Streaming Algorithms for Ellipsoidal Approximation of Convex Polytopes

Yury Makarychev YURY@TTIC.EDU
Naren Sarayu Manoj NSM@TTIC.EDU
Max Ovsiankin MAXOV@TTIC.EDU
Toyota Technological Institute Chicago

Editors: Po-Ling Loh and Maxim Raginsky

Abstract

We give efficient deterministic one-pass streaming algorithms for finding an ellipsoidal approximation of a symmetric convex polytope. The algorithms are near-optimal in that their approximation factors differ from that of the optimal offline solution only by a factor sub-logarithmic in the aspect ratio of the polytope.

Keywords: Ellipsoid, streaming, sketching, online algorithms

1. Introduction

Let X be a centrally symmetric convex body in $\mathbb{R}^d$. We say that an ellipsoid $\mathcal{E}$ is an *α-ellipsoidal approximation for X* if $\mathcal{E}/\alpha \subseteq X \subseteq \mathcal{E}$ (where $\alpha \geq 1$). Calculating ellipsoidal approximations has applications to problems in machine learning and data science, including sampling and volume estimation (see, e.g., [CV15] and [JLLV21]), obstacle collision detection in robotics (see [RB97]), differential privacy (see [NTZ13]), and online learning (see [LLS19]). Further, the ellipsoid $\mathcal{E}$ provides a very succinct approximate representation of X – $\mathcal{E}$ can be stored using only $\binom{d+1}{2} = O(d^2)$ floats, while the exact representation of X may be arbitrarily large.

John's theorem [Joh48] states that the minimum-volume outer ellipsoid, called *John's Ellipsoid of X*, is a $\sqrt{d}$-ellipsoidal approximation when X is symmetric – that is, when $X = -X$. Similarly, John's theorem implies that the maximum-volume inner ellipsoid also yields a $\sqrt{d}$-ellipsoidal approximation when X is symmetric. Furthermore, the approximation factor of $\sqrt{d}$ given by John's Ellipsoid cannot be improved in the worst case (e.g. for the hypercube or cross-polytope). John's result can be made algorithmic: Cohen, Cousins, Lee, and Yang designed an highly efficient algorithm for computing the maximum volume interior ellipsoid when X is a symmetric polytope defined by linear constraints, giving a $\sim \sqrt{d}$ approximation in the offline setting [CCLY19].

Streaming Algorithm. A natural follow-up question is whether a similar approximation guarantee exists for convex polytopes given in the *streaming setting*. Specifically, suppose that we are given points or constraints defining a symmetric convex polytope one-at-a-time. Our goal is to design an algorithm that computes an α-ellipsoidal approximation to the polytope and uses as little memory as possible. Such an algorithm will be useful in a memory-constrained environment. For instance, consider a streaming data summarization task in which a user wishes to obtain an approximation of a dataset that is too large to fit in memory. By computing a good ellipsoidal approximation, the

user can summarize the dataset in only $\binom{d+1}{2}$ floating point numbers, whereas to store all vertices of the polytope X we may need nd floats. To our knowledge, existing solutions (such as that of [CCLY19]) require the entire dataset to be stored in memory and therefore cannot be applied to a streaming setting.

Related Works and Applications The problem of calculating an ellipsoidal approximation to a convex body has been well-studied; see [Tod16] for an overview of the area. The recent paper of [CCLY19] presents an $\widetilde{O}\left(nd^2\right)$-time algorithm for computing a $\sim \sqrt{d}$-approximation for X when X is specified by symmetric linear constraints.

The problem of approximating a *non-symmetric* convex hull with an ellipsoid was introduced in [MSS10]. The authors present a greedy algorithm for this problem and show that its approximation factor is unbounded for every $d \geq 2$. However, they do not provide any upper bounds on the approximation ratio.

Efficiently calculating ellipsoidal approximations has implications to the problem of estimating the volume of a convex body. For instance, it is known (see [CV15]) that if a convex body X satisfies $B_2^d \subseteq X \subseteq R \cdot B_2^d$, then its volume can be approximated in time $\widetilde{O}\left(\max\left\{d^2R^2, d^3\right\}\right)$ using a procedure known as *Gaussian cooling*. Observe that computing an α-ellipsoidal approximation yields a linear transformation T that transforms a convex body X into a position such that $B_2^d \subseteq TX \subseteq \alpha \cdot B_2^d$. Hence, an efficient algorithm to compute an α-ellipsoidal approximation for small α yields an efficient algorithm for estimating the volume of a convex polytope.

Ellipsoidal approximations have also been used as *exploration bases* in some online optimization problems. For instance, the work of [LLS19] considers a stochastic linear optimization setting with adversarial corruptions. Here, a learner observes noisy evaluations of a linear function with the goal of maximizing this function over a convex constraint set. The algorithm used in [LLS19] uses an ellipsoidal rounding of the constraint set to construct an exploration basis. The learner then uses this exploration basis to sample actions during its exploration phases. The approximation factor of the rounding plays a role in the expected regret of the algorithm. Thus, a good, efficiently computable ellipsoidal approximation can be used as a black box to obtain a more efficient, lower-regret algorithm for this setting.

It is highly desirable to give a streaming algorithm for problems in compute-constrained scenarios. For instance, the work of [RB97] studies a problem in which a robot must estimate the distance between a collection of obstacles and itself. At a high level, their workflow involves computing an ellipsoidal approximation of the convex hull of the set of obstacles. The robot then calculates the distance between itself and this ellipsoidal approximation. Now, the robot could be operating using a microcontroller or some other device with limited computing capabilities and at the same time, the amount of data may be very large. Therefore, we need a memory-efficient algorithm for this setting.

Finally, a streaming algorithm to compute ellipsoidal approximations would yield an algorithm that can adapt to certain changes in a dataset over time while maintaining a consistent approximation guarantee. For instance, suppose that the robot in the setting of [RB97] has only a limited sight distance. As the robot moves, it acquires knowledge of new obstacles. A streaming algorithm for ellipsoidal approximations would allow the robot to quickly update its summary of the set of obstacles as it moves.

1.1. Main Result

In this paper, we study the following formalization of the *Ellipsoidal Approximation Problem*, stated below.

Problem 1 (Ellipsoidal Approximation Problem) *Given a symmetric convex body X, find an ellipsoid $\mathcal{E}$ so that $\mathcal{E}/\alpha \subseteq X \subseteq \mathcal{E}$ for some $\alpha \geq 1$. The goal is to find an approximation with a small value of α. We say that $\mathcal{E}$ is an α-approximation to X.*

We study the Ellipsoidal Approximation Problem in the following streaming model. We assume that points $x_1, \ldots, x_n \in X$ arrive one-by-one. After the algorithm receives point x_t, it must output an ellipsoid $\mathcal{E}_t$ centered at the origin such that all points $x_1, \ldots, x_t$ lie in $\mathcal{E}_t$. The *approximation factor* of the algorithm is the smallest α such that $\mathcal{E}_n/\alpha \subseteq \mathsf{conv}(\{\pm x_1, \ldots, \pm x_n\}) \subseteq \mathcal{E}_n$ for every input sequence. The value of n might not be known to the algorithm beforehand.

Problem 2 (Formal Problem Statement) *We observe a stream of points $x_1, \ldots, x_n$ where points arrive one-by-one and n might not be known beforehand. Upon receiving point x_t, find an ellipsoid $\mathcal{E}_t$ such that:*

- *For all $t \in \{1, \ldots, n\}$, we have $\mathsf{conv}(\{\pm x_1, \ldots, \pm x_t\}) \subseteq \mathcal{E}_t$;*
- *At the end of the stream, we have $\mathcal{E}_n/\alpha \subseteq \mathsf{conv}(\{\pm x_1, \ldots, \pm x_n\}) \subseteq \mathcal{E}_n$.*

The primary results of our work are algorithms presented in Theorem 3.

Theorem 3 (Main Result) *Let $X = \mathsf{conv}(\{\pm x_1, \ldots, \pm x_n\})$. Assume that X contains a ball of radius r and is contained in a ball of radius R.*

1. *There is a streaming algorithm that given r and a stream of points $x_1, \ldots, x_n$ provides a solution for Problem 2 with $\alpha = O(\sqrt{d \log(R/r + 1)})$.*
2. *There is a streaming algorithm that given R/r (but not r and R) and a stream of points $x_1, \ldots, x_n$ provides a solution for Problem 2 with $\alpha = O(\sqrt{d \log(R/r + 1)})$.*

The algorithms run in time $\widetilde{O}(nd^2)$ and store $O(d^2)$ floating-point numbers.

The ratio R/r gives an upper bound to the *aspect ratio* $\kappa(X)$, a quantity we formally define in Definition 6. We present and analyze the algorithm for item 1 in Theorem 9 and the one for item 2 in Theorem 18. Our results provide a nearly optimal approximation, since $\sqrt{d \log(R/r + 1)}$ is worse than $\sqrt{d}$ (the best possible factor for the offline setting) by only a factor of $O\left(\sqrt{\log(R/r + 1)}\right)$. Our algorithms store only $O(d^2)$ floats – this amount of memory is necessary to represent an ellipsoid in $\mathbb{R}^d$. We note that the algorithm from item 1 is similar to the algorithm for the non-symmetric case of the problem presented in [MSS10].

Hyperplane representation of a polytope. Our algorithms also work in the setting where instead of receiving points x_i, we receive hyperplanes (or, more precisely, slabs) $\{y : |\langle x_i, y\rangle| \leq 1\}$ defining body X. We describe this setting in Appendix C.

Applications to volume estimation. Our algorithms directly yield concrete results for the problem of estimating the volume of a symmetric convex polytope. In particular, composing our ellipsoidal approximation procedures with the Gaussian cooling algorithm of [CV15] yields an

$\widetilde{O}\left(nd^2 + d^3 \log\left(R/r\right) + d^3\right)$-time algorithm for approximating the volume of a symmetric convex polytope. This guarantee is comparable to that obtained by using the algorithm of [CCLY19] as a preprocessing step prior to Gaussian cooling. Furthermore, this improves upon the best guarantee for this problem suggested by [JLLV21], which is $\widetilde{O}\left(nd^{3.2}\text{polylog}\left(R/r\right)\right)$ time ([JLLV21] assumes the KLS hyperplane conjecture but does not require that X be symmetric).

Limitations to approximating John's ellipsoid. We note that one could try to obtain a good ellipsoidal approximation $\mathcal{E}$ for X in the streaming model by finding a β-approximation for John's ellipsoid. Such an ellipsoid $\mathcal{E}$ would provide a $\beta\sqrt{d}$ approximation for X. However, we give a lower bound on the approximability of John's ellipsoid for a natural class of streaming algorithms. We show that none of the algorithms in this class can find a better than $\sqrt{d}$-approximation for John's ellipsoid. Thus, the approach of approximating X via approximating John's ellipsoid potentially may only yield an $O(\sqrt{d} \cdot \sqrt{d}) = O(d)$ approximation.

Independent and concurrent work on convex hull approximation. Independent of and concurrent to our work, Woodruff and Yasuda [WY22] consider a problem of maintaining a coreset for ℓ_∞ subspace embedding: given a stream of points, choose a small subset of them so that the symmetric convex hull of this subset is a good approximation to the convex hull of the original points. This problem and the one we study are closely related. In both of them, we need to maintain an approximation for the symmetric convex hull of a stream of points. However, Woodruff and Yasuda approximate the convex hull with a convex hull of a small coreset, while we approximate the convex hull with an ellipsoid. The approximation guarantee of [WY22] is $O(\sqrt{d \log\left(n\kappa^{\mathsf{OL}}(X)\right)})$, where $\kappa^{\mathsf{OL}}(X)$ is the online condition number. It has the same dependence on d as our guarantee; $\kappa^{\mathsf{OL}}(X)$ is closely related to but different from the parameter $\kappa(X)$ we use. Also, by applying John's Theorem, the authors of [WY22] get an $O(d\sqrt{\log\left(n\kappa^{\mathsf{OL}}(X)\right)})$ ellipsoidal approximation to the convex hull. Our approximation guarantee is better than this one by a factor of $\sim \sqrt{d}$ and does not depend on n.

A Proof Sketch of the Main Result. We now give a proof outline of Theorem 3.

Our first algorithm, Algorithm 1, maintains an ellipsoid $\mathcal{E}_t$ covering $\mathsf{conv}\left(\{\pm x_1, \dots, \pm x_t\}\right)$. Initially, $\mathcal{E}_0$ is simply a ball of radius r. Upon receiving point x_{t+1}, the algorithm updates $\mathcal{E}_t$ to $\mathcal{E}_{t+1}$ by computing the minimal volume ellipsoid containing $\mathcal{E}_t$ and $\pm x_{t+1}$.

We need to prove that this algorithm achieves an $\alpha = O(\sqrt{d \log\left(R/r + 1\right)})$ approximation. By construction, $\mathcal{E}_n$ contains X. Now we need to prove that $\mathcal{E}_n \subseteq \alpha \cdot X$. We first prove a different statement. We compare ellipsoid $\mathcal{E}_n$ not to X but rather to an ellipsoid $\mathcal{E}^*$ that contains X. We show that $\mathcal{E}_n \subseteq \alpha' \mathcal{E}^*$ for every ellipsoid $\mathcal{E}^*$ such that (i) $\mathcal{E}^*$ contains X and (ii) $\mathcal{E}^*$ has aspect ratio (the ratio of its longest to shortest semi-axis) at most R/r. To this end, we define a potential function Φ (see Definition 12) with the following properties:

- $\mathcal{E}_n \subseteq \alpha' \mathcal{E}^*$ for $\alpha' = O(\sqrt{\Phi})$ (see Lemma 13)
- Initially, Φ is $O(d \log\left(R/r + 1\right))$ (see Lemma 17).
- The value of the potential function is non-increasing over time (see Lemma 14).

These properties imply that $\mathcal{E}_n \subseteq \alpha' \mathcal{E}^*$ for $\alpha' = O(\sqrt{d \log\left(R/r + 1\right)})$. Then we prove in Theorem 10, that this implies that $\mathcal{E}_n \subseteq \alpha X$ with $\alpha = \sqrt{2}\alpha'$, as required.

For this algorithm to perform well, it must know r or a reasonable estimate for r. However, if we do not have any estimate on r, the performance of the algorithm may be arbitrarily bad. On the technical level, the challenge is that the initial value of potential Φ may be arbitrarily large.

Algorithm 2 does not need to know r but instead needs to have an estimate ξ for the aspect ratio of X. Algorithm 2 updates $\mathcal{E}_i$ in two steps: first it performs the update step from Algorithm 1 and then ensures that the aspect ratio of the obtained ellipsoid is roughly at most ξ (if it is more than that, it expands the semiaxes of the ellipsoid appropriately). The analysis of Algorithm 2 is based on that of Algorithm 1 but is substantially more complex. We use a pair of potential function S and P and keep track of their evolutions over time.

Outline The rest of our paper is organized as follows. In Section 2, we present definitions and notation used in this paper. In Section 3, we describe the first algorithm from Theorem 1. The algorithm itself is very simple but its analysis is insightful and captures the core technical ideas used later. In Section 4, we present the second algorithm from Theorem 2. The analyses of Theorems 1 and 2 relies on the fact that every centrally symmetric convex body is well approximated by an intersection of ellipsoids with bounded aspect ratio. We prove this fact in Appendix A. In Appendix B, we present our lower bound on the approximability of John's ellipsoid. Finally, in Appendix C, we discuss the equivalence between the problem we study and an alternate formulation wherein we receive linear constraints one-at-a-time instead of points.

2. Preliminaries and Notation

Notation Consider a sequence of points $\{x_1, \dots, x_n\} \subset \mathbb{R}^d$. We denote the symmetric convex hull of the first t points by $X_t = \mathsf{conv}(\{\pm x_1, \dots, \pm x_t\})$ and the symmetric convex hull of all points $\{\pm x_i\}$ by $X = X_n$. We denote the standard Euclidean norm of a vector v by $\|v\|$ and the Frobenius norm of a matrix A by $\|A\|_F$. We denote the singular values of a matrix $A \in \mathbb{R}^{d\times d}$ by $\sigma_1(A), \dots, \sigma_d(A)$. Let $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ be the largest and smallest singular values of A, respectively. We say that X is centrally symmetric if $X = -X$.

Denote the ℓ_p-unit ball by $B_p^d = \{x \in \mathbb{R}^d : \|x\|_p \le 1\}$. Given a set $S \subseteq \mathbb{R}^d$, its polar is $S^\circ := \{y \in \mathbb{R}^d \colon \sup_{x\in S} |\langle x, y\rangle| \le 1\}$. We use natural logarithms unless otherwise specified.

In this paper, we will work extensively with ellipsoids. We will always assume that all ellipsoids and balls we consider are centered at the origin; we will not explicitly state that. We use the following representation of ellipsoids. For a non-singular matrix $A \in \mathbb{R}^{d\times d}$, let $\mathcal{E}_A := \{x : \|Ax\| \le 1\}$. In other words, matrix A defines a bijective map of $\mathcal{E}_A$ to the unit ball B_2^d. Every ellipsoid (centered at the origin) has such a representation. We note that this representation is not unique as matrices A and MA define the same ellipsoid if matrix M is orthogonal (since $\|Av\| = \|MAv\|$ for every vector v). Now consider the singular value decomposition of A: $A = U\Sigma^{-1}V^T$ (it will be convenient for us to write Σ^{-1} instead of standard Σ in the decomposition). The diagonal entries of Σ are exactly the semi-axes of $\mathcal{E}_A$. As mentioned above, matrices $U\Sigma^{-1}V^T$ and $U'\Sigma^{-1}V^T$ define the same ellipsoid for any orthogonal $U' \in \mathbb{R}^{d\times d}$; in particular, every ellipsoid can be represented by a matrix of the form $A = \Sigma^{-1}V^T$.

Our goal is to design an algorithm for Problem 2 that achieves a good approximation α and at the same time uses as little memory as possible. To understand what value of α is achievable in the offline case, recall John's Theorem.

Theorem 4 (John's Theorem, [Joh48]) *I. Let X be a centrally symmetric convex body. Consider the minimum volume ellipsoid $\mathcal{E}$ containing X. Then we have $\mathcal{E}/\sqrt{d} \subseteq X \subseteq \mathcal{E}$.*

II. There exists a centrally symmetric body X in $\mathbb{R}^d$ (e.g. hypercube B_∞^d and cross-polytope B_1^d) such that there is no ellipsoid $\mathcal{E}$ that approximates X within a factor of $\alpha < \sqrt{d}$: $\mathcal{E}/\sqrt{d} \subseteq X \subseteq \mathcal{E}$.

In this work, we consider a natural class of one-pass streaming algorithms that we call *monotonic algorithms*.

Definition 5 (Monotonic Algorithm) *We call an algorithm for Problem 2* monotonic *if it produces a sequence of ellipsoids $\mathcal{E}_t$ satisfying $X_t \subseteq \mathcal{E}_t$ and $\mathcal{E}_t \subseteq \mathcal{E}_{t+1}$ for all timestamps t.*

Monotonic algorithms have the advantage that once they decide that a certain point x belongs to ellipsoid $\mathcal{E}_t$, they commit to this decision: all consecutive ellipsoids $\mathcal{E}_{t+1}, \mathcal{E}_{t+2}, \dots$ also contain point x. The approximation factor of our algorithms depend sublogarithmically on the *aspect ratio* of the convex body X.

Definition 6 (Aspect Ratio) *Consider a centrally symmetric convex body X. Let r be the radius of the largest ball $r \cdot B_2^d$ contained in X and R be the radius of the smallest ball $R \cdot B_2^d$ that contains X. Then the aspect ratio of X is written as $\kappa(X) = R/r$.*

Logarithmic dependences on the aspect ratio have previously appeared for algorithms on convex bodies; for example, the algorithms in [JLLV21] for rounding and computing the volume of a convex body have a runtime that depends on $\log \kappa(X)$. We also recall the condition number of a matrix:

Definition 7 (Condition Number of a Matrix) *The condition number $\kappa(A)$ of a symmetric non-singular matrix A is the ratio of its largest to smallest singular values: $\kappa(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$.*

The notions of aspect ratio of a convex body and condition number of a matrix are closely related. It is immediate that the aspect ratio of an ellipsoid $\mathcal{E}$ equals the ratio of its longest to shortest semi-axes. Consequently, $\kappa(\mathcal{E}_A) = \kappa(A)$.

3. Scale-Dependent Algorithm for Ellipsoid Approximation

In this section, we present and analyze a simple algorithm for Problem 2; see Algorithm 1. This algorithm must be given a radius r such that $r \cdot B_2^d \subseteq X$. The approximation guarantee of the algorithm linearly depends on $\sqrt{\log(R/r)}$ where $R = \max_t \|x_t\|$ is the radius of the smallest ball that contains X.

The key line in the algorithm is Line 6 which updates $\mathcal{E}_t$ to contain x_t; we refer to it as the "update rule." Claim 8, which we prove in Appendix D, shows how to compute the update.

Claim 8 *Given a matrix A_{t-1} for $\mathcal{E}_{t-1}$, the updated matrix A_t for $\mathcal{E}_t$ can be updated using the following formula: $A_t = \widehat{A} A_{t-1}$, where*

$$\widehat{A} = \left(I - \left(1 - \frac{1}{\|A_{t-1} x_t\|} \right) \left(\frac{(A_{t-1} x_t)(A_{t-1} x_t)^T}{\|A_{t-1} x_t\|^2} \right) \right). \tag{3.1}$$

We first show that $\mathcal{E}_n$ provides a good approximation to every ellipsoid $\mathcal{E}^*$ with aspect ratio R/r containing X.

Algorithm 1 Streaming Ellipsoidal Approximation – Scale-Dependent Algorithm

Input: A stream of points $x_1, \ldots, x_n$ and a value r such that:

$$r \cdot B_2^d \subset X = \mathsf{conv}\left(\{\pm x_1, \ldots, \pm x_n\}\right)$$

Output: Ellipsoid $\mathcal{E}_n$ that covers X.
Initialize $\mathcal{E}_0 = r \cdot B_2^d$.
for $t = 1, \ldots, n$ **do**
    Read point x_t from the stream.
    Let $\mathcal{E}_t$ be the ellipsoid of smallest volume (centered at 0) that contains both $\mathcal{E}_{t-1}$ and x_t.
end for
Output: $\mathcal{E}_n$.

Theorem 9 *Let $\mathcal{E}^* \supseteq X$ be an ellipsoid containing X with aspect ratio at most R/r. Then the output of Algorithm 1 $\mathcal{E}_n$ satisfies:*

$$\mathcal{E}_n \subseteq \alpha \cdot \mathcal{E}^*$$

where $\alpha = O\left(\sqrt{d\left(\log\left(R/r\right)+1\right)}\right)$. Algorithm 1 runs in time $O(nd^2)$ and stores at most $O(d^2)$ floating point numbers.

As stated, Theorem 9 does not say that $\mathcal{E}_n$ provides an α approximation for X. However, the statement of Theorem 9 holds simultaneously for all ellipsoids $\mathcal{E}^*$ whose aspect ratio is at most that of X. As the following theorem shows, this is sufficient to get the desired result that $\mathcal{E}_n \subseteq \alpha\sqrt{2} \cdot X$.

Theorem 10 *Consider a centrally symmetric convex body X and an ellipsoid $\mathcal{E}$. Assume that every ellipsoid $\mathcal{E}^*$ that satisfies properties (i) and (ii)*

(i) $\mathcal{E}^*$ contains X and (ii) $\mathcal{E}^*$ has aspect ratio of at most $\kappa(X)$

also contains $\mathcal{E}$. Then $\mathcal{E} \subseteq \sqrt{2} \cdot X$.

We prove Theorem 10 in Appendix A. Combining these theorems, we have Theorem 11.

Theorem 11 *Algorithm 1 gets an $\alpha = O\left(\sqrt{d\log\left(R/r+1\right)}\right)$ approximation: $\mathcal{E}_n/\alpha \subseteq X \subseteq \mathcal{E}_n$.*

We focus now on proving Theorem 9.

Proof [Proof of Theorem 9]

Runtime and Memory Complexity The key observation is that Algorithm 1 only stores the matrix A_t representing the ellipsoid $\mathcal{E}_t$ between iterations and does not require additional memory within an iteration. Hence, the memory complexity of Algorithm 1 is $O(d^2)$. Next, observe that computing the update rule as per Claim 8 requires only three matrix-vector products, which takes time $O(d^2)$. It immediately follows that the runtime of Algorithm 1 is $O(nd^2)$, as desired.

Correctness We assume without loss of generality that for all $t \geq 1$, $\|A_{t-1}x_t\| > 1$, since Algorithm 1 ignores all points x_t with $\|A_{t-1}x_t\| \leq 1$. In particular, when Algorithm 1 encounters such a

point, it simply lets $A_t = A_{t-1}$. Additionally, we assume that the shortest semi-axis of $\mathcal{E}^*$ is at most R. If not, we prove the statement for $\mathcal{E}^{**} = R \cdot B_2^d$ (which is contained in $\mathcal{E}^*$ by our assumption) and get $\mathcal{E}_n \subseteq \alpha\mathcal{E}^{**} \subseteq \alpha\mathcal{E}^*$, as required.

Let J^* be a matrix that defines the ellipsoid $\mathcal{E}^*$: $\mathcal{E}^* = \{x \ : \ \|J^*x\| \le 1\}$. Since all points x_t lie in $X \subseteq \mathcal{E}^*$, we have $\|J^*x_t\| = \|J^*(-x_t)\| \le 1$. Since the shortest semi-axis of $\mathcal{E}^*$ is at most R, we have $\sigma_{\max}(J^*) \ge 1/R$. Further, as $r \cdot B_2^d \subseteq X \subseteq \mathcal{E}^*$, we have $\sigma_{\max}(J^*) \le 1/r$.

Now we define a potential function $\Phi_{J^*}(\mathcal{E}_t)$. We will show that the value of this function does not increase over time. We will then upper bound the approximation factor α in terms of $\Phi_{J^*}(\mathcal{E}_n)$.

Definition 12 (Potential Function $\Phi_{J^*}(\cdot)$) *We define the potential function $\Phi_{J^*}(A_t)$ as:*

$$S_t := \left\|J^* \cdot A_t^{-1}\right\|_F^2 = \sum_{i=1}^{d} \sigma_i(J^*A_t)^2$$

$$P_t := 2\log\det\left(J^* \cdot A_t^{-1}\right) = \log\prod_{i=1}^{d} \sigma_i(J^*A_t)^2$$

$$\Phi_{J^*}(A_t) := S_t - P_t = \left\|J^* \cdot A_t^{-1}\right\|_F^2 - 2\log\det\left(J^* \cdot A_t^{-1}\right)$$

Let us see how we use $\Phi_{J^*}(\cdot)$ to upper bound the singular values of $J^*A_n^{-1}$ and the approximation factor α.

Lemma 13

1. $\sigma_{\max}(J^* \cdot A_n^{-1})^2 \le \frac{e}{e-1} \cdot \Phi_{J^*}(A_n)$
2. $\mathcal{E}_n \subseteq \alpha\mathcal{E}^*$, *where* $\alpha = \sigma_{\max}(J^* \cdot A_n^{-1})$.

Proof 1. Let $M = J^* \cdot A_n^{-1}$. Note that $f(t) := t^2 - \log t^2 \ge \frac{e-1}{e} \cdot t^2 > 0$ for $t > 0$. Therefore,

$$\Phi_{J^*}(A_n) = \sum_{i=1}^{d} \left(\sigma_i^2(M) - \log\sigma_i^2(M)\right) = \sum_{i=1}^{d} f(\sigma_i(M)) \ge f(\sigma_{\max}(M)) \ge \frac{e-1}{e} \cdot \sigma_{\max}^2(M)$$

as desired.

2. Consider a point $x \in \mathcal{E}_n$. Then $\|A_nx\| \le 1$. We get

$$\|J^*x\| = \left\|J^*A_n^{-1} \cdot A_nx\right\| \le \sigma_{\max}\left(J^* \cdot A_n^{-1}\right) \cdot \|A_nx\| \le \alpha.$$

We conclude that $x \in \alpha\mathcal{E}^*$. Thus, $\mathcal{E}_n \subseteq \alpha\mathcal{E}^*$. ■

Let us describe the plan for the rest of the proof.

- In Lemma 14, we will prove that $\Phi_{J^*}(A_t) \le \Phi_{J^*}(A_{t-1})$ for all $t \ge 1$; that is, values $\Phi_{J^*}(A_{t-1})$ are non-increasing.
- In Lemma 17, we give an upper bound $\Phi_{J^*}(A_0) \le O(d\log(R/r + 1))$.

From Lemma 13, we get that $\mathcal{E}_n \subseteq \alpha \cdot \mathcal{E}^*$ with

$$\alpha \leq O(\sqrt{\Phi_{J^*}(A_n)}) \leq O(\sqrt{\Phi_{J^*}(A_0)}) \leq O(\sqrt{d \log(R/r + 1)}).$$

It remains to prove Lemmas 14 and 17 mentioned above. We start with Lemma 14.

Lemma 14 *Under the update rule for Algorithm 1, we have $\Phi_{J^*}(A_t) \leq \Phi_{J^*}(A_{t-1})$ for all $t \geq 1$.*

Proof We first derive formulas that express P_t and S_t in terms of P_{t-1} and S_{t-1}.

Lemma 15 *We have,*

$$P_t = 2 \log \|A_{t-1} x_t\| + P_{t-1}$$
$$S_t = S_{t-1} + \left(1 - \frac{1}{\|A_{t-1} x_t\|^2}\right) \|J^* \cdot x_t\|^2$$

Proof By Claim 8, $A_t = \widehat{A} A_{t-1}$ where $\hat{A}$ is given by (3.1).

Determinant Update We start by calculating $\det\left(\widehat{A}\right)$. Note that $\left(1 - \frac{1}{\|A_{t-1}x_t\|}\right)\left(\frac{(A_{t-1}x_t)(A_{t-1}x_t)^T}{\|A_{t-1}x_t\|^2}\right)$ is a symmetric rank-1 matrix, whose only non-zero eigenvalue equals $\left(1 - \frac{1}{\|A_{t-1}x_t\|}\right)$. Therefore, the spectrum of $\widehat{A}$ consists of 1 with multiplicity $d-1$ and $\|A_{t-1}x_t\|^{-1}$ with multiplicity 1. Thus, $\det\left(\widehat{A}\right) = \|A_{t-1}x_t\|^{-1}$ and we have:

$$\det\left(J^* A_t^{-1}\right) = \det\left(J^* A_{t-1}^{-1} \widehat{A}^{-1}\right) = \det\left(J^* A_{t-1}^{-1}\right) \cdot \det\left(\widehat{A}^{-1}\right)$$

Frobenius Norm Update It is well-known (see, e.g., [HJ91]) that for any matrix A and orthonormal matrix V (with $v_1, \ldots, v_d$ as its columns):

$$\|A\|_F^2 = \|AV\|_F^2 = \sum_{i=1}^{d} \|A v_i\|^2$$

Let V be a matrix consisting of the eigenvectors of $\widehat{A}$. Observe that one of these vectors must be $v_1 := A_{t-1}x_t / \|A_{t-1}x_t\|$; let $v_2, \ldots, v_d$ denote the remaining eigenvectors, all of which have an associated eigenvalue of 1. Now we calculate:

$$\begin{aligned}
S_t &= \left\|J^* \cdot A_t^{-1}\right\|_F^2 = \left\|J^* \cdot A_{t-1}^{-1} \widehat{A}^{-1}\right\|_F^2 = \sum_{i=1}^{d} \left\|J^* \cdot A_{t-1}^{-1} \widehat{A}^{-1} v_i\right\|^2 \\
&= \|A_{t-1}x_t\|^2 \left\|J^* \cdot A_{t-1}^{-1} v_1\right\|^2 + \sum_{i=2}^{d} \left\|J^* \cdot A_{t-1}^{-1} v_i\right\|^2 \\
&= \left(\|A_{t-1}x_t\|^2 - 1\right) \left\|J^* \cdot A_{t-1}^{-1} v_1\right\|^2 + \sum_{i=1}^{d} \left\|J^* \cdot A_{t-1}^{-1} v_i\right\|^2 \\
&= \left(\|A_{t-1}x_t\|^2 - 1\right) \left\|J^* \cdot A_{t-1}^{-1} \cdot \frac{A_{t-1}x_t}{\|A_{t-1}x_t\|}\right\|^2 + \left\|J^* \cdot A_{t-1}^{-1}\right\|_F^2
\end{aligned}$$

$$= S_{t-1} + \left(1 - \frac{1}{\|A_{t-1}x_t\|^2}\right) \|J^* \cdot x_t\|^2$$

■

Now are ready to prove Lemma 16, which implies Lemma 14. In fact, it is stronger than what we need to prove Lemma 14; however, it will be necessary in the sequel.

Lemma 16 *For all $t \in \{1, \dots, n\}$, we have:*

$$S_t - S_{t-1} \leq (P_t - P_{t-1}) \|J^* \cdot x_t\|^2$$

Proof From Lemma 15, we have:

$$\frac{S_t - S_{t-1}}{P_t - P_{t-1}} = \frac{(1 - 1/\|A_{t-1}x_t\|^2) \|J^* \cdot x_t\|^2}{2 \log (\|A_{t-1}x_t\|)} \leq \|J^* x_t\|^2$$

where the inequality follows from the fact that $\frac{(1-1/y^2)}{2 \log y} < 1$ for all $y > 1$. We now multiply both sides by $P_t - P_{t-1}$ to obtain $S_t - S_{t-1} \leq (P_t - P_{t-1}) \cdot \|J^* x_t\|^2$. ■

As $x_t \in \mathcal{E}^*$, we have $\|J^* x_t\| \leq 1$. Using Lemma 14 and rearranging, we find:

$$\Phi_{J^*}(A_t) = S_t - P_t \leq S_{t-1} - P_{t-1} = \Phi_{J^*}(A_{t-1})$$

This concludes the proof of Lemma 14. ■

Lemma 17 *We have $\Phi_{J^*}(A_0) \leq d\,(1 + 4 \log (R/r))$.*

Proof Recall that $R = \max_{x \in X} \|x\|$.

Let $f(t) := t^2 - \log t^2$ as in the proof of Lemma 13. By the definition of the potential function, we have

$$\Phi_{J^*}(A_0) = \left\|J^* \cdot A_0^{-1}\right\|_F^2 - 2 \log \det \left(J^* \cdot A_0^{-1}\right) = \sum_{i=1}^{d} f(\sigma_i(J^* \cdot A_0^{-1}))$$

Now we bound the range of $\sigma_i(J^* \cdot A_0^{-1})$. Observe that we have $\mathcal{E}_0 = r \cdot B_2^d$ and, accordingly, $A_0 = 1/r \cdot I$. Hence, $J^* \cdot A_0^{-1} = rJ^*$ and $\sigma_i(J^* \cdot A_0^{-1}) = r\sigma_i(J^*)$. Now recall that by assumption $1/R \leq \sigma_{\max}(J^*) \leq 1/r$, and the condition number of J^* is at most R/r. Thus $\sigma_{\min}(J^*) \geq \frac{\sigma_{\max}(J^*)}{R/r} \geq r/R^2$, and consequently $r^2/R^2 \leq \sigma_i(J^* \cdot A_0^{-1}) \leq 1$. Now we bound $f(\sigma_i(J^* \cdot A_0^{-1}))$. Since f is convex,

$$f(t) \leq \max \{f\,(r^2/R^2)\,, f(1)\} \leq 1 + 2 \log (R^2/r^2) = 1 + 4 \log (R/r) \quad \text{for } t \in [r^2/R^2, 1]$$

We conclude that

$$\Phi_{J^*}(A_0) = \sum_{i=1}^{d} f(\sigma_i(J^* \cdot A_0^{-1})) \leq d\,(1 + 4 \log (R/r))\,.$$

■

We have proven Lemmas 14 and 17. This concludes the proof of Theorem 9. ■

4. Scale-Independent Algorithm

To use Algorithm 1, we need to know some lower bound r on the radius of the largest ball contained in X. The approximation guarantee of the algorithm linearly depends on $\sqrt{\log(R/r)}$, so as long as we have a reasonable estimate on r, we can use Algorithm 1. However, if we have no prior information about the scale of X and do not have any reasonable estimate r, we cannot use Algorithm 1. In this section, we present Algorithm 2 that only requires an upper bound $\xi \geq \kappa(X)$ on the aspect ratio of X. We present Algorithm 2 with its full implementation details. However,

Algorithm 2 Streaming Ellipsoidal Approximation

1: **Input**: A stream of points $x_1, \ldots, x_n$, and an aspect ratio estimate ξ
2: **Output**: Matrix A_n that defines ellipsoid $\mathcal{E}_n = \{x \ : \ \|A_n x\| \leq 1\}$.
3: Receive point x_1.
4: Set V_1 to be an orthonormal matrix satisfying $V_1^T x_1 = \|x_1\| \cdot e_1$.
5: $U_1 = I$, $\Sigma_1 = \mathsf{diag}\left(\|x_1\|, {}^{\|x_1\|}/_{\xi}, \ldots, {}^{\|x_1\|}/_{\xi}\right)$
6: $A_1 = U_1 \Sigma_1^{-1} V_1^T$
7: **for** $t = 2, \ldots, n$ **do**
8: Receive point x_t
9: **if** $\|A_{t-1} x_t\| > 1$ **then**
10: $a = {}^{A_{t-1}x_t}/_{\|A_{t-1}x_t\|}$ and $b = A_{t-1}^T a$.
11: $M_t = \max(M_{t-1}, \|x_t\|)$ *(that is, $M_t = \max_{1 \leq i \leq t} \|x_i\|$)*
12: $(U_t, (\Sigma_t')^{-1}, V_t) = \textsc{SVDRankOneUpdate}((U_{t-1}, \Sigma_{t-1}^{-1}, V_{t-1}), -\left(1 - {}^{1}/_{\|A_{t-1}x_t\|}\right) a, b)$
13: $\Sigma_t = \mathsf{diag}\left(\tau_{1,t}, \ldots, \tau_{d,t}\right)$, where $\tau_{i,t} = \max([\Sigma_t']_{ii}, {}^{M_t}/_{\xi})$ for every $i \in [d]$
14: **else**
15: $U_t = U_{t-1}, V_t = V_{t-1}, \Sigma_t = \Sigma_{t-1}$
16: **end if**
17: $A_t = U_t \Sigma_t^{-1} V_t^T$
18: **end for**
19: **Output**: $\mathcal{E}_n = \{x \ : \ \|A_n x\| \leq 1\}$

conceptually the only difference between Algorithms 1 and 2 is a new "singular value correction step", presented on line 13. As in Algorithm 1, we perform the basic update rule (line 12) – compute the minimum volume ellipsoid $\mathcal{E}_t'$ containing the current ellipsoid $\mathcal{E}_{t-1}$ and the new point x_t. Then, we increase the semi-axes of $\mathcal{E}_t'$ (if necessary) so that all of them are at least M_t/ξ, where $M_t = \max(\|x_1\|, \ldots, \|x_t\|)$ is the length of the longest vector we have received so far. This ensures that the matrix A_t is well-conditioned.

Formally, we compute matrix $A_t' = \widehat{A} A_{t-1}$ where $\widehat{A}$ is given by (3.1), and update its singular values. To implement this efficiently, we use a procedure

$$(U', \Sigma', V') = \textsc{SVDRankOneUpdate}((U, \Sigma, V), y, z),$$

which gives U', Σ', V' as the SVD of the matrix $U\Sigma V^T + yz^T$. Per [Sta08], this can be implemented in time $O(d^2 \log^2 d)$. As noted in Section 2, the U_ts have no effect on the definition of $\mathcal{E}_t$s and thus will not factor into our analysis. In fact, the algorithm may discard the value of U_t after it computes

the SVD for A_t and later use the identity matrix instead of U_t. However, we keep them in the algorithm so that the exposition is closer to that of Algorithm 1.

The remainder of this section is devoted to proving Theorem 18.

Theorem 18 *Algorithm 2 outputs an ellipsoid $\mathcal{E}_n$ satisfying:*

$$\frac{\mathcal{E}_n}{\sqrt{6+28d\log\xi+16d}} \subseteq X \subseteq \mathcal{E}_n$$

assuming $\xi \geq \kappa(X)$. Moreover, Algorithm 2 runs in time $O(nd^2\log^2 d)$ and stores at most $O(d^2)$ floats.

Proof We first compute the running time and then prove the correctness of the algorithm.

Runtime and Memory Complexity It is easy to see that Algorithm 2 keeps track of three matrices U_t, Σ_t, V_t at every iteration, in addition to the new point x_t. The algorithm does not explicitly store the matrices $A_t = U_t\Sigma_t^{-1}V_t$. Instead, when it is asked to compute $A_t w$ or $A_t^T w$, it simply consecutively performs 3 matrix-vector multiplications. Algorithm 2 does not store anything else in between iterations, so the memory complexity is $O(d^2)$ floating-point numbers, as desired.

We now analyze the running time. The matrix-vector multiplications require time $O(d^2)$. With the rank-one update, each iteration takes time $O(d^2\log^2 d)$. Hence, Algorithm 2 has time complexity $O(nd^2\log^2 d) = \widetilde{O}\left(nd^2\right)$ as desired.

Correctness As in the analysis of Algorithm 1, it will in fact be enough to show that for all ellipsoids $\mathcal{E}^*$ that cover X and have aspect ratio at most ξ, we have:

$$\mathcal{E}_n \subseteq \sqrt{3+14d\log\xi+8d}\cdot\mathcal{E}^*$$

Then, by Theorem 10, we will have that the intersection of all ellipsoids $\mathcal{E}^*$ covering X with aspect ratio at most ξ is itself a $\sqrt{2}$-approximation to X. Hence, $\mathcal{E}_n$ will be a $\sqrt{2}\cdot\sqrt{3+14d\log\xi+8d}$ to X, as required. Similarly, as in the analysis of Algorithm 1, we assume without loss of generality that for all t, $\|A_{t-1}x_t\| > 1$.

Observe that Lemma 13 implies that it is sufficient to analyze $\sigma_{\max}\left(J^*\cdot A_n^{-1}\right)$ for our choice of J^* defining the ellipsoid $\mathcal{E}^* = \{x \;:\; \|J^*x\| \leq 1\}$ satisfying $\kappa(\mathcal{E}^*) \leq \xi$. Specifically, our goal is now to show that $\sigma_{\max}\left(J^*\cdot A_n^{-1}\right)^2 \leq 3+14d\log\xi+8d$.

Define $Q_t = \max_{1\leq i\leq t}\|J^*x_i\|^2$. Clearly, $Q_1 \leq \cdots \leq Q_n$. Now we prove a version of Lemma 16 that applies to Algorithm 2. We give the proofs of all the lemmas stated below in Appendix E.

Lemma 19 *Under the update rule for Algorithm 2, we have the following*

- *For any two timestamps u, t, such that $1 \leq u \leq t \leq n$:*

$$S_t \leq S_u + Q_t\cdot(P_t - P_u) \tag{4.1}$$

- *For any timestamp $2 \leq t \leq n$,*

$$S_t \leq S_{t-1} + d\cdot Q_t \tag{4.2}$$

Let $\sigma_{i,j}$ be the ith singular value of $J^* A_j^{-1}$ and $\sigma_{\max,j} = \max_i \sigma_{i,j}$. By Lemma 13 (part 2), $\alpha \leq \sigma_{\max,n} \leq \sqrt{\sum_{i=1}^d \sigma_{i,n}^2}$, hence upper bounding $S_n = \sum_{i=1}^d \sigma_{i,n}^2$ is sufficient to finish the analysis. We do this by bounding S_1 and then applying Lemma 19, as described below.

Lemma 20 *We have $S_1 \leq d$.*

By applying the equations from Lemma 19 in sequence, we can relate S_n to S_1. Only applying (4.2) repeatedly is insufficient, as then the resulting upper bound on $S_n - S_1$ grows linearly with n. Even though (4.1) is efficient for large stretches where t is much later than u, it also does not give a good bound when Q_t is more than a constant factor larger than Q_u (this phenomenon may not be apparent from the equation itself, but becomes clear from the upcoming analysis). In order to obtain the desired bound on S_n, we split the Q_ts into contiguous groups such that Q_t does not increase significantly within any group, and differs by at least e outside the groups.

Lemma 21 *After running Algorithm 2 for n steps, we have*

$$S_n = \sum_{i=1}^d \sigma_{i,n}^2 \leq 3 + 14d \log \xi + 8d$$

Finally, we write $\sigma_{\max,n}^2 \leq \sum_{i=1}^d \sigma_{i,n}^2 \leq 3 + 14d \log \xi + 8d$. Recall that this is sufficient to conclude the proof of Theorem 18 – specifically, since we have $\sigma_{\max,n} = \sigma_{\max}\left(J^* \cdot A_n^{-1}\right) \leq \sqrt{3 + 14d \log \xi + 8d}$, we can invoke part 2 of Lemma 13 to arrive at $\mathcal{E}_n \subseteq \sqrt{3 + 14d \log \xi + 8d} \cdot \mathcal{E}^*$. ■

5. Acknowledgments

YM and MO were supported in part by NSF awards CCF-1718820, CCF-1955173, and CCF-1934843. NSM was supported by a United States NSF Graduate Research Fellowship. We thank Meghal Gupta, Darshan Thaker, Taisuke Yasuda, and the anonymous reviewers for helpful feedback and discussions.

References

[BC17] Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer International Publishing, 2nd edition, 2017. ISBN: 978-3-319-48310-8.

[CCLY19] Michael B. Cohen, Ben Cousins, Yin Tat Lee, and Xin Yang. A near-optimal algorithm for approximating the John ellipsoid. In *Proceedings of the Conference on Learning Theory*, volume 99, pages 849–873, 2019.

[CV15] Benjamin Cousins and Santosh Vempala. Bypassing KLS: Gaussian cooling and an $O^*(N^3)$ volume algorithm. In *Proceedings of the Symposium on Theory of Computing*, pages 539–548, 2015.

[HJ91] Roger A. Horn and Charles R. Johnson. *Topics in matrix analysis*. In Cambridge University Press, 1991.

[JLLV21] He Jia, Aditi Laddha, Yin Tat Lee, and Santosh Vempala. Reducing isotropy and volume to kls: an $O^*(n^3\psi^2)$ volume algorithm. In *Proceedings of the Symposium on Theory of Computing*, pages 961–974, 2021.

[Joh48] Fritz John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays Presented to R. Courant on his 60th Birthday*, pages 187–204. Interscience Publishers, Inc, 1948.

[LLS19] Yingkai Li, Edmund Y. Lou, and Liren Shan. Stochastic linear optimization with adversarial corruption, 2019. arXiv: 1909.02109 [cs.LG].

[MSS10] Asish Mukhopadhyay, Animesh Sarker, and Tom Switzer. Approximate ellipsoid in the streaming model. In *Proceedings of the International Conference on Combinatorial Optimization and Applications*, pages 401–413, 2010.

[NTZ13] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. In *Proceedings of the Symposium on Theory of Computing*, pages 351–360, 2013.

[RB97] Elon Rimon and Stephen P. Boyd. Obstacle collision detection using best ellipsoid fit. *J. Intell. Robotics Syst.*, 18(2):105–126, 1997.

[Sta08] Peter Stange. On the efficient update of the singular value decomposition. *PAMM*, 8(1):10827–10828, 2008.

[Tod16] Michael J. Todd. *Minimum-Volume Ellipsoids: Theory and Algorithms*. SIAM-Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2016. ISBN: 1611974372.

[Vis21] Nisheeth K. Vishnoi. *Algorithms for Convex Optimization*. Cambridge University Press, 2021.

[WY22] David P Woodruff and Taisuke Yasuda. High-dimensional geometric streaming in polynomial space. *arXiv preprint arXiv:2204.03790*, 2022.

Appendix A. Proof of Theorem 10: Approximating Convex Polytopes with Ellipsoids

In this section, we prove Theorem 10. To this end, we show that every centrally symmetric convex polytope X is well-approximated by the intersection of all ellipsoids $\mathcal{E}$ containing X with comparable aspect ratio. This will immediately imply Theorem 10. Let $q \geq 1/\kappa(X)$ and $\delta = \sqrt{1+1/q^2} - 1$. Define

$$E_q = \{\mathcal{E} \ : \ X \subseteq \mathcal{E} \text{ and } \kappa(\mathcal{E}) \leq q \cdot \kappa(X)\} \quad \text{and} \quad \mathcal{A}_q = \bigcap_{\mathcal{E} \in E_q} \mathcal{E}$$

We show that $\mathcal{A}_q$ provides a good approximation for X.

Lemma 22 *We have* $\frac{1}{1+\delta} \cdot \mathcal{A}_q \subseteq X \subseteq \mathcal{A}_q$.

Proof The first inclusion $X \subseteq \mathcal{A}_q$ is trivial, since all ellipsoids $\mathcal{E}$ in E_q contain X. We now prove the the first inclusion:

$$cA_q \subseteq (1+\delta)X. \tag{A.1}$$

Consider the set of all slabs of the form $\{x : |\langle a, x\rangle| \leq 1\}$ that contain X. Since X is a centrally symmetric convex body, the intersection of all the slabs in S equals X. Further, let set S' consist of the slabs in S expanded by a factor of $(1+\delta)$: for every slab $|\langle a, x\rangle| \leq 1$ in S, there is a slab $|\langle a, x\rangle| \leq (1+\delta)$ in S'. Then the intersection of slabs in S' equals $(1+\delta)X$. Thus, to prove inclusion (A.1), it is sufficient to prove that $|\langle a, x\rangle| \leq (1+\delta)$ in S. To this end, we construct an ellipsoid $\mathcal{E}$ in E_q that lies in the slab $|\langle a, x\rangle| \leq (1+\delta)$ (and contains $\mathcal{A}_q$, by the definition of $\mathcal{A}_q$).

We complement vector $a/\|a\|$ to an orthonormal basis for $\mathbb{R}^d$: $w_1 = a/\|a\|, w_2, \ldots, w_d$. Recall that the condition number $\kappa(X)$ equals the ratio of the radius r of the largest inscribed ball to the radius R of the smallest circumscribed ball. Since the width of slab $|\langle a, x\rangle| \leq 1$ containing X is $2/\|a\|$, we have $r \leq 1/\|a\|$. Accordingly, $R \leq \kappa(X)/\|a\|$. Therefore, for $x \in X \subseteq R \cdot B_2^d$, $\sum_{i=1}^d \langle w_i, x\rangle^2 = \|x\|^2 \leq R^2 \leq (\kappa(X)/\|a\|)^2$. Also note that $\langle w_1, x\rangle^2 = \langle a, x\rangle^2/\|a\|^2 \leq 1/\|a\|^2$. We are ready to define ellipsoid $\mathcal{E}$:

$$\mathcal{E} = \left\{ x : \langle w_1, x\rangle^2 + \sum_{i=2}^d \frac{\langle w_i, x\rangle^2}{q^2\kappa(X)^2} \leq \frac{1}{\|a\|^2}\left(1 + \frac{1}{q^2}\right) \right\}.$$

Now we verify that (i) $\mathcal{E} \in E_q$ and (ii) all points in $\mathcal{E}$ satisfy $|\langle a, x\rangle| \leq (1+\delta)$. First, note that the aspect ratio of $\mathcal{E}$ is $q\kappa(X)$. Using the bounds we derived above, we get that for all $x \in X$

$$\langle w_1, x\rangle^2 + \sum_{i=2}^d \frac{\langle w_i, x\rangle^2}{q^2\kappa(X)^2} \leq \frac{1}{\|a\|^2} + \frac{\kappa(X)^2}{\|a\|^2} \cdot \frac{1}{q^2\kappa(X)^2} = \frac{1}{\|a\|^2}\left(1 + \frac{1}{q^2}\right).$$

Therefore, all points from X lie in $\mathcal{E}$; that is, $X \subseteq \mathcal{E}$. We conclude that $\mathcal{E} \in E_q$, as required.

Finally, if $x \in \mathcal{E}$, then

$$\langle x, a\rangle = \sqrt{\|a\|^2\langle w_1, x\rangle^2} \leq \sqrt{1 + 1/q^2} = 1 + \delta.$$

This concludes the proof. ■

To get Theorem 10, we apply Theorem 22 with $q = 1$. Clearly, an ellipsoid $\mathcal{E}$ that is contained in every $\mathcal{E}^* \in E_1$ is also contained in $\mathcal{A} \subseteq \sqrt{2} \cdot X$.

Appendix B. Tracking the Minimum-Volume Outer Ellipsoid

Observe that the guarantee of Theorem 18 gives a guarantee similar to that given by John's Theorem for centrally-symmetric convex bodies. Therefore, a natural question is, "how closely can any one-pass monotonic algorithm approximate the minimum-volume outer ellipsoid for a centrally-symmetric convex body?" We formalize this notion below.

Definition 23 (Approximation to Minimum Volume Outer Ellipsoid) *We say a streaming algorithm A α-approximates the minimum volume outer ellipsoid if A outputs an ellipsoid $\mathcal{E}_n$ satisfying $\mathcal{E}_n \subseteq \alpha \cdot J(X)$, where $J(X)$ is the minimum volume outer ellipsoid for X.*

Theorem 24 asserts that for a natural class of streaming algorithms, it is not possible to approximate the minimum volume outer ellipsoid up to factor $< \sqrt{d}$ in the worst case.

Theorem 24 *Every one-pass monotonic deterministic streaming algorithm for Problem 2 has approximation factor to the minimum volume outer ellipsoid of at least $\sqrt{d}$, for infinitely many d.*

Proof Before we begin, recall that a Hadamard basis is a set of vectors $v_1, \dots, v_d$ such that:

- $\|v_i\| = 1$, for all $i \in [d]$;
- For all $i \neq j$, $\langle v_i, v_j \rangle = 0$;
- Every entry of v_i is in $\{\pm 1/\sqrt{d}\}$.

Our family of hard instances proceeds in two phases.

Phase 1 Let d be such that there exists a Hadamard basis for $\mathbb{R}^d$. Consider a corresponding Hadamard basis $v_1, \dots, v_d$. The adversary gives the algorithm the points $v_1, \dots, v_d$.

Phase 2 The adversary selects $i \in [d]$ arbitrarily and $\varepsilon \in (0, d-1)$ arbitrarily. They then define the vectors $w_i := e_i \cdot 1/\sqrt{d-\varepsilon}$ and $w_j := e_j \cdot \sqrt{d-1/\varepsilon}$ for all $j \neq i$. The adversary gives the algorithm the points $w_1, \dots, w_d$. Call the outcome here "Outcome (i)."

It is easy to see that at the end of Phase 1, the minimum volume outer ellipsoid is simply B_2^d. Furthermore, the algorithm's solution $\widehat{\mathcal{E}}$ contains $\mathsf{conv}\,(\pm v_1, \dots, \pm v_d)$. On the other hand, consider the following claim.

Lemma 25 *The following ellipsoid is the minimum-volume outer ellipsoid for Outcome (i):*

$$\mathcal{E}_{OPT(i)} = \left\{ x \ : \ 1 \geq \frac{x_i^2}{\left(1/\sqrt{d-\varepsilon}\right)^2} + \sum_{j \neq i}^{d} \frac{x_j^2}{\left(\sqrt{d-1/\varepsilon}\right)^2} \right\}$$

Proof Notice that all the points w_j are orthogonal. Thus, the minimum-volume outer ellipsoid containing all the w_j must be the one whose axes are along the directions of w_j and whose poles are located on w_j. Observe that $\mathcal{E}_{OPT(i)}$ satisfies this, so it must be the minimum-volume outer ellipsoid for the convex body whose vertices are determined by the w_j.

It now remains to show that every Hadamard basis vector is on the surface of $\mathcal{E}_{OPT(i)}$:

$$\frac{\left(1/\sqrt{d}\right)^2}{\left(1/\sqrt{d-\varepsilon}\right)^2} + \sum_{j\neq i}^{d} \frac{\left(1/\sqrt{d}\right)^2}{\left(\sqrt{d-1/\varepsilon}\right)^2} = \frac{1}{d}\left((d-\varepsilon) + (d-1)\cdot\frac{\varepsilon}{d-1}\right) = 1$$

Since the minimum volume ellipsoid containing all the w_j also contains the Hadamard basis vectors, it (i.e., $\mathcal{E}_{OPT(i)}$) must be the minimum-volume outer ellipsoid for Outcome (i). ■

We will now show that any that outputs an ellipsoid $\widehat{\mathcal{E}}$ at the end of Phase 1 must have an approximation factor of at least $\sqrt{d-\varepsilon}$ on at least one of Outcomes $(1,\ldots,i)$. Suppose that in each of Outcome (i), we obtain an ellipsoid $\widehat{\mathcal{E}}_i$ that satisfies $C\cdot\mathcal{E}_{OPT(i)} \supseteq \widehat{\mathcal{E}}_i$. We now have:

$$\mathsf{conv}\left(\{\pm v_1,\ldots,v_d\}\right) \subseteq \widehat{\mathcal{E}} \subseteq \bigcap_{i=1}^{d}\widehat{\mathcal{E}}_i \subseteq C\cdot\bigcap_{i=1}^{d}\mathcal{E}_{OPT(i)}$$

We therefore want to argue about $\widehat{\mathcal{E}}$ given that it must contain $\mathsf{conv}\left(\{\pm v_1,\ldots,v_d\}\right)$ and be contained by $C\cdot\bigcap_{i=1}^{d}\mathcal{E}_{OPT(i)}$. Let A be a matrix mapping $\widehat{\mathcal{E}}$ to the unit ball. Then, notice that we can write for all $i\in[d]$:

$$\|Av_i\| \leq 1 \qquad\qquad \left\|A\cdot\frac{Ce_i}{\sqrt{d-\varepsilon}}\right\| \geq 1$$

In particular, the rightmost exclusion follows from the fact that $Ce_i/\sqrt{d-\varepsilon}$ lies on the boundary of $C\cdot\bigcap_{i=1}^{d}\mathcal{E}_{OPT(b.i)}$. Now, recall the well-known fact that for any unitary matrix W, we have $\|AW\|_F = \|A\|_F$ (see, e.g., [HJ91]), and observe that we have:

$$d \geq \sum_{i=1}^{d}\|Av_i\|^2 = \|AV\|_F^2 = \|A\|_F^2 = \|AI\|_F^2 = \sum_{i=1}^{d}\|Ae_i\|^2 \geq \frac{d(d-\varepsilon)}{C^2}$$

Rearranging gives $C\geq\sqrt{d-\varepsilon}$, as desired. ■

Appendix C. Dual Problem – Inner Ellipsoidal Approximation

We design our algorithms for Problem 2 in the setting where we receive points $x_1,\ldots,x_n$ defining $X=\mathsf{conv}\left(\{\pm x_1,\ldots,\pm x_n\}\right)$. Alternatively, we may define a centrally symmetric convex polytope is by providing a set of its faces, or more generally, a set of slabs of the form $\{x:|a^Tx|\leq 1\}$. Accordingly, we may consider a different online model where inequalities $\{x:|a^Tx|\leq 1\}$ arrive one-by-one and the resulting polytope is their intersection. Using the notion of a polar set, we show that this model is essentially equivalent to the model we study in this paper. All our results equally apply to it.

Thus, another possible formulation for Problem 2 involves the algorithm receiving the linear constraints one-at-a-time instead of points from the body.

In this section, we show that this choice of formulation does not matter. Specifically, an algorithm for one of these variants yields an algorithm for the other. In fact, as written in Table 1, these problems are dual to one another.

Primal	**Dual**
Find $\mathcal{E}$ (outer ellipsoid) such that:	Find $\mathcal{E}$ (inner ellipsoid) such that:
$\mathcal{E}/\alpha \subseteq X \subseteq \mathcal{E}$	$\mathcal{E} \subseteq Y \subseteq \alpha \cdot \mathcal{E}$
where $X = \mathsf{conv}(\{\pm x_1, \dots, \pm x_n\})$	where $Y = \{y \ : \ \lvert\langle x_i, y\rangle\rvert \leq 1 \text{ for all } i \in [n]\}$

Table 1: The primal and dual version of the ellipsoidal approximation problem.

We first address the duality between the two problems in Table 1. To do so, observe the following useful facts regarding convex polars.

- If A and B are convex bodies, and if $A \subseteq B$, then $B^\circ \subseteq A^\circ$ (see Proposition 7.16(iv) in [BC17]).
- In Table 1, $X^\circ = Y$, and $Y^\circ = X$.
- If an ellipsoid $\mathcal{E} = \{x \ : \ \|Ax\| \leq 1\}$, then $\mathcal{E}^\circ = \left\{x \ : \ \left\|A^{-T}x\right\| \leq 1\right\}$ (see Definition 2.17 in [Vis21]).

The first and third are well-known, and the second follows from the definition of the polar and that if X is closed, convex, and contains the origin, then $(X^\circ)^\circ = X$ (see Corollary 7.19(i) in [BC17]).

We now put these facts together to show that a solution to the primal problem can be converted to one for the dual problem. First, notice that we have $\mathcal{E}$ and α such that $\mathcal{E}/\alpha \subseteq X \subseteq \mathcal{E}$. Using the first fact, we have $\mathcal{E}^\circ \subseteq X^\circ \subseteq (\mathcal{E}/\alpha)^\circ$. Using the second fact, we have $\mathcal{E}^\circ \subseteq Y \subseteq (\mathcal{E}/\alpha)^\circ$. Finally, using the third fact, we have $(\mathcal{E}/\alpha)^\circ = \alpha \cdot \mathcal{E}^\circ$, which yields $\mathcal{E}^\circ \subseteq Y \subseteq \alpha \cdot \mathcal{E}^\circ$. A similar argument shows that a solution to the dual yields a solution to the primal.

We now address how a solution to the streaming variant of the primal problem can be converted to a solution to the streaming variant of the dual problem. Specifically, suppose we are in the dual setting, wherein we receive linear constraints one-at-a-time. Our task is to find an α-ellipsoidal approximation to $Y_t = \{y \ : \ \lvert\langle x_i, y\rangle\rvert \leq 1 \text{ for all } i \in [t]\}$. Observe that every incoming linear constraint $\{y \ : \ \lvert\langle x_i, y\rangle\rvert \leq 1\}$ can be treated as an incoming point $\pm x_i$ in the primal space. This means that we can apply our algorithm in the primal setting to obtain a solution in the primal space, which for all t gives an ellipsoid such that $X_t \subseteq \mathcal{E}_t$. We then compute the polar of the outer ellipsoid we obtain in the primal space (i.e., $\mathcal{E}_t$) to obtain an inner ellipsoid in the dual space (i.e., $\mathcal{E}_t^\circ$), which yields $\mathcal{E}_t^\circ \subseteq Y_t$. As per our previous argument, this preserves the approximation factor – at the end of the stream, we have $\mathcal{E}_n^\circ \subseteq Y \subseteq \alpha \cdot \mathcal{E}_n^\circ$, as desired.

Appendix D. Proof of Claim 8

Note that the volume of the ellipsoid determined by A_t is proportional to $\det\left(A_t^{-1}\right)$. Therefore, A_t is the solution to the following optimization problem, where we use that the volume of the ellipsoid determined by A_t is proportional to $\det\left(A_t^{-1}\right)$.

$$\max \det\left(A_t\right) \text{ such that } A_t \preceq A_{t-1} \text{ and } \|A_t x_t\| \leq 1$$

Additionally, since $\det(AB) = \det(A) \cdot \det(B)$, we have that this objective is invariant under linear transformations. It thus follows that our objective can be rewritten as:

$$\begin{aligned}
&\max \det(A_t) && \text{such that } A_t \preceq A_{t-1} \text{ and } \|A_t x_t\| \leq 1 \\
\equiv &\max \det\left(A_t \cdot A_{t-1}^{-1}\right) && \text{such that } A_t \cdot A_{t-1}^{-1} \preceq I \text{ and } \left\|\left(A_t \cdot A_{t-1}^{-1}\right) A_{t-1} x_t\right\| \leq 1 \\
\equiv &\max \det\left(\widehat{A}\right) && \text{such that } \widehat{A} \preceq I \text{ and } \left\|\widehat{A} A_{t-1} x_t\right\| \leq 1
\end{aligned}$$

where the last line follows from using the intermediate variable $\widehat{A} = A_t \cdot A_{t-1}^{-1}$.

In other words, after the transformation, the problem is equivalent to finding the minimum volume ellipsoid that contains (i) the unit ball and (ii) point $A_{t-1}x_t$. Geometrically, it is clear what the optimal ellipsoid for this problem is: one of its semi-axes is $A_{t-1}x_t$; all others are orthogonal to $A_{t-1}x_t$ and have length 1 (this can be formally proved using symmetrization). However, we do not use this observation and derive a formula for $\widehat{A}$ using linear algebra.

We first give an upper bound on the objective value of the above optimization problem. Since $\widehat{A} \preceq I$, we have that all its singular values must be at most 1. Additionally, since $1 \geq \left\|\widehat{A} A_{t-1} x_t\right\| \geq \sigma_{\min}\left(\widehat{A}\right) \cdot \|A_{t-1}x_t\|$, we have that at least one singular value of $\widehat{A}$ must be $\leq 1/\|A_{t-1}x_t\|$. Putting everything together and using the fact that the determinant is the product of the singular values gives $\det\left(\widehat{A}\right) \leq 1/\|A_{t-1}x_t\|$.

We now show that there exists a setting of $\widehat{A}$ that achieves this upper bound. Let v_1 be a unit vector in the direction of $A_{t-1}x_t$ and $v_2, \ldots, v_d$ complete the orthonormal basis for $\mathbb{R}^d$ from v_1, and write $\widehat{A} = \frac{1}{\|A_{t-1}x_t\|} v_1 v_1^T + \sum_{i=2}^{d} v_i v_i^T$. We will show that $\widehat{A}$ satisfies the constraints imposed by the optimization problem. Since we have $\|A_{t-1}x_t\| \geq 1$ (as we impose that $x_t \notin \mathcal{E}_{A_{t-1}}$), the fact that $\widehat{A} \preceq I$ follows immediately. For the second constraint, we write:

$$\left\|\widehat{A} A_{t-1} x\right\| = \left\|\left(\frac{1}{\|A_{t-1}x_t\|} v_1 v_1^T + \sum_{i=2}^{d} v_i v_i^T\right) A_{t-1} x_t\right\| = \left\|\frac{A_{t-1}x_t}{\|A_{t-1}x_t\|}\right\| = 1$$

Furthermore, it is easy to see that $\det\left(\widehat{A}\right) = 1/\|A_{t-1}x_t\|$, which achieves our upper bound.

Finally, recall that we wrote $\widehat{A} = A_t \cdot A_{t-1}^{-1}$; rearranging this gives us what we want.

Appendix E. Proofs from Section 4

For the purposes of our analysis, we will "simulate" the singular value correction step using the following procedure. Let $w_1, \ldots, w_d$ be the the i-th column of V_t (note that the w_i-s are unit vectors that are the directions of the semi-axes of $\mathcal{E}_t$). Let $\mathcal{E}'_t$ be the ellipsoid obtained prior to Line 13. i.e., $\mathcal{E}'_t = \left\{x : \left\|(\Sigma'_t)^{-1} V_t^T x\right\| \leq 1\right\}$. We create "ghost" points $z_1, \ldots, z_d$: $z_i = \tau_{i,t} w_i = \max([\Sigma'_t]_{ii}, M_t/\xi) w_i$ (see line 13 in Algorithm 2). Note that X contains the ball of radius M_t/ξ centered at 0, since the aspect ratio of X is at most ξ. Thus, each point z_i either lies in X (if $\|z_i\| = M_t/\xi$) or in $\mathcal{E}'_t$ (if $\|z_i\| = [\Sigma'_t]_{ii}$). We finally start with matrix A_{t-1} and consecutively apply the update rule from Algorithm 1 for each of the points $x_t, z_1, \ldots, z_d$.

Next, we show Lemma 26, which states that this simulation of the singular value correction step yields A_t, the same matrix that we obtain when we perform the update rule from Algorithm 2.

Lemma 26 *The process described above yields matrix A_t.*

Proof Consider the executions of Algorithms 1 and 2. After Algorithm 1 processes point x_t and Algorithm 2 executes Line 12, both algorithms are in the same state. Namely they store matrix A'_t given by Claim 8,

$$A'_t = A_{t-1} - \left(1 - \frac{1}{\|A_{t-1}x_t\|}\right)\left(\frac{(A_{t-1}x_t)\,(A_{t-1}x_t)^T}{\|A_{t-1}x_t\|^2}\right)A_{t-1}$$

It remains to show that executing Line 13 in Algorithm 2 is equivalent to injecting these "ghost" points $z_1, \ldots, z_d$ into Algorithm 1.

It is sufficient to consider the effect of injecting one point z_i. Assume we started with matrix A and obtained matrix A' by injecting z_i. Observe that if $\sigma_i(A) \le 1/\|z_i\|$ (that is, z_i lies in the ellipsoid defined by A; in particular, if $z_i \in \mathcal{E}'_t$), then $A' = A$. We prove that if $\sigma_i(A) > 1/\|z_i\|$, then $A' = U\Sigma'^{-1}V^T$ where $A = U\Sigma^{-1}V^T$, entry $\Sigma'_{ii} = \|z_i\|$, and all other entries of Σ' are equal to the corresponding entries of Σ. We have,

$$A' = \widehat{A}A = \left(I - \left(1 - \frac{1}{\|Az_i\|}\right)\left(\frac{(Az_i)\,(Az_i)^T}{\|Az_i\|^2}\right)\right)A$$

Since $z_i = \tau_{i,t}w_i$, we have $V^Tz_i = \|z_i\|e_i = \tau_{i,t}e_i$. Accordingly, $Az_i = U\Sigma^{-1}\tau_{i,t}e_i = \frac{\tau_{i,t}}{\Sigma_{ii}}Ue_i$. Thus, $\widehat{A} = I - (1 - \frac{\Sigma_{ii}}{\tau_{i,t}})Ue_ie_i^TU^T$. Since U is an orthogonal matrix, $UU^T = I$ and thus

$$A' = \left(I - \left(1 - \frac{\Sigma_{ii}}{\tau_{i,t}}\right)Ue_ie_i^TU^T\right)U\Sigma^{-1}V^T = U\underbrace{\left(\Sigma^{-1}\left(I - \left(1 - \frac{\Sigma_{ii}}{\tau_{i,t}}\right)e_ie_i^T\right)\right)}_{\Sigma'^{-1}}V^T.$$

Note that $I - \left(1 - \frac{\Sigma_{ii}}{\tau_{i,t}}\right)e_ie_i^T$ is a diagonal matrix; all of its diagonal entries are equal to 1 except for the i-th diagonal entry, which is $\Sigma_{ii}/\tau_{i,t}$. We get that Σ'^{-1} differs from Σ^{-1} only in the i-th diagonal entry: $(\Sigma'^{-1})_{ii} = 1/\tau_{it}$, as required. ■

Proof [Proof of Lemma 19] By Lemma 26, we can break the evolution of our potential functions into two main steps: that after Algorithm 1 gets x_t and that after Algorithm 1 gets the ghost points. Let S'_t, P'_t, Q_t represent $\left\|J^* \cdot (A'_t)^{-1}\right\|_F^2$, $2\log\det\left((A'_t)^{-1}\right)$, and $\max_{i\in[t]}\|J^*x_i\|^2$, respectively, where A'_t is the matrix defined in Lemma 26.

By Lemma 16, for all t, we have $S'_t - S_{t-1} \le (P'_t - P_{t-1}) \cdot Q_t$. Similarly, as $\|J^*x_t\| \le 1$ and $\|A_{t-1}x_t\| > 1$, we have:

$$S'_t - S_{t-1} = \left(1 - \frac{1}{\|A_{t-1}x_t\|^2}\right)\|J^*x_t\|^2 \le \left(1 - \frac{1}{\|A_{t-1}x_t\|^2}\right)Q_t \le Q_t$$

We now analyze the singular value correction step. By Lemma 26,it can be simulated by adding the ghost points $z_1, \ldots, z_d$. First, observe that we only need to analyze points z_i with $z_i = M_t/\xi w_i$

(because other points are in $\mathcal{E}'_t$ and do not cause any update). We now show that $\|J^* z_i\| \leq \max_{i\in[t]} \|J^* x_t\| = \sqrt{Q_t}$. We have:

$$\|J^* z_i\| = \frac{M_t}{\xi} \cdot \|J^* w_i\| \leq M_t \cdot \frac{\sigma_{\max}(J^*)}{\xi} \leq \sigma_{\min}(J^*) \cdot M_t$$

Let $p \coloneqq \operatorname{argmax}_{i\in[t]} \|x_i\|$. Then:

$$\|J^* z_i\| \leq \sigma_{\min}(J^*) \cdot M_t = \sigma_{\min}(J^*) \cdot \|x_p\| \leq \|J^* x_p\| \leq \max_{i\in[t]} \|J^* x_t\| = \sqrt{Q_t}.$$

Consider the state of Algorithm 1 in the simulation after it gets points $z_1, \ldots, z_i$. Let $A'_{t,i}$ be the resulting state matrix, and $S'_{t,i}$ and $P'_{t,i}$ be the values of functions S and P. Observe that $S'_{t,0} = S'_t$ and $P'_{t,0} = P'_t$ and that $S_t = S'_{t,d}$ and $P_t = P'_{t,d}$. We now invoke Lemma 16 repeatedly:

$$\begin{aligned}
S'_{t,0} - S_{t-1} &\leq (P'_{t,0} - P_{t-1}) \cdot Q_t \\
S'_{t,1} - S'_{t,0} &\leq (P'_{t,1} - P'_{t,0}) \cdot Q_t \\
&\vdots \\
S'_{t,i} - S'_{t,i-1} &\leq \left(P'_{t,i} - P'_{t,i-1}\right) \cdot Q_t \\
&\vdots \\
S_t - S'_{t,d-1} &\leq \left(P_t - P'_{t,d-1}\right) \cdot Q_t
\end{aligned}$$

Adding all these inequalities yields $S_t - S_{t-1} \leq (P_t - P_{t-1}) \cdot Q_t$. Since numbers Q_t are non-decreasing, this implies that for all $u \leq t$, we have $S_t \leq S_u + Q_t \cdot (P_t - P_u)$.

Similarly, we repeatedly write:

$$\begin{aligned}
S'_{t,0} - S_{t-1} &\leq Q_t \\
S'_{t,1} - S'_{t,0} &\leq Q_t \\
&\vdots \\
S'_{t,i} - S'_{t,i-1} &\leq Q_t \\
&\vdots \\
S_t - S'_{t,d-1} &\leq Q_t
\end{aligned}$$

Note that at least one of the semi-axes of $\mathcal{E}'_t$ has length at least M_t (since all points $x_1, \ldots, x_t$ are in $\mathcal{E}'_t$). That is, $[\Sigma']_{ii} \geq M_t \geq M_t/\xi_i$ for some point z_i. This means that we do not perform any updates for z_i and $S'_{t,i} = S'_{t,i-1}$. Summing up the inequalities above and taking into account that $S'_{t,i} = S'_{t,i-1}$ for at least one i yields $S_t \leq S_{t-1} + d \cdot Q_t$, as required. ■

Proof [Proof of Lemma 20] Consider ellipsoid $\mathcal{E}_1$. It has semi-axes $\tilde{w}_1 = x_1$ and some $\tilde{w}_2, \ldots, \tilde{w}_d$. Since $\|\tilde{w}_i\| = \|x_1\|/\xi$ for $i \geq 2$ (see step 5 of Algorithm 2), all points $\tilde{w}_i$ lie inside $X \subset \mathcal{E}^*$. In particular, $\|J^* \tilde{w}_i\| \leq 1$. Finally, note that $A_1 \tilde{w}_i = e_i$. We have:

$$S_1 = \|J^* A_1^{-1}\|_F = \sum_{i=1}^{d} \|J^* A_1^{-1} e_i\|^2 = \sum_{i=1}^{d} \|J^* \tilde{w}_i\|^2 \leq d$$

■

Proof [Proof of Lemma 21] Set $t_0 = n$, and define $t_i \in \{1, \dots, n\}$ for $i \geq 1$ recursively as follows:

1. Set $t_i = \max\{j \in \{1, \dots, n\} \colon Q_j < {}^{Q_{t_{i-1}}}/_{e}\}$
2. If there is no such j, then finish.

Let $t_0 > \ldots > t_m$ be the indices defined by this process. Observe that for each $0 \leq i \leq m$, we have $Q_{t_i} \leq e^{-i}$. Further, for each $0 \leq i \leq m-1$, we have ${}^{Q_{t_i}}/_{Q_{t_{i+1}+1}} \leq e$, and ${}^{Q_{t_m}}/_{Q_1} \leq e$.

Now we apply bound (4.1). For every $0 \leq i \leq m-1$, we have

$$S_{t_i} \leq S_{t_{i+1}+1} + Q_{t_i}(P_{t_i} - P_{t_{i+1}+1})$$

From bound (4.2), we get

$$S_{t_{i+1}+1} \leq S_{t_{i+1}} + d \cdot Q_{t_{i+1}+1}.$$

Combining these equations, we obtain for each $0 \leq i \leq m-1$:

$$S_{t_i} - S_{t_{i+1}} \leq d \cdot Q_{t_{i+1}+1} + Q_{t_i} \cdot (P_{t_i} - P_{t_{i+1}+1}) \tag{E.1}$$

Now we add up inequalities (E.1) for all $0 \leq i \leq m-1$ and get:

$$S_{t_m} - S_{t_0} = \sum_{i=0}^{m-1} \left(S_{t_i} - S_{t_{i+1}}\right) \leq d \cdot \sum_{i=0}^{m-1} Q_{t_{i+1}+1} + \sum_{i=0}^{m-1} Q_{t_i} \cdot P_{t_i} + \sum_{i=0}^{m-1} Q_{t_i} \cdot \left(-P_{t_{i+1}+1}\right)$$

By (4.1) we have $S_{t_m} - S_1 \leq Q_{t_m} \cdot (P_{t_m} - P_1)$. Combining this with the previous inequality, we get

$$S_n - S_1 \leq d \cdot \underbrace{\sum_{i=0}^{m-1} Q_{t_{i+1}+1}}_{\text{I}} + \underbrace{\sum_{i=0}^{m} Q_{t_i} \cdot P_{t_i}}_{\text{II}} + \underbrace{(-Q_{t_m} \cdot P_1) + \sum_{i=0}^{m-1} Q_{t_i} \cdot \left(-P_{t_{i+1}+1}\right)}_{\text{III}} \tag{E.2}$$

Now, we bound each individual term in (E.2). For I, observe that as $t_{i+1} + 1 \leq t_i$, we have $Q_{t_{i+1}+1} \leq Q_{t_i} \leq e^{-i}$. Thus

$$\sum_{i=0}^{m-1} Q_{t_{i+1}+1} \leq \sum_{i=0}^{m-1} e^{-i} \leq \sum_{i=0}^{\infty} e^{-i} = \frac{e}{e-1}$$

For II, we have $P_{t_i} \leq P_{t_0}$ for all i, therefore

$$\sum_{i=0}^{m} Q_{t_i} P_{t_i} \leq P_{t_0} \cdot \sum_{i=0}^{m} Q_{t_i} \leq P_{t_0} \cdot \frac{e}{e-1}$$

Now we bound III. First, we prove the following lemma to bound terms of the form $-Q_t \cdot P_u$:

Lemma 27 *Consider indices $u \le t$. If $Q_t/Q_u \le e$, then*

$$-Q_t \cdot P_u \le Q_t \cdot \left(d + 4d \log \xi + d \log \left(\frac{1}{Q_t} \right) \right)$$

Proof By definition, we have $-P_u = -\sum_{i=1}^{d} \log(\sigma_{i,u}^2)$. Next, notice that for all $t \in \{1, \ldots, u\}$:

$$\begin{aligned}
\sigma_{\min}\left(J^* \cdot A_u^{-1}\right) &\ge \sigma_{\min}\left(J^*\right) \cdot \sigma_{\min}\left(A_u^{-1}\right) \\
&\ge \sigma_{\min}\left(J^*\right) \cdot \frac{\|x_t\|}{\xi} \qquad \text{since the correction step ensures that } \sigma_{\min}\left(A_u^{-1}\right) \ge \frac{\|x_t\|}{\xi} \\
&\ge \frac{\sigma_{\max}\left(J^*\right)}{\xi} \cdot \frac{\|x_t\|}{\xi} \\
&\ge \frac{\|J^* x_t\|}{\xi^2}
\end{aligned}$$

Since this is true for all $t \in [u]$, we can maximize the RHS over $t \in [u]$, and we obtain $\sigma_{i,u}^2 \ge Q_u/\xi^4$. Hence, $-P_u \le d \log\left(1/Q_u\right) + 4d \log \xi$. As $1/Q_u \le e/Q_t$, we get $-P_u \le d + 4d \log \xi + d \log\left(1/Q_t\right)$. Multiplying by Q_t gives the claim. ■

Applying Lemma 27 to each term in III, we obtain

$$-Q_{t_m} \cdot P_1 + \sum_{i=0}^{m-1} Q_{t_i} \cdot -P_{t_{i+1}+1} \le (d + 4d \log \xi) \cdot \sum_{i=0}^{m} Q_{t_i} + d \cdot \sum_{i=0}^{m} Q_{t_i} \log\left(\frac{1}{Q_{t_i}}\right)$$

As before, we can bound the first term with $\sum_{i=0}^{m} Q_{t_i} \le \frac{e}{e-1}$. For the second term, observe that $y \mapsto y \log(1/y)$ is increasing on $[0, 1/e]$. Thus, $Q_{t_i} \log(1/Q_{t_i}) \le e^{-i} \log(1/e^{-i}) = e^{-i} \cdot i$ for $i \ge 1$. The maximum of $y \log\left(1/y\right))$ is $1/e$; therefore $Q_{t_0} \log(1/Q_{t_0}) \le 1/e$. Thus we can bound the second term by

$$d \cdot \left(\frac{1}{e} + \sum_{i=1}^{m} i \cdot e^{-i} \right) \le d \cdot \left(\frac{1}{e} + \sum_{i=1}^{\infty} i \cdot e^{-i} \right) = d \cdot \left(\frac{1}{e} + \frac{e}{(1-e)^2} \right).$$

To summarize, we bound term III by

$$d \cdot \left(\frac{1}{e} + \frac{e}{e-1} + \frac{e}{(1-e)^2} \right) + 4d \log \xi \cdot \frac{e}{e-1}$$

Combining the bounds on the terms of (E.2) results in:

$$S_n - S_1 \le d \cdot \left(\frac{1}{e} + \frac{2 \cdot e}{e-1} + \frac{e}{(1-e)^2} \right) + P_{t_0} \cdot \frac{e}{e-1} + 2d \log \xi \cdot \frac{e}{e-1}$$

Rearranging and applying Theorem 20, we obtain

$$S_n - P_n \cdot \frac{e}{e-1} \le d \cdot \left(1 + \frac{1}{e} + \frac{2 \cdot e}{e-1} + \frac{e}{(1-e)^2} \right) + 2d \log \xi \cdot \frac{e}{e-1} \tag{E.3}$$

Observe that $S_n - P_n \cdot \frac{e}{e-1} = \sum_{i=1}^{d} (\sigma_{i,n}^2 - \frac{e}{e-1} \log(\sigma_{i,n}^2))$. As $y - \frac{e}{e-1} \log y \geq \frac{e-2}{e-1}\, y$ for all $y > 0$, we get

$$\sum_{i=1}^{d} \sigma_{i,n}^2 \leq \frac{e-1}{e-2} \left(S_n - P_n \cdot \frac{e}{e-1} \right)$$

Using (E.3) and replacing constants with their integer ceilings, we finish with

$$\sum_{i=1}^{d} \sigma_{i,n}^2 \leq 3 + 14d \log \xi + 8d$$

■