On Adaptive Distance Estimation

Yeshwanth Cherapanamjeri* Jelani Nelson†

October 22, 2020

Abstract

We provide a static data structure for distance estimation which supports *adaptive* queries. Concretely, given a dataset $X = \{x_i\}_{i=1}^n$ of n points in $\mathbb{R}^d$ and $0 < p \leq 2$, we construct a randomized data structure with low memory consumption and query time which, when later given any query point $q \in \mathbb{R}^d$, outputs a $(1+\varepsilon)$-approximation of $\|q - x_i\|_p$ with high probability for all $i \in [n]$. The main novelty is our data structure's correctness guarantee holds even when the sequence of queries can be chosen adaptively: an adversary is allowed to choose the jth query point q_j in a way that depends on the answers reported by the data structure for $q_1, \ldots, q_{j-1}$. Previous randomized Monte Carlo methods do not provide error guarantees in the setting of adaptively chosen queries [JL84, Ind06, TZ12, IW18]. Our memory consumption is $\tilde{O}((n+d)d/\varepsilon^2)$, slightly more than the $O(nd)$ required to store X in memory explicitly, but with the benefit that our time to answer queries is only $\tilde{O}(\varepsilon^{-2}(n+d))$, much faster than the naive $\Theta(nd)$ time obtained from a linear scan in the case of n and d very large. Here $\tilde{O}$ hides $\log(nd/\varepsilon)$ factors. We discuss applications to nearest neighbor search and nonparametric estimation.

Our method is simple and likely to be applicable to other domains: we describe a generic approach for transforming randomized Monte Carlo data structures which do not support adaptive queries to ones that do, and show that for the problem at hand, it can be applied to standard nonadaptive solutions to ℓ_p norm estimation with negligible overhead in query time and a factor d overhead in memory.

1 Introduction

In recent years, much research attention has been directed towards understanding the performance of machine learning algorithms in adaptive or adversarial environments. In diverse application domains ranging from malware and network intrusion detection [BCM+17, CBK09] to strategic classification [HMPW16] to autonomous navigation [PMG16, LCLS17, PMG+17], the vulnerability of machine learning algorithms to malicious manipulation of input data has been well documented. Motivated by such considerations, we study the problem of designing efficient data structures for distance estimation, a basic primitive in algorithms for nonparametric estimation and exploratory data analysis, in the adaptive setting where the sequence of queries made to the data structure may be adversarially chosen. Concretely, the distance estimation problem is defined as follows:

Problem 1.1 (Approximate Distance Estimation (ADE))**.** For a known norm $\|\cdot\|$ we are given a set of vectors $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$ and an accuracy parameter $\varepsilon \in (0,1)$, and we must produce some data structure $\mathcal{D}$. Then later, given only $\mathcal{D}$ stored in memory with no direct access to X, we must respond to queries that specify $q \in \mathbb{R}^d$ by reporting distance estimates $\tilde{d}_1, \ldots, \tilde{d}_n$ satisfying

$$\forall i \in [n],\ (1-\varepsilon)\|q - x_i\| \leq \tilde{d}_i \leq (1+\varepsilon)\|q - x_i\|.$$

*UC Berkeley. yeshwanth@berkeley.edu. Supported by a Microsoft Research BAIR Commons Research Grant

†UC Berkeley. minilek@berkeley.edu. Supported by NSF award CCF-1951384, ONR grant N00014-18-1-2562, ONR DORECG award N00014-17-1-2127, and a Google Faculty Research Award.

The quantities we wish to minimize in a solution to ADE are (1) pre-processing time (the time to compute $\mathcal{D}$ given X), (2) memory required to store $\mathcal{D}$ (referred to as "space complexity"), and (3) query time (the time required to answer a single query). The trivial solution is to, of course, simply store X in memory explicitly, in which case pre-processing time is zero, the required memory is $O(nd)$, and the query time is $O(nd)$ (assuming the norm can be computed in linear time, which is the case for the norms we focus on in this work).

Standard solutions to ADE are via *randomized linear sketching*: one picks a random "sketching matrix" $\Pi \in \mathbb{R}^{m \times d}$ for some $m \ll d$ and stores $y_i = \Pi x_i$ in memory for each i. Then to answer a query, q, we return some estimator applied to $\Pi(q - x_i) = \Pi q - y_i$. Specifically in the case of ℓ_2, one can use the Johnson-Lindenstrauss lemma [JL84], AMS sketch [AMS99], or CountSketch [CCF04, TZ12]. For ℓ_p norms $0 < p < 2$, one can use Indyk's p-stable sketch [Ind06] or that of [KNPW11]. Each of these works specifies some distribution over such Π, together with an estimation procedure. All these solutions have the advantage that $m = \tilde{O}(1/\varepsilon^2)$, so that the space complexity of storing $y_1, \ldots, y_n$ would only be $\tilde{O}(n/\varepsilon^2)$ instead of $O(nd)$. The runtimes for computing Πx for a given x range from $O(d/\varepsilon^2)$ to $O(d)$ for ℓ_2 and from $O(d/\varepsilon^2)$ to $\tilde{O}(d)$ for ℓ_p ([KNPW11]). However, estimating $\|q - x_i\|_p$ from $\Pi q - y_i$ takes $\tilde{O}(n/\varepsilon^2)$ time in all cases. Notably, the recent work of Indyk and Wagner [IW18] is not based on linear sketching and attains the optimal space complexity in bits required to solve ADE in Euclidean space, up to an $O(\log(1/\varepsilon))$ factor.

One downside of all the prior work mentioned in the previous paragraph is that they give Monte Carlo randomized guarantees that do *not* support adaptive queries, i.e. the ability to choose a query vector based on responses by the data structure given to previous queries. Specifically, all these data structures provide a guarantee of the form

$$\forall q \in \mathbb{R}^d,\ \mathbb{P}_s(\text{data structure correctly responds to } q) \geq 1 - \delta,$$

where s is some random "seed", i.e. a random string, used to construct the data structure (for linear sketches specifically, s specifies Π). The main point is that q is not allowed to depend on s; q is first fixed, then s is drawn independently. Thus, in a setting in which we want to support a potentially adversarial sequence of queries $q_1, q_2, \ldots$, where q_j may depend on the data structure's responses to $q_1, \ldots, q_{j-1}$, the above methods do not provide any error guarantees since responses to previous queries are correlated with s. Thus, if q_j is a function of those responses, it, in turn, is correlated with s. In fact, far from being a technical inconvenience, explicit attacks exploiting such correlations were constructed against all approaches based on linear sketching ([HW13]), rendering them open to exploitation in the adversarial scenario. We present our results in the above context:

Our Main Contribution. We provide a new data structure for ADE in the adaptive setting, for ℓ_p norms ($0 < p \leq 2$) with memory consumption $\tilde{O}((n+d)d/\varepsilon^2)$, slightly more than the $O(nd)$ required to store X in memory explicitly, but with the benefit that our query time is only $\tilde{O}(\varepsilon^{-2}(n+d))$ as opposed to the $O(nd)$ query time of the trivial algorithm. The pre-processing time is $\tilde{O}(nd^2/\varepsilon^2)$. Our solution is randomized and succeeds with probability $1 - 1/\,poly(n)$ for each query. Unlike the previous work discussed, the error guarantees hold even in the face of adaptive queries.

In the case of Euclidean space ($p = 2$), we are able to provide sharper bounds with fewer logarithmic factors. Our formal theorem statements appear later as Theorems 4.1 and B.1. Consider for example the setting where ε is a small constant, like 0.1 and $n > d$. Then, the query time of our algorithm is optimal up to logarithmic factors; indeed just reading the input then writing the output of the distance estimates takes time $\Omega(n+d)$. Secondly, a straightforward encoding argument implies that any such approach must have space complexity at least $\Omega(nd)$ bits (see Section C) which means that our space complexity is nearly optimal as well. Finally, pre-processing time for the data structure can be improved by using fast algorithms for rectangular matrix multiplication (See Section 4 for further discussion).

1.1 Related Work

As previously discussed, there has been growing interest in understanding risks posed by the deployment of algorithms in potentially adversarial settings ([BCM+17, HMPW16, GSS15, YHZL19, LCLS17, PMG16]). In addition, the problem of preserving statistical validity in exploratory data analysis has been well explored [DFH+15a, BNS+16, DFH+15b, DFH+15c, DSSU17] where the goal is to maintain coherence with an unknown distribution from which one obtains data samples. There has also been previous work studying linear sketches in adversarial scenarios quite different from those appearing here ([MNS11, GHR+12, GHS+12]).

Specifically on data structures, it is, of course, the case that deterministic data structures provide correctness guarantees for adaptive queries automatically, though we are unaware of any non-trivial deterministic solutions for ADE. For the specific application of approximate nearest neighbor, the works of [Kle97, KOR00] provide non-trivial data structures supporting adaptive queries; a comparison with our results is given in Subsection 1.2. In the context of streaming algorithms (i.e. sublinear memory), the very recent work of Ben-Eliezer et al. [BEJWY20] considers streaming algorithms with both adaptive queries *and* updates. One key difference is they considered the insertion-only model of streaming, which does not allow one to model computing some function of the difference of two vectors (e.g. the norm of $q - x_i$).

1.2 More on applications

Nearest neighbor search: Obtaining efficient algorithms for Nearest Neighbor Search (NNS) has been a topic of intense research effort over the last 20 years, motivated by diverse applications spanning computer vision, information retrieval and database search [BM01, SDI08, DIIM04]. While fast algorithms for *exact* NNS have impractical space complexities ([Cla88, Mei93]), a line of work, starting with the foundational results of [IM98, KOR00], have resulted in query times sub-linear in n for the approximate variant. Formally, the Approximate Nearest Neighbor Problem (ANN) is defined as follows:

Problem 1.2 (Approximate Nearest Neighbor)**.** Given $X = \{x_i\}_{i\in[n]} \subset \mathbb{R}^d$, norm $\|\cdot\|$, and approximation factor $c > 1$, create a data structure $\mathcal{D}$ such that in the future, for any query point $q \in \mathbb{R}^d$, $\mathcal{D}$ will output some $i \in [n]$ satisfying $\|q - x_i\| \leq c \cdot \min_{j\in[n]} \|q - x_j\|$.

The above definition requires the algorithm to return a point from the dataset whose distance to the query point is close to the distance of the exact nearest neighbor. The Locality Sensitive Hashing (LSH) approach of [IM98] gives a Monte Carlo randomized approach with low memory and query time, but it does not support adaptive queries. There has also been recent interest in obtaining Las Vegas versions of such algorithms [Ahl17, Wei19, Pag18, SW17]. Unfortunately, those works also do not support adaptive queries. More specifically, these Las Vegas algorithms always answer (even adaptive) queries correctly, but their query times are random variables that are guaranteed to be small in expectation only when queries are made non-adaptively.

The algorithms of [KOR00, Kle97] *do* support adaptive queries. However, those algorithms though they have small query time, use large space; [KOR00] uses $\Omega(n^{O(1/\varepsilon^2)})$ space for $c = 1 + \varepsilon$, and [Kle97] uses $\Omega(n^d)$ space. The work of [Kle97] also presents another algorithm with memory and query/pre-processing times similar to our ADE data structure though specifically for Euclidean space. While both of these works provide algorithms with runtimes sublinear in n (at the cost of large space complexity), they are specifically for finding the approximate single nearest neighbor ("1-NN") and do not provide distance estimates to *all* points in the same query time (e.g. if one wanted to find the k approximate nearest neighbors for a k-NN classifier).

Nonparametric estimation: While NNS is a vital algorithmic primitive for some fundamental methods in nonparametric estimation, it is inadequate for others, where a few near neighbors do not suffice or the number of required neighbors is unknown. For example, consider the case of kernel regression where a prediction for a query point, q, is given by $\hat{y} = \sum_{i=1}^n w_i y_i$ where $w_i = K(q,x_i)/\sum_{i=1}^n K(q,x_i)$ for some kernel function K and y_i is

the label for the i^{th} data point. For this and other nonparametric models including SVMs, distance estimates to potentially every point in the dataset may be required [WJ95, HSS08, Alt92, Sim96, AMS97]. Even for simpler tasks like k-nearest neighbor classification or database search, it is often unclear what the right value of k should be and is frequently chosen at test time based on the query point. Unfortunately, modifying previous approaches to return k nearest neighbors instead of 1, results in a factor k increase in query time. Due to the ubiquity of such methods in practical applications, developing efficient versions deployable in adversarial settings is an important endeavor for which an adaptive ADE procedure is a useful primitive.

1.3 Overview of Techniques

Our main idea is quite simple and generic, and thus, we believe it could be widely applicable to a number of other problem domains. In fact, it is so generic that it is most illuminating to explain our approach from the perspective of an arbitrary data structural problem instead of focusing on ADE specifically. Suppose we have a randomized Monte Carlo data structure $\mathcal{D}$ for some data structural problem that supports answering nonadaptive queries from some family $\mathcal{Q}$ of potential queries (in the case of ADE, $\mathcal{Q} = \mathbb{R}^d$, so that the allowed queries are the set of all $q \in \mathbb{R}^d$). Suppose further that l independent instantiations of $\mathcal{D}$, $\{\mathcal{D}_i\}_{i=1}^{l}$, satisfy the following:

$$\forall q \in \mathcal{Q} : \sum_{i=1}^{l} \mathbf{1}\left\{\mathcal{D}_i \text{ answers } q \text{ correctly}\right\} \geq 0.9l \tag{Rep}$$

with high probability. Since the above holds *for all* $q \in \mathcal{Q}$, it is true even for any query in an adaptive sequence of queries. The Chernoff bound implies that to answer a query successfully with probability $1 - \delta$, one can sample $r = \Theta(\log(1/\delta))$ indices $i_1, \ldots, i_r \in [l]$, query each $\mathcal{D}_{i_j}$ for $j \in [r]$, then return the majority vote (or e.g. median if the answer is numerical and correctness guarantees are approximate). Note that the runtime of this procedure is at most r times the runtime of an individual $\mathcal{D}_i$ and this constitutes the main benefit of the approach: during queries not all copies of $\mathcal{D}$ must be queried, but only a random *sample*. Now, defining for any data structure, $\mathcal{D}$:

$$l^*(\mathcal{D}, \delta) := \inf\{l > 0 : \text{Rep holds for } \mathcal{D} \text{ with probability at least } 1 - \delta\},$$

the argument in the preceding paragraph now yields the following general theorem:

Theorem 1.3. *Let $\mathcal{D}$ be a randomized Monte Carlo data structure over a query space $\mathcal{Q}$, and $\delta \in (0, 1/2)$ be given. If $l^*(\mathcal{D}, \delta)$ is finite, then there exists a data structure $\mathcal{D}'$, which correctly answers any $q \in \mathcal{Q}$, even in a sequence of adaptively chosen queries, with probability at least $1 - 2\delta$. Furthermore, the query time of $\mathcal{D}'$ is at most $O(\log 1/\delta \cdot t_q)$ where t_q is the query time of $\mathcal{D}$ and its space complexity and pre-processing time are at most $l^*(\mathcal{D}, \delta)$ times those of $\mathcal{D}$.*

In the context of ADE, the underlying data structures will be randomized linear sketches and the data structural problem we require them to solve is length estimation; that is, given a vector v, we require that at least $0.9l$ of the linear sketches accurately represent its length (See Section 4). In our applications, we select a random sample of $r = \Theta(\log n/\delta)$ linear sketches, use them obtain r estimates of $\|q - x_i\|_p$ and aggregate these estimates by computing their median. The argument in the previous paragraph along with a union bound shows that this strategy succeeds in returning accurate estimates of $\|q - x_i\|_p$ with probability at least $1 - \delta$.

The main impediment to deploying Theorem 1.3 in a general domain is obtaining a reasonable bound on l such that Rep holds with high probability. In the case that $\mathcal{Q}$ is finite, the Chernoff bound implies the upper bound $l^*(\mathcal{D}, \delta) = O(\log(|\mathcal{Q}|/\delta))$. However, since $\mathcal{Q} = \mathbb{R}^d$ in our context, this is nonsensical. Nevertheless, we show that a bound of $l = \tilde{O}(d)$ suffices for the ADE problem for ℓ_p norms with $0 < p \leq 2$ and can be tightened to $O(d)$ for the Euclidean case. We believe that reasonable bounds on l establishing Rep can be obtained for a number of other applications yielding a generic procedure for constructing adaptive data structures from nonadaptive ones in all these scenarios. Indeed, the vast literature on Empirical Process Theory yields bounds of precisely this nature which we exploit as a special case in our setting.

Organization: For the remainder for the paper, we review preliminary material and introduce necessary notation in Section 2, formally present our algorithms in Section 3 and analyze their run time and space complexities in Section 4 before concluding our discussion in Section 6.

2 Preliminaries and Notation

We use d to represent dimension, and n is the number of data points. For a natural number k, $[k]$ denotes $\{1, \ldots, k\}$. For $0 < p \leq 2$, we will use $\|v\|_p = (\sum_{i=1}^{d} |v_i|^p)^{1/p}$ to denote the ℓ_p "norm" of v (For $p < 1$, this is technically not a norm). For a matrix $M \in \mathbb{R}^{l \times m}$, $\|M\|_F$ denotes the Frobenius norm of M: $(\sum_{i,j} M_{i,j}^2)^{1/2}$. Henceforth, when the norm is not specified, $\|v\|$ and $\|M\|$ denote the standard Euclidean ($p = 2$) and spectral norms of v and M respectively. For a vector $v \in \mathbb{R}^d$ and real valued random variable Y, we will abuse notation and use $\text{Median}(v)$ and $\text{Median}(Y)$ to denote the median of the entries of v and the distribution of Y respectively. For a probabilistic event, $\mathcal{E}$, we use $\mathbf{1}\{\mathcal{E}\}$ to denote the indicator random variable for $\mathcal{E}$. Finally, we will use $\mathbb{S}_p^d = \{x \in \mathbb{R}^d : \|x\|_p = 1\}$

One of our algorithms makes use of the p-stable sketch of Indyk [Ind06]. Recall the following concerning p-stable distributions:

Definition 2.1. [Zol86, Nol18] For $p \in (0, 2]$, there exists a probability distribution, $\text{Stab}(p)$, called the p-stable distribution with $\mathbb{E}[e^{-itZ}] = e^{-|t|^p}$ for $Z \sim \text{Stab}(p)$. Furthermore, for any n, vector $v \in \mathbb{R}^n$ and $Z_1, \ldots Z_n$ are iid samples from $\text{Stab}(p)$, we have $\sum_{i=1}^{n} v_i Z_i \sim \|v\|_p Z$ with $Z \sim \text{Stab}(p)$.

Note the Gaussian distribution is 2-stable, and hence, these distributions can be seen as generalizing the stable properties of a gaussian distribution for norms other than Euclidean. These distributions have found applications in steaming algorithms and approximate nearest neighbor search [Ind06, DIIM04] and moreover, it is possible to efficiently obtain samples from them [CMS76]. We will use Med_p to denote $\text{Median}(|Z|)$ where $Z \sim \text{Stab}(p)$. Finally, we will use $\mathcal{N}(0, \sigma^2)$ to denote the distribution function of a normal random variable with mean 0 and variance σ^2.

3 Algorithms

As previously mentioned, our construction combines the approach outlined in Subsection 1.3 with known linear sketches [JL84, Ind06] and a net argument. Both Theorem Theorems 4.1 and B.1 are proven using this recipe, with the only differences being a swap in the underlying data structure (or linear sketch) being used and the argument used to establish a bound on l satisfying Rep (discussed in Section 1.3). Concretely, in our solutions to ADE we pick $l = \tilde{O}(d)$ linear sketch matrices $\Pi_1, \ldots, \Pi_l \in \mathbb{R}^{m \times d}$ for $m = O(1/\varepsilon^2)$. The data structure stores $\Pi_i x_j$ for all $i \in [l], j \in [n]$. Then to answer a query $q \in \mathbb{R}^d$:

1. Select a set of $r = O(\log n)$ indices $j_k \in [l]$ uniformly at random, with replacement
2. For each $i \in [n], k \in [r]$, obtain distance estimates $\tilde{d}_{i,k}$ based on $\Pi_{j_k} x_i$ (stored) and $\Pi_{j_k} q$. These estimates are obtained from the underlying sketching algorithm (for ℓ_2 it is $\|\Pi_{j_k} q - \Pi_{j_k} x_i\|_2$ [JL84], and for ℓ_p for $0 < p < 2$ it is $\text{Median}(|\Pi_{j_k} q - \Pi_{j_k} x_i|)/\text{Med}_p$ [Ind06] where $|\cdot|$ for a vector denotes entry-wise absolute value.
3. Return distance estimates $\tilde{d}_1, \ldots, \tilde{d}_n$ with $\tilde{d}_i = \text{Median}(\tilde{d}_{i,1}, \ldots, \tilde{d}_{i,r})$

As seen above, the only difference between our algorithms for the Euclidean and ℓ_p norm cases are the distributions used for the Π_j, as well as the method for distance estimation in Step 2. Since the algorithms are quite similar (though the analysis for the Euclidean case is sharpened to remove logarithmic factors), we discuss the case of ℓ_p ($0 < p < 2$) in this section and defer our special treatment of the Euclidean case to Appendix B.

Algorithm 1 constructs the linear embeddings for the case ℓ_p norms, $0 < p < 2$, using [Ind06]. The algorithm takes as input the dataset X, an accuracy parameter ε and failure probability δ, and it constructs a data structure containing the embedding matrices and the embeddings $\Pi_j x_i$ of the points in the dataset. The linear embeddings are subsequently used in Algorithm 2 to answer queries.

Algorithm 1 Compute Data Structure (ℓ_p space for $0 < p < 2$, based on [Ind06])

Input: $X = \{x_i \in \mathbb{R}^d\}_{i=1}^n$, Accuracy $\varepsilon \in (0,1)$, Failure Probability $\delta \in (0,1)$
$l \leftarrow O\left((d + \log 1/\delta) \log d/\varepsilon\right)$
$m \leftarrow O\left(\frac{1}{\varepsilon^2}\right)$
For $j \in [l]$, let $\Pi_j \in \mathbb{R}^{m \times d}$ with entries drawn iid from Stab(p)
Output: $\mathcal{D} = \{\Pi_j, \{\Pi_j x_i\}_{i=1}^n\}_{j \in [l]}$

Algorithm 2 Process Query (ℓ_p space for $0 < p < 2$, based on [Ind06])

Input: Query q, Data Structure $\mathcal{D} = \{\Pi_j, \{\Pi_j x_i\}_{i=1}^n\}_{j \in [l]}$, Failure Probability $\delta \in (0,1)$
$r \leftarrow O(\log(n/\delta))$
Sample $j_1, \ldots j_r$ iid with replacement from $[l]$
For $i \in [n], k \in [r]$, let $\tilde{d}_{i,k} \leftarrow \frac{\text{Median}(|\Pi_{j_k}(q - x_i)|)}{\text{Med}_p}$
For $i \in [n]$, let $\tilde{d}_i \leftarrow \text{Median}(\{\tilde{d}_{i,k}\}_{k=1}^r)$
Output: $\{\tilde{d}_i\}_{i=1}^n$

4 Analysis

In this section we prove our main theorem for ℓ_p spaces, $0 < p < 2$. We then prove Theorem B.1 for the Euclidean case in Appendix B.

Theorem 4.1. *For any $0 < \delta < 1$ and any $0 < p < 2$, there is a data structure for the ADE problem in ℓ_p space that succeeds on any query with probability at least $1 - \delta$, even in a sequence of adaptively chosen queries. Furthermore, the time taken by the data structure to process each query is $\tilde{O}\left(\varepsilon^{-2}(n+d)\log 1/\delta\right)$, the space complexity is $\tilde{O}\left(\varepsilon^{-2}(n+d)(d+\log 1/\delta)\right)$, and the pre-processing time is $\tilde{O}\left(\varepsilon^{-2}nd(d+\log 1/\delta)\right)$.*

The data structures, $\mathcal{D}_j$, that we use in our instantiation of the strategy described in Subsection 1.3 are the random linear sketches, Π_j, and the data structural problem they implicitly solve is length estimation; that is, given any vector $v \in \mathbb{R}^d$, at least $0.9l$ of the linear sketches, Π_j, accurately represent its length. To ease exposition, we will now directly reason about the matrices Π_j through the rest of the proof. We start with a theorem of [Ind06]:

Theorem 4.2 ([Ind06])**.** *Let $0 < p < 2$, $\epsilon \in (0,1)$ and m as in Algorithm 1. Then, for $\Pi \in \mathbb{R}^{m \times d}$ with entries drawn iid from Stab(p) and any $v \in \mathbb{R}^d$:*

$$\mathbb{P}\left\{\left(1 - \frac{\varepsilon}{2}\right)\|v\|_p \leq \frac{\textit{Median}(|\Pi v|)}{\textit{Med}_p} \leq \left(1 + \frac{\varepsilon}{2}\right)\|v\|_p\right\} \geq 0.975$$

One can also find an analysis of Theorem 4.2 that only requires Π to be pseudorandom and thus require less memory to store; see the proof of Theorem 2.1 in [KNW10]. We now formalize the Rep requirement on the data structures $\{\Pi_j\}_{j=1}^l$ in our particular context:

Definition 4.3. Given $\varepsilon > 0$ and $0 < p < 2$, we say that a set of matrices $\{\Pi_j \in \mathbb{R}^{m \times d}\}_{j=1}^l$ is (ε, p)-*representative* if:

$$\forall \|x\|_p = 1 : \sum_{j=1}^{l} \mathbf{1}\left\{(1-\varepsilon) \leq \frac{\text{Median}(|\Pi_j x|)}{\text{Med}_p} \leq (1+\varepsilon)\right\} \geq 0.9l.$$

We now show that $\{\Pi_j\}_{j=1}^l$, output by Algorithm 1, are (ε, p)-representative.

Lemma 4.4. *For $p \in (0,2), \varepsilon, \delta \in (0,1)$ and l as in Algorithm 1, the collection $\{\Pi_j\}_{j=1}^l$ output by Algorithm 1 are (ε, p)-representative with probability at least $1 - \delta/2$.*

Proof. We start with the simple lemma:

Lemma 4.5. *Let $0 < p < 2$, $C > 0$ a sufficiently large constant. Suppose $\Pi \in \mathbb{R}^{m \times d}$ is distributed as follows: the Π_{ij} are iid with $\Pi_{ij} \sim Stab(p)$, with m as in Algorithm 1. Then*

$$\mathbb{P}(\|\Pi\|_F \leq C(dm)^{(2+p)/(2p)}) \geq 0.975$$

Proof. From Corollary A.2 a p-stable random variable Z satisfies $\mathbb{P}(|Z| \geq t) = O(t^{-p})$. Thus by the union bound for large enough constant C:

$$\mathbb{P}\left\{\exists i,j : |\Pi_{ij}| \geq C(dm)^{1/p}\right\} \leq \sum_{i,j} \mathbb{P}\left\{|\Pi_{ij}| \geq C(dm)^{1/p}\right\} \leq \frac{1}{40}.$$

Therefore, we have that with probability probability at least 0.975, $\|\Pi\|_F \leq C(dm)^{(2+p)/(2p)}$.

□

Now, let $\theta = C(dm)^{(2+p)/(2p)}$ and define $\mathcal{B} \subset \mathbb{R}^d$ to be a γ-net (Definition A.3) of the unit sphere $\mathbb{S}_p^d$ under ℓ_2 distance, with $\gamma = \Theta(\varepsilon(dm)^{-(2+p)/(2p)})$. Recall that $\mathcal{B}$ being a γ-net under ℓ_2 means for all $x \in \mathbb{S}_p \exists x' \in \mathcal{B}$ *s.t.* $\|x - x'\|_2 \leq \gamma$ and that we may assume $|\mathcal{B}| \leq (3/\gamma)^d$ (Lemma A.7).

By Theorems 4.2 and A.8 and our setting of l, we have that for any $v \in \mathcal{B}$:

$$\mathbb{P}\left\{\sum_{j=1}^l \mathbf{1}\left\{(1-\varepsilon/2)\text{Med}_p \leq \text{Median}(|\Pi_j v|) \leq (1+\varepsilon/2)\text{Med}_p\right\} \geq 0.95l\right\} \geq 1 - \frac{\delta}{4|\mathcal{B}|}.$$

Therefore, the above condition holds for all $v \in \mathcal{B}$ with probability at least $1 - \delta/4$. Also, for a fixed $j \in [l]$ Lemma 4.5 yields $\mathbb{P}\left\{\|\Pi_j\|_F \leq \theta\right\} \geq 0.975$. Thus, by a Chernoff bound, with probability at least $1 - \exp(-\Omega(l)) = 1 - \delta/4$, at least $0.95l$ of the Π_j have $\|\Pi_j\|_F \leq \theta$. Thus by a union bound:

$$\forall v \in \mathcal{B} : \sum_{j=1}^l \mathbf{1}\left\{(1-\varepsilon/2)\text{Med}_p \leq \text{Median}(|\Pi_j v|) \leq (1+\varepsilon/2)\text{Med}_p \cap \|\Pi_j\|_F \leq \theta\right\} \geq 0.9l.$$

with probability at least $1 - \delta/2$. We condition on this event and now, extend from $\mathcal{B}$ to the whole ℓ_p ball. Consider any $\|x\|_p = 1$. From the definition of $\mathcal{B}$, there exists $v \in \mathcal{B}$ such that $\|x - v\|_2 \leq \gamma$. Let $\mathcal{J}$ be defined as:

$$\mathcal{J} = \left\{j : (1-\varepsilon/2)\text{Med}_p \leq \text{Median}(|\Pi_j v|) \leq (1+\varepsilon/2)\text{Med}_p \text{ and } \|\Pi_j\|_F \leq \theta\right\}.$$

From the previous discussion, we have $|\mathcal{J}| \geq 0.9l$. For $j \in \mathcal{J}$:

$$\|\Pi_j x - \Pi_j v\|_\infty \leq \|\Pi_j x - \Pi_j v\|_2 = \|\Pi_j(x-v)\|_2 \leq \|\Pi_j\|_F \|x - v\|_2 \leq \frac{\varepsilon}{2}\text{Med}_p$$

from our definition of γ and the bound on $\|\Pi_j\|_F$. Therefore, we have:

$$|\text{Median}(|\Pi x|) - \text{Median}(|\Pi v|)| \leq \frac{\varepsilon}{2}\text{Med}_p.$$

From this, we may conclude that for all $j \in \mathcal{J}$:

$$(1-\varepsilon)\text{Med}_p \leq \text{Median}(|\Pi_j x|) \leq (1+\varepsilon)\text{Med}_p.$$

Since x is an arbitrary vector in $\mathbb{S}_p^d$ and $|\mathcal{J}| \geq 0.9l$, the statement of the lemma follows. □

We prove the correctness of Algorithm 2 assuming that $\{\Pi_j\}_{j=1}^l$ are (ε, p)-representative.

Lemma 4.6. *Let $0 < \varepsilon$ and $\delta \in (0,1)$. Then, Algorithm 2 when given as input any query point $q \in \mathbb{R}^d$, $\mathcal{D} = \{\Pi_j, \{\Pi_j x_i\}_{i=1}^n\}_{j=1}^l$ where $\{\Pi_j\}_{j=1}^l$ are (ε, p)-representative, ε and δ, outputs distance estimates $\{\tilde{d}_i\}_{i=1}^n$ satisfying:*

$$\forall i \in [n] : (1-\varepsilon)\|q - x_i\|_p \leq \tilde{d}_i \leq (1+\varepsilon)\|q - x_i\|_p$$

with probability at least $1 - \delta/2$.

Proof. Let $i \in [n]$ and $W_k = \mathbf{1}\{\tilde{d}_{i,k} \in [(1-\varepsilon)\|q - x_i\|_p, (1+\varepsilon)\|q - x_i\|_p]\}$. We have from the fact that $\{\Pi_j\}_{j=1}^l$ are (ε, p)-representative and the scale invariance of Definition 4.3 that $\mathbb{E}[W_k] \geq 0.9$. Furthermore, W_k are independent for distinct k. Therefore, we have by Theorem A.8 that with probability at least $1 - \delta/(2n)$, $\sum_{k=1}^r W_k \geq 0.6r$. From the definition of $\tilde{d}_i$, $\tilde{d}_i$ satisfies the desired accuracy requirements when $\sum_{k=1}^r W_k \geq 0.6r$ and hence, with probability at least $1 - \delta/(2n)$. By a union bound over all $i \in [n]$, the conclusion of the lemma follows. □

Finally, we analyze the runtimes of Algorithms 1 and 2 where $\mathsf{MM}(a, b, c)$ is the runtime to multiply an $a \times b$ matrix with a $b \times c$ matrix. Note $\mathsf{MM}(a, b, c) = O(abc)$, but is in fact lower due to the existence of fast rectangular matrix multiplication algorithms [GU18]; since the precise bound depends on a case analysis of the relationship between a, b, and c, we do not simplify the bound beyond simply stating "$\mathsf{MM}(a, b, c)$" since it is orthogonal to our focus.

Lemma 4.7. *The query time of Algorithm 2 is $\tilde{O}((n+d)\log(1/\delta)/\varepsilon^2)$, and for Algorithm 1 the space is $\tilde{O}((n+d)d\log(1/\delta)/\varepsilon^2)$ and pre-processing time is $O(\mathsf{MM}(\varepsilon^{-2}(d+\log(1/\delta))\log(d/\varepsilon), d, n))$ (which is naively $\tilde{O}(nd(d+\log(1/\delta))/\varepsilon^2)$).*

Proof. The space required to store the matrices $\{\Pi_j\}_{j=1}^l$ is $O(mld)$ and the space required to store the projections $\Pi_j x_i$ for all $i \in [n], j \in [l]$ is $O(nml)$. For our settings of m, l, the space complexity of the algorithms follows. The query time follows from the time required to compute $\Pi_{j_k} q$ for $k \in [r]$ with $r = O(\log n/\delta)$, the n median computations in Algorithm 2 and our setting of m. For the pre-processing time, it takes $O(mdl) = \tilde{O}(\varepsilon^{-2}d(d+\log(1/\delta)))$ time to generate all the Π_j. Then we have to multiply $\Pi_j x_i$ for all $j \in [l], i \in [n]$. Naively this would take time $O(nlmd) = \tilde{O}(\varepsilon^{-2}nd(d+\log(1/\delta)))$. This can be improved though using fast matrix multiplication. If we organize the x_i as columns of a $d \times n$ matrix A, and stack the Π_j row-wise to form a matrix $\Pi \in \mathbb{R}^{ml \times d}$, then we wish to compute ΠA, which we can do in $\mathsf{MM}(ml, d, n)$ time. □

Remark 4.8. In the case $p = 2$ one can instead use the CountSketch instead of Indyk's p-stable sketch, which supports multiplying Πx in $O(d)$ time instead of $O(d/\varepsilon^2)$ [CCF04, TZ12]. Thus one could improve the ADE query time in Euclidean space to $\tilde{O}(d + n/\varepsilon^2)$, i.e. the $1/\varepsilon^2$ term need not multiply d. Since for the CountSketch matrix, one has $\|\Pi\|_F \leq \sqrt{d}$ with probability 1, the same argument as above allows one to establish $(\varepsilon, 2)$-representativeness for CountSketch matrices as well. It may also be possible to improve query time similarly for $0 < p < 2$ using [KNPW11], though we do not do so in the present work.

We now assemble our results to prove Theorem 4.1. The proof of Theorem 4.1 follows by using Algorithm 1 to construct our adaptive data structure, $\mathcal{D}$, and Algorithm 2 to answer any query, q. The correctness guarantees follow from Lemmas 4.4 and 4.6 and the runtime and space complexity guarantees follow from Lemma 4.7. This concludes the proof of the theorem.

□

5 Experimental Evaluation

In this section, we provide empirical evidence of the efficacy of our scheme. We have implemented both the vanilla Johnson-Lindenstrauss (JL) approach to distance estimation and our own along with an attack

designed to compromise the correctness of the JL approach. Recall that in the JL approach, one first selects a matrix $\Pi \in \mathbb{R}^{k \times d}$ with $k = \Theta(\varepsilon^{-2} \log n)$ whose entries have been drawn from a sub-gaussian distribution with variance $1/k$. Given a query point q, the distance to x_i is approximated by computing $\|\Pi(q - x_i)\|$. We now describe our evaluation setup starting with the description of the attack.

Our Attack: The attack we describe can be carried out for any database of at least two points; for the sake of simplicity, we describe our attack applied to the database of three points $\{-e_1, 0, e_1\}$ where e_1 is the 1^{st} standard basis vector. Now, consider the set S defined as follows:

$$S := \{x : \|\Pi(x + e_1)\| \leq \|\Pi(x - e_1)\|\} = \{x : \langle x, \Pi\Pi^\top e_1 \rangle \leq 0\}.$$

When Π is drawn from say a gaussian distribution as in the JL-approach, the vector $y = \Pi^\top \Pi e_1$, with high probability, has length $\Omega(\sqrt{d/k})$ while $y_1 \approx 1$. Therefore, when $k \ll d$, the overlap of y with e_1 is small. Conditioned on this high probability event, we sample a sequence of iid random vectors $\{z_i\}_{i=1}^r \sim \mathcal{N}(0, I)$ and compute $z \in \mathbb{R}^d$ defined as:

$$z := \sum_{i=1}^{r} (-1)^{W_i} z_i \text{ where } W_i = \mathbf{1}\left\{\|\Pi(z_i - e_1)\| \leq \|\Pi(z_i + e_1)\|\right\}. \tag{1}$$

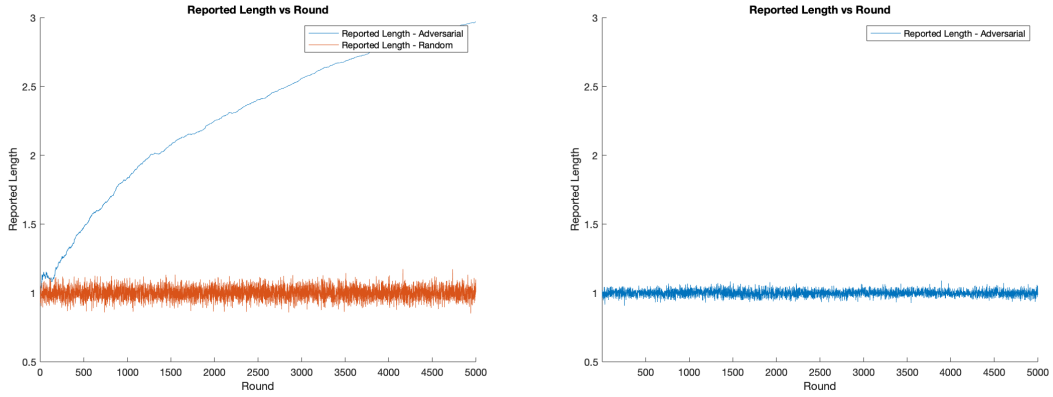
Through simple concentration arguments, z can be shown to be a good approximation of y (in terms of angular distance) and noticing that $\|\Pi y\|$ is $\Omega(d/k)$, we get that $\|\Pi z\| \geq \Omega(\sqrt{d/k})\|z\|$ so that z makes a good adversarial query. Note that the above attack can be implemented solely with access to two points from the dataset and the values $\|\Pi(q - e_1)\|$ and $\|\Pi(q + e_1)\|$. Perhaps even more distressingly, the attack consists of a series of *random* inputs and concludes with a *single* adaptive choice. That is, the JL approach to distance estimation can be broken with a *single* round of adaptivity.

In Figure 1, we illustrate the results of our attack on the JL sketch as well as an implementation of our algorithm when $d = 5000$ and $k = 250$ (for computational reasons, we chose a much smaller value of $l = 200$ to implement our data structure). To observe how the performance of the JL approach degrades with the number of rounds, we plotted the reported length of z as in Eq. (1) for r ranging from 1 to 5000. Furthermore, we compare this to the results that one would obtain if the inputs to the sketch were random points in $\mathbb{R}^d$ as opposed to adversarial ones. From Subfigure 1a, the performance of the JL approach drops drastically as as soon as a few hundred random queries are made which is significantly smaller than the ambient dimension. In contrast, Subfigure 1b shows that our ADE data structure is unaffected by the previously described attack corroborating our theoretical analysis.

6 Conclusion

In this paper, we studied the problem of adaptive distance estimation where one is required to estimate the distance between a sequence of possibly adversarially chosen query points and the points in a dataset. For the aforementioned problem, we devised algorithms for all ℓ_p norms with $0 < p \leq 2$ with nearly optimal query times and whose space complexities are nearly optimal. Prior to our work, the only previous result with comparable guarantees is an algorithm for the Euclidean case which only returns *one* near neighbor [Kle97] and does not estimate all distances. Along the way, we devised a novel framework for building adaptive data structures from non-adaptive ones and leveraged recent results from heavy tailed estimation for one analysis. We now present some open questions:

1. Our construction can be more broadly viewed as a specific instance of *ensemble learning* [Die00]. Starting with the influential work of [Bre96, Bre01], ensemble methods have been a mainstay in practical machine learning techniques. Indeed, the matrices stored in our ensemble have $O(\varepsilon^{-2})$ rows while using a single large matrix would require a model with $O(d)$ rows. Are there other machine learning tasks for which such trade-offs can be quantified?



(a) Johnson-Lindenstrauss Adaptive vs Random

(b) ADE with Adaptive Inputs

Figure 1: Subfigure 1a illustrates the impact of adaptivity on the performance of the JL approach to distance estimation and contrasts its performance to what one would obtain if the inputs were random. In contrast, Subfigure 1b shows that the ADE approach described in this paper is unaffected by the attack described here.

2. The main drawback our results is the time taken to compute the data structure, $O(nd^2)$ (this could be improved using fast rectangular matrix multiplication, but would still be $\omega(nd)$, i.e. superlinear). One question is thus whether nearly linear time pre-processing is possible.

7 Acknowledgements

We would like to thank Sam Hopkins, Sidhanth Mohanty and Nilesh Tripuraneni for helpful comments in the course of this project and in the preparation of this manuscript.

References

[Ahl17] Thomas Dybdahl Ahle. Optimal Las Vegas locality sensitive data structures. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 938–949. IEEE Computer Society, 2017. 3

[AK17] Noga Alon and Bo'az Klartag. Optimal compression of approximate inner products and dimension reduction. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 639–650, 2017. 21

[Alt92] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *Amer. Statist.*, 46(3):175–185, 1992. 4

[AMS97] Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning. *Artif. Intell. Rev.*, 11(1-5):11–73, 1997. 4

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999. 2

[BCM+17] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. *CoRR*, abs/1708.06131, 2017. 1, 3

[BEJWY20] Omri Ben-Eliezer, Rajesh Jayaram, David P. Woodruff, and Eylon Yogev. A framework for adversarially robust streaming algorithms. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, 2020. 3

[BLM13] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities*. Oxford University Press, Oxford, 2013. A nonasymptotic theory of independence, With a foreword by Michel Ledoux. 16

[BM01] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In Doheon Lee, Mario Schkolnick, Foster J. Provost, and Ramakrishnan Srikant, editors, *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001*, pages 245–250. ACM, 2001. 3

[BNS+16] Raef Bassily, Kobbi Nissim, Adam D. Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1046–1059. ACM, 2016. 3

[Bre96] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 1996. 9

[Bre01] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001. 9

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, 2009. 1

[CCF04] Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004. 2, 8

[Cla88] Kenneth L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comput.*, 17(4):830–847, 1988. 3

[CMS76] J. M. Chambers, C. L. Mallows, and B. W. Stuck. A method for simulating stable random variables. *J. Amer. Statist. Assoc.*, 71(354):340–344, 1976. 5

[DFH+15a] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Proceedings of the 28th Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 2350–2358, 2015. 3

[DFH+15b] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015. 3

[DFH+15c] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing (STOC)*, pages 117–126, 2015. 3

[Die00] Thomas G. Dietterich. Ensemble methods in machine learning. In Josef Kittler and Fabio Roli, editors, *Multiple Classifier Systems, First International Workshop, MCS 2000, Cagliari, Italy, June 21-23, 2000, Proceedings*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2000. 9

[DIIM04] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In Jack Snoeyink and Jean-Daniel Boissonnat, editors, *Proceedings of the 20th ACM Symposium on Computational Geometry, Brooklyn, New York, USA, June 8-11, 2004*, pages 253–262. ACM, 2004. 3, 5

[DSSU17] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed! a survey of attacks on private data. *Annual Review of Statistics and Its Application*, 4(1):61–84, 2017. 3

[GHR+12] Anna C. Gilbert, Brett Hemenway, Atri Rudra, Martin J. Strauss, and Mary Wootters. Recovering simple signals. In *2012 Information Theory and Applications Workshop, ITA 2012, San Diego, CA, USA, February 5-10, 2012*, pages 382–391. IEEE, 2012. 3

[GHS+12] Anna C. Gilbert, Brett Hemenway, Martin J. Strauss, David P. Woodruff, and Mary Wootters. Reusable low-error compressive sampling schemes through privacy. In *IEEE Statistical Signal Processing Workshop, SSP 2012, Ann Arbor, MI, USA, August 5-8, 2012*, pages 536–539. IEEE, 2012. 3

[GSS15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3

[GU18] Francois Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the Coppersmith-Winograd tensor. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1029–1046, 2018. 8

[HMPW16] Moritz Hardt, Nimrod Megiddo, Christos H. Papadimitriou, and Mary Wootters. Strategic classification. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 111–122. ACM, 2016. 1, 3

[HSS08] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *Ann. Statist.*, 36(3):1171–1220, 2008. 4

[HW13] Moritz Hardt and David P. Woodruff. How robust are linear sketches to adaptive inputs? In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 121–130. ACM, 2013. 2

[IM98] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 604–613. ACM, 1998. 3

[Ind06] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006. 1, 2, 5, 6

[IW18] Piotr Indyk and Tal Wagner. Approximate nearest neighbors in limited space. In *Proceedings of the Conference On Learning Theory (COLT)*, pages 2012–2036, 2018. 1, 2

[JL84] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemp. Math.*, pages 189–206. Amer. Math. Soc., Providence, RI, 1984. 1, 2, 5, 17

[Kle97] Jon M. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 599–608. ACM, 1997. 3, 9

[KNPW11] Daniel M. Kane, Jelani Nelson, Ely Porat, and David P. Woodruff. Fast moment estimation in data streams in optimal space. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC)*, pages 745–754, 2011. 2, 8

[KNW10] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the exact space complexity of sketching and streaming small norms. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1161–1178, 2010. 6

[KOR00] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM J. Comput.*, 30(2):457–474, 2000. 3

[LCLS17] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1, 3

[LM19] Gábor Lugosi and Shahar Mendelson. Near-optimal mean estimators with respect to general norms. *Probab. Theory Related Fields*, 175(3-4):957–973, 2019. 18, 19

[LT11] Michel Ledoux and Michel Talagrand. *Probability in Banach spaces*. Classics in Mathematics. Springer-Verlag, Berlin, 2011. Isoperimetry and processes, Reprint of the 1991 edition. 16

[Mei93] S. Meiser. Point location in arrangements of hyperplanes. *Inform. and Comput.*, 106(2):286–303, 1993. 3

[MNS11] Ilya Mironov, Moni Naor, and Gil Segev. Sketching in adversarial environments. *SIAM J. Comput.*, 40(6):1845–1870, 2011. 3

[MZ18] Shahar Mendelson and Nikitz Zhivotovskiy. Robust covariance estimation under L_4-L_2 norm equivalence. *arXiv preprint arXiv:1809.10462*, 2018. 18

[Nol18] J. P. Nolan. *Stable Distributions - Models for Heavy Tailed Data*. Birkhauser, Boston, 2018. In progress, Chapter 1 online at http://fs2.american.edu/jpnolan/www/stable/stable.html. 5, 15

[Pag18] Rasmus Pagh. Coveringlsh: Locality-sensitive hashing without false negatives. *ACM Trans. Algorithms*, 14(3):29:1–29:17, 2018. 3

[PMG16] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016. 1, 3

[PMG+17] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Ramesh Karri, Ozgur Sinanoglu, Ahmad-Reza Sadeghi, and Xun Yi, editors, *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pages 506–519. ACM, 2017. 1

[SDI08] Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk. Nearest-neighbor methods in learning and vision. *IEEE Trans. Neural Networks*, 19(2):377, 2008. 3

[Sim96] Jeffrey S. Simonoff. *Smoothing methods in statistics*. Springer Series in Statistics. Springer-Verlag, New York, 1996. 4

[SW17] Piotr Sankowski and Piotr Wygocki. Approximate nearest neighbors search without false negatives for $ł_2$ for $c > \sqrt{\log \log n}$. In Yoshio Okamoto and Takeshi Tokuyama, editors, *28th International Symposium on Algorithms and Computation, ISAAC 2017, December 9-12, 2017, Phuket, Thailand*, volume 92 of *LIPIcs*, pages 63:1–63:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. 3

[TZ12] Mikkel Thorup and Yin Zhang. Tabulation-based 5-independent hashing with applications to linear probing and second moment estimation. *SIAM J. Comput.*, 41(2):293–331, 2012. 1, 2, 8

[Ver18] Roman Vershynin. *High-dimensional probability*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2018. An introduction with applications in data science, With a foreword by Sara van de Geer. 15, 20

[Wei19] Alexander Wei. Optimal las vegas approximate near neighbors in ℓ_p. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1794–1813, 2019. 3

[WJ95] M. P. Wand and M. C. Jones. *Kernel smoothing*, volume 60 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, Ltd., London, 1995. 4

[YHZL19] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Networks Learn. Syst.*, 30(9):2805–2824, 2019. 3

[Zol86] V. M. Zolotarev. *One-dimensional stable distributions*, volume 65 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, RI, 1986. Translated from the Russian by H. H. McFaden, Translation edited by Ben Silver. 5

A Miscellaneous Results and Supporting

A.1 Properties of Stable Distributions

We will use the following property of stable distributions:

Lemma A.1. *[Nol18] For fixed $0 < p < 2$, the probability density function of a p stable distribution is $\Theta(|x|^{-p-1})$ for large $|x|$.*

By integrating the tail bound from the previous result, we get the following simple corollary.

Corollary A.2. *For fixed $0 < p < 2$ and $Z \sim Stab(p)$ and t large:*

$$\mathbb{P}\{|Z| \geq t\} = \Theta(t^{-p}).$$

A.2 Probability and High-dimensional Concentration Tools

We recall here standard definitions in empirical process theory from [Ver18].

Definition A.3 (ε-net [Ver18])**.** Let (T, d) be a metric space, $K \subset T$ and $\varepsilon > 0$. Then, a subset $\mathcal{N} \subset K$ is an ε-net of K if very point in K is within a distance of ε to some point in $\mathcal{N}$. That is:

$$\forall x \in K, \exists y \in \mathcal{N} : d(x, y) \leq \varepsilon.$$

From this, we obtain the definition of a covering number:

Definition A.4 (Covering Number [Ver18])**.** Let (T, d) be a metric space, $K \subset T$ and $\varepsilon > 0$. The smallest possible cardinality of an ε-net of K is called the *covering number* of K and is denoted by $\mathcal{N}(K, d, \varepsilon)$.

In the most general set up, we also recall the definition of a covering number.

Definition A.5 (Packing Number [Ver18])**.** Let (T, d) be a metric space, $K \subset T$ and $\varepsilon > 0$. A subset $\mathcal{P}$ of T is *ε-separated* if for all $x, y \in \mathcal{P}$, we have $d(x, y) > \varepsilon$. The largest possible cardinality of an ε-separated set in K is called the packing number of K and is denoted by $\mathcal{P}(K, d, \varepsilon)$.

We finally recall the following simple fact relating packing and covering numbers.

Lemma A.6 ([Ver18])**.** *Let (T, d) be a metric space, $K \subset T$ and $\varepsilon > 0$. Then:*

$$\mathcal{P}(K, d, 2\varepsilon) \leq \mathcal{N}(K, d, \varepsilon) \leq \mathcal{P}(K, d, \varepsilon).$$

In all our applications, we will take $d(\cdot, \cdot)$ to be the Euclidean distance and the sets K will always be ℓ_p balls for $0 < p \leq 2$. The following lemma follows from a standard volumetric argument.

Lemma A.7. *Let $K = \mathbb{S}_p^d$ for $0 < p \leq 2$ and $0 < \varepsilon \leq 1$. Then, we have:*

$$\mathcal{N}(K, \|\cdot\|_2, \varepsilon) \leq \left(\frac{3}{\varepsilon}\right)^d.$$

Proof. Note from Lemma A.6 that it is sufficient to prove:

$$\mathcal{P}(K, \|\cdot\|_2, \varepsilon) \leq \left(\frac{3}{\varepsilon}\right)^d.$$

Let T be any ε-separated set in K and let $T_\varepsilon = \{x : \exists y \in T, \|x - y\|_2 \leq \varepsilon/2\}$. Note from the triangle inequality and the fact that T is ε-separated, that for any point $x \in T_\varepsilon$, there exists a unique point $y \in T_\varepsilon$ such that $\|x - y\|_2 \leq \varepsilon/2$. Now, for any point $x \in \mathbb{S}_p^d$, we have:

$$\|x\|_2^2 = \sum_{i=1}^{d} |x_i|^2 \leq \sum_{i=1}^{d} |x_i|^p = 1$$

where the inequality follows from the fact that $|x_i| \leq 1$. Therefore, we have $T \subset \mathbb{B}_2(0, 1, d)$ where $\mathbb{B}_2(x, r, d) = \{y \in \mathbb{R}^d : \|y - x\| \leq r\}$. From this, we obtain from the triangle inequality that $T_\varepsilon \subset \mathbb{B}_2(0, 1 + \varepsilon/2, d)$. From the fact that the sets $\mathbb{B}_2(x, \varepsilon/2, d)$ and $\mathbb{B}_2(y, \varepsilon/2, d)$ are disjoint for distinct $x, y \in T$, we have:

$$\text{Vol}\,(T_\varepsilon) = |T|\,\text{Vol}\,(\mathbb{B}_2(0, \varepsilon/2, d)) \leq \text{Vol}\,(\mathbb{B}_2(0, 1 + \varepsilon/2, d)).$$

By dividing both sides and by using that fact that $\text{Vol}\,(\mathbb{B}_2(0, l, d)) = l^d\,\text{Vol}\,(\mathbb{B}_2(0, 1, d))$, we get:

$$|T| \leq \frac{(1 + \frac{\varepsilon}{2})^d}{(2/\varepsilon)^d} = \left(1 + \frac{2}{\varepsilon}\right)^d \leq \left(\frac{3}{\varepsilon}\right)^d$$

as $\varepsilon \leq 1$ and this concludes the proof of the lemma. □

We will also make use of Hoeffding's Inequality:

Theorem A.8. *[BLM13] Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \in [a_i, b_i]$ almost surely for $i \in [n]$ and let $S = \sum_{i=1}^n X_i - \mathbb{E}[X_i]$. Then, for every $t > 0$:*

$$\mathbb{P}\{S \geq t\} \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

We will also require the bounded differences inequality:

Theorem A.9. *[BLM13] Let $\{X_i \in \mathcal{X}\}_{i=1}^n$ be n independent random variables and suppose $f : \mathcal{X}^n \to \mathbb{R}$ satisfies the bounded differences condition with constants $\{c_i\}_{i=1}^n$; i.e f satisfies:*

$$\forall i \in [n] : \sup_{\substack{x_1, \ldots, x_n \in \mathcal{X} \\ x_i' \in \mathcal{X}}} |f(x_1, \ldots, x_n) - f(x_1, \ldots, x_i', \ldots, x_n)| \leq c_i.$$

Then, we have for the random variable $Z = f(X_1, \ldots, X_n)$:

$$\mathbb{P}\{Z - \mathbb{E}[Z] \geq t\} \leq \exp\left(-\frac{t^2}{2v}\right)$$

where $v = \frac{\sum_{i=1}^n c_i^2}{4}$.

We also present the Ledoux-Talagrand Contraction Inequality:

Theorem A.10 ([LT11])**.** *Let $X_1, \ldots, X_n \in \mathcal{X}$ be i.i.d. random vectors, $\mathcal{F}$ be a class of real-valued functions on $\mathcal{X}$ and $\sigma_i, \ldots, \sigma_n$ be independent Rademacher random variables. If $\phi : \mathbb{R} \to \mathbb{R}$ is an L-Lipschitz function with $\phi(0) = 0$, then:*

$$\mathbb{E} \sup_{f \in \mathcal{F}} \sum_{i=1}^n \sigma_i \phi(f(X_i)) \leq 2L \cdot \mathbb{E} \sup_{f \in \mathcal{F}} \sum_{i=1}^n \sigma_i f(X_i).$$

B ADE Data Structure for Euclidean Case

Algorithm 3 Compute Data Structure (Euclidean space, based on [JL84])

Input: Data points $X = \{x_i \in \mathbb{R}^d\}_{i=1}^n$, Accuracy ε, Failure Probability δ
$m \leftarrow O\left(\frac{1}{\varepsilon^2}\right)$, $l \leftarrow O\left((d + \log(1/\delta))\log(d/\varepsilon)\right)$
For $j \in [l]$, let $\Pi_j \in \mathbb{R}^{m \times d}$ be such that each entry is drawn iid from $\mathcal{N}(0, 1/m)$
Output: $\mathcal{D} = \{\Pi_j, \{\Pi_j x_i\}_{i=1}^n\}_{j=1}^l$

Algorithm 4 Process Query (Euclidean space, based on [JL84])

Input: Query Point q, Data Structure $\mathcal{D} = \{\Pi_j, \{\Pi_j x_i\}_{i=1}^n\}_{j=1}^l$, Failure Probability δ
$r \leftarrow O(\log n + \log 1/\delta)$
Sample $j_1, \ldots j_r$ iid with replacement from $[l]$
For $i \in [n], k \in [r]$, let $y_{i,k} \leftarrow \|\Pi_{j_k}(q - x_i)\|$
For $i \in [n]$, let $\tilde{d}_i \leftarrow \text{Median}(\{y_{i,k}\}_{k=1}^r)$
Output: $\{\tilde{d}_i\}_{i=1}^n$

In this section we show that logarithmic factors may be improved in an ADE for Euclidean space specifically. Our main theorem of this section is the following.

Theorem B.1. *For any $0 < \delta < 1$ there is a data structure for the ADE problem in Euclidean space that succeeds on any query with probability at least $1 - \delta$, even in a sequence of adaptively chosen queries. Furthermore, the time taken by the data structure to process each query is $O\left(\varepsilon^{-2}(n + d)\log n/\delta\right)$, the space complexity is $O\left(\varepsilon^{-2}(n + d)(d + \log 1/\delta)\right)$, and the pre-processing time is $O\left(\varepsilon^{-2}nd(d + \log 1/\delta)\right)$.*

In the remainder of this section, we prove Theorem B.1. We start by introducing the formal guarantee required of the matrices, Π_j, returned by Algorithm 3:

Definition B.2. Given $\varepsilon > 0$, we say a set of matrices $\{\Pi_j \in \mathbb{R}^{m \times d}\}_{j=1}^l$ is ε-representative if:

$$\forall \|v\| = 1 : \sum_{j=1}^{l} \mathbf{1}\left\{(1 - \varepsilon) \leq \|\Pi_j v\| \leq (1 + \varepsilon)\right\} \geq 0.9l.$$

Intuitively, the above definition states that for any any vector, v, most of the projections, $\Pi_j v$, approximately preserve its length. In our proofs, we will often instantiate the above definition by setting $v_i = \frac{q - x_i}{\|q - x_i\|}$, for a query point q and a dataset point x_i. As a consequence the above definition, this means that most of the projections $\Pi_j(q - x_i)$ have length approximately $\|q - x_i\|$. By using standard concentration arguments this also holds for the matrices sampled in Algorithm 4 and the correctness of Algorithm 4 follows. The following lemma formalizes this intuition:

Lemma B.3. *Let $\varepsilon > 0$ and $0 < \delta < 1$. Then, Algorithm 4, when given as input query point $q \in \mathbb{R}^d$, $\mathcal{D} = \{\Pi_j, \{\Pi_j x_i\}_{i=1}^n\}_{j=1}^l$ for an ε-representative set of matrices $\{\Pi_j\}_{j=1}^l$, ε and δ outputs a set of estimates $\{\tilde{d}_i\}_{i=1}^n$ satisfying:*

$$\forall i \in [n] : (1 - \varepsilon)\|q - x_i\| \leq \tilde{d}_i \leq (1 + \varepsilon)\|q - x_i\|$$

with probability at least $1 - \delta$. Furthermore, Algorithm 4 runs in time $O\left((n + d)m(\log n + \log 1/\delta)\right)$.

Proof. We will first prove that $\tilde{d}_i$ is a good estimate of $\|q - x_i\|$ with high probability and obtain the guarantee for all $i \in [n]$ by a union bound. Now, let $i \in [n]$. From the definition of $\tilde{d}_i$, we see that the conclusion is

trivially true for the case where $q = x_i$. Therefore, assume that $q \neq x_i$ and let $v = \frac{q-x_i}{\|q-x_i\|}$. From the fact that $\{\Pi_j\}_{j=1}^l$ is ε-representative, the set $\mathcal{J}$, defined as:

$$\mathcal{J} = \left\{j : (1-\varepsilon) \leq \|\Pi_j v\| \leq (1+\varepsilon)\right\}$$

has size at least $0.9l$. We now define the random variables $\tilde{y}_{i,k} = \|\Pi_{j_k} v\|$ and $\tilde{z}_i = \text{Median}\{\tilde{y}_{i,k}\}_{k=1}^r$ with $r, \{j_k\}_{k=1}^r$ defined in Algorithm 4. We see from the definition of $\tilde{d}_i$ that $\tilde{d}_i = \|q - x_i\|\tilde{z}_i$. Therefore, it is necessary and sufficient to bound the probability that $\tilde{z}_i \in [1-\varepsilon, 1+\varepsilon]$. To do this, let $W_k = \mathbf{1}\{j_k \in \mathcal{J}\}$ and $W = \sum_{k=1}^r W_k$. Furthermore, we have $\mathbb{E}[W] \geq 0.9r$ and since $W_k \in \{0,1\}$, we have by Hoeffding's Inequality (Theorem A.8):

$$\mathbb{P}\{W \leq 0.6r\} \leq \exp\left(-\frac{2(0.3r)^2}{r}\right) \leq \frac{\delta}{n}$$

from our definition of r. Furthermore, for all k such that $j_k \in \mathcal{J}$, we have:

$$1 - \varepsilon \leq \tilde{y}_{i,k} \leq 1 + \varepsilon.$$

Therefore, in the event that $W \geq 0.6r$, we have $(1-\varepsilon) \leq \tilde{z}_i \leq (1+\varepsilon)$. Hence, we get:

$$\mathbb{P}\left\{(1-\varepsilon)\|q - x_i\| \leq \tilde{d}_i \leq (1+\varepsilon)\|q - x_i\|\right\} \geq 1 - \frac{\delta}{n}.$$

From the union bound, we obtain:

$$\mathbb{P}\left\{\forall i : (1-\varepsilon)\|q - x_i\| \leq \tilde{d}_i \leq (1+\varepsilon)\|q - x_i\|\right\} \geq 1 - \delta.$$

This concludes the proof of correctness of the output of Algorithm 4. The runtime guarantees follow from the fact that the runtime is dominated by the cost of computing the projections $\Pi_{j_k} v$ and the cost of computing $\{y_{i,k}\}_{i\in[n],k\in[r]}$ which take time $O(dmr)$ and $O(nmr)$ respectively. □

Therefore, the runtime of Algorithm 4, is determined by the dimension of the matrices, Π_j. The subsequent lemma bounds on this quantity as well as the number of matrices, l. In our proof of the following lemma, we use recent techniques developed in the context of heavy-tailed estimation [LM19, MZ18] to obtain sharp bounds on both l and m avoiding extraneous log factors.

Lemma B.4. *Let $0 < \varepsilon, 0 < \delta < 1$ and m, l be defined as in Algorithm 3. Then, the output $\{\Pi_j\}_{j=1}^l$ of Algorithm 3 satisfies:*

$$\forall \|v\| = 1 : \sum_{j=1}^{l} \mathbf{1}\left\{(1-\varepsilon) \leq \|\Pi_j v\| \leq (1+\varepsilon)\right\} \geq 0.9l$$

with probability at least $1 - \delta$. Furthermore, Algorithm 3 runs in time $O(\mathsf{MM}(ml, d, n))$.

Proof. We must show that for any $x \in \mathbb{R}^d$, a large fraction of the Π_j approximately preserve its length. Concretely, we will analyze the following random variable where l, m are defined in Algorithm 3:

$$Z = \max_{\|v\|=1} \sum_{j=1}^{l} \mathbf{1}\left\{\left|\|\Pi_j v\|^2 - 1\right| \geq \varepsilon\right\}.$$

Intuitively, Z searches for a unit vector v whose length is well approximated by the fewest number of sample projection matrices Π_j. We first notice that Z satisfies a bounded differences condition.

Lemma B.5. *Let $k \in [l]$, $\Pi'_k \in \mathbb{R}^{m\times d}$ and Z' be defined as:*

$$Z' = \max_{\|v\|=1} \mathbf{1}\left\{|\|\Pi'_k v\|^2 - 1| \geq \varepsilon\right\} + \sum_{\substack{1\leq j\leq l \\ i\neq k}} \mathbf{1}\left\{|\|\Pi_j v\|^2 - 1| \geq \varepsilon\right\}.$$

Then, we have:

$$|Z - Z'| \leq 1.$$

Proof. Let $Y_j(v) = \mathbf{1}\left\{|\|\Pi_j v\|^2 - 1| \geq \varepsilon\right\}$ and $Y'_k(v) = \mathbf{1}\left\{|\|\Pi'_k v\|^2 - 1| \geq \varepsilon\right\}$. The proof follows from the following manipulation:

$$\begin{aligned} Z - Z' &= \max_{\|v\|=1} \sum_{j=1}^{l} Y_j(v) - \max_{\|v\|=1} Y'_k(v) + \sum_{\substack{1\leq j\leq l \\ i\neq k}} Y_j(v) \\ &\leq \max_{\|v\|=1} \sum_{j=1}^{l} Y_j(v) - Y'_k(v) - \sum_{\substack{1\leq j\leq l \\ i\neq k}} Y_j(v) \\ &= \max_{\|v\|=1} Y_k(v) - Y'_k(v) \leq 1. \end{aligned}$$

Through a similar manipulation, we get $Z' - Z \leq 1$ and this concludes the proof of the lemma. □

As a consequence of Theorem A.9, it now suffices for us to bound the expected value of Z.

Lemma B.6. *We have $\mathbb{E}[Z] \leq 0.05l$.*

Proof. We bound the expected value of Z as follows, using an approach of [LM19] (see the proof of their Theorem 2):

$$\begin{aligned} \mathbb{E}[Z] &\leq \frac{1}{\varepsilon} \cdot \mathbb{E}\left[\max_{\|v\|=1} \sum_{j=1}^{l} |\|\Pi_j v\|^2 - 1|\right] \\ &\leq \frac{1}{\varepsilon} \cdot \left(\mathbb{E}\left[\max_{\|v\|=1} \sum_{j=1}^{l} |\|\Pi_j v\|^2 - 1| - \mathbb{E}|\|\Pi'_j v\|^2 - 1|\right] + l \max_{v} \mathbb{E}\left[|\|\Pi v\|^2 - 1|\right]\right) \end{aligned}$$

where $\{\Pi'_j\}_{j=1}^{l}, \Pi$ are mutually independent and independent of $\{\Pi_j\}_{j=1}^{l}$ with the same distribution. We first bound the second term in the above display. We have for all $\|v\| = 1$:

$$\begin{aligned} \mathbb{E}\left[|\|\Pi v\|^2 - 1|\right] &\leq \sqrt{\mathbb{E}\left[(\|\Pi v\|^2 - 1)^2\right]} = \sqrt{m\mathbb{E}[\sum_{i=1}^{m}(\langle w_i, v\rangle^2 - m^{-1})^2]} \\ &\leq \sqrt{m\mathbb{E}\left[\sum_{i=1}^{m}\langle w_i, v\rangle^4\right]} = \sqrt{\frac{3}{m}}. \end{aligned}$$

where $w_i \sim \mathcal{N}(0, I/m)$ are the rows of the matrix Π. For the first term, we have:

$$
\begin{aligned}
&\mathbb{E}_{\Pi_j}\left[\max_{\|v\|=1}\sum_{j=1}^{l}|\|\Pi_j v\|^2-1|-\mathbb{E}_{\Pi_j'}\left[|\|\Pi_j' v\|^2-1|\right]\right] \\
&\leq \mathbb{E}_{\Pi_j,\Pi_j'}\left[\max_{\|v\|=1}\sum_{j=1}^{l}|\|\Pi_j v\|^2-1|-|\|\Pi_j' v\|^2-1|\right] \\
&= \mathbb{E}_{\Pi_j,\Pi_j',\sigma_j}\left[\max_{\|v\|=1}\sum_{j=1}^{l}\sigma_j\left(|\|\Pi_j v\|^2-1|-|\|\Pi_j' v\|^2-1|\right)\right] && \sigma_j \overset{iid}{\sim} \{\pm 1\} \\
&\leq 2\mathbb{E}_{\Pi_j,\sigma_j}\left[\max_{\|v\|=1}\sum_{j=1}^{l}\sigma_j|\|\Pi_j v\|^2-1|\right] \\
&\leq 4\mathbb{E}_{\Pi_j,\sigma_j}\left[\max_{\|v\|=1}\sum_{j=1}^{l}\sigma_j\left(\|\Pi_j v\|^2-1\right)\right] && \text{Theorem A.10} \\
&= 4\mathbb{E}_{\Pi_j,\sigma_j}\left[\max_{\|v\|=1}\sum_{j=1}^{l}\sigma_j\left(\left(\|\Pi_j v\|^2-1\right)-\mathbb{E}_{\Pi_j'}\left[\|\Pi_j' v\|^2-1\right]\right)\right] \\
&\leq 4\mathbb{E}_{\Pi_j,\Pi_j',\sigma_j}\left[\max_{\|v\|=1}\sum_{j=1}^{l}\sigma_j\left(\left(\|\Pi_j v\|^2-1\right)-\left(\|\Pi_j' v\|^2-1\right)\right)\right] \\
&= 4\mathbb{E}_{\Pi_j,\Pi_j'}\left[\max_{\|v\|=1}\sum_{j=1}^{l}\left(\left(\|\Pi_j v\|^2-1\right)-\left(\|\Pi_j' v\|^2-1\right)\right)\right] \\
&\leq 4\mathbb{E}_{\Pi_j}\left[\max_{\|v\|=1}\sum_{j=1}^{l}\left(\|\Pi_j v\|^2-1\right)\right]+4\mathbb{E}_{\Pi_j'}\left[\max_{\|v\|=1}-\sum_{j=1}^{l}\left(\|\Pi_j' v\|^2-1\right)\right] \\
&\leq 8l\mathbb{E}_{\Pi_j}\left[\left\|\frac{\sum_{j=1}^{l}\Pi_j^\top\Pi_j}{l}-I\right\|\right] \leq \frac{l\varepsilon}{40}
\end{aligned}
$$

where the final inequality follows from the fact that $\frac{\sum_{j=1}^{l}\Pi_j^\top\Pi_j}{l}$ is the empirical covariance matrix of ml standard gaussian vectors and the final result follows from standard results on the concentration of empirical covariance matrices of sub-gaussian random vectors (See, for example, Theorem 4.6.1 from [Ver18]) From the previous two bounds, we conclude the proof of the lemma. □

Now we complete the proof of Lemma B.4. From Lemmas B.5 and B.6 and Theorem A.9, we have with probability at least $1-\delta$:

$$
\forall\|v\|=1:\sum_{j=1}^{l}\mathbf{1}\left\{|\|\Pi_j v\|^2-1|\leq\varepsilon\right\}\geq 0.9l.
$$

Now, condition on the above event. Let $\|v\|=1$ and let $\mathcal{J}=\{j:|\|\Pi_j v\|^2-1|\leq\varepsilon\}$. For $j\in\mathcal{J}$:

$$
1-\varepsilon\leq\|\Pi_j v\|^2\leq 1+\varepsilon \implies 1-\varepsilon\leq\|\Pi_j v\|\leq 1+\varepsilon.
$$

This concludes the proof of correctness of the output of Algorithm 3. The runtime guarantees follow from our setting of m, l and the fact that the runtime is dominated by the time taken to compute $\Pi_j x_i$ for $j\in[l]$ and $i\in[n]$ which can be done by stacking the projection matrices into a single large matrix $\Pi=[\Pi_1^\top\Pi_2^\top\ldots\Pi_l^\top]^\top$ and performing a matrix-matrix multiplication with the matrix containing the data points along the columns. □

Lemmas B.3 and B.4 now imply Theorem B.1. An algorithm satisfying the guarantees of Theorem B.1 follows by first constructing a data structure, $\mathcal{D}$, using Algorithm 3 with failure probability set to $\delta/2$ and accuracy requirement set to ε. Each query can now be answered by Algorithm 4 with $\mathcal{D}$ by setting the failure probability to $\delta/2$. The correctness and runtime guarantees of this construction follow from Lemmas B.3 and B.4 and the union bound.

□

C Lower Bound

Here we show that any Monte Carlo randomized data structure for handling adaptive ADE queries in Euclidean space with $> 1/2$ success probability needs to use $\Omega(nd)$ space. Since this will be a lower bound on the space complexity in bits yet thus far we have been talking about vectors in $\mathbb{R}^d$, we need to make an assumption on the precision being used. Fix $\eta \in (0, 1/2)$ and define $B_\eta := \{x \in \mathbb{R}^d : \|x\|_2 \leq 1, \forall i \in [d],\ x_i \text{ is an integer multiple of } \eta/\sqrt{d}\}$. That is, B_η is the subset of the Euclidean ball in which all vector coordinates are integer multiples of $\eta/\sqrt{d}$ for some $\eta \in (0, 1/2)$. We will show that the lower bound holds even in the special case that all database and query vectors are in B_η.

Lemma C.1. $\forall \eta \in (0, 1/2)$, $|B_\eta| = \exp(\Theta(d \log(1/\eta)))$

Proof. A proof of the upper bound appears in [AK17]. For the lower bound, observe that if $x_i = c_i \eta/\sqrt{d}$ for $c_i \in \{0, 1, \ldots, \lfloor 1/\eta \rfloor\}$, then $\|x\|_2 \leq 1$ so that $x \in B_\eta$. Thus $|B_\eta| \geq \lfloor 1/\eta \rfloor^d$. □

We now prove the space lower bound using a standard encoding-type argument.

Theorem C.2. *Fix $\eta \in (0, 1/2)$. Then any data structure for ADE in Euclidean space which always halts within some finite time bound T when answering a query, with failure probability $\delta < 1/2$ and $\varepsilon \in (0, 1)$, requires $\Omega(nd \log(1/\eta))$ bits of memory. This lower bound holds even if promised that all database and query vectors are elements of B_η.*

Proof. Let $\mathcal{D}$ be such a data structure using S bits of memory. We will show that the mere existence of $\mathcal{D}$ implies the existence of a randomized encoding/decoding scheme where the encoder and decoder share a common public random string, with $\mathsf{Enc} : B_\eta^n \to \{0, 1\}^S$. The decoder will succeed with probability 1. Thus encoding length s needs to be at least the entropy of the input distribution, which will be the uniform distribution over B_η^n, and thus $S \geq \lceil n \log_2 |B_\eta| \rceil$, which is at least $\Omega(nd \log(1/\eta))$ by Lemma C.1.

We now define the encoding: we map $X = (x_i)_{i=1}^n \in B_\eta^n$ to the memory state of the data structure after pre-processing with database X (this memory state is random since the pre-processing procedure may be randomiezd). The encoding length is thus S bits. We now give an exponential-time decoding algorithm which can recover X precisely given only $\mathsf{Enc}(X)$. To decode, we iterate over all $q \in B_\eta$ to discover which x_i equal q (if any). Note $\|q - x_i\|_2 = 0$ iff $q = x_i$, and thus a multiplicative $1 + \varepsilon$-approximation to all distances would reveal which x_i are equal to q. To circumvent the nonzero failure probability of querying the data structure, we simply iterate over all possibilities for the random string used by the data structure (since $\mathcal{D}$ runs in time at most T it always flips at most T coins, and there are at most 2^T possibilities to check). Since the failure probability is at most $1/2$, the estimate of q to x_i will be zero more than half the time iff $q = x_i$. □