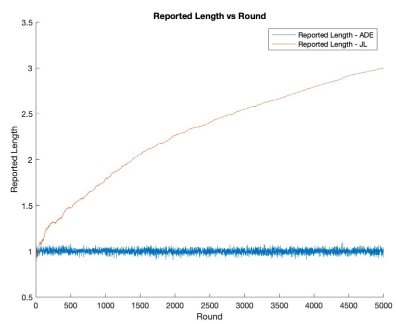
We thank the reviewers for their careful readings of Submission # 4484, title “On Adaptive Distance Estimation”.

Reviewer 1: In response to your suggestion, we have implemented a vanilla Johnson-Lindenstrauss (JL) sketch vs. our structure and shown the results of adaptive querying on both with the following experimental setup:



We have three database vectors $e_1, -e_1, 0$ in 5000 dimensions. We JL sketch down to 250 dimensions. We always query unit norm vectors q. We do a sequence of queries, and the x-axis specifies which query number we are at in the sequence, and the y-axis is the sketch’s reported distance to 0 for that query (so, q’s length). We do an adaptive attack, where we pick the next query vector from a distribution based on previous queries’ distance estimations to $e_1, -e_1$ (attack description included in revision). The orange curve shows the vanilla JL’s length estimate, which deviates more from the true length as we do more adaptive queries. The blue curve is our structure’s estimates of length, which is correctly always near 1. For our structure, we took the median of 5 randomly selected sketches of 200, which only increased query time by a factor 5.

Reviewer 2: **Utility for e.g. kernel regression:** In kernel regression the database also has y_i’s and for a query q we must (approximately) output $\sum_i k(x, x_i)y_i$ for some kernel $k(\cdot, \cdot)$. For kernels based on Euclidean distances, e.g. RBF’s, it is thus natural to want to have all distances to then approximately compute this sum. We note that for the slightly similar problem of kernel density estimation, there are *sublinear* time algorithms (e.g. see papers of Charikar and Siminelakis), but they are not designed to handle adaptive queries, plus we are unaware of similar solutions for kernel regression. We also remark that there *are* studies of “coreset” constructions for kernel regression which reduce n to some $n' \ll n$ (e.g. (Zheng, Phillips KDD’17)); this is a (weighted) subset of the data that gives approximately the same answer to any query. Coresets provide the approximation property for all queries and thus support adaptive queries. Thus the naive $O(nd)$ time query algorithm becomes $O(n'd)$ for adaptive queries. This is orthogonal to our approach though, and can in fact be combined: one can build our data structure *on the coreset* to get query time $\tilde{O}(n' + d)$.

Empirical evaluation: See response to Reviewer 1.

Novelty compared to [Kle97,KOR00] : Our work, as well as these two works, all do use the idea of having some random process (henceforth we will call a “test”) that does something useful with good probability for some fixed vector, amplifying via repetition to work with high probability for all vectors, then doing some form of sampling of tests at query time. [Kle97,KOR00] descriptions and analyses are tailored to the specific processes, whereas we strive for a completely general meta theorem (Theorem 1.3) that converts any non-robust structure into a robust one that can handle adaptive queries; our ADE result is essentially then a corollary. There are other differences; our “tests” are all different, and furthermore in [KOR00] one test (per level of binary search) is sampled at query time (a “test” there is the sequence of dot products over $\mathbb{F}_2$, after reducing to Hamming space on the hypercube, of a point with a collection of random binary vectors from some distribution); sampling only one sketch per query cannot work in our setting unless we blow the space up by an undesirable $poly(n)/\delta$ factor.

Reviewer 3: Thank you for your review.

Reviewer 4: We would like to address this reviewer’s question regarding correctness. As stated, our results guarantee correctness with high probability for *any* individual query even in a sequence of adaptively chosen queries. For example, the 100$^{\text{th}}$ query may be chosen depending on the answers to the first 99, but our correctness guarantees still hold. A union bound does in fact then guarantee correctness for an entire sequence of T adaptive queries with probability $1 - \delta$ with per query runtime $O(\epsilon^{-2}(n + d)\log(T/\delta))$ by instantiating the data structure with failure probability parameter $\delta' := \delta/T$. The reason this union bound is permissible: our analysis conditions on a certain event occurring during the (random) pre-processing stage of the data structure: namely that the set of (random) sketches generated are “representative” (Definition 4.3). Once we condition on this event of having “representative” sketches, our data structure actually allows the answering of each subsequent adaptive query in a sequence correctly with 100% success probability, just by returning the median output of *every* sketch. Doing so though unfortunately leads to slow query time, which is why our query procedure instead samples only a few ($O(\log(n/\delta))$) random sketches and output the median result from them. Conditioned on the sketches being representative, this works with high probability even in adaptive settings.

Reviewer 5: We thank you for finding typographical errors and for the question about Theorem A.8; in the revision we will make sure to consistently say “Chernoff bound” in all applications; the generalization to a_i, b_i bounds on the r.v.’s is indeed not needed for us. We also thank you for the reference to the work of Hardt and Ullman.