-off. By using an *unbalanced* data structure – to be discussed in detail in Section 3.1 – with Laplace noise, we can achieve a more accurate result at the cost of larger running time and more space. In fact, for any parameter, our unbalanced data structure solves DP-ANNC with an additive error $n^{\sigma+o(1)}$ for $\sigma < \rho$. The main insight is that the sum of $n^{\rho+o(1)}$ noisy, unbiased, and uncorrelated counters, provided by the Laplace mechanism, scales with $n^{\rho/2+o(1)}$ by concentration arguments. This makes it suboptimal to provide the same upper bound for both sources of error.

Comparison to Andoni et al. (NeurIPS 2023)

For the convenience of the reader, we now provide a quick description of the solution in [4], and highlight differences to the present work. Let η_u , η_q , T and K be suitable parameters that depend on α and β . The data structure consists of a tree with degree T and height K ; each internal node v is associated to a d -dimensional random Gaussian vector $\mathbf{g}_v$ and

to a subset of the input set $\mathcal{S}$. At the beginning, the root contains the entire input set $\mathcal{S}$. Then, for each internal node with a top-down approach, we partition the assigned vectors into T groups: each input vector $\mathbf{x}$ is assigned to the node v with smallest index such that the inner product is $\langle \mathbf{x}, \mathbf{g}_v \rangle \geq \eta_u$. (If no such index exists, the point is not stored in the data structure.) Once the input points have been processed, we replace the list in each leaf with its noisy size by adding Truncated Laplace noise [17] (if the final value is below a given threshold, we replace the value with 0). Given a query $\mathbf{q}$, we collect the counts of all leaves v for which $\langle \mathbf{q}, \mathbf{g}_{v'} \rangle \geq \eta_q$ for all nodes v' on the unique path from the root to v , and return as a final estimate the sum of these counts. Using this tree data structure circumvents the evaluation problem of a too large number of filters mentioned for our variant above.

As the goal of their paper is to address Euclidean distance, the range of the α and β parameters is limited; their analysis works only for an α value that corresponds to Euclidean distance $\Theta(\log^{-1/8} n)$ and all other distances are only supported through embedding and scaling, which adds an additional distortion to the distance values. In contrast, our solution allows for a wider range of these parameters, increasing the applicability for inner product similarity on the sphere and still gets a data structure that holds for Euclidean distance. Furthermore, by removing the tree structure, we are able to design an algorithm with little data dependencies that is likely to exploit hardware accelerators (e.g., Nvidia Tensor Core, Intel AMX) for neural networks that are optimized for batches of inner products.

1.2 Previous work

Near Neighbor Search

Locality Sensitive Hashing (LSH) [22] is one of the most used approaches for solving ANN with rigorous guarantees. However, it suffers of large space requirements. Indeed, LSH requires $O(nd + n^{1+\rho})$ memory words, where ρ is a parameter describing the “power” of the used LSH (e.g., $\rho = O(1/c^2)$ for Euclidean distance [3]): indeed the data structure requires to create n^ρ hash tables, each storing all n points. Both Panigrahy [25] and Kapralov [23] provided linear space solutions using variants of LSH. An interesting technique to achieve smooth space-time trade-offs is given by *Locality Sensitive Filters* (LSF) [9, 5]. In the context of this work, the interesting space-time trade-off to focus on is the linear space regime [6]. Besides offering optimal space, this regime has many additional interesting properties for downstream applications. For example, very recently, Andoni et al. [4] showed their application in the context of differentially private range counting in high dimensional data. As mentioned above, a linear space data structure only involves at most one time a point of the dataset, so the absence or presence of a data point only affects a small part of the data structure; in comparison, with traditional LSH-based approaches a single point is stored in many different tables in the data structure.

Differentially Private Counting Queries

Counting queries require, except for a few classes of queries, a polynomial error in n and in the space dimension to guarantee privacy [20, 24]. This incentivized Huang and Yao [21] to relax the condition, allowing for the release of any count within a fuzzy range of the query. For ball queries in $\mathbb{R}^d$, this is essentially the problem to release any count between $|\mathcal{S} \cap B_D(\mathbf{q}, r)|$ and $|\mathcal{S} \cap B_D(\mathbf{q}, cr)|$, that we will identify as the (c, r) -Approximate Nearest Neighbor Count (ANNC) problem. One of the main results in [21] is that there exists a differential private solution of the problem with poly-logarithmic error in n at the price of an exponential dependence in the dimension d . A solution for the high dimensional case

was proposed in [4], where Andoni et al. proposed a linear space data structure for the (c, r) -ANN, to solve the differential private (c, r) -ANNC. The authors developed a Locality Sensitive Filtering data structure with $\rho = \frac{4c^2}{(c^2+1)^2}$, for the differential private (c, r) -ANNC in the Euclidean space with additive error $O(n^{\rho+o(1)})$ and multiplicative error $1 - o(1)$, getting rid of the dependence on the dimension. The proposed data structure is based on a more general theory for data structures with space-time trade-offs [5], making the analysis more involved. In this paper, we will show that our data structure offers the same guarantees with a more streamlined analysis.

2 Preliminaries

2.1 Notation

We let $[m]$ be the set of integers $\{1, \dots, m\}$. We denote with $\mathbf{q} \in \mathbb{S}^{d-1}$ a query point, and with $\mathbf{x}_\varrho$ a point of the dataset such that $\langle \mathbf{x}_\varrho, \mathbf{q} \rangle = \varrho$. We set $\mathbf{a}_\mathbf{x}$ as the vector associated to $\mathbf{x}$, and define $X_\mathbf{x} := \langle \mathbf{a}_\mathbf{x}, \mathbf{x} \rangle$, and $Q_\mathbf{x} = \langle \mathbf{a}_\mathbf{x}, \mathbf{q} \rangle$ as the concomitant – to be defined in Section 2.3 – of $X_\mathbf{x}$. If $X_\mathbf{x} = \max_{i \in [m]} \langle \mathbf{a}_i, \mathbf{x} \rangle$ then it is denoted as $X_{\mathbf{x},(m)}$ and so the concomitant as $Q_{\mathbf{x},[m]}$. The threshold for the query filter is $\eta = \alpha\sqrt{2\log m} - \sqrt{2(1-\alpha^2)\log\log m}$. A ball in the hyper-sphere under inner product similarity centered in $\mathbf{q}$ is denoted as $B(\mathbf{q}, \alpha) := \{\mathbf{x} \in \mathbb{S}^{d-1} : \langle \mathbf{q}, \mathbf{x} \rangle \geq \alpha\}$. We call a point $\mathbf{x}$ in $\mathcal{S}$ *close* to $\mathbf{q}$ if $\langle \mathbf{q}, \mathbf{x} \rangle \geq \alpha$, and *far* if $\langle \mathbf{q}, \mathbf{x} \rangle < \beta$. We consider n to be the number of points in the dataset $\mathcal{S} \subset \mathbb{S}^{d-1}$. We denote the Gaussian distribution of mean μ and variance σ^2 as $\mathcal{N}(\mu, \sigma^2)$.

2.2 Problem Definition

► **Definition 2** $((\alpha, \beta)$ -ANN). *Consider a set $\mathcal{S} \subseteq \mathbb{S}^{d-1}$ of n points. The Approximate Nearest Neighbor Search ANN problem asks to construct a data structure for $\mathcal{S}$ that for a given query $\mathbf{q} \in \mathbb{S}^{d-1}$, such that $B(\mathbf{q}, \alpha)$ contains a point in $\mathcal{S}$, returns a point in $\mathcal{S} \cap B(\mathbf{q}, \beta)$.*

We will study a data structure that solves this problem with asymptotically high probability, hence at least $1 - o(1)$. The inner product similarity is related to the Euclidean distance, as $\|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{2(1 - \langle \mathbf{x}, \mathbf{y} \rangle)}$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{S}^{d-1}$. Therefore, for $\alpha = 1 - \frac{r^2}{2}$ and $\beta = 1 - \frac{(cr)^2}{2}$, a (c, r) -ANN in $(\mathbb{S}^{d-1}, \|\cdot\|_2)$ is equivalent to the (α, β) -ANN defined above.

► **Definition 3** $((\alpha, \beta)$ -ANNC). *Consider a set $\mathcal{S} \subseteq \mathbb{S}^{d-1}$ of n points. The Approximate Near Neighbor Counting (ANNC) problem asks to construct a data structure for $\mathcal{S}$ that, for a given query $\mathbf{q} \in \mathbb{S}^{d-1}$, returns a number between $|\mathcal{S} \cap B(\mathbf{q}, \alpha)|$ and $|\mathcal{S} \cap B(\mathbf{q}, \beta)|$.*

This problem is the counting equivalent of the well-studied *spherical range reporting problem* (see for example [1]) that asks to *enumerate* all points at a certain distance from $\mathbf{q}$.

2.3 Concomitant Order Statistics

The theory of concomitant order statistics offers a very elegant and intuitive tool for random projections in $\mathbb{S}^{d-1}$, as highlighted in [15, 26, 27]. Let $(X_1, Y_1), \dots, (X_m, Y_m)$ be m random samples from a bivariate distribution. We order the values according to X such that $X_{(1)} \leq \dots \leq X_{(i)} \leq \dots \leq X_{(m)}$. The Y -variate associated with $X_{(r)}$ is denoted as $Y_{[r]}$ and it is called the *concomitant* of the r -th order statistic.

Relation With Random Projections

Let $\mathbf{x}, \mathbf{y} \in \mathbb{S}^{d-1}$ such that $\langle \mathbf{x}, \mathbf{y} \rangle = \varrho$ and $\mathbf{a} \sim \mathcal{N}(0, 1)^d$. Consider the random variables $X = \langle \mathbf{x}, \mathbf{a} \rangle$ and $Y = \langle \mathbf{y}, \mathbf{a} \rangle$, then $(X, Y) \sim \mathcal{N}(0, 0, 1, 1, \varrho)$, which is a standard bivariate normal distribution with correlation coefficient ϱ ⁴. The relation between concomitant and order statistics for the normal bivariate distribution is given by the following lemma.

► **Lemma 4** ([10]). *Given m samples $\{(X_i, Y_i)\}_{i=1, \dots, m}$ from the standard bivariate normal distribution $\mathcal{N}(0, 0, 1, 1, \varrho)$, for any $r \in \{1, \dots, m\}$ we have that $Y_{[r]} = \varrho X_{(r)} + Z_r$, where Z_r is a random variable distributed as $\mathcal{N}(0, 1 - \varrho^2)$ and independent of $X_{(r)}$.*

A standard result of concomitant order statistics states that $Y_{[r]} - \mathbb{E}[Y_{[r]}]$ weakly converges to a Gaussian distribution $\mathcal{N}(0, 1 - \varrho^2)$ [10]. Thus, defining $F_{Y_{(m)}}$ as the probability density function of $Y_{[m]}$, we have that $\lim_{m \rightarrow \infty} F_{Y_{[m]}} = \mathcal{N}(\varrho \mathbb{E}[X_{(m)}], 1 - \varrho^2)$ [16]. By adding the fact that $\mathbb{E}[X_{(m)}] = \sqrt{2 \log m} - o(1)$ [18] we get the following theorem.

► **Theorem 5** ([10, 18]). *Let $\{(X_i, Y_i)\}_{i=1, \dots, m}$ be m i.i.d. samples from $\mathcal{N}(0, 0, 1, 1, \varrho)$. Then $Y_{[m]}$ weakly converges to $\mathcal{N}(\varrho \sqrt{2 \log m}, 1 - \varrho^2)$.*

This asymptotic result serves as the basis of the intuition for our data structure: if we associate to each point in $\mathbf{x} \in \mathcal{S}$ the closest Gaussian vector $\mathbf{a}_{\mathbf{x}} = \arg \max_{\mathbf{a} \in \{\mathbf{a}_1, \dots, \mathbf{a}_m\}} \langle \mathbf{a}, \mathbf{x} \rangle$, then a query $\mathbf{q} \in \mathbb{S}^{d-1}$, such that $\langle \mathbf{q}, \mathbf{x} \rangle = \varrho$, will find $\mathbf{x}$ associated to a Gaussian vector with inner product similarity $\langle \mathbf{q}, \mathbf{a}_{\mathbf{x}} \rangle \sim \varrho \sqrt{2 \log m}$.

2.4 Differential Privacy

Differential Privacy (DP) is a definition on indistinguishability for the outputs of protocols applied to *neighboring* datasets. Two datasets are neighbor $\mathcal{S} \sim \mathcal{S}'$ by *addition/removal* if they differ by the addition or a removal of one point, instead they are neighbor by *substitution* if $|\mathcal{S}| = |\mathcal{S}'|$ and they differ in one point.

► **Definition 6** (Approximate Differential Privacy [14]). *For $\varepsilon > 0$ and $\delta \in [0, 1)$, we say that a randomized algorithm $\mathcal{M}$ is (ε, δ) -differentially private if for any two neighboring datasets $\mathcal{S} \sim \mathcal{S}'$, and any possible outcome of the algorithm $Y \subseteq \text{range}(\mathcal{M})$, we have $\Pr[\mathcal{M}(\mathcal{S}) \in Y] \leq e^\varepsilon \Pr[\mathcal{M}(\mathcal{S}') \in Y] + \delta$.*

We are mainly interested in *histogram queries* $f : \mathcal{X}^n \rightarrow \mathbb{N}^{|\mathcal{X}|}$ [13], where $\mathcal{X}$ is the data universe and n is the size of the data set. The most common way to privatize $f(\mathcal{S})$ is to obfuscate the true values by adding noise scaled on the *sensitivity* of the query $\Delta_f = \max_{\mathcal{S} \sim \mathcal{S}'} \|f(\mathcal{S}) - f(\mathcal{S}')\|_1$ [13]. In our context, each data point contributes to exactly one counter. For the addition/removal neighboring relationship, $\Delta_f = 1$; for substitution, $\Delta_f = 2$. We consider three different DP mechanisms when privatizing counters, for example, in the function `make_private` in Algorithm 1: Truncated Laplace Mechanism [17], Laplace Mechanism [17], and Max Projection [7]. More details can be found in Appendix A.1.

3 Top-1 Data Structure for ANN

Algorithm 2 describes the Top-1 data structure, which is the variant of Algorithm 1 targeting the (α, β) -ANN problem. Let $\mathcal{A}^m = (\mathbf{a}_1, \dots, \mathbf{a}_m)$ be a set of m random vectors from $\mathcal{N}(0, 1)^d$. The data structure consists of a hash table that stores the input vectors assigned to each

⁴ The general notation for a bivariate Gaussian distribution is $\mathcal{N}(\mu_X, \mu_Y, \text{Var}[X], \text{Var}[Y], \text{Cov}[X, Y])$, while $\text{Cov}[X, Y] = \sum_{i=1}^d x_i y_i = \varrho$.

■ **Algorithm 2** Top-1 Data Structure.

1: procedure CONSTRUCTION($\mathcal{S} \subset \mathbb{S}^{d-1}, \alpha, \beta, \theta$)	1: procedure SEARCH($\mathbf{q}$)
2: $m \leftarrow \lceil n^{\frac{\theta}{1-\alpha^2}} \rceil$	2: return $\{i \in [m] : \langle \mathbf{a}_i, \mathbf{q} \rangle \geq \eta\}$
3: $\mathcal{A}^m = (\mathbf{a}_1, \dots, \mathbf{a}_m)$ with $\mathbf{a}_i \sim \mathcal{N}(0, 1)^d$	3: procedure QUERY($\mathbf{q}$)
4: $H \leftarrow$ empty hash table	4: $B \leftarrow \mathcal{D}.\text{search}(\mathbf{q}, \eta)$
5: for $\mathbf{x} \in \mathcal{S}$ do	5: for $i \in B$ do
6: $i \leftarrow \arg \max_{i \in [m]} \langle \mathbf{a}_i, \mathbf{x} \rangle$	6: for $\mathbf{x} \in H[i]$ do
7: $H.\text{insert}(i, \mathbf{x})$	7: if $\langle \mathbf{q}, \mathbf{x} \rangle \geq \beta$ then
8: $\eta \leftarrow \alpha\sqrt{2\log m} - \sqrt{2(1-\alpha^2)\log \log m}$	8: return $\mathbf{x}$
9: return $\mathcal{D} = (H, \mathcal{A}^m, m, \eta)$	9: return $\perp$

random vector in $\mathcal{A}^m$: more specifically, we assign each input vector $\mathbf{x} \in \mathcal{S}$ to the random vector in $\mathcal{A}^m$ with the largest inner product. For a given query vector $\mathbf{q}$, the query algorithm selects all random vectors with an inner product larger than η with $\mathbf{q}$. Then, it searches for an approximate near neighbor in the lists of points associated with these vectors in the hash table. We call *buckets* the indices of the hash table (i.e. the random vectors), and *filters* the function used to query the hash table (i.e. the inner product). In this section, we consider the asymptotic regime for $n \rightarrow \infty$, so to use the limiting distribution of the extreme concomitant in Theorem 5.

► **Lemma 7** (Probability to Find a Close Point). *For $n \rightarrow \infty$, Top-1 search contains a bucket with a close point, if it exists, with at least $1 - o(1)$ probability.*

Proof. Consider a close point $\mathbf{x}_\alpha$, associated to the bucket $\mathbf{a}_{\mathbf{x}_\alpha}$. That bucket is found in search if $\langle \mathbf{a}_{\mathbf{x}_\alpha}, \mathbf{q} \rangle = Q_{\mathbf{x}_\alpha, [m]} \geq \eta$. From Proposition 20 and Theorem 5, we observe that

$$\Pr [Q_{\mathbf{x}_\alpha, [m]} \leq \eta] = \Pr_{\mathcal{N}(0, 1-\alpha^2)} [Z \leq -\sqrt{2(1-\alpha^2)\log \log m}] \leq \frac{1}{\log m} = O\left(\frac{1}{\log n}\right),$$

where we use $m = n^{\frac{\theta}{1-\alpha^2}}$. Thus, with probability at least $1 - o(1)$, $\mathbf{x}_\alpha$ is associated to a vector that exceeds the threshold. ◀

► **Lemma 8** (Expected Number of Buckets and Far Points). *For $n \rightarrow \infty$, $0 \leq \beta < \alpha < 1$ such that $(\alpha - \beta) = \Omega(\sqrt{\log \log n / \log n})$, Top-1 search returns in expectation at most $n^{\theta+o(1)}$ buckets, containing in expectation at most $n^{1-\theta \frac{(\alpha-\beta)^2}{(1-\alpha^2)(1-\beta^2)} + o(1)}$ far points.*

Proof. We observe that by setting $m = n^{\frac{\theta}{1-\alpha^2}}$ we get $\frac{\log \log m}{\log m} = O((1-\alpha^2) \frac{\log \log n}{\log n})$. Thus, the threshold is $\eta \geq \alpha\sqrt{2\log m}(1 - O(\frac{1-\alpha^2}{\alpha} \sqrt{\frac{\log \log n}{\log n}}))$, which is positive for $\alpha \geq \alpha - \beta = \Omega(\sqrt{\log \log n / \log n})$. From Proposition 20, the probability that a filter exceeds the threshold is

$$\Pr[\langle \mathbf{a}, \mathbf{q} \rangle \geq \eta] \leq \Pr_{\mathcal{N}(0, 1)} \left[Z \geq \alpha\sqrt{2\log m} \left(1 - \frac{1-\alpha^2}{\alpha} o(1) \right) \right] \leq m^{-\alpha^2 + (1-\alpha^2)o(1)}, \quad (1)$$

as the projection over a Gaussian vector is a normal random variable. In expectation, a query inspects at most $m^{1-\alpha^2+o(1-\alpha^2)}$ buckets. The claim follows by setting $m = n^{\frac{\theta}{1-\alpha^2}}$. For the analysis of far points, we may write $\eta \geq \beta\sqrt{2\log m} + (\alpha - \beta)\sqrt{2\log m} \left(1 - \frac{1-\alpha^2}{\alpha-\beta} O(\sqrt{\frac{\log \log n}{\log n}}) \right)$. The second factor is positive for $(\alpha - \beta) = \Omega(\sqrt{\log \log n / \log n})$. Thus, by applying Theorem 5 and Proposition 20, the probability to inspect a far point $\mathbf{x}_\beta$ is

$$\begin{aligned}
\Pr[Q_{\mathbf{x}_{\beta}, [m]} \geq \eta] &\leq \frac{\Pr}{\mathcal{N}(\beta\sqrt{2\log m}, 1-\beta^2)} \left[Z \geq \beta\sqrt{2\log m} + (\alpha - \beta)\sqrt{2\log m} \left(1 - \frac{1-\alpha^2}{\alpha - \beta} o(1) \right) \right] \\
&\leq \exp \left[- \frac{(\alpha - \beta)^2}{1 - \beta^2} \left(1 - \frac{1-\alpha^2}{\alpha - \beta} o(1) \right)^2 \cdot \log m \right] \\
&= m^{-\frac{(\alpha - \beta)^2}{1 - \beta^2} + \frac{(1 - \alpha^2)(\alpha - \beta)}{1 - \beta^2} o(1)}, \tag{2}
\end{aligned}$$

By inserting $m = n^{\frac{\theta}{1-\alpha^2}}$ in the previous inequality, we obtain $n^{-\theta \frac{(\alpha - \beta)^2}{(1 - \alpha^2)(1 - \beta^2)} + o(1)}$, as $\frac{\alpha - \beta}{1 - \beta^2} = O(1)$. Since we have at most n far points, the expected number of inspected far points is at most $n^{1 - \theta \frac{(\alpha - \beta)^2}{(1 - \alpha^2)(1 - \beta^2)} + o(1)}$. ◀

The proposed data structure **Top-1** (Algorithm 2) is a naive solution with high space, pre-processing time and query time, for the (α, β) -ANN.

► **Theorem 9.** *Consider $n \rightarrow \infty$. For any $0 \leq \beta < \alpha < 1$ such that $(\alpha - \beta) = \Omega(\sqrt{\log \log n / \log n})$, $0 < \theta \leq O(1)$, and for any dataset $\mathcal{S} = \{x_i\}_{i=1, \dots, n}$ in $\mathbb{S}^d$, **Top-1** solves with at least $1 - o(1)$ probability the (α, β) -ANN using pre-processing time $O(d \cdot n^{1 + \frac{\theta}{1 - \alpha^2}})$, space $O(d \cdot \max\{n, n^{\frac{\theta}{1 - \alpha^2}}\})$, and expected query time $O(d \cdot \max\{n^{\frac{\theta}{1 - \alpha^2}}, n^{1 - \theta \frac{(\alpha - \beta)^2}{(1 - \alpha^2)(1 - \beta^2)} + o(1)}\})$.*

Proof. The pre-processing time is given by $O(d \cdot n \cdot m) = O(d \cdot n^{1 + \frac{\theta}{1 - \alpha^2}})$ as for each of the n points we need to look at m random vectors of dimensionality d . Each point is assigned to only one random vector, so the space needed to store the data structure is $O(d \cdot (n + m)) = O(d \cdot \max\{n, n^{\frac{\theta}{1 - \alpha^2}}\})$. The running time is given by summing the running time of **search** and **query**. The buckets in **search**(**q**) are found in time $O(d \cdot m)$ while the expected running time of **query**(**q**) is given by the expected number of far points present in these buckets, which are at most $n^{1 - \theta \frac{(\alpha - \beta)^2}{(1 - \alpha^2)(1 - \beta^2)} + o(1)}$, and the expected number of buckets returned by **search**, which are at most $n^{\theta + o(1)}$, due to Lemma 8. Thus, the expected running time is at most $O(d \cdot \max\{n^{\frac{\theta}{1 - \alpha^2}}, n^{1 - \theta \frac{(\alpha - \beta)^2}{(1 - \alpha^2)(1 - \beta^2)} + o(1)}, n^{\theta + o(1)}\})$. The problem is solved with at least $1 - o(1)$ probability due to Lemma 7. ◀

3.1 Balanced and Unbalanced Top-1

The standard way to minimize the expected query time of an algorithm that solves (α, β) -ANN is to balance the number of buckets that have to be inspected with the number of far points (“error”) that are associated with those buckets. To balance the contribution of far points and the number of buckets inspected we choose $\theta = \frac{(1 - \alpha^2)(1 - \beta^2)}{(1 - \alpha\beta)^2}$, which solves the equation $\theta = 1 - \theta \frac{(\alpha - \beta)^2}{(1 - \alpha^2)(1 - \beta^2)}$. We denote this specific solution as ρ to highlight its connection to standard ANN analysis. However, alternative values of θ can be chosen to achieve different trade-offs. The next corollary follows from Lemma 8.

► **Corollary 10** (Balanced and Unbalanced Top-1). *Consider $n \rightarrow \infty$. For any $0 \leq \beta < \alpha < 1$ such that $(\alpha - \beta) = \Omega(\sqrt{\log \log n / \log n})$, consider $\sigma = 2 \frac{(1 - \alpha^2)(1 - \beta^2)}{(1 - \alpha\beta)^2 + (\alpha - \beta)^2}$ and $\rho = \frac{(1 - \alpha^2)(1 - \beta^2)}{(1 - \alpha\beta)^2}$. Define balanced and unbalanced **Top-1** as the data structures initialized with $\theta = \rho$ and $\theta = \sigma$ respectively. Then,*

1. *The expected number of buckets inspected by **search**(**q**) is at most $n^{\rho + o(1)}$ for balanced **Top-1**, and $n^{\sigma + o(1)}$ for unbalanced **Top-1**.*

■ **Algorithm 3** From ANN to ANNC using a space-partitioning data structure.

<pre> 1: procedure CONSTRUCTION($\mathcal{L}$) 2: $T[1, \dots, m] \leftarrow (0, \dots, 0)$ 3: for each $L_i \in \mathcal{L}$ do 4: $T[i] \leftarrow L_i$ 5: return $\mathcal{D} = (T, \mathcal{L})$ </pre>	<pre> 1: procedure QUERY($\mathbf{q}, \mathcal{Q}$) 2: $I(\mathbf{q}) \leftarrow \mathcal{Q}(\mathbf{q})$ 3: $\widehat{\text{ans}} \leftarrow 0$ 4: for $i \in I(\mathbf{q})$ do 5: $\widehat{\text{ans}} \leftarrow \widehat{\text{ans}} + T[i]$ 6: return $\widehat{\text{ans}}$ </pre>
--	---

2. The buckets from $\text{search}(\mathbf{q})$ contain, in expectation, at most $n^{\rho+o(1)}$ far points for balanced **Top-1** and $n^{\frac{\sigma}{2}+o(1)}$ far points for unbalanced **Top-1**.
3. For any $0 \leq \beta < \alpha \leq 1$ we have $\frac{\sigma}{2} < \rho < \sigma$.

Proof. It follows by a simple computation from Lemma 8. ◀

As $\frac{\rho}{1-\alpha^2} \geq 1$ the space and the running time of balanced **Top-1** is $O(d \cdot n^{\frac{\rho}{1-\alpha^2}})$, the latter considers the worst case as the expected number of far points is $n^{\rho+o(1)} \ll n^{\frac{\rho}{1-\alpha^2}}$. Space and running time follow directly for unbalanced **Top-1** as $\sigma > \rho$. The unusual behavior of unbalanced **Top-1** will be further clarified in the next section, where its utility for (α, β) -DP-ANNC will be discussed. Although **Top-1** benefits from a clean and straightforward analysis (due to the assumption $n \rightarrow \infty$), it involves significant preprocessing, space, and query time requirements. These limitations will be addressed in Section 5.2 through the use of *tensorization*. Additionally, the assumption of $n \rightarrow \infty$ will be lifted with a minor modification to the algorithm, as detailed in Section 5.1. Nevertheless, as we will show in the next section, **Top-1** still provides a meaningful solution for (α, β) -ANNC under differential privacy constraints.

4 From ANN to DP-ANNC

In this section, we study the relationship between ANN and DP-ANNC. We expose a general way to solve ANNC starting from a *space-partitioning data structure* for ANN, and discuss different differentially private mechanisms to privatize the ANNC data structure. We also show how the unbalanced data structure from the previous section can be used to increase the accuracy for (α, β) -DP-ANNC, by paying an increase in query time, pre-processing time, and space usage.

4.1 From ANN to ANNC

Inspired by the *list-of-points* data structure developed in [5], we define a family of data structures suitable for a general reduction from ANN to ANNC.

► **Definition 11** (Space Partitioning Data Structure). *Given a set $\mathcal{S} \subseteq \mathbb{S}^{d-1}$ and an integer m , a space-partitioning data structure for the ANN problem is defined as follows:*

- *The data structure is a partition⁵ of $\mathcal{S}$ into m sets $\mathcal{L} = (L_1, \dots, L_m)$ such that $\bigcup_{i \in [m]} L_i \subseteq \mathcal{S}$ and $L_i \cap L_j = \emptyset$ for $i \neq j$, and a function $\mathcal{Q}$ that maps $\mathbb{S}^{d-1} \ni \mathbf{q} \mapsto I(\mathbf{q}) \subseteq [m]$.*
- *For a query $\mathbf{q}$, we obtain the set $I(\mathbf{q}) \leftarrow \mathcal{Q}(\mathbf{q})$ and scan all points in $L_i, i \in I(\mathbf{q})$. If there exists a point with inner product at least β , we return it. Otherwise we return $\perp$.*

⁵ Technically we define the data structure by using a not full partition, as we allow some points to not be stored.

The total space is $|\mathcal{Q}| + O(d \cdot n)$, where $|\mathcal{Q}|$ is the space necessary to store the function $\mathcal{Q}$. The query time is at most $T_{\mathcal{Q}}(\mathbf{q}) + O(d \sum_{i \in I(\mathbf{q})} |L_i|)$, where $T_{\mathcal{Q}}$ is the time taken to compute $I(\mathbf{q})$ given query $\mathbf{q} \in \mathbb{S}^{d-1}$, and $O(d \sum_{i \in I(\mathbf{q})} |L_i|)$ is the worst-case time needed to check all the points.

For example, for Algorithm 2, L_i represents the points that achieve their maximum inner product with $\mathbf{a}_i$. $\mathcal{Q}$ consists of all $\mathbf{a}_1, \dots, \mathbf{a}_m$ (its size is dm) and computes the indices $I(\mathbf{q})$ of all filters that are above the query threshold. For the data structure of Andoni et al. [4] discussed in the introduction, L_i are the leaves of the tree, and $\mathcal{Q}$ represents the navigation tree-based data structure. Algorithm 3 presents a simple transformation for a space-partitioning data structure: It indeed suffices to substitute the actual points with their amount in each list. The new (α, β) -ANNC query returns the sum of the elements contained in these lists. Since each point is stored at most once, summing the cardinality of each bucket ensures that no point is counted more than once.

► **Lemma 12** (From ANN to ANNC). *Let $\mathcal{S} = \{x_i\}_{i=1, \dots, n} \subseteq \mathbb{S}^{d-1}$. Consider a space-partitioning data structure for $\mathcal{S}$ such that for each $\mathbf{q} \in \mathbb{S}^{d-1}$: (i) $I(\mathbf{q})$ contains a list with a close point with probability at least $1 - o(1)$, and (ii) the expected number of far points in $\bigcup_{i \in I(\mathbf{q})} L_i$ is at most $\mathcal{K}$. Then query in Algorithm 3 returns a value $\widehat{\text{ans}}$ that, with probability at least $2/3$, satisfies the following inequality:*

$$(1 - o(1))|\mathcal{S} \cap B(\mathbf{q}, \alpha)| \leq \widehat{\text{ans}} \leq |\mathcal{S} \cap B(\mathbf{q}, \beta)| + \mathcal{K}.$$

The data structure in Algorithm 3 uses space $|\mathcal{Q}| + O(n)$ and the query time is $T_{\mathcal{Q}}(\mathbf{q}) + |I(\mathbf{q})|$.

4.2 From ANNC to DP-ANNC

The data structure returned by Algorithm 3 uses *counters*, which is essentially a histogram. This histogram can be privatized using the algorithms presented in Section 2.4. To achieve differential privacy, we need to analyze the sensitivity of the counters $T[1..m]$. Let $\mathcal{S}$ and $\mathcal{S}'$ be two neighboring datasets that differ in exactly one point, and let $\mathcal{D}$ and $\mathcal{D}'$ be the data structures constructed for these data sets, respectively. Applying Algorithm 3 to $\mathcal{D}$ and $\mathcal{D}'$ will result in two counters T and T' that differ by at most 1 in at most one position. If $\mathcal{Q}$ is data independent, i.e., $\mathcal{Q}$ does not depend on the actual data set $\mathcal{S}$, it is sufficient to privatize the counter T , which can be done using any differentially private mechanism `make_private` for histograms, as shown in the aforementioned `DPTop-1` (see Algorithm 1). The next theorem states the guarantees for two specific privacy mechanisms.

► **Theorem 13** (DP-ANNC with Truncated Laplace or Max Projection). *Let $\mathcal{S} = \{x_i\}_{i=1, \dots, n} \subseteq \mathbb{S}^{d-1}$, $\mathbf{q} \in \mathbb{S}^{d-1}$. Consider a space partitioning data structure for $\mathcal{S}$ satisfying the assumptions of Lemma 12, with the addition of $\mathbb{E}[|I(\mathbf{q})|] \leq \mathcal{K}$ and $\mathcal{Q}$ being data independent. When T is privatized using the truncated Laplace mechanism, the data structure is (ε, δ) -DP, for any $\varepsilon \leq 1$, requires an additional $O(n)$ term in space and pre-processing time (compared to Lemma 12), and the query algorithm returns a value $\widehat{\text{ans}}$ that satisfies the following inequality with probability at least $2/3$:*

$$(1 - o(1))|\mathcal{S} \cap B(\mathbf{q}, \alpha)| - O\left(\frac{\log(1/\delta)}{\varepsilon} \mathcal{K}\right) \leq \widehat{\text{ans}} \leq |\mathcal{S} \cap B(\mathbf{q}, \beta)| + O\left(\frac{\log(1/\delta)}{\varepsilon} \mathcal{K}\right). \quad (3)$$

When T is privatized with the **Max Projection** mechanism, then the data structure is $(\varepsilon, 0)$ -DP, requires an additional $O(\varepsilon \cdot n)$ term in space and pre-processing time, and the additive error in Equation 3 becomes $O(\frac{\mathcal{K}}{\varepsilon})$.

Due to Lemma 7 and Corollary 10, balanced **Top-1** satisfies the requirements for Theorem 13 with $\mathcal{K} = n^{\rho+o(1)}$, which proves Theorem 1. Moreover, the tree-based data structure described by Andoni et al. [4] is another data structure that satisfies these requirements of Theorem 13.⁶

4.2.1 Usefulness of Unbalanced Data Structure

We now study how an unbiased and uncorrelated differentially private estimator of T , from the unbalanced **Top-1** ANNC data structure, can be used to reduce the error compared to Theorem 13. The construction leverages the concentration of the sum of i.i.d. Laplace random variables.

► **Theorem 14** (DP-ANNC with Laplace Noise and Unbalanced Data Structure). *Let $\mathcal{S} = \{x_i\}_{i=1,\dots,n} \subseteq \mathbb{S}^{d-1}$, $\mathbf{q} \in \mathbb{S}^{d-1}$. Consider a space partitioning data structure for $\mathcal{S}$ satisfying the assumptions of Lemma 12, with the addition of $\mathbb{E}[I(\mathbf{q})] \leq \mathcal{K}^2$ and $\mathcal{Q}$ being data independent. When T is privatized by using the Laplace mechanism, the data structure is $(\varepsilon, 0)$ -DP, for any $\varepsilon \leq 1$, and **query** returns a value $\widetilde{\text{ans}}$ that satisfies the following inequality with probability at least $2/3$:*

$$(1 - o(1))|\mathcal{S} \cap B(\mathbf{q}, \alpha)| - O\left(\frac{\mathcal{K}}{\varepsilon}\right) \leq \widetilde{\text{ans}} \leq |\mathcal{S} \cap B(\mathbf{q}, \beta)| + O\left(\frac{\mathcal{K}}{\varepsilon}\right).$$

The privatized data structure requires an additional $O(m)$ space, and pre-processing time.

Due to Lemma 7 and Corollary 10, unbalanced **Top-1** satisfies the requirements for Theorem 14 with $\mathcal{K} = n^{\frac{\sigma}{2}+o(1)}$. As $\frac{\sigma}{2} < \rho$ unbalanced **Top-1** is always more accurate than **Top-1** for DP-ANNC. However, the pre-processing time and the space increase.

In the next section we provide several improvements for **Top-1**, aiming to get rid of the asymptotic assumption $n \rightarrow \infty$ used to apply the Theorem 5 for concomitant statistics, and reduce the pre-processing time, the query time, and the space. These improvements regard only the balanced ANN data structure; the additional requirements in space and pre-processing time of $O(m)$ for DP-ANNC with unbalanced data structures will still be present and are an interesting open question for future work. Finally, we highlight that for errors of the form $n^{C+o(1)}$ we may increase the range of the privacy budget to $\varepsilon \leq n^{o(1)}$ in Theorems 13 and 14, for the same argument provided by Andoni et al. [4].

5 Improving the Top-1 Data Structure

In this section we propose two improvements of **Top-1**. With **CloseTop-1** we get rid of the assumption of $n \rightarrow \infty$,⁷ while with **TensorCloseTop-1** we reduce the pre-processing time to $d \cdot n^{1+o(1)}$, the space to $O(d \cdot n)$, and obtain an expected query time of $d \cdot n^{\theta+o(1)}$. In addition, we discuss how **TensorCloseTop-1** can solve the (r, c) -ANN in the Euclidean space in Appendix C. These improvements do not alter the core of the data structure (a hash table of points with a **search** function), allowing them to be utilized for DP-ANNC as discussed in the previous section.

Algorithm 4 CloseTop-1 Data Structure.

```

1: procedure CONSTRUCTION( $S \subset \mathbb{S}^{d-1}, \alpha, \beta, \theta$ )
2:    $m \leftarrow \lceil n^{\frac{\theta}{1-\alpha^2}} \rceil$ 
3:    $\mathcal{A}^m = (\mathbf{a}_1, \dots, \mathbf{a}_m)$  with  $\mathbf{a}_i \sim \mathcal{N}(0, 1)^d$  ▷ Sample  $m$  Gaussian Random Vectors
4:    $H \leftarrow$  empty hash table
5:   for  $\mathbf{x} \in S$  do
6:     for  $i \in \{1, \dots, m\}$  do
7:       if  $\sqrt{2 \log m} - \frac{3}{2} \frac{\log \log m}{\sqrt{2 \log m}} \leq \langle \mathbf{a}_i, \mathbf{x} \rangle \leq \sqrt{2 \log m}$  then
8:          $H.\text{insert}(i, \mathbf{x})$ 
9:       break
10:   $\eta \leftarrow \alpha \sqrt{2 \log m} - \sqrt{2(1 - \alpha^2) \log \log m}$ 
11:  return  $\mathcal{D} = (H, \mathcal{A}^m, m, \eta)$ 

```

12: ▷ **search** and **query** are the same in Algorithm 2.

5.1 CloseTop-1

We now study **CloseTop-1** (see Algorithm 4), a practical implementation of the previous asymptotic data structure. In **Top-1** we associate to each point of the dataset the random vector with the highest inner product. This is an intuitive choice that leads to a simple and clear analysis by analyzing Gaussian tails of concomitant statistics (Theorem 5). However, this is an asymptotic theorem, results from the fact that $X_{\mathbf{x}} = \max_{\mathbf{a} \in \mathcal{A}^m} \langle \mathbf{a}, \mathbf{x} \rangle = \sqrt{2 \log m} - o(1)$ and $\text{Var}[X_{\mathbf{x}}] = o(1)$ [18]. In fact, it can be obtained by Lemma 4 by setting $X_{\mathbf{x}} = \sqrt{2 \log m}$. The intuition of **CloseTop-1** is to provide a lower and an upper bound for $X_{\mathbf{x}}$ by construction, by associating to each point of the dataset a random vector with an inner product *close* to the expected maximum, so the name of the data structure. In the proposed construction, we sample m random vectors $\mathbf{a} \sim \mathcal{N}(0, 1)^d$, and we associate to any $\mathbf{x}$ the first random vector such that $\sqrt{2 \log m} - \frac{3}{2} \frac{\log \log m}{\sqrt{2 \log m}} \leq \langle \mathbf{a}, \mathbf{x} \rangle \leq \sqrt{2 \log m}$. If at least one random vector succeeds in the association, then we say that $\mathbf{x}$ collided. The key property of **CloseTop-1** is that a point collides with high probability (Lemma 23), which allows to state the following lemma.

► **Lemma 15.** *Lemma 7 and Lemma 8 remain valid for **CloseTop-1** under the same assumptions, without relying on the $n \rightarrow \infty$ condition.*

As Theorem 9 and Corollary 10 are derived from Lemmas 7 and 8, **CloseTop-1** give the same results as **Top-1**, without relying on the assumption that the concomitants follow a limiting distribution.

5.2 TensorCloseTop-1

In this section, we propose **TensorCloseTop-1** (see Algorithm 5), to reduce the pre-processing time to $d \cdot n^{1+o(1)}$, space to $O(d \cdot n)$, and expected query time to $d \cdot n^{\rho+o(1)}$ (for the balanced data structure). The data structure uses a technique developed in [9] called *tensoring* that essentially allows to simulate an exponential number of vectors by concatenating a polynomial number of data structures. The same expedient was used in [6] to get a pre-processing time of $n^{1+\rho+o(1)}$. This technique is similar to creating a tree, yet this data structure allows parallel

⁶ Technically, as stated earlier, [4] analyze their data structure for a fixed choice of α .

⁷ We observe that, although the time and space complexities are expressed in big-O notation (i.e., with a notation asymptotic in n), the correctness of this algorithm does not require assuming a limiting distribution for the concomitants, while this was the case for **Top-1**.

■ **Algorithm 5** TensorCloseTop-1 Data Structure.

1: procedure CONSTRUCTION($S \subset \mathbb{S}^{d-1}, \alpha, \beta, \theta$)	1: procedure SEARCH($\mathbf{q}$)
2: $t \leftarrow \lceil \frac{\log^{1/8} n}{1-\alpha^2} \rceil$	2: for $i \in (1, \dots, t)$ do
3: $\tilde{m} \leftarrow \lceil n^{\frac{1}{t} \frac{\theta}{1-\alpha^2}} \rceil$	3: $\tilde{B}_i \leftarrow \mathcal{D}_i.\text{search}(\mathbf{q}, \eta)$
4: $\{\mathcal{D}_i\}_{i=1, \dots, t} \leftarrow \{\text{CloseTop-1}(S, \tilde{m})\}_{i=1, \dots, t}$	4: return $\tilde{B}_1 \times \dots \times \tilde{B}_t \triangleright$ Cartesian Product
5: $H \leftarrow$ empty hash table	5: procedure QUERY($\mathbf{q}$)
6: for $\mathbf{x} \in S$ do	6: $B \leftarrow H.\text{search}(\mathbf{q})$
7: $i \leftarrow (H_1.\text{hash}(\mathbf{x}), \dots, H_t.\text{hash}(\mathbf{x}))$	7: for $i \in B$ do
8: $H.\text{insert}(i, \mathbf{x})$	8: for $\mathbf{x} \in H[i]$ do
9: $\eta \leftarrow \alpha \sqrt{2 \log \tilde{m}} - \sqrt{2(1-\alpha^2) \log \log \tilde{m}}$	9: if $\langle \mathbf{q}, \mathbf{x} \rangle \geq \beta$ then
10: return $(H, \{\mathcal{D}\}_{i=1, \dots, t}, \eta)$	10: return $\mathbf{x}$
	11: return None

evaluation for the hashes (Line 2-3 Algorithm 5 `search`). Define the concatenation factor $t \in \mathbb{N}$ and assume $m^{1/t}$ is an integer; consider t independent `CloseTop-1` data structures $\mathcal{D}_1, \dots, \mathcal{D}_t$ each using $m^{1/t}$ Gaussian vectors $\mathbf{a}_{i,j}$, where $i \in [t]$ indicates the data structure and $j \in [m^{1/t}]$ indicates the vector. For each point $\mathbf{x} \in S$ consider the t colliding vectors in each data structure $(\mathbf{a}_{1,i_1}, \dots, \mathbf{a}_{t,i_t})$, then map the point to a bucket $(i_1, \dots, i_t) \in [m^{1/t}]^t$ using a hash table. Given a query $\mathbf{q} \in \mathbb{S}^{d-1}$, for each data structure the indices $\tilde{B}_i$ of the random vectors are selected such that $\langle \mathbf{a}, \mathbf{q} \rangle \geq \eta$, hence $\tilde{B}_i := \{j \in [m^{1/t}] : \langle \mathbf{a}_{i,j}, \mathbf{q} \rangle \geq \eta\}$, and a search in all the buckets $\tilde{B}_1 \times \dots \times \tilde{B}_t$ is performed. The number of random vectors to sample is $t \cdot m^{1/t}$ which is sub-linear in n provided the data structure is not designed to search for points with exceedingly high inner product similarity.

► **Proposition 16** (Tensorization). *For any constant $C > 0$ assume $\frac{1}{1-\alpha^2} \leq (\log n)^C$. Then for $t = \frac{\log^{1/8} n}{1-\alpha^2}$, $m = n^{\frac{\theta}{(1-\alpha^2)}}$, and $\theta = O(1)$, we have $t \cdot m^{1/t} = n^{o(1)}$*

Proof. Just a simple computation: $m^{1/t} = n^{\frac{\theta}{\log^{1/8} n}} = n^{o(1)}$ as $\theta = O(1)$, while $t = \frac{\log^{1/8} n}{1-\alpha^2} \leq (\log n)^{O(1)} = n^{o(1)}$, then $t \cdot m^{1/t} = n^{o(1)}$. ◀

In practice $t, m^{1/t}$ and m are all integers. However, this does not affect the asymptotic behavior since $\lceil t \rceil m^{1/\lceil t \rceil} \leq (t+1) \cdot m^{1/t} = n^{o(1)}$. With this trick, we reduce the query time due to `search` (Line 2,3,4 Algorithm 5, procedure `search`) to $n^{o(1)}$. Therefore, the query time is mainly affected by how many times the data structure needs to access the hash table, which can be bounded in expectation. Under similar assumptions for α we can prove that `TensorCloseTop-1` finds with high probability a close point.

► **Lemma 17** (Probability to Find a Close Point). *For any $0 \leq \beta < \alpha < 1$ such that $1 - \alpha^2 = \omega(\log^{-3/4} n)$, `TensorCloseTop-1` finds a close point, if it exists, with at least $1 - o(1)$ probability.*

► **Lemma 18** (Expected Number of Buckets and Far Points). *For any $0 \leq \beta < \alpha < 1$ such that $(1 - \alpha^2) = \omega(\log^{-3/4} n)$, and $(\alpha - \beta) = \Omega(\sqrt{\frac{\log \log n}{\log^{7/8} n}})$, `TensorCloseTop-1 search` finds in expectation at most $n^{\theta+o(1)}$ buckets containing at most $n^{1-\theta \frac{(1-\alpha^2)(1-\beta^2)}{(\alpha-\beta)^2} + o(1)}$ far points.*

The previous Lemma states that θ has the same function it has in `CloseTop-1`, so it can be used to construct balanced and unbalanced `TensorCloseTop-1`. We now argue for the query, space and pre-processing time.

► **Theorem 19.** *For any $0 \leq \beta < \alpha < 1$ such that $(1 - \alpha^2) = \omega(\log^{-3/4} n)$, $(\alpha - \beta) = \Omega(\sqrt{\frac{\log \log n}{\log^{7/8} n}})$, and $0 < \theta \leq O(1)$. For any dataset $\mathcal{S} = \{x_i\}_{i=1, \dots, n}$ in $\mathbb{S}^d$, **TensorCloseTop-1** solves with at least $1 - o(1)$ probability the (α, β) -ANN using space $O(d \cdot n)$, preprocessing time $d \cdot n^{1+o(1)}$, and expected query time $d \cdot \max\{n^{\theta+o(1)}, n^{1-\theta \frac{(\alpha-\beta)^2}{(1-\alpha^2)(1-\beta)^2} + o(1)}\}$.*

Proof. Due to Proposition 16, the data structure needs to store $t \cdot m^{1/t} = n^{o(1)}$ random vectors. As each point $\mathbf{x}$ is stored in at most one bucket, the space is $O(d \cdot (n + n^{o(1)})) = O(d \cdot n)$. As to each point it is necessary to compute $m^{1/t}$ inner products at most t times, the pre-processing time is $O(t \cdot d \cdot n \cdot m^{1/t}) = O(d \cdot n^{1+o(1)})$. The buckets in **search**($\mathbf{q}$) can be computed in time $O(d \cdot n^{o(1)})$, so the expected query time is at most $d \cdot \max\{n^{\theta+o(1)}, n^{1-\theta \frac{(\alpha-\beta)^2}{(1-\alpha^2)(1-\beta)^2} + o(1)}\}$ due to Lemma 18. The problem is solved with at least $1 - o(1)$ probability due to Lemma 17. ◀

As Corollary 10 applies to **TensorCloseTop-1** due to Lemma 18, the parameters ρ and σ respectively characterize the balanced and unbalanced versions of **TensorCloseTop-1**. The balanced version has an expected query time of $O(d \cdot n^{\rho+o(1)})$ and, when combined with Theorem 13, achieves an additive error of $O(n^{\rho+o(1)}/\varepsilon)$ for differentially private approximate nearest neighbor search (DP-ANNC). For $\varepsilon = O(1)$, the additional space and preprocessing requirements are $O(n)$. In contrast, the unbalanced version of **TensorCloseTop-1** has an expected query time of $O(d \cdot n^{\sigma+o(1)})$ and, when used with Theorem 14, yields an asymptotically smaller additive error $O(n^{\frac{\sigma}{2}+o(1)}/\varepsilon)$ for DP-ANNC. However, this approach incurs a significant drawback: the Laplace noise introduces an additional space and preprocessing overhead of $O(m) = O(\tilde{m}^t) = O(n^{\frac{\sigma}{1-\alpha^2}})$. This overhead becomes the dominant cost as $\frac{\sigma}{1-\alpha^2} \geq 1$, especially in the Euclidean ANN problem, where $(1 - \alpha^2)^{-1} = \text{polylog}(n)$. For further details on how this data structure is applied in Euclidean space, see Appendix C.

6 Conclusion and Open Problems

This paper introduced and analyzed simple linear space data structures that solve the (α, β) -ANN problem and can be transformed into efficient solutions for its counting variant under differential privacy. This provides an alternative data structure to the one proposed recently by Andoni et al. [4] with a simpler data structure and analysis. We provided general black-box transformations from approximate near neighbor problems to their counting variant under privacy constraints and showed that interesting error/time trade-offs are possible via *unbalanced* ANN data structures. The most intriguing open question was already posed by Andoni et al. [4]: Can one obtain better accuracy guarantees for range counting than by transforming near neighbor data structures that have well-understood lower bounds [5]? For example, [1] describes a sampling based range counting algorithm that could be a good starting point for further investigation. For the presented data structures, one should further investigate the relation of the noise error due to differential privacy and the error due to including “far points” which could give interesting trade-offs. We initiated such a study through unbalanced ANN data structures; the main obstacle for a space-efficient solution is to store “small counts” in a data structure that uses space $O(f(\varepsilon)n^{1+o(1)})$ and provides unbiased counters such that the expected error of the sum of $\mathcal{K}$ counters is only a factor $O(\sqrt{\mathcal{K}})$ larger than the expected per-point error. Finally, while we believe that our algorithms are simple and straightforward, an experimental comparison between the different solutions presented here and in the literature seems necessary, not only for approximate range counting, but also filtering-based approximate near neighbor search. In fact, only the work of Pham et al. [27] provided evidence of the practical impact of filtering-based near neighbor search, and they achieve their result by a combination of LSH and LSF.

References

- 1 Thomas D. Ahle, Martin Aumüller, and Rasmus Pagh. Parameter-free locality sensitive hashing for spherical range reporting. In *SODA*, pages 239–256. SIAM, 2017. doi:10.1137/1.9781611974782.16.
- 2 Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 136–150. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.18.
- 3 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008. doi:10.1145/1327452.1327494.
- 4 Alexandr Andoni, Piotr Indyk, Sepideh Mahabadi, and Shyam Narayanan. Differentially private approximate near neighbor counting in high dimensions. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA, 2023. Curran Associates Inc.
- 5 Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '17*, pages 47–66, USA, 2017. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611974782.4.
- 6 Martin Aumüller, Sarel Har-Peled, Sepideh Mahabadi, Rasmus Pagh, and Francesco Silvestri. Sampling a near neighbor in high dimensions — who is the fairest of them all? *ACM Trans. Database Syst.*, 47(1), April 2022. doi:10.1145/3502867.
- 7 Martin Aumüller, Christian Janos Lebeda, and Rasmus Pagh. Representing sparse vectors with differential privacy, low error, optimal space, and fast access. *Journal of Privacy and Confidentiality*, 12(2), November 2022. doi:10.29012/jpc.809.
- 8 Roberto J. Bayardo, Yiming Ma, and Ramakrishnan Srikant. Scaling up all pairs similarity search. In *Proceedings of the 16th International Conference on World Wide Web, WWW '07*, pages 131–140, New York, NY, USA, 2007. Association for Computing Machinery. doi:10.1145/1242572.1242591.
- 9 Tobias Christiani. A framework for similarity search with space-time tradeoffs using locality-sensitive filtering. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '17*, pages 31–46, USA, 2017. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611974782.3.
- 10 H. A. David and J. Galambos. The asymptotic theory of concomitants of order statistics. *Journal of Applied Probability*, 11(4):762–770, 1974. URL: <http://www.jstor.org/stable/3212559>.
- 11 Devdatt Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, USA, 1st edition, 2009.
- 12 Cynthia Dwork. Differential privacy and the us census. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS '19*, page 1, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3294052.3322188.
- 13 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography, TCC'06*, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag. doi:10.1007/11681878_14.
- 14 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014. doi:10.1561/04000000042.
- 15 Kave Eshghi and Shyamsundar Rajaram. Locality sensitive hash functions based on concomitant rank order statistics. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08*, pages 221–229, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1401890.1401921.

- 16 Thomas R Fleming and David P Harrington. *Counting processes and survival analysis*. John Wiley & Sons, 2013.
- 17 Quan Geng, Wei Ding, Ruiqi Guo, and Sanjiv Kumar. Tight analysis of privacy and utility tradeoff in approximate differential privacy. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 89–99. PMLR, 26–28 August 2020. URL: <https://proceedings.mlr.press/v108/geng20a.html>.
- 18 Peter Hall. On the rate of convergence of normal extremes. *Journal of Applied Probability*, 16(2):433–439, 1979. URL: <http://www.jstor.org/stable/3212912>.
- 19 Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70, 2010. doi:10.1109/FOCS.2010.85.
- 20 Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, pages 705–714, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1806689.1806786.
- 21 Ziyue Huang and Ke Yi. Approximate Range Counting Under Differential Privacy. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 45:1–45:14, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2021.45.
- 22 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. Association for Computing Machinery. doi:10.1145/276698.276876.
- 23 Michael Kapralov. Smooth tradeoffs between insert and query complexity in nearest neighbor search. In *PODS*, pages 329–342. ACM, 2015. doi:10.1145/2745754.2745761.
- 24 S. Muthukrishnan and Aleksandar Nikolov. Optimal private halfspace counting via discrepancy. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 1285–1292, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2213977.2214090.
- 25 Rina Panigrahy. Entropy based nearest neighbor search in high dimensions. In *SODA*, pages 1186–1195. ACM Press, 2006. URL: <http://dl.acm.org/citation.cfm?id=1109557.1109688>.
- 26 Ninh Pham. Simple yet efficient algorithms for maximum inner product search via extreme order statistics. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pages 1339–1347, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3447548.3467345.
- 27 Ninh Pham and Tao Liu. Falconn++: a locality-sensitive filtering approach for approximate nearest neighbor search. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.
- 28 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Asell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. URL: <http://proceedings.mlr.press/v139/radford21a.html>.
- 29 Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. Association for Computing Machinery. doi:10.1145/371920.372071.
- 30 Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing)*. The MIT Press, 2006.

- 31 Stanislaw J. Szarek and Elisabeth Werner. A nonsymmetric correlation inequality for gaussian measure. 68(2):193–211, February 1999. doi:10.1006/jmva.1998.1784.
- 32 Salil Vadhan. *The Complexity of Differential Privacy*, pages 347–450. Springer, Yehuda Lindell, ed., 2017. doi:10.1007/978-3-319-57048-8_7.
- 33 Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *CoRR*, abs/1408.2927, 2014. arXiv:1408.2927.
- 34 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2):357–365, 2005. Automata, Languages and Programming: Algorithms and Complexity (ICALP-A 2004). doi:10.1016/j.tcs.2005.09.023.

A

 Useful inequalities and Additional Definitions

A.1 Differentially Private Mechanisms

In this work we considered three differentially private mechanisms:

- The *Truncated Laplace mechanism* [17], used also by Andoni et al. [4], which obfuscate each *positive* entry of $f(\mathcal{S})$ by adding truncated Laplace noise. The mechanism is (ε, δ) -DP and produces a biased estimator with expected absolute error $O(\log(1/\delta)/\varepsilon)$. In our context, it has the advantage that it only needs to sample and store the counts of the non-zero entries, i.e., at most n random variables.
- The *Laplace mechanism* [13], which adds independent Laplace noise to each entry of $f(\mathcal{S})$. The mechanism is $(\varepsilon, 0)$ -DP and produces an unbiased estimator with uncorrelated entries with absolute expected error $O(1/\varepsilon)$. The estimator behaves well for *range queries* (i.e. $\sum_{i \in B} f_i(\mathcal{S})$ for some $B \subseteq [\mathcal{X}]$) obtaining an expected absolute error $O(\sqrt{B}/\varepsilon)$. However, it requires to sample and store $|\mathcal{X}|$ random variables.
- The *Max Projection mechanism* [7] which stores all “small counts” in a sketching-based data structure. The mechanism is $(\varepsilon, 0)$ -DP and produces a data structure with $O(1)$ access time returning a biased estimator with expected absolute error $O(1/\varepsilon)$. The additional space it needs is $O(\varepsilon \cdot n)$, making it a valid pure-DP alternative to the Truncated Laplace mechanism.

A.2 Tail Bounds

We will make use of the following Gaussian tail bounds.

► **Proposition 20** (Gaussian Tail Bounds [11]). *Let $Z \sim \mathcal{N}(\mu, \sigma^2)$. Then, for any $t \geq 0$, we have that $\Pr[|Z - \mu| \geq t] \leq e^{-\frac{t^2}{2\sigma^2}}$.*

► **Proposition 21** (Proposition 3, [31]). *Let Z be a standard normal random variable. Then, for any $t > -1$, we have that*

$$\frac{2\sqrt{2\pi}}{t + \sqrt{t^2 + 4}} e^{-\frac{t^2}{2}} \leq \Pr[Z \geq t] \leq \frac{4\sqrt{2\pi}}{3t + \sqrt{t^2 + 8}} e^{-\frac{t^2}{2}},$$

From the previous proposition, it may be more useful to use the following loose bounds:

► **Proposition 22.** *Let Z be a standard normal random variable. Then, for any $t > 1$, we have that $\frac{2\sqrt{2\pi}}{(1+\sqrt{5})t} e^{-\frac{t^2}{2}} \leq \Pr[Z \geq t] \leq \frac{4\sqrt{2\pi}}{3t} e^{-\frac{t^2}{2}}$*

Proof. The bounds follow from Proposition 21. The upper bound is trivial, while the lower bound follows by noticing that $\sqrt{t^2 + 4} \leq t\sqrt{5}$ as $t > 1$. ◀

B Omitted Proofs

B.1 Omitted Proofs in Section 4

Proof of Lemma 12. We start with the lower bound. Let X be the random variable indicating the number of close points in $\mathcal{S}$ not included in $\widehat{\text{ans}}$. Due to requirement (i) the probability to not find, and so to not count, a close point is at most $o(1)$, then $\mathbb{E}[X] \leq |\mathcal{S} \cap B(\mathbf{q}, \alpha)|o(1)$. Using Markov's inequality we have that $X \leq |\mathcal{S} \cap B(\mathbf{q}, \alpha)|o(1)$ with constant probability. Consider now the number of close points counted $\widehat{\text{ans}}_{\text{close}}$, clearly $\widehat{\text{ans}} \geq \widehat{\text{ans}}_{\text{close}}$ and $\widehat{\text{ans}}_{\text{close}} = |\mathcal{S} \cap B(\mathbf{q}, \alpha)| - X$. Therefore, with constant probability we have $\widehat{\text{ans}}_{\text{close}} \geq |\mathcal{S} \cap B(\mathbf{q}, \alpha)| - |\mathcal{S} \cap B(\mathbf{q}, \alpha)|o(1) = |\mathcal{S} \cap B(\mathbf{q}, \alpha)|(1 - o(1))$ which concludes the proof for the lower bound.

We proceed with the upper bound. Let Y be the random variable indicating the number of far points in $\mathcal{S}$ included in $\widehat{\text{ans}}$, then $\widehat{\text{ans}} \leq |\mathcal{S} \cap B(\mathbf{q}, \beta)| + Y$. Due to requirement (ii) we have that $\mathbb{E}[Y] \leq \mathcal{K}$. Thus, by using Markov's inequality $\widehat{\text{ans}} \leq |\mathcal{S} \cap B(\mathbf{q}, \beta)| + \mathcal{K}$ with constant probability. Combining these two bounds, we arrive at the desired result.

As the algorithm substitute d dimensional point with a number, the space to store these number reduces to $O(n)$. The query does not search for a ANN, but sums all the numbers stored in the counter on the indices $I(\mathbf{q})$, so the running time is $T_Q(\mathbf{q}) + |I(\mathbf{q})|$. ◀

Proof of Theorem 13. As $\mathcal{Q}$ is data independent, on neighboring datasets the data structures differ only in the counters. We start by considering the truncated Laplace noise. Let T be the counter from Algorithm 3 and $\tilde{T}$ be the differentially private version. The error due to differential privacy in the counts is $|\tilde{T}[i] - T[i]| \leq O\left(\frac{\log(1/\delta)}{\epsilon}\right)$, as the truncated Laplace mechanism adds bounded noise sampled from $[-C\frac{\log(1/\delta)}{\epsilon}, C\frac{\log(1/\delta)}{\epsilon}]$ for some $C > 0$. Therefore, the expected error between $\widetilde{\text{ans}}$ and $\widehat{\text{ans}}$ is at most

$$\mathbb{E}[|\widetilde{\text{ans}} - \widehat{\text{ans}}|] = \mathbb{E}\left[\left|\sum_{i \in I(\mathbf{q})} (\tilde{T}[i] - T[i])\right|\right] \leq O\left(\frac{\log(1/\delta)}{\epsilon} \mathbb{E}[|I(\mathbf{q})|]\right) \leq O\left(\frac{\log(1/\delta)}{\epsilon} \mathcal{K}\right).$$

Thus, by Markov's inequality we have $|\widetilde{\text{ans}} - \widehat{\text{ans}}| \leq O\left(\frac{\log(1/\delta)}{\epsilon} \mathcal{K}\right)$ with constant probability. The claim follows by Lemma 12 and $\epsilon \leq 1$. As the Truncated Laplace mechanism only needs to sample at most n random variables, the additional factor in space and pre-processing time is $O(n)$.

Max Projection returns a $(\epsilon, 0)$ -DP counter $\tilde{T}$ with constant access time, using space and pre-processing time $O(\epsilon n)$, and with error $\mathbb{E}[|T[i] - \tilde{T}[i]|] \leq O(1/\epsilon)$ (Corollary 8.3 [7]). The analysis then follows identically. ◀

Proof of Theorem 14. Let T be the counter from Algorithm 3, $\tilde{T}$ its differential private version, and $I(\mathbf{q})$ be the set of indices of the buckets the algorithms needs to inspect, then $\widetilde{\text{ans}} = \sum_{i \in I(\mathbf{q})} \tilde{T}[i]$ and $\widehat{\text{ans}} = \sum_{i \in I(\mathbf{q})} T[i]$. The application of Laplace noise leads to an unbiased and uncorrelated estimator $\tilde{T}[i]$ so the variance of the error is

$$\text{Var}[\widetilde{\text{ans}} - \widehat{\text{ans}}] = \mathbb{E}\left[\left(\sum_{i \in I(\mathbf{q})} (\tilde{T}[i] - T[i])\right)^2\right] = \mathbb{E}[|I(\mathbf{q})|] \cdot \text{Var}[\text{Lap}(1/\epsilon)] \leq O\left(\frac{\mathcal{K}^2}{\epsilon^2}\right),$$

as $\tilde{T}[i] = T[i] + Z$ where $Z \sim \text{Lap}(1/\epsilon)$ and each noise is sampled independently. Therefore, by Jensen's inequality $\mathbb{E}[|\widetilde{\text{ans}} - \widehat{\text{ans}}|] \leq \sqrt{\text{Var}[\widetilde{\text{ans}} - \widehat{\text{ans}}]} \leq O\left(\frac{\mathcal{K}}{\epsilon}\right)$ and then by Markov's inequality $|\widetilde{\text{ans}} - \widehat{\text{ans}}| \leq O\left(\frac{\mathcal{K}}{\epsilon}\right)$ holds with constant probability. The claim follows by Lemma

12 and $\varepsilon \leq 1$. The additional $O(m)$ space and pre-preprocessing time is necessary to store and sample m i.i.d. independent Laplace random variables, one for each element of the partition $\mathcal{L} = (L_1, \dots, L_m)$. ◀

B.2 Omitted Proofs in Section 5

► **Lemma 23.** *The probability that a point $\mathbf{x} \in \mathbb{S}^{d-1}$ collides during **CloseTop-1** construction is at least $1 - \frac{1}{m^{\Omega(1)}}$.*

For the proof of Lemma 23 we first need the following technical lemma.

► **Lemma 24.** *Let m be any integer greater than 4. Define $a = \sqrt{2 \log m} - \frac{3}{2} \frac{\log \log m}{\sqrt{2 \log m}}$ and $b = \sqrt{2 \log m}$, then for $Z \sim \mathcal{N}(0, 1)$, we have $\Pr[Z \in (a, b)] \geq \frac{2\sqrt{\pi}}{3} \frac{\log m}{m}$.*

Proof. Using Proposition 22 we may bound $\Pr[Z \geq b] \leq \frac{4\sqrt{\pi}}{3} \frac{1}{m\sqrt{\log m}}$. For the left side of the interval, we first need to check if $a \geq 1$. We have that $\sqrt{2 \log m} - \frac{3}{2} \frac{\log \log m}{\sqrt{2 \log m}} \geq 1$ only if $3 \leq \frac{4 \log m - 2\sqrt{2 \log m}}{\log \log m}$. But $\frac{4 \log m - 2\sqrt{2 \log m}}{\log \log m} > 5$ for any $m \geq 5$. Thus, by applying Proposition 22 we get

$$\begin{aligned} \Pr[Z \geq a] &\geq \frac{2\sqrt{2\pi}}{(1 + \sqrt{5})} \frac{1}{\sqrt{2 \log m} - \frac{3}{2} \frac{\log \log m}{\sqrt{2 \log m}}} \exp \left[-\frac{1}{2} \left(\sqrt{2 \log m} - \frac{3}{2} \frac{\log \log m}{\sqrt{2 \log m}} \right)^2 \right] \\ &\geq \frac{2\sqrt{\pi}}{(1 + \sqrt{5})} \frac{1}{m\sqrt{\log m}} \log^{3/2} m \exp \left[-\frac{9}{16} \frac{(\log \log m)^2}{\log m} \right] \geq \frac{4\sqrt{\pi}}{3} \frac{\log m}{m}. \end{aligned}$$

Where the last inequality holds if $\frac{9}{16} \frac{(\log \log m)^2}{\log m} \leq \log(2(1 + \sqrt{5})/3)$. The right-hand side is greater than $1/2$, thus, it is sufficient to check $\frac{(\log \log m)^2}{\log m} \leq 8/9$. For $m \geq 5$ the left-hand side is always smaller⁸ than $1/3$, thus, the inequality is satisfied. Putting these two bounds together, we conclude

$$\Pr[Z \in (a, b)] = \Pr[Z \geq a] - \Pr[Z \geq b] \geq \frac{4\sqrt{\pi}}{3} \frac{\log m}{m} \left(1 - \frac{1}{(\log m)^{3/2}} \right) \geq \frac{2\sqrt{\pi}}{3} \frac{\log m}{m}.$$

The last inequality follows from $(\log m)^{3/2} \geq (\log 5)^{3/2} \geq 2$. ◀

Proof of Lemma 23. If the probability that a random vector succeeds in the assignment is p , then a point will not collide with probability $(1 - p)^m$. Then for $p = \Omega(\frac{\log m}{m})$ (from Proposition 24) the probability to not collide is at most $(1 - p)^m \leq e^{-pm} = \frac{1}{m^{\Omega(1)}}$. ◀

Proof of Lemma 15. The probability to not find a close point $\mathbf{x}_\alpha$ is

$$\begin{aligned} \Pr[Q_{\mathbf{x}_\alpha} \leq \eta] &= \Pr_{Z \sim \mathcal{N}(0, 1 - \alpha^2)} [Z \leq \eta - \alpha X_{\mathbf{x}_\alpha}] \\ &\leq \Pr_{Z \sim \mathcal{N}(0, 1 - \alpha^2)} \left[Z \leq -\sqrt{2(1 - \alpha^2) \log \log m} \left(1 - \frac{3}{4} \sqrt{\frac{\alpha^2}{1 - \alpha^2} \frac{\log \log m}{\log m}} \right) \right] \\ &\leq \Pr \left[Z \leq -\sqrt{2(1 - \alpha^2) \log \log m} \left(1 - O \left(\sqrt{\frac{\log \log n}{\log n}} \right) \right) \right] \\ &\leq \log m^{-1} (\log m)^{O(\sqrt{\log \log n / \log n})} \leq O(\log^{-1} m), \end{aligned} \tag{4}$$

⁸ The maximum is reached at $m = 11$.

where in the first equality we used Lemma 4, in the second inequality we use the fact that $X_{\mathbf{x}} \geq \sqrt{2 \log m} - \frac{3}{2} \frac{\log \log m}{\sqrt{2 \log m}}$ by construction, in the third inequality $\frac{\log \log m}{\log m} = O((1 - \alpha^2) \frac{\log \log n}{\log n})$ for $m = n^{\frac{\theta}{1 - \alpha^2}}$, and lastly $\lim_{n \rightarrow \infty} (\log n)^{O(\sqrt{\log \log n / \log n})} = 1$. The probability to find a close point is the probability of the joint event $[Q_{\mathbf{x}_\alpha} \geq \eta]$ and $\mathbf{x}_\alpha$ is stored in the data structure. Thus, by Lemma 23, we have that $\Pr[\text{find } \mathbf{x}_\alpha] \geq (1 - O(\log^{-1} m))(1 - m^{-\Omega(1)}) = 1 - o(1)$. This proves Lemma 7 for **CloseTop-1**. The probability to inspect a far point $\mathbf{x}_\beta$ is

$$\begin{aligned} \Pr[Q_{\mathbf{x}_\beta} \geq \eta] &= \Pr_{Z \sim \mathcal{N}(0, 1 - \beta^2)}[Z \geq \eta - \beta X_{\mathbf{x}_\beta}] \\ &\leq \Pr_{Z \sim \mathcal{N}(0, 1 - \beta^2)}[Z \geq (\alpha - \beta) \sqrt{2 \log m} - \sqrt{2(1 - \alpha^2) \log \log m}] \end{aligned} \quad (5)$$

where in the first equality we used Lemma 4, while in the following inequality we used the fact that $X_{\mathbf{x}} \leq \sqrt{2 \log m}$ by construction. The analysis then follows the same step of Lemma 8. As the analysis of the expected number of buckets to inspect is the same, Lemma 8 holds under the same assumption. $\blacktriangleleft$

Proof of Lemma 17. Let's consider one data structure $\mathcal{D}_i$, due to Equation 4 we have an upper bound of $O(\frac{t}{\log m})$ to not find a close point in $\mathcal{D}_i$. By applying a union bound over t data structures we have that

$$\Pr\left[\bigcup_{i=1}^t \{\text{not find } \mathbf{x}_\alpha \text{ in } \mathcal{D}_i\}\right] \leq O\left(\frac{t^2}{\log m}\right) = O\left(\frac{1}{(1 - \alpha^2) \log^{3/4} n}\right) = o(1),$$

where we used $m = n^{\frac{\theta}{1 - \alpha^2}}$, and $(1 - \alpha^2) = \omega(\log^{-3/4} n)$. We now study the probability to not store a point. Due to Lemma 23 the probability to not store a point in $\mathcal{D}_i$ is $\frac{1}{m^{\Omega(1/t)}}$, then by a union bound we have

$$\Pr\left[\bigcup_{i=1}^t \{\mathbf{x} \text{ is not stored in } \mathcal{D}_i\}\right] \leq \frac{t}{m^{\Omega(1/t)}} = o(1) \underbrace{\log^{7/8} n \cdot e^{-\Omega(\log^{7/8} n)}}_{=o(1)} = o(1),$$

as $t = o(\log^{7/8} n)$ for $1 - \alpha^2 = \omega(\log^{-3/4} n)$, and $m^{\Omega(1/t)} = n^{\Omega(\log^{-1/8} n)} = e^{\Omega(\log^{7/8} n)}$ for $m = n^{\frac{\theta}{1 - \alpha^2}}$. Therefore, a close point is found with at least $1 - o(1)$ probability. $\blacktriangleleft$

Proof of Lemma 18. Consider one **CloseTop-1** data structure $\mathcal{D}_i$ with $\tilde{m} = n^{\frac{1}{t} \frac{\theta}{1 - \alpha^2}} = n^{\frac{\theta}{\log^{1/8} n}}$ random vectors. Thus $\frac{\log \log \tilde{m}}{\log \tilde{m}} = O(\frac{\log \log n}{\log^{7/8} n})$, so that the threshold may be written as $\eta \geq \alpha \sqrt{2 \log \tilde{m}} (1 - \frac{\sqrt{1 - \alpha^2}}{\alpha} O(\sqrt{\frac{\log \log n}{\log^{7/8} n}}))$, which is positive for $\alpha \geq \alpha - \beta = \Omega(\sqrt{\frac{\log \log n}{\log^{7/8} n}})$. Therefore, by following the same computation of Lemma 8 (Equation 1), the expected number of buckets to inspect in $\mathcal{D}_i$ is at most $\tilde{m}^{1 - \alpha^2 + \sqrt{1 - \alpha^2} o(1)} = n^{\frac{1}{t} (\theta + \frac{1}{\sqrt{1 - \alpha^2}} o(1))}$. By assumption, we have $1/\sqrt{1 - \alpha^2} = o(\sqrt{\log^{3/4} n})$, thus $\frac{1}{\sqrt{1 - \alpha^2}} O(\sqrt{\frac{\log \log n}{\log^{7/8} n}}) = o(\sqrt{\frac{\log \log n}{\log^{1/8} n}}) = o(1)$. By tensorization of t independent data structures, we conclude that the expected number of buckets is at most $n^{\theta + o(1)}$.

Analogously, starting from the computation in Lemma 15 (Equation 5) and substituting m with $\tilde{m}$, we may lower bound the threshold with $(\alpha - \beta) \sqrt{2 \log \tilde{m}} (1 - \sqrt{\frac{1 - \alpha^2}{(\alpha - \beta)^2}} O(\sqrt{\frac{\log \log n}{\log^{7/8} n}}))$, which is positive for $(\alpha - \beta) = \Omega(\sqrt{\frac{\log \log n}{\log^{7/8} n}})$. Thus, by following the computation in Lemma 8 (Equation 2), the probability to find a far point is at most $\tilde{m}^{-\frac{(\alpha - \beta)^2}{1 - \beta^2} + \frac{\sqrt{1 - \alpha^2}(\alpha - \beta)}{1 - \beta^2} o(1)}$. The

probability to find a far point in all the t independent data structures is at most

$$\begin{aligned} \Pr \left[\bigcap_{i=1}^t \{ \mathbf{x}_\beta \text{ is found in } \mathcal{D}_i \} \right] &\leq \tilde{m} \left(-\frac{(\alpha-\beta)^2}{1-\beta^2} + \frac{\sqrt{1-\alpha^2}(\alpha-\beta)}{1-\beta^2} o(1) \right) \\ &= n^{-\theta \frac{(\alpha-\beta)^2}{(1-\alpha^2)(1-\beta^2)} + \frac{(\alpha-\beta)}{\sqrt{(1-\alpha^2)(1-\beta^2)}} o(1)}. \end{aligned}$$

The last addend is still $o(1)$ as $\frac{1}{\sqrt{1-\alpha^2}} O\left(\sqrt{\frac{\log \log n}{\log^{7/8} n}}\right) = o(1)$ and $\frac{\alpha-\beta}{1-\beta^2} = O(1)$. Thus, as there are at most n far points, the expected number of far points that are inspected is $n^{1-\theta \frac{(\alpha-\beta)^2}{(1-\alpha^2)(1-\beta^2)}}$. $\blacktriangleleft$

C Data Structures for the Euclidean Space

In this section, we prove that balanced **TensorCloseTop-1** solves the (c, r) -ANN in the Euclidean space and reproduces the results for (c, r) -ANNC in [4]. Due to standard embedding techniques (see Lemma A.1 and Corollary A.1 in [4]), a (c, r) -ANN in $\mathbb{R}^d$ can be mapped into a $(c \cdot \frac{1-\gamma}{1+\gamma}, r(1+\gamma))$ -ANN in $\mathbb{S}^{d'}$, in time $O(d \cdot d')$, with $d' = O(\frac{\log n}{\gamma^2})$, if $(cr)^2 \leq \gamma/2$. Thus, the embedding preserves asymptotically the metric only for small distances $r = o(1)$ in $\mathbb{S}^{d'-1}$, that can be obtained in the original space $\mathbb{R}^d$ after an appropriate scaling. The relation between inner product similarity and Euclidean distance for small distances are

$$(1 - \alpha^2) = \Theta(r^2), \quad (1 - \beta)^2 = \Theta(r^2), \quad (\alpha - \beta) = \Theta(r^2), \quad (1 - \alpha\beta) = \Theta(r^2), \quad (6)$$

and the concatenation factor $t = \frac{\log^{1/8} n}{1-\alpha^2}$ is in $\Theta(\frac{\log^{1/8} n}{r^2})$.

Theorem 25. *For any $r > 0$, constant $c > 1$, and a dataset $\mathcal{S} = \{x_i\}_{i=1,\dots,n}$ in $\mathbb{R}^d$, there exists a data structure that solves with at least $1 - o(1)$ probability the (c, r) -ANN using almost linear space $n^{1+o(1)}$, pre-processing time $d \cdot n^{1+o(1)}$, and query time in expectation at most $d \cdot n^{o(1)} + n^{\rho+o(1)}$ for $\rho = \frac{4c^2}{(c^2+1)^2}$.*

Proof. To apply **TensorCloseTop-1** to the embedded dataset we need to satisfy the assumptions in Theorem 19 which are: (i) $\rho = \frac{(1-\alpha^2)(1-\beta^2)}{(1-\alpha\beta)^2} = O(1)$, (ii) $\alpha - \beta = \Omega\left(\sqrt{\frac{\log \log n}{\log^{7/8} n}}\right)$, and (iii) $1 - \alpha^2 = \omega(\log^{-3/4} n)$. Requirement (i) is satisfied due to the asymptotic Equations 6. More precisely, by substituting $\alpha = 1 - \frac{r^2}{2}$ and $\beta = 1 - \frac{(cr)^2}{2}$ we get

$$\rho = \frac{(1 - \alpha^2)(1 - \beta^2)}{(1 - \alpha\beta)^2} = \frac{c^2(4 - r^2)(c^2r^2 - 4)}{(c^2(r^2 - 2) - 2)^2} = \frac{4c^2}{(c^2 + 1)^2} + O(r^2).$$

Requirement (ii) and (iii) are satisfied for any distance $r = \omega(\log^{-3/8} n)$, due to the asymptotic relations in Equation 6. Therefore, for any $C < 3/8$ by setting $\gamma = \log^{-2C} n$ and $r = \log^{-C} n$ we can scale the dataset $\mathcal{S} \subset \mathbb{R}^d$, apply the standard embedding techniques to get a dataset in $\mathbb{S}^{d'-1}$ with $d' = \log^{O(1)} n$, and invoke **TensorCloseTop-1** to solve the (α, β) -ANN in $\mathbb{S}^{d'}$ by paying an asymptotically small γ factor.⁹ As the mapping can be computed in $O(d \cdot d') = O(d \cdot (\log n)^{O(1)}) = d \cdot n^{o(1)}$ time, the pre-processing time is $d \cdot n^{1+o(1)}$. The space is $O(d' \cdot (n + n^{o(1)})) = n^{1+o(1)}$ and the query time is in expectation at most $d \cdot n^{o(1)} + n^{\rho+o(1)}$, given by the time to embed the query $d \cdot n^{o(1)}$ in the hyper-sphere plus the query time of the data structure $d' \cdot n^{\rho+o(1)} = n^{\rho+o(1)}$. $\blacktriangleleft$

⁹ Andoni et al. [4] set $r = \Theta(\log^{-1/8} n)$ and $\gamma = \Theta(\log^{-1/8} n)$. Our analysis demonstrates more clearly that there is a broader range of possible values.

The Unbalanced TensorCloseTop-1

Unbalanced `TensorCloseTop-1` can be used to solve the Euclidean DP-ANNC problem as well. The proof is the same as Theorem 25, with the distinction that

$$\frac{\sigma}{2} = \frac{(1 - \alpha^2)(1 - \beta^2)}{(1 - \alpha\beta)^2 + (1 - \alpha\beta)} = \frac{2c^2}{1 + c^4} + O(r^2)$$

The space and pre-processing time needed is $n^{\frac{2\sigma}{1-\alpha^2}} = n^{\frac{2\sigma}{\Theta(r^2)}} = n^{\text{polylog}(n)}$.