

One-Pass Randomized Algorithm with Practical Rangefinder for Low-Rank Approximation to Quaternion Matrices

Chao Chang*

Yuning Yang^{*†}

April 24, 2024

Abstract

As its real/complex counterparts, randomized algorithms for low-rank approximation to quaternion matrices received attention recently. For large-scale problems, however, existing quaternion orthogonalization methods are not efficient, leading to slow rangefinders. By relaxing orthonormality while maintaining favorable condition numbers, this work proposes two practical quaternion rangefinders that take advantage of mature scientific computing libraries to accelerate heavy computations. They are then incorporated into the quaternion version of a well-known one-pass algorithm. Theoretically, we establish the probabilistic error bound, and demonstrate that the error is proportional to the condition number of the rangefinder. Besides Gaussian, we also allow quaternion sub-Gaussian test matrices. Key to the latter is the derivation of a deviation bound for extreme singular values of a quaternion sub-Gaussian matrix. Numerical experiments indicate that the one-pass algorithm with the proposed rangefinders work efficiently while only sacrificing little accuracy. In addition, we tested the algorithm in an on-the-fly 3D Navier-Stokes equation data compression to demonstrate its efficiency in large-scale applications.

Keywords: Randomized Algorithm, Rangefinder, One-pass, Quaternion matrix, Sketching, Low-rank approximation, sub-Gaussian

1 Introduction

1.1 Background

Low-rank matrix approximation (LRMA) has been applied in various applications. In the big data era, large amounts of data are being captured and generated through various channels, such as high-definition color video, scientific simulations, and artificial intelligence training sets. This trend poses challenges in terms of computation time, storage, and memory costs to LRMA. In 2011, a randomized SVD algorithm (HMT) [14] was proposed by Halko, Martinsson, and Tropp, which uses a random sketch to obtain an oversampling approximation before implementing the truncated SVD. Compared to the deterministic SVD, the randomized one runs faster with adjustable precision loss. It is robust, but there is still a need to revisit the original data during the low-rank approximation. In 2017, Tropp et al. [41] proposed a one-pass randomized algorithm using two sketches, which needs to visit the data only once. It is more effective for managing data with limited storage, arithmetic, and communication capabilities. Later on, the

*College of Mathematics and Information Science, Guangxi University, Nanning, 530004, China

[†]Corresponding author: Yuning Yang, yyang@gxu.edu.cn.

authors developed also a one-pass algorithm using three sketches and applied it to streaming data [42]. Prior to these work, randomized algorithms have been studied extensively in the literature; see, e.g., [1, 46, 8, 30, 45, 9, 4], and the recent surveys [40, 20, 21, 34, 31, 32].

Despite the noncommutativity in quaternion multiplications, quaternion matrices have been widely used in various applications such as signal processing [11], color image analysis [39, 38], and machine learning [48, 33] in recent years. Randomized quaternion low-rank matrix approximation has garnered increasing attention very recently. Liu et al. [27] developed a randomized quaternion SVD algorithm based on the HMT framework by using structure-preserving quaternion QR and quaternion SVD, and studied its error bound. This algorithm was later applied to nonnegative pure quaternion matrix approximation [29]. Ren et al. [35] proposed a randomized quaternion QLP decomposition algorithm. Li et al. [23] also proposed a randomized block Krylov subspace algorithm with improved approximation accuracy. Very recently, a fixed-precision randomized quaternion SVD was studied in [28].

The framework of the HMT algorithm [14] (also [41, 42], and the quaternion randomized algorithms [27, 35, 23]) can be divided into a randomized QB approximation stage and a truncation stage. Initially, the QB stage involves generating a sketch of the input data matrix through randomized oversampling with Gaussian or alternative embeddings. This then leads to the formation of the Q matrix, representing an orthonormal basis for the range of the sketch, known as the *rangefinder* step. The B matrix is then determined, either exactly [14] or approximately [41, 42]. In the truncation stage, truncated SVD or other deterministic methods are performed on the B matrix to find a more accurate fixed-rank approximation.

1.2 Quaternion rangefinders

In the real/complex case, constructing an orthonormal rangefinder is cheap. Fast and stable algorithms for orthogonalization have matured, and highly optimized implementations are available, such as MATLAB’s built-in function `qr`, and various QR routines in LAPACK [3] and the Intel Math Kernel Library (MKL) [10].

In the quaternion case, orthogonalization approaches are developing. Classical QR decomposition methods can be extended to the quaternion arithmetic with little modifications. For example, the quaternionic Householder QR was proposed for quaternion eigenvalue computations [5]; this was implemented by Sangwine and Le Bihan in the quaternion toolbox for MATLAB (QTFM) [36] with the function name `qr`. To speed up quaternion matrix computations, in a series of papers [18, 24, 19, 7], the authors proposed structure-preserving algorithms. This type of algorithms is promising, as its basic idea is to only operate on the compact real representation of a quaternion matrix, avoiding quaternion operations and smartly reducing computational complexity. The structure-preserving quaternion Householder QR (QHQR) [19, 26] is numerically stable and accurate; the structure-preserving quaternionic modified Gram-Schmit (QMGS) [44] is more economic but may lose accuracy. These methods together with QTFM’s `qr` function were employed for orthonormal rangefinders by the quaternion randomized algorithms [27, 23, 35].

1.3 Limitation and motivation

The above quaternionic orthogonalization methods are efficient in small and moderate problems; however, for large-scale matrices, they are still expensive. For instance, even QMGS (which is the fastest one mentioned in the previous subsection) is at least two order of magnitude slower than `qr` in MATLAB for 5000×500 matrices; see Fig. 1¹. One of the possible reasons is that

¹The code of QHQR was downloaded from http://maths.jsnu.edu.cn/_upload/article/files/40/5c/0abecd234d2c909be8b4fac9c4ad/ and that of QMGS was implemented from [2]. MATLAB’s `qr` and `svd` are applied to the full complex representation of the quaternion matrix, which is of size $2m \times 0.2m$; see the next section. As complex QR and

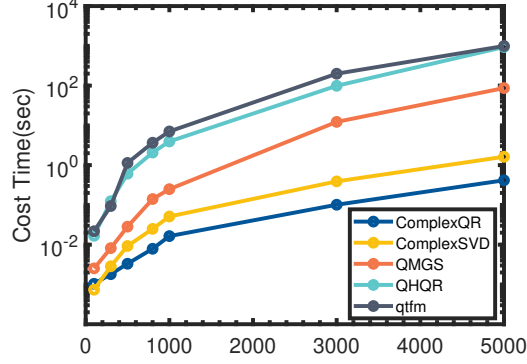


Figure 1: Running time comparisons of QTFM’s `qr`, `QHQR`, `QMGS`, and MATLAB’s `qr` and `svd` (applied to the full complex representation). m from 100 to 5000, $n = 0.1m$

highly optimized implementations of these quaternion orthogonalization methods are far from developed. For example, MATLAB’s built-in functions such as `qr` and `svd` do not support quaternions yet; nor MKL. Without an elegant implementation, even an advanced method cannot take advantage of the features such as parallelism and memory management in modern computing architecture, holding back its advancement.

To pursue speed, a common design philosophy is to trade accuracy, akin to randomized algorithms. Following this vein, it is possible to consider a non-orthonormal rangefinder. If so, what criteria should the new rangefinder meet, and how to quantify the loss of orthonormality?

Within the HMT and RQSVD frameworks, the orthonormal rangefinder plays two roles. One is to improve the numerical stability in matrix decompositions [14, Section 4.1]. The other is to explode the singular values and their corresponding singular vectors in the truncation stage, which helps in better approximating the original matrix. To quantify the accuracy, it is natural to take the condition number of the rangefinder into account. Obviously, an orthonormal rangefinder gives the optimal condition number, while deviations from orthonormality should maintain a condition number that does not indicate severe ill-conditioning.

When the criterion above is met, it is expected that the new rangefinder is built upon mature scientific computing libraries or advanced algorithms in the real/complex arithmetic to fully utilize the modern computing architecture.

1.4 This work

The general idea throughout this work is to transform heavy quaternion computations to QR, SVD, and solving linear equations in the complex arithmetic. To this end, we employ compact or full complex representations of quaternion matrices as intermediaries, accepting potential reductions in orthonormality and accuracy to maintain favorable condition numbers. The influence of condition numbers on approximation accuracy will be examined theoretically. Specifically:

In Section 3, we provide two range-preserving rangefinders, termed pseudo-QR and pseudo-SVD. Theoretically, Pseudo-QR can reduce the condition number of the sketch from 10^8 within 10. Pseudo-SVD on the other hand generates an orthonormal matrix even with very ill-

SVD cannot directly generate an orthonormal rangefinder in the quaternion domain, the purpose of this rough comparison is to show that the speed of computing the rangefinder still has a large room to improve.

conditioned sketch. Almost all the computations are built upon established scientific computing libraries in the complex arithmetic to ensure their efficiency.

In Section 4, the proposed rangefinders are incorporated into the quaternion version of the one-pass framework of [41].

Our theoretical findings in Sections 4, 5, and 6 are summarized as follows:

- In the QB approximation stage, the approximation error, measured by the tail energy, is independent of the condition number of the rangefinder;
- In the truncation stage, the truncation error is proportional to the condition number of the rangefinder. This and the previous point ensure the reasonability of using a non-orthonormal yet well-conditioned rangefinder in theory.
- We derive a deviation bound for extreme singular values of a quaternion sub-Gaussian matrix, which may be of independent interest. This theoretically justifies the use of quaternion sub-Gaussian test matrices.

Some comments are in order.

- The result of the first two points applies to any range-preserving while non-orthonormal rangefinders.
- The deviation bound of extreme singular values of a quaternion Gaussian matrix was first studied in [27], while ours in the sub-Gaussian case generalizes a result in the real case by Vershynin [43]. Specifically, with high probability, the singular values of a tall $m \times n$ quaternion sub-Gaussian matrix are shown to lie in the interval $[2\sqrt{m} - 2C\sqrt{n} - t, 2\sqrt{m} + 2C\sqrt{n} + t]$ for some $C, t > 0$.

Finally, Section 7 evaluates the performance of our algorithm and other ones through various experiments using synthetic data. We apply our method to compression of scientific simulation data from 3D Navier-Stokes equations and a 4D Lorenz-type chaotic system to demonstrate its efficiency in large-scale problems.

Our implementation in MATLAB is available at github.com/Mitchell-Cxyk/RQLRMA, which can be ran with either CPU or GPU.

2 Preliminaries on Quaternions

Quaternions were invented by Sir William Rowan Hamilton in 1843. A quaternion scalar is of the form $a = a_w + a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k}$ where a_w, a_x, a_y, a_z are real numbers. The sum of quaternions is defined component-wise and the their multiplication is determined by the following rules along with the associative and distributive laws $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. For $a, b \in \mathbb{Q}$, $ab \neq ba$ in general. A quaternion matrix $\mathbf{Q} \in \mathbb{Q}^{m \times n}$ is defined as $\mathbf{Q} = Q_w + Q_x\mathbf{i} + Q_y\mathbf{j} + Q_z\mathbf{k}$, where $Q_i (i = w, x, y, z) \in \mathbb{R}^{m \times n}$. The conjugate and conjugate transpose of $\mathbf{Q}$ are respectively denoted as $\overline{\mathbf{Q}} := Q_w - Q_x\mathbf{i} - Q_y\mathbf{j} - Q_z\mathbf{k}$ and $\mathbf{Q}^* := Q_w^T - Q_x^T\mathbf{i} - Q_y^T\mathbf{j} - Q_z^T\mathbf{k}$. For two quaternion matrices $\mathbf{P}$ and $\mathbf{Q}$ of proper size, $\mathbf{P}^*\mathbf{Q} = (\mathbf{Q}^*\mathbf{P})^*$ while $\overline{\mathbf{P}^*\mathbf{Q}} \neq \overline{\mathbf{P}}^* \cdot \overline{\mathbf{Q}}$. More properties are referred to [47].

2.1 Quaternion vector space

Considering vectors with quaternion coordinates, a module over the ring $\mathbb{Q}$ is usually called the quaternion right vector space under the summation and the right scalar multiplication. Given quaternion vectors $\mathbf{v}_1, \dots, \mathbf{v}_r$, they are right linearly independent if for quaternions $k_1, \dots, k_r$,

$$\mathbf{v}_1 k_1 + \mathbf{v}_2 k_2 + \dots + \mathbf{v}_r k_r = 0 \quad \text{implies} \quad k_i = 0, \quad i = 1, \dots, r.$$

Most linear algebra concepts and results can be transplanted to the right vector space in parallel [47] and throughout this work, we always omit the prefix “right”.

Given $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_r]$, its range is defined as

$$\mathcal{R}(\mathbf{V}) = \text{span}(\mathbf{v}_1, \dots, \mathbf{v}_r) = \left\{ \sum_{i=1}^r \mathbf{v}_i k_i \mid k_i \in \mathbb{Q}, i = 1, \dots, r \right\}.$$

The inner product between $\mathbf{u}$ and $\mathbf{v}$ is given by $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^* \mathbf{v}$ [17, 6]. They are orthogonal if $\langle \mathbf{u}, \mathbf{v} \rangle = 0$ and written as $\mathbf{u} \perp \mathbf{v}$. If $\langle \mathbf{u}, \mathbf{u} \rangle = \langle \mathbf{v}, \mathbf{v} \rangle = 1$, they are called orthonormal quaternion vectors. Considering a quaternion linear space $\mathcal{V}$, for a subspace $\mathcal{L} \subset \mathcal{V}$, its orthogonal complement is defined as $\mathcal{L}^\perp = \{\mathbf{v} \in \mathcal{V} : \langle \mathbf{v}, \mathbf{w} \rangle = 0, \forall \mathbf{w} \in \mathcal{L}\}$.

2.2 Complex representation

$\mathbf{Q} \in \mathbb{Q}^{m \times n}$ can be represented as $\mathbf{Q} = Q_0 + Q_1 \mathbf{j}$, where $Q_0, Q_1 \in \mathbb{C}^{m \times n}$ with $Q_0 = Q_w + Q_x \mathbf{i}$ and $Q_1 = Q_y + Q_z \mathbf{i}$. The (full) complex representation of $\mathbf{Q}$ is defined as [47]:

$$\chi_{\mathbf{Q}} := \begin{bmatrix} Q_0 & Q_1 \\ -Q_1 & Q_0 \end{bmatrix} \in \mathbb{C}^{2m \times 2n}.$$

$\chi_{\mathbf{Q}}$ has several nice properties that it is useful in the study of quaternions:

Proposition 2.1 ([47]). *Let $\mathbf{P}, \mathbf{Q}$ be quaternion matrices of proper size. Then*

$$\chi_{k_1 \mathbf{P} + k_2 \mathbf{Q}} = k_1 \chi_{\mathbf{P}} + k_2 \chi_{\mathbf{Q}} \quad (k_1, k_2 \in \mathbb{R}), \quad \chi_{\mathbf{P}\mathbf{Q}} = \chi_{\mathbf{P}} \chi_{\mathbf{Q}},$$

$$\chi_{\mathbf{Q}^*} = \chi_{\mathbf{Q}}^*, \quad \chi_{\mathbf{Q}^{-1}} = \chi_{\mathbf{Q}}^{-1} \text{ if } \mathbf{Q}^{-1} \text{ exists,}$$

$\chi_{\mathbf{Q}}$ is (partially) unitary/Hermitian if and only if $\mathbf{Q}$ is (partially) unitary/Hemitian.

$\chi_{\mathbf{Q}}$ can be partitioned as two blocks:

$$\chi_{\mathbf{Q}} = [\mathbf{Q}_c, \mathbf{Q}_a], \text{ with } \mathbf{Q}_c := \begin{bmatrix} Q_0 \\ -Q_1 \end{bmatrix}, \quad \mathbf{Q}_a := \begin{bmatrix} Q_1 \\ Q_0 \end{bmatrix}.$$

We call $\mathbf{Q}_c$ the *compact* complex representation of $\mathbf{Q}$. $\mathbf{Q}_a$ can be generated from $\mathbf{Q}_c$ by using the symplectic matrix $\mathcal{J} := \begin{bmatrix} 0 & -I_m \\ I_m & 0 \end{bmatrix}$, with $\mathbf{Q}_a = \mathcal{J} \overline{\mathbf{Q}_c}$. The relation between $\mathbf{Q}_c$ and $\mathbf{Q}_a$ is important in the design and analysis of our rangefinders. One can directly check that $\mathcal{J}$ admits the following properties:

Lemma 2.1. *$\mathcal{J}$ -adjoint satisfies:*

$$\begin{aligned} \mathcal{J}^* &= \mathcal{J}^{-1} = -\mathcal{J}; \quad \chi_{\mathbf{Q}} = [\mathbf{Q}_c, \mathcal{J} \overline{\mathbf{Q}_c}]; \quad \mathcal{J}^* \chi_{\mathbf{Q}} \mathcal{J} = \mathcal{J} \chi_{\mathbf{Q}} \mathcal{J}^* = \overline{\chi_{\mathbf{Q}}}; \\ \mathcal{J}^* [v, \mathcal{J} \overline{v}] \mathcal{J} &= \mathcal{J} [v, \mathcal{J} \overline{v}] \mathcal{J}^* = \overline{[v, \mathcal{J} \overline{v}]}, \quad \text{and} \quad \langle v, \mathcal{J} \overline{v} \rangle = 0, \forall v \in \mathbb{C}^{2m}. \end{aligned}$$

The quaternion Moore-Penrose (MP) inverse can be defined similarly as its real/complex counterpart [2, section 1.6]. For $\mathbf{A} \in \mathbb{Q}^{m \times n}$, there exists a unique solution $\mathbf{X}$, denoted as $\mathbf{A}^\dagger$, that satisfies the following four matrix equations

$$\mathbf{A}\mathbf{X}\mathbf{A} = \mathbf{A}, \quad \mathbf{X}\mathbf{A}\mathbf{X} = \mathbf{X}, \quad (\mathbf{A}\mathbf{X})^* = \mathbf{A}\mathbf{X}, \quad (\mathbf{X}\mathbf{A})^* = \mathbf{X}\mathbf{A}.$$

Lemma 2.2. Let $\mathbf{A} \in \mathbb{Q}^{m \times n}$; then $\chi_{\mathbf{A}^\dagger} = (\chi_{\mathbf{A}})^\dagger$.

Proof. The lemma can be proved by checking the MP inverse directly:

$$\begin{aligned}\chi_{\mathbf{A}^\dagger} \chi_{\mathbf{A}} \chi_{\mathbf{A}^\dagger} &= \chi_{\mathbf{A}^\dagger \mathbf{A} \mathbf{A}^\dagger} = \chi_{\mathbf{A}^\dagger}, \\ \chi_{\mathbf{A}} \chi_{\mathbf{A}^\dagger} \chi_{\mathbf{A}} &= \chi_{\mathbf{A} \mathbf{A}^\dagger \mathbf{A}} = \chi_{\mathbf{A}}, \\ (\chi_{\mathbf{A}} \chi_{\mathbf{A}^\dagger})^* &= \chi_{\mathbf{A} \mathbf{A}^\dagger}^* = \chi_{(\mathbf{A} \mathbf{A}^\dagger)^*} = \chi_{\mathbf{A} \mathbf{A}^\dagger} = \chi_{\mathbf{A}} \chi_{\mathbf{A}^\dagger}, \\ (\chi_{\mathbf{A}^\dagger} \chi_{\mathbf{A}})^* &= \chi_{\mathbf{A}^\dagger \mathbf{A}}^* = \chi_{(\mathbf{A}^\dagger \mathbf{A})^*} = \chi_{\mathbf{A}^\dagger \mathbf{A}} = \chi_{\mathbf{A}^\dagger} \chi_{\mathbf{A}}.\end{aligned}$$

□

Lemma 2.3. Let $U \in \mathbb{C}^{2m \times n}$; denote $M := [U, \mathcal{J}\bar{U}]$ and $P_M := MM^\dagger$ the orthogonal projection on to $\mathcal{R}(M)$. Then $\mathcal{J}P_M\mathcal{J}^* = \mathcal{J}^*P_M\mathcal{J} = \overline{P_M}$.

Proof. Denote $\mathbf{U} \in \mathbb{Q}^{m \times n}$ such that $\mathbf{U}_c = U$. Then $\chi_{\mathbf{U}} = M$. Thus

$$\mathcal{J}P_M\mathcal{J}^* = \mathcal{J}\chi_{\mathbf{U}}\chi_{\mathbf{U}}^\dagger\mathcal{J}^* = \mathcal{J}\chi_{\mathbf{U}\mathbf{U}^\dagger}\mathcal{J}^* = \overline{\chi_{\mathbf{U}\mathbf{U}^\dagger}} = \overline{\chi_{\mathbf{U}}\chi_{\mathbf{U}}^\dagger} = \overline{MM^\dagger} = \overline{P_M},$$

where the second and fourth equalities are due to Lemma 2.2 and Proposition 2.1, and the third one comes from Lemma 2.1. Verifying $\mathcal{J}^*P_M\mathcal{J} = \overline{P_M}$ is similar. □

Analogously to the real/complex case, quaternion matrices admit SVD:

Theorem 2.1. (compact QSVD [47, Theorem 7.2]) Let $\mathbf{A} \in \mathbb{Q}^{m \times n}$ ($m \geq n$) be of rank r . Then there exists unitary quaternion matrices $U \in \mathbb{Q}^{m \times m}$, $V \in \mathbb{Q}^{n \times n}$ and diagonal real matrix $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r, 0, \dots, 0) \in \mathbb{R}^{n \times n}$ with $\sigma_1 \geq \dots \geq \sigma_r > 0$, such that $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*$.

The following property, which can be deduced from [47], reveals the relation between the SVD of $\mathbf{A} \in \mathbb{Q}^{m \times n}$ and its complex representation $\chi_{\mathbf{A}}$:

Proposition 2.2. Under the notations in Theorem 2.1, if $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*$, then $\chi_{\mathbf{A}} = \chi_{\mathbf{U}}S\chi_{\mathbf{V}}^*$ with $S = \text{diag}(\Sigma, \Sigma) \in \mathbb{R}^{2n \times 2n}$ is a compact SVD of $\chi_{\mathbf{A}}$, and vice versa.

Lemma 2.4. [2, Section 1.6] Under the notations in Theorem 2.1, if $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*$, then $\mathbf{A}^\dagger = \mathbf{U}\Sigma^\dagger\mathbf{V}^*$ with $\Sigma^\dagger = \text{diag}(\sigma_1^{-1}, \dots, \sigma_r^{-1}, 0, \dots, 0) \in \mathbb{R}^{n \times n}$.

3 Practical Quaternion Rangefinders

Given a data matrix $A \in \mathbb{F}^{m \times n}$ ($\mathbb{F} = \mathbb{R}, \mathbb{C}, \mathbb{Q}$), a randomized rangefinder first draws a random test matrix $\Omega \in \mathbb{F}^{n \times s}$ with $s \ll n$ [41] (s is close to the target rank and often can be regarded as a constant), takes a sketch $Y = A\Omega$, and then orthogonalizes it, i.e.,

$$Y = A\Omega, \quad Q = \text{orth}(Y),$$

where Q is orthonormal and preserves the range of Y . In the real/complex case, computing Q is cheap by using QR decomposition, while things change in the quaternion setting, especially for large-scale problems, as discussed in the introduction. To better fit into the modern need, we present two practical rangefinders in this section by trading accuracy or space for time cost. To achieve this, we mostly employ mature libraries such as QR, SVD, and linear equation solvers in complex arithmetic for heavy computations. Finally, we will compare the running time of the proposed rangefinders with those in the literature.

3.1 Pseudo-QR

Given a quaternion matrix $\mathbf{X} = X_0 + X_1\mathbf{j} \in \mathbb{Q}^{m \times n}$, previous work (cite) indicates that full information has been contained in its compact representation $\mathbf{X}_c = \begin{bmatrix} X_0 \\ -\overline{X_1} \end{bmatrix} \in \mathbb{C}^{2m \times n}$, while full representation $\chi_{\mathbf{X}}$ further preserves its structure as an operator. Although X_c does not capture the entire structure in the same way as $\chi_{\mathbf{X}}$, the fact that $\chi_{\mathbf{X}}$ can be easily derived from $\mathbf{X}_c$ implies that underlying structure may still be preserved within $\mathbf{X}_c$. Starting from this observation, we prefer to operate on $\mathbf{X}_c$ to obtain our first rangefinder.

Let $\mathbf{Y} = \mathbf{A}\mathbf{\Omega} = Y_0 + Y_1\mathbf{j} \in \mathbb{Q}^{m \times s}$ with $m > s$ be the sketch of the input data matrix $\mathbf{A} \in \mathbb{Q}^{m \times n}$, with $\mathbf{\Omega} \in \mathbb{Q}^{n \times s}$ the random test matrix. Let $\mathbf{Y}_c = \begin{bmatrix} Y_0 \\ -\overline{Y_1} \end{bmatrix} \in \mathbb{C}^{2m \times s}$ be its compact complex representation. Then, a cheap thin QR in the complex arithmetic can be directly applied to $\mathbf{Y}_c$:

$$\mathbf{Y}_c = \begin{bmatrix} Y_0 \\ -\overline{Y_1} \end{bmatrix} =: QR, \quad (1)$$

where $Q \in \mathbb{C}^{2m \times s}$ is orthonormal in the complex space, and $R \in \mathbb{C}^{s \times s}$ is upper triangular. Then, we partition Q as $Q = \begin{bmatrix} Q_0 \\ Q_1 \end{bmatrix}$ with $Q_0, Q_1 \in \mathbb{C}^{m \times s}$. Furthermore, denote

$$H_0 := Q_0, \quad H_1 := -\overline{Q_1} \quad \text{and} \quad \mathbf{H} := H_0 + H_1\mathbf{j} \in \mathbb{Q}^{m \times s}. \quad (2)$$

It then follows from (2) that the compact representation of $\mathbf{H}$ is exactly Q :

$$\mathbf{H}_c = \begin{bmatrix} H_0 \\ -\overline{H_1} \end{bmatrix} = \begin{bmatrix} Q_0 \\ Q_1 \end{bmatrix} = Q.$$

This together with (1) shows that $\mathbf{H}_c$ and R gives the QR decomposition of $\mathbf{Y}_c$:

$$\mathbf{Y}_c = \mathbf{H}_c R, \quad \mathbf{H}_c \in \mathbb{C}^{2m \times s}, \quad R \in \mathbb{C}^{s \times s}. \quad (3)$$

The following result shows that $\mathbf{H}$ has the same range as $\mathbf{Y}$:

Theorem 3.1. *Assume that $\mathbf{Y} \in \mathbb{Q}^{m \times s}$ has full column rank. Then,*

$$\mathcal{R}(\mathbf{H}) = \mathcal{R}(\mathbf{Y}). \quad (4)$$

Proof. It suffices to show that there exists an invertible matrix $\mathbf{R} \in \mathbb{Q}^{s \times s}$ such that $\mathbf{Y} = \mathbf{H}\mathbf{R}$. Let $\mathbf{Y}_c = \mathbf{H}_c R$ be as in (3). As $\mathbf{Y}$ is of full column rank, Proposition 2.1 indicates that $\chi_{\mathbf{Y}}$ is also of full column rank, and so is $\mathbf{Y}_c$. Thus R is invertible. By Lemma 2.1, $\chi_{\mathbf{Y}}$ can be represented as:

$$\begin{aligned} \chi_{\mathbf{Y}} &= [\mathbf{Y}_c, \mathcal{J}\overline{\mathbf{Y}_c}] = [\mathbf{H}_c R, \mathcal{J}\overline{\mathbf{H}_c R}] \\ &= [\mathbf{H}_c, \mathcal{J}\overline{\mathbf{H}_c}] \text{diag}(R, \overline{R}) = \chi_{\mathbf{H}} \text{diag}(R, \overline{R}). \end{aligned}$$

Transforming back to quaternions, the above identity is equivalent to $\mathbf{Y} = \mathbf{H}\mathbf{R}$. Thus (4) follows from the invertibility of R . $\square$

Remark 3.1. *Even if $\mathbf{Y}$ is rank-deficient, we still have $\mathcal{R}(\mathbf{H}) \supset \mathcal{R}(\mathbf{Y})$, i.e., the range of $\mathbf{H}$ captures the range of the sketch $\mathbf{Y}$.*

The theorem belows shows that, although $\mathbf{H}$ may be non-orthonormal in the quaternion domain, its singular values are structured.

Theorem 3.2. *All the singular values of $\mathbf{H}$ takes the form of*

$$\sqrt{1 \pm \mu_1}, \sqrt{1 \pm \mu_2}, \dots,$$

with $0 \leq \mu_i \leq 1$, and the largest singular value is upper bounded by $\sqrt{2}$.

Proof. By Lemma 2.2, It suffices to consider the singular values of its complex representation

$$\chi_{\mathbf{H}} = \begin{bmatrix} H_0 & H_1 \\ -\overline{H_1} & \overline{H_0} \end{bmatrix} = [\mathbf{H}_c, \mathbf{H}_a] = [\mathbf{H}_c, \mathcal{J}\overline{\mathbf{H}_c}].$$

It follows from the (3) that $\mathbf{H}_c^* \mathbf{H}_c = I_s$; $\mathbf{H}_a^* \mathbf{H}_a = \overline{\mathbf{H}_c}^* \mathcal{J}^* \mathcal{J} \overline{\mathbf{H}_c} = \overline{\mathbf{H}_c}^* \overline{\mathbf{H}_c} = I_s$ as well. Then we have:

$$\chi_{\mathbf{H}}^* \chi_{\mathbf{H}} = \begin{bmatrix} I_s & H_0^* H_1 - \overline{H_1}^* \overline{H_0} \\ H_1^* H_0 - \overline{H_0}^* \overline{H_1} & I_s \end{bmatrix}$$

and it is positive semi-definite. Denote $T := \mathbf{H}_c^* \mathbf{H}_a = H_0^* H_1 - \overline{H_1}^* \overline{H_0}$; then

$$\chi_{\mathbf{H}}^* \chi_{\mathbf{H}} = \begin{bmatrix} I & T \\ T^* & I \end{bmatrix}. \quad (5)$$

Consider the characteristic polynomial of $\chi_{\mathbf{H}}^* \chi_{\mathbf{H}}$:

$$\begin{aligned} \det \left(\begin{bmatrix} (1-\lambda)I_s & T \\ T^* & (1-\lambda)I \end{bmatrix} \right) &= \det((1-\lambda)I_s) \det((1-\lambda)I_s - (1-\lambda)^{-1} T^* T) \\ &= \det((1-\lambda)^2 I_s - T^* T). \end{aligned}$$

Let μ^2 be an eigenvalue of $T^* T$. Then the above relation shows that $\lambda = 1 \pm \mu$ are a pair of eigenvalues of $\chi_{\mathbf{H}}^* \chi_{\mathbf{H}}$, i.e., $\sqrt{1 \pm \mu}$ are a pair of singular values of $\chi_{\mathbf{H}}$, which together with Lemma 2.2 shows that they are also singular values of $\mathbf{H}$. Finally, it is easily seen from (5) that $I_s \succeq T^* T$, namely, $\mu \in [0, 1]$, which implies that the largest singular value of $\mathbf{H}$ is smaller than $\sqrt{2}$. □

Remark 3.2. *The two properties above are not enough to control the condition number $\kappa(\mathbf{H})$. In fact, from the analysis above, we see that the smallest singular value of $\mathbf{H}$ depends on $T = \mathbf{H}_c^* \mathbf{H}_a$. If the angle between $\text{span}(\mathbf{H}_c)$ and $\text{span}(\mathbf{H}_a)$ is very small, then T tends to be an identity matrix and so the smallest eigenvalue of $\chi_{\mathbf{H}}^* \chi_{\mathbf{H}}$ tends to zero; on the contrary, if $\text{span}(\mathbf{H}_c)$ and $\text{span}(\mathbf{H}_a)$ are perpendicular to each other, then T is exactly 0 and the smallest eigenvalue of $\chi_{\mathbf{H}}^* \chi_{\mathbf{H}}$ is 1. However, by the construction (3), $\text{span}(\mathbf{H}_c) = \text{span}(\mathbf{Y}_c)$ and $\text{span}(\mathbf{H}_a) = \text{span}(\mathbf{Y}_a)$. As $[\mathbf{Y}_c, \mathbf{Y}_a] = \chi_{\mathbf{Y}}$ and $\mathbf{Y}$ is data-dependent, we cannot make any assumption on the angle between $\text{span}(\mathbf{Y}_c)$ and $\text{span}(\mathbf{Y}_a)$. Thus the smallest singular value of $\mathbf{H}$ cannot be estimated, nor $\kappa(\mathbf{H})$.*

Nevertheless, empirically we usually observe that $\kappa(\mathbf{H})$ is at least two times smaller than $\kappa(\mathbf{Y})$. To further reduce $\kappa(\mathbf{H})$, we can perform a range-preserving correction step:

$$\mathbf{H}_{new} \leftarrow (1 - \epsilon)\mathbf{H} + \epsilon(\mathbf{H}^\dagger)^* \quad (6)$$

The next theorem shows that, if ϵ is chosen close to the smallest singular value of $\mathbf{H}$, and if $\kappa(\mathbf{H})$ is not small, then it will be reduced rapidly.

Theorem 3.3. *Suppose in (6), one chooses ϵ such that $\epsilon \in [\sigma_s(\mathbf{H}), \delta\sigma_s(\mathbf{H})]$, where $\delta \in [1, \sqrt{7}/2]$. If $\kappa(\mathbf{H}) > \max\{2\sqrt{2}\delta, 2\delta^2 + 1/2\}$, then $\kappa(\mathbf{H}_{new}) < \sqrt{\kappa(\mathbf{H})}$.*

The role of δ above means that one can compute $\sigma_s(\mathbf{H})$ inexactly; in practice, one usually performs two or three power iterates of approximating $\sigma_1(\mathbf{H}^\dagger)$ (namely, $\sigma_s(\mathbf{H})^{-1}$) to obtain ϵ . The upper bound of $\kappa(\mathbf{H}) > \max\{2\sqrt{2}\delta, 2\delta^2 + 1/2\}$ is 4, i.e, when $\kappa(\mathbf{H}) > 4$, it will be reduced to its square root. Empirically, an $\mathbf{H}$ with $\kappa(\mathbf{H}) < 10$ is enough for obtaining a desirable low-rank approximation.

In practice, one can execute the correction step (6) at most two or three times to obtain a desirable $\mathbf{H}$. The following corollary shows that, if $\mathbf{H}$ is generated by (1) and (2) with $\kappa(\mathbf{H}) < 10^8$, then $\kappa(\mathbf{H})$ will not exceed 10 after at most three correction steps.

Corollary 3.1. *Let $\mathbf{H}_{k+1} \leftarrow (1 - \epsilon_k)\mathbf{H}_k + \epsilon_k(\mathbf{H}_k^\dagger)^*$ with $\mathbf{H}_0$ be given by (1) and (2). If $\epsilon_k \in [\sigma_s(\mathbf{H}_k), \delta\sigma_s(\mathbf{H}_k)]$, where $\delta \in [1, \sqrt{7}/2]$, and if $\kappa(\mathbf{H}_k) > \max\{2\sqrt{2}\delta, 2\delta^2 + 1/2\}$, then $\kappa(\mathbf{H}_{k+1}) < \sqrt{\kappa(\mathbf{H}_k)}$.*

The proofs of Theorem 3.3 and Corollary 3.1 are given in the appendix. The iterative scheme above is essentially the quaternion version of the quadratically convergent Newton method for computing polar decomposition [15, Section 3.3] but with different parameters.

The remaining question is how to compute $(\mathbf{H}^\dagger)^*$. We convert it to solving linear equations in the complex arithmetic such that highly efficient solvers can be used. To this end, assume that $\mathbf{Y}$ has full column rank (and by Theorem 3.1, so does $\mathbf{H}$); further assume that $\mathbf{H}^*\mathbf{H}$ is not too ill-conditioned (say, $\kappa(\mathbf{H}^*\mathbf{H}) < 10^{16}$). Since $\mathbf{H}^\dagger = (\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*$, we can solve the equation $\mathbf{H}^*\mathbf{H}\mathbf{X} = \mathbf{H}^*$ to obtain $(\mathbf{H}^\dagger)^*$. The following idea comes from [47]. For a general quaternion linear equation $\mathbf{A}\mathbf{X} = \mathbf{B}$ with $\mathbf{A} \in \mathbb{Q}^{m \times n}$, $\mathbf{X} \in \mathbb{Q}^{n \times s}$, $\mathbf{B} \in \mathbb{Q}^{m \times s}$,

$$\mathbf{A}\mathbf{X} = \mathbf{B} \Leftrightarrow \chi_{\mathbf{A}}\chi_{\mathbf{X}} = \chi_{\mathbf{B}} \Leftrightarrow \chi_{\mathbf{A}}[\mathbf{X}_c, \mathbf{X}_a] = [\mathbf{B}_c, \mathbf{B}_a].$$

In fact, using the relation $\mathbf{X}_a = \mathcal{J}\overline{\mathbf{X}_c}$, solving the complex equation $\chi_{\mathbf{A}}Z = \mathbf{B}_c$ is enough to give a solution to $\mathbf{A}\mathbf{X} = \mathbf{B}$. To see this, let $\hat{Z} \in \mathbb{C}^{2n \times s}$ be a solution to $\chi_{\mathbf{A}}Z = \mathbf{B}_c$ and partition it as $\hat{Z} = [\hat{Z}_0^*, \hat{Z}_1^*]^*$ with $\hat{Z}_0, \hat{Z}_1 \in \mathbb{C}^{n \times s}$. Similar to (2), let $\mathbf{X} := \hat{Z}_0 - \hat{Z}_1\mathbf{j}$; then $\mathbf{X}_c = \hat{Z}$, namely, $\chi_{\mathbf{A}}\mathbf{X}_c = \mathbf{B}_c$, and it follows from Lemma 2.1 that

$$\chi_{\mathbf{A}}\mathbf{X}_a = \chi_{\mathbf{A}}\mathcal{J}\overline{\mathbf{X}_c} = \mathcal{J}\overline{\chi_{\mathbf{A}}\mathcal{J}^*\mathcal{J}\overline{\mathbf{X}_c}} = \mathcal{J}\overline{\chi_{\mathbf{A}}\mathbf{X}_c} = \mathcal{J}\overline{\mathbf{B}_c} = \mathbf{B}_a,$$

which together with $\chi_{\mathbf{A}}\mathbf{X}_c = \mathbf{B}_c$ means that $\mathbf{A}\mathbf{X} = \mathbf{B}$.

Therefore, it suffices to solve $\chi_{\mathbf{H}^*\mathbf{H}}Z = (\mathbf{H}^*)_c$ to obtain $\mathbf{H}^\dagger$. Note that $\chi_{\mathbf{H}^*\mathbf{H}} \in \mathbb{C}^{2s \times 2s}$ and $(\mathbf{H}^*)_c \in \mathbb{C}^{2s \times m}$. As the sampling size s is usually small, solving this equation in the complex arithmetic can be efficient by using mature solvers.

Algorithm 1 Quaternion structure-preserving equations solver

Require: $\mathbf{A} \in \mathbb{Q}^{n_1 \times n_2}$ and $\mathbf{B} \in \mathbb{Q}^{n_1 \times l}$

Ensure: Quaternion matrix $\mathbf{X} \in \mathbb{Q}^{n_2 \times l}$ satisfy $\mathbf{A}\mathbf{X} = \mathbf{B}$

- 1: Construct $\chi_{\mathbf{A}} \in \mathbb{C}^{2n_1 \times 2n_2}$ and $\mathbf{B}_c \in \mathbb{C}^{2n_1 \times l}$.
 - 2: Compute $\mathbf{X}_c = \chi_{\mathbf{A}} \backslash \mathbf{B}_c$.
 - 3: Construct $\mathbf{X} = \mathbf{X}_c(1 : n_2, :) - \overline{\mathbf{X}_c(n_2 + 1 : 2n_2, :)}\mathbf{j}$
-

The whole algorithm in this subsection is summarized in Algorithm 2.

3.2 Pseudo-SVD

In the complex QR step of pseudo-QR, we only enforce orthonormality on $\mathbf{H}_c$. To obtain a better well-conditioned $\mathbf{H}$, the relationship between $\mathbf{H}_c$ and $\mathbf{H}_a = \mathcal{J}\overline{\mathbf{H}_c}$ should be taken into account. A motivation is from the following lemma.

Algorithm 2 (pseudo-QR) quaternion pseudo-QR implementation

Require: Sketch matrix $\mathbf{Y} \in \mathbb{Q}^{m \times n}$.

Ensure: A quaternion rangefinder $\mathbf{H} \in \mathbb{Q}^{m \times s}$

- 1: Construct the $2m \times s$ complex matrix $\mathbf{Y}_c$ from $\mathbf{Y}$.
 - 2: Compute complex QR $[U, \sim] = \text{qr}(\mathbf{Y}_c, 0)$
 - 3: Construct $\mathbf{H} = H_0 + H_1\mathbf{j}$ from U with $H_0 = U(1:m, :)$ and $H_1 = -\overline{U(m+1:2m, :)}\mathbf{j}$.
 - 4: Execute the correction step (6) a few times (often 2 to 3 times). One can use Algorithm 1 to solve $\chi_{\mathbf{H}^*\mathbf{H}}\mathbf{Z} = (\mathbf{H}^*)_c$ to obtain $\mathbf{H}^\dagger$.
-

Lemma 3.1 (c.f. [47]). *Let $\mathbf{A} \in \mathbb{Q}^{m \times m}$ be Hermitian. If v is an eigenvector of $\chi_{\mathbf{A}}$ corresponding to the eigenvalue λ , then $\lambda = \overline{\lambda}$, and $\mathcal{J}\overline{v}$ is also an eigenvector associated to λ . Moreover, v is perpendicular to $\mathcal{J}\overline{v}$.*

The above lemma implies that in the ideal situation, if every columns of $\mathbf{H}_c$ are from different eigenspaces, then $\mathbf{H}_c^*\mathbf{H}_c = I$, $\mathbf{H}_c^*\mathbf{H}_a = \mathbf{H}_c^*\mathcal{J}\overline{\mathbf{H}_c} = 0$, which means that $\chi_{\mathbf{H}^*}\chi_{\mathbf{H}} = I$, and $\mathbf{H}$ is orthonormal. To this end, we resort to (and modify) the method introduced in [22] to find a suitable pair $(\mathbf{H}_c, \mathbf{H}_a)$. The idea is to find an SVD of $\mathbf{Y}$ via computing the complex SVD of $\chi_{\mathbf{Y}}$. Let

$$\chi_{\mathbf{Y}} := USV^*, \quad U \in \mathbb{C}^{2m \times 2s}, \quad V \in \mathbb{C}^{2s \times 2s}, \quad S = \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma \end{bmatrix} \in \mathbb{R}^{2s \times 2s} \quad (7)$$

be a compact SVD of $\chi_{\mathbf{Y}}$, where Σ is diagonal. Partition

$$U = \begin{bmatrix} U_{ul} & U_{ur} \\ U_{dl} & U_{dr} \end{bmatrix}, \quad V = \begin{bmatrix} V_{ul} & V_{ur} \\ V_{dl} & V_{dr} \end{bmatrix}, \quad \text{with } U_{ul} \in \mathbb{C}^{m \times s}, \quad V_{ul} \in \mathbb{C}^{s \times s}. \quad (8)$$

Denote

$$\mathbf{H} := U_{ul} - \overline{U_{dl}}\mathbf{j} \in \mathbb{Q}^{m \times s} \quad \text{and} \quad \mathbf{V} := V_{ul} - \overline{V_{dl}}\mathbf{j} \in \mathbb{Q}^{s \times s}. \quad (9)$$

Theorem 3.4 ensures that conditionally, $\mathbf{H}\Sigma\mathbf{V}^*$ is a compact QSVD of $\mathbf{Y}$.

Theorem 3.4. *Let $\mathbf{H}, \Sigma, \mathbf{V}$ be given by (7) and (9). If all the singular values of $\mathbf{Y}$ are distinct, then $\mathbf{H}\Sigma\mathbf{V}^*$ is a compact QSVD of $\mathbf{Y}$.*

We first present the following lemma.

Lemma 3.2. *Let $\sigma_1 > \dots > \sigma_r$ be r singular values of $\chi_{\mathbf{Y}}$, each of which has multiplicity two, and $U_\sigma = [u_1, \dots, u_r]$ be r left singular vectors corresponding to $\sigma_1, \dots, \sigma_r$. Then $[U_\sigma, \mathcal{J}\overline{U_\sigma}]$ is orthonormal and spans the left invariant subspace of $\sigma_1, \dots, \sigma_r$.*

Proof. By the multiplicity assumption on $\sigma_1, \dots, \sigma_r$ and noting $u_j \perp \mathcal{J}\overline{u_j}$, Lemma 3.1 shows that $\text{span}(u_j, \mathcal{J}\overline{u_j})$ is the invariant subspace of σ_j . On the other hand, each u_j belongs to distinct σ_j , and so $u_i \perp u_j$, $\mathcal{J}\overline{u_i} \perp \mathcal{J}\overline{u_j}$, $\mathcal{J}\overline{u_i} \perp u_j$, $i \neq j$. Thus the results follow. $\square$

Proof of Theorem 3.4. The assumption shows that Σ consists of distinct singular values. By Lemma 2.2, it suffices to prove that

$$\chi_{\mathbf{Y}} = \chi_{\mathbf{H}}S\chi_{\mathbf{V}}^* = [\mathbf{H}_c, \mathcal{J}\overline{\mathbf{H}_c}] \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} \mathbf{V}_c^* \\ (\mathcal{J}\overline{\mathbf{V}_c})^* \end{bmatrix} \quad (10)$$

with $[\mathbf{H}_c, \mathcal{J}\overline{\mathbf{H}_c}]$ and $[\mathbf{V}_c, \mathcal{J}\overline{\mathbf{V}_c}]$ orthonormal. It follows from the construction of $\mathbf{H}$ that $\mathbf{H}_c = \begin{bmatrix} U_{ul} \\ U_{dl} \end{bmatrix}$, which, by Lemma 3.2, demonstrates the orthonormality of $[\mathbf{H}_c, \mathcal{J}\overline{\mathbf{H}_c}]$. Similarly, $[\mathbf{V}_c, \mathcal{J}\overline{\mathbf{V}_c}]$ is orthonormal. It then follows from $\chi_{\mathbf{Y}}\mathbf{V}_c = \mathbf{H}_c\Sigma$ that

$$\chi_{\mathbf{Y}}\mathcal{J}\overline{\mathbf{V}_c} = \mathcal{J}\mathcal{J}^*\chi_{\mathbf{Y}}\mathcal{J}\overline{\mathbf{V}_c} = \mathcal{J}\overline{\chi_{\mathbf{Y}}\mathbf{V}_c} = \mathcal{J}\overline{\mathbf{H}_c\Sigma} = \mathcal{J}\overline{\mathbf{H}_c}\Sigma,$$

and so $\chi_{\mathbf{Y}}[\mathbf{V}_c, \mathcal{J}\overline{\mathbf{V}_c}] = [\mathbf{H}_c, \mathcal{J}\overline{\mathbf{H}_c}]\text{diag}(\Sigma, \Sigma)$, which together with the orthonormality of $[\mathbf{V}_c, \mathcal{J}\overline{\mathbf{V}_c}] \in \mathbb{C}^{2s \times 2s}$ yields (10). $\square$

Remark 3.3. The above proof implies why the assumption in Theorem 3.4 is necessary. Consider the counterexample $\mathbf{Y} = I_2 \in \mathbb{Q}^{2 \times 2}$, and so $\chi_{\mathbf{Y}} = I_4 \in \mathbb{C}^{4 \times 4}$. Any normalized $u \in \mathbb{C}^4$ is a singular vector. If $\mathbf{H}_c = [u, \mathcal{J}\overline{u}]$, it is clear that $\mathcal{J}\overline{\mathbf{H}_c} = \mathbf{H}_c$ and so $[\mathbf{H}_c, \mathcal{J}\overline{\mathbf{H}_c}]$ is rank-deficient.

However, some issues should be addressed. Firstly, as $\chi_{\mathbf{Y}}$ is two times larger than $\mathbf{Y}$ (in terms of the real elements), directly computing the SVD of $\chi_{\mathbf{Y}}$ seems to be redundant. Nevertheless, recall that $\mathbf{Y} \in \mathbb{Q}^{m \times s}$ is a sketch, whose column size s is usually much smaller than m [41]; thus the SVD of $\chi_{\mathbf{Y}}$ scales well. As shown in subsection 3.3, rangefinder based on SVD of $\chi_{\mathbf{Y}}$ is still much faster than the competitors.

A much criticized flaw is that when $\mathbf{Y}$ is too ill-conditioned (say, $\kappa(\mathbf{Y}) > 10^{13}$), due to rounding errors, numerically, the small singular values of $\chi_{\mathbf{Y}}$ may not appear twice, and so doing (9) may not generate the correct singular vectors for $\mathbf{H}$ corresponding to the small singular values [5, p. 84]. The same situation also occurs when $\mathbf{Y}$ has duplicated singular values. In this two cases, $\mathbf{H}$ given by (9) is no longer orthonormal and may not even span the correct range.

Fortunately, the above issue indeed can be tackled because the task is to find a well-conditioned range of $\mathbf{Y}$ instead of a QSVD. When this issue occurs, numerically computing an SVD of $\chi_{\mathbf{Y}}$ may exhibit the following form

$$\begin{aligned} \chi_{\mathbf{Y}} &= U\tilde{S}V^*, \quad U \in \mathbb{C}^{2m \times 2s}, \quad V \in \mathbb{C}^{2s \times 2s}, \quad \tilde{S} = \begin{bmatrix} S_g & 0 \\ 0 & \Sigma_b \end{bmatrix} \in \mathbb{R}^{2s \times 2s}, \\ U^*U &= I_{2s}, \quad V^*V = I_{2s}, \quad \Sigma_b \in \mathbb{R}^{2t \times 2t} (t < s), \quad S_g = \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma \end{bmatrix} \in \mathbb{R}^{2(s-t) \times 2(s-t)}; \end{aligned} \quad (11)$$

i.e., now the singular values $\tilde{S}$ can be partitioned as the “good” part S_g and the “bad” part Σ_b . S_g consists of singular values of $\chi_{\mathbf{Y}}$ still appearing exactly twice, i.e., Σ consists of distinct singular values. Σ_b represents those small singular values, which, due to rounding errors, are distinct and the order is disturbed (the latter is more severe), as well as the singular values with multiplicity larger than 2.

Given $\chi_{\mathbf{Y}} = U\tilde{S}V^*$ as in (11), if still generating $\mathbf{H}$ by (9), then $\mathbf{H}^*\mathbf{H}$ will exhibit the form of

$$\mathbf{H}^*\mathbf{H} = \begin{bmatrix} I_{2(s-t)} & 0 \\ 0 & \times \end{bmatrix} \in \mathbb{Q}^{2s \times 2s},$$

where “ $\times$ ” is a matrix not equal to identity. Based on this observation, correcting $\mathbf{H}$ can be executed as follows: first write $U = [U_g, U_b]$, with $U_g \in \mathbb{C}^{2m \times 2(s-t)}$ and $U_b \in \mathbb{C}^{2m \times 2t}$ corresponding to S_g and Σ_b respectively; $V = [V_g, V_b]$ is partitioned accordingly. From (11) denote

$$\chi_{\mathbf{Y}} := Y_g + Y_b, \quad Y_g = U_g S_g V_g^*, \quad Y_b = U_b \Sigma_b V_b^*. \quad (12)$$

Further partition U_g as in (8) and generate $\mathbf{H}_g$ from U_g as in (9). According to the definition of S_g , Lemma 3.2 ensures that

$$\mathbf{H}_g \text{ is orthonormal and } \mathcal{R}(\chi_{\mathbf{H}_g}) = \mathcal{R}(U_g). \quad (13)$$

For Σ_b , although it is not structured, we can still select a representation basis from U_b . Denote $U_b := [u_1, \dots, u_{2t}] \in \mathbb{C}^{2m \times 2t}$. Then:

Proposition 3.1. *One can find $U_{\tilde{b}} := [u_{i_1}, \dots, u_{i_t}] \subset U_b = [u_1, \dots, u_{2t}]$, such that*

$$\text{span}(U_{\tilde{b}}, \mathcal{J}\overline{U_{\tilde{b}}}) = \text{span}(U_b) = 2t. \quad (14)$$

The proof is given in appendix. Given this and let $\mathbf{H}_{\tilde{b}} \in \mathbb{Q}^{m \times t}$ be constructed such that $\chi_{\mathbf{H}_{\tilde{b}}} = [U_{\tilde{b}}, \mathcal{J}\overline{U_{\tilde{b}}}]$. Then

Proposition 3.2. *If $\mathbf{Y}$ has full column rank, then $\mathcal{R}([\mathbf{H}_g, \mathbf{H}_{\tilde{b}}]) = \mathcal{R}(\mathbf{Y})$.*

Proof. This can be achieved by verifying that the direct sum $\mathcal{R}(\mathbf{H}_g) \oplus \mathcal{R}(\mathbf{H}_{\tilde{b}}) = \mathcal{R}(\mathbf{Y})$, as $\mathbf{H}_g \perp \mathbf{H}_{\tilde{b}}$. As (12) holds, we only need to verify $\mathcal{R}(\chi_{\mathbf{H}_{\tilde{b}}}) = \mathcal{R}(U_b)$, which by the construction of $\mathbf{H}_{\tilde{b}}$ is true because $\mathcal{R}(\chi_{\mathbf{H}_{\tilde{b}}}) = \text{span}((\mathbf{H}_{\tilde{b}})_c, \mathcal{J}(\overline{(\mathbf{H}_{\tilde{b}})_c})) = \text{span}(U_{\tilde{b}}, \mathcal{J}\overline{U_{\tilde{b}}}) = \text{span}(U_b)$ using (14). $\square$

Denote $\mathbf{H}_{\text{new}} = [\mathbf{H}_g, \mathbf{H}_{\tilde{b}}]$, it remains to adjust $\mathbf{H}_{\text{new}}$ such that it is orthonormal. Owing to (13) and that $\mathbf{H}_g \perp \mathbf{H}_{\tilde{b}}$, we only need to adjust $\mathbf{H}_{\tilde{b}}$, which in fact can be simultaneously done during the selection of $U_{\tilde{b}}$ using modified Gram-Schmit orthogonalization [12].

However, in case that $t \ll s$ and $\mathbf{Y}$ is too ill-conditioning, we find that the following process is more accurate and efficient. The idea still resorts to complex SVD. Denote $\mathbf{H}_b \in \mathbb{Q}^{m \times 2t}$ such that $\chi_{\mathbf{H}_b} = [U_b, \mathcal{J}\overline{U_b}]$ (this $\mathbf{H}_b$ need not be explicitly constructed). Then Lemma 3.2 shows that $\mathcal{R}(\chi_{\mathbf{H}_b}) = \text{span}(U_b)$. In addition, let

$$\mathbf{H}_b \leftarrow \mathbf{H}_b \text{diag}(f), \text{ with } f_j \sim \text{Uniform}(0, 1) + 1, \quad j = 1, \dots, 2t. \quad (15)$$

Empirically, multiplying $\text{diag}(f)$ avoids $\mathbf{H}_b$ to have duplicated singular values. In this case, applying the complex SVD to $\chi_{\mathbf{H}_b} \in \mathbb{C}^{2m \times 4t}$ yields

$$\chi_{\mathbf{H}_b} = [U_{h1}, U_{h2}, U_{hb}] \text{diag}(\Sigma_h, \Sigma_h, 0) V_h^*, \quad \Sigma_h \in \mathbb{R}^{t \times t}, \quad U_{h1} \in \mathbb{C}^{2m \times t}, \quad (16)$$

which reduces to the case of (11). We can construct the new $\mathbf{H}_{\tilde{b}} \in \mathbb{Q}^{m \times t}$ from U_{h1} such that $\chi_{\mathbf{H}_{\tilde{b}}} = [U_{h1}, \mathcal{J}\overline{U_{h1}}]$. By Lemma 3.2, $\mathbf{H}_{\tilde{b}}$ is orthonormal and $\mathcal{R}(\chi_{\mathbf{H}_{\tilde{b}}}) = \text{span}(U_b)$. Denote $\mathbf{H}_{\text{new}} = [\mathbf{H}_g, \mathbf{H}_{\tilde{b}}]$. We still have $\mathcal{R}(\mathbf{H}_{\text{new}}) = \mathcal{R}(\mathbf{Y})$ and $\mathbf{H}_{\text{new}}$ is orthonormal.

The whole computation is summarized in Algorithm 3.

Algorithm 3 (pseudo-SVD) quaternion pseudo-SVD implementation

Require: Sketch matrix $\mathbf{Y} \in \mathbb{Q}^{m \times s}$.

Ensure: Quaternion rangefinder $\mathbf{H} \in \mathbb{Q}^{m \times s}$.

- 1: Compute complex SVD $[U, \sim, \sim] = \text{svd}(\chi_{\mathbf{Y}}, 0)$ and partition $U = [U_g, U_b]$ with $U_b \in \mathbb{C}^{2m \times 2t}$ corresponding to the “bad” part.
 - 2: Construct $\mathbf{H}_g = H_0 + H_1 \mathbf{j}$ from U_g such that $\chi_{\mathbf{H}_g} = [U_g, \mathcal{J}\overline{U_g}]$.
 - 3: **if** $t \ll s$ **then**
 - 4: Construct $\mathbf{H}_b$ from (15). Compute $[U_h, \sim, \sim] = \text{svd}(\chi_{\mathbf{H}_b}, 0)$ with $U_h = [U_{h1}, U_{h2}, U_{hb}]$ as (16), and construct $\mathbf{H}_{\tilde{b}}$ such that $\chi_{\mathbf{H}_{\tilde{b}}} = [U_{h1}, \mathcal{J}\overline{U_{h1}}]$.
 - 5: **else**
 - 6: Sequentially select linearly independent $u_{i_1}, \dots, u_{i_t}$ from U_b and simultaneously do orthogonalization such that $u_{i_j} \perp [u_{i_1}, \mathcal{J}\overline{u_{i_1}}, \dots, u_{i_{j-1}}, \mathcal{J}\overline{u_{i_{j-1}}}]$, $j = 2, \dots, t$. Let $U_{\tilde{b}} = [u_{i_1}, \dots, u_{i_t}]$ and construct $\mathbf{H}_{\tilde{b}}$ such that $\chi_{\mathbf{H}_{\tilde{b}}} = [U_{\tilde{b}}, \mathcal{J}\overline{U_{\tilde{b}}}]$.
 - 7: **end if**
 - 8: $\mathbf{H} = [\mathbf{H}_g, \mathbf{H}_{\tilde{b}}]$.
-

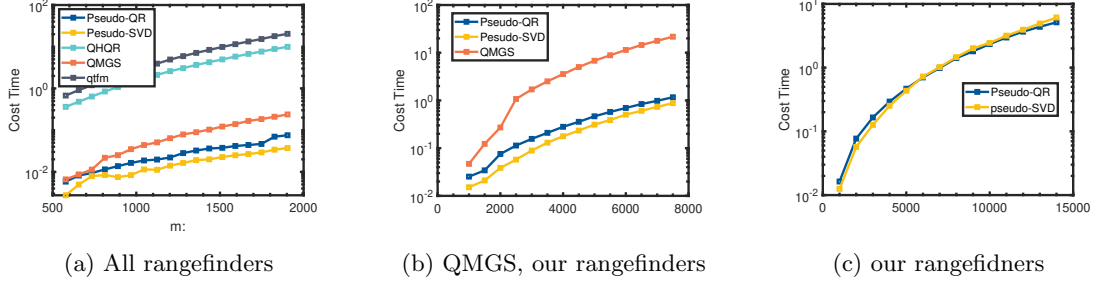


Figure 2: Running Time $m = 25n$, x-axis is m and y-axis is running time with Logarithmic scale.

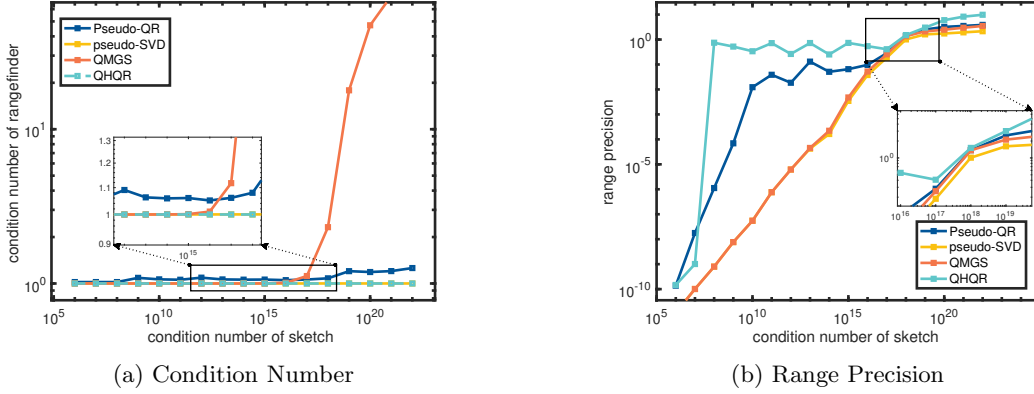


Figure 3: $m = 1000$, $n = 200$, condition number of sketch from $1e6$ to $1e22$.

3.3 Comparisons

For a sketch $\mathbf{Y} \in \mathbb{Q}^{m \times s}$, the computational complexity of Pseudo-QR and Pseudo-SVD is $O(ms^2 + s^3)$ and $O(ms^2)$, respectively. In practice, these algorithms are suited to different criteria. Pseudo-QR performs well when the condition number $\kappa(\mathbf{Y}) < 10^8$. Conversely, Pseudo-SVD is more accurate for highly ill-conditioned sketch; however, it necessitates to compute the SVD of $\chi(\mathbf{Y})$, which demands twice the memory compared to Pseudo-QR.

Subsequently, we will juxtapose these algorithms with other rangefinders, including the $\mathbf{qr}$ function in QTFM, the structure-preserving Quaternion Householder QR (QHQR) [19], and the structure-preserving modified Gram-Schmidt QR (QMGS) [2]. Figure 2a illustrates that our rangefinders exhibit significantly lower computational costs compared to $\mathbf{qr}$ in QTFM and the QHQR. The disparity in their time complexity can span two to three orders of magnitude, escalating with an increase in the size of the sketches from 600 to 2000. As the size continues to grow, Figure 2b demonstrates that our rangefinders outperform the QMGS algorithm markedly, while the $\mathbf{qr}$ function in QTFM and QHQR become prohibitively time-consuming. Finally, Figure 2c reveals that Pseudo-QR operates marginally faster than Pseudo-SVD when the data size exceeds 10^4 .

We then fix the size of $\mathbf{Y}$ to be 1000×200 while vary $\kappa(\mathbf{Y})$ and compare their precision as illustrated in Fig. 3. In terms of the condition number of the rangefinder, all the rangefinders perform well when $\kappa(\mathbf{Y}) < 10^{16}$, and the orthonormal ones (Pseudo-SVD, QMGS, QHQR) keep their orthogonality. In terms of the range precision $\|\mathbf{H}\mathbf{H}^\dagger - \mathbf{Y}\mathbf{Y}^\dagger\|_F$, Pseudo-SVD and MGSQR outperform the competitors, while Pseudo-QR is still valuable when $\kappa(\mathbf{Y}) < 10^8$.

4 One-Pass Algorithm

In this section, we consider the one-pass randomized algorithm proposed by Tropp et al. [41] with a range-preserving while possibly non-orthonormal rangefinder. The one-pass algorithm can reduce storage cost and ensure linear update of streaming data, where the latter can save multiplication flops during sketching [41, 42].

Our theoretical result shows that the accuracy loss of the truncation approximation is proportional to the condition number $\kappa(\mathbf{H})$. More detailed estimation with Gaussian and sub-Gaussian embeddings will be left to Sections 5 and 6.

4.1 The algorithm

For a given data matrix $\mathbf{A} \in \mathbb{Q}^{m \times n}$ and the target rank r , the purpose is to find a rank- r approximation. First draw two random quaternion $\Psi \in \mathbb{Q}^{l \times m}$ and $\Omega \in \mathbb{Q}^{m \times s}$ independently where the sketch size satisfy: $r \leq s \leq l \leq \min\{m, n\}$. Then main information can be captured by two sketches:

$$\mathbf{Y} = \mathbf{A}\Omega \in \mathbb{Q}^{s \times m} \quad \text{and} \quad \mathbf{W} = \Psi\mathbf{A} \in \mathbb{Q}^{l \times n}, \quad (17)$$

where $\mathbf{Y}$ is used to generate the range $\mathbf{H}$ by using any range-preserving rangefinders such as pseudo-QR or pseudo-SVD. Then, a rank- s QB approximation can be obtained by solving

$$\mathbf{H}\mathbf{X} = \mathbf{H}(\Psi\mathbf{H})^\dagger \mathbf{W}. \quad (18)$$

Finally, a truncated QSVD is applied to $\mathbf{X}$ to further obtain the final rank- r approximation. In the recovery algorithm, only $\mathbf{Y}$, $\mathbf{W}$, and Ψ are required, which means that $\mathbf{A}$ is not exposed to the recovery process. The pseudocode is given in Algorithms 4 and 5.

Algorithm 4 QB Stage

Require: Quaternion sketch $\mathbf{Y} \in \mathbb{Q}^{m \times s}$, $\mathbf{W} \in \mathbb{Q}^{l \times n}$, test matrices $\Psi \in \mathbb{Q}^{l \times m}$,
Ensure: Rank- s approximation of the form $\hat{\mathbf{A}} = \mathbf{H}\mathbf{X} \in \mathbb{Q}^{m \times n}$ with $\mathbf{H} \in \mathbb{Q}^{m \times s}$ and $\mathbf{X} \in \mathbb{Q}^{s \times n}$
1: $\mathbf{H} \leftarrow \mathcal{F}(\mathbf{Y})$ $\triangleright \mathcal{F}$ is a rangefinder map.
2: $\mathbf{X} \leftarrow (\Psi\mathbf{H}) \setminus \mathbf{W}$ $\triangleright$ Solving overdetermined quaternion linear equations
3: return $(\mathbf{H}, \mathbf{X})$

Algorithm 5 Truncation Stage

Require: $(\mathbf{H}, \mathbf{X})$ from Algorithm 4
Ensure: Rank- r approximation $\hat{\mathbf{A}} = \mathbf{U}\Sigma\mathbf{V}^*$ with $\mathbf{U} \in \mathbb{Q}^{m \times r}$, orthonormal $\mathbf{V} \in \mathbb{Q}^{n \times r}$, and real diagonal matrix $\Sigma \in \mathbb{R}^{r \times r}$.
1: $(\mathbf{U}, \Sigma, \mathbf{V}) = \text{QSVD}(\mathbf{X})$
2: $\Sigma = \Sigma(1:r, 1:r)$, $\mathbf{U} = \mathbf{U}(:, 1:r)$, $\mathbf{V} = \mathbf{V}(:, 1:r)$
3: $\mathbf{U} = \mathbf{H}\mathbf{U}$
4: return $(\mathbf{U}, \Sigma, \mathbf{V})$

Remark 4.1. To solve $(\Psi\mathbf{H}) \setminus \mathbf{W}$, one can use Algorithm 1 in the complex arithmetic. QSVD in the truncation stage can be any quaternion SVD algorithms [26, 37, 16, 25] or directly computing the SVD of $\chi_{\mathbf{X}}$.

Remark 4.2. *Even though $\mathbf{H}$ may be non-orthonormal, empirically, we find that a truncation stage is still helpful in improving the recovery accuracy than only doing a QB approximation. This is the same as the orthonormal case [41, 14, 27].*

4.2 Deterministic error

This subsection is only concerned with the deterministic error without assuming the distribution of the test matrices at first. More detailed probabilistic bounds will be given in Sections 5 and 6 when a randomized embedding is selected. The following basic requirements on the rangefinder $\mathbf{H}$ and the sketch size are made throughout this subsection:

$$\mathcal{R}(\mathbf{H}) = \mathcal{R}(\mathbf{Y}), \text{ and } \mathbf{H} \text{ has full column rank;} \quad (19)$$

$$r \leq s \leq l \leq \min\{m, n\}. \quad (20)$$

To analyze the approximation error with a non-orthonormal $\mathbf{H}$, the idea is to use QB decomposition as a bridge such that existing error analysis can be applied. One can represent $\mathbf{H}$ as:

$$\mathbf{H} = \mathbf{Q}\mathbf{B}, \quad \mathbf{Q} \in \mathbb{Q}^{m \times s}, \quad \mathbf{Q}^* \mathbf{Q} = I_s. \quad (21)$$

Under (19), $\mathcal{R}(\mathbf{Q}) = \mathcal{R}(\mathbf{H}) = \mathcal{R}(\mathbf{Y})$ and $\mathbf{B} \in \mathbb{Q}^{s \times s}$ is invertible. Here (21) can be thin QR, compact QSVD, or polar decomposition.

We introduce notations used in this subsection. Define the partially orthonormal matrix $\mathbf{Q}_\perp \in \mathbb{Q}^{m \times (m-s)}$ with $\mathbf{Q}_\perp \mathbf{Q}_\perp^* := I_m - \mathbf{Q}\mathbf{Q}^*$ such that $\mathcal{R}(\mathbf{Q}) \perp \mathcal{R}(\mathbf{Q}_\perp)$. Then we define submatrices:

$$\Psi_1 = \Psi \mathbf{Q}_\perp \in \mathbb{Q}^{l \times (m-s)}, \quad \Psi_2 = \Psi \mathbf{Q} \in \mathbb{Q}^{l \times s}. \quad (22)$$

Let the QSVD of $\mathbf{A} \in \mathbb{Q}^{m \times n}$ be

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^* = [\mathbf{U}_1 \quad \mathbf{U}_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^* \\ \mathbf{V}_2^* \end{bmatrix}, \quad \Sigma_1 \in \mathbb{R}^{r \times r}, \quad \mathbf{U}_1 \in \mathbb{Q}^{m \times r}, \quad \mathbf{V}_1 \in \mathbb{Q}^{n \times r}; \quad (23)$$

and define:

$$\Omega_1 = \mathbf{V}_1^* \Omega \in \mathbb{Q}^{r \times s}, \quad \Omega_2 = \mathbf{V}_2^* \Omega \in \mathbb{Q}^{(n-r) \times s}. \quad (24)$$

The error of Algorithm 4, which is independent of $\kappa(\mathbf{H})$, is provided.

Theorem 4.1 (QB error). *Let $(\mathbf{H}, \mathbf{X})$ be generated by Algorithm 4 and (19) holds. With notations in (22) and (24), if Ω_1^* and Ψ_2 have full column rank, we have the following error estimation, where “a” $\in \{F, 2\}$:*

$$\|\mathbf{H}\mathbf{X} - \mathbf{A}\|_a \leq \|\mathbf{A} - \mathbf{Q}\mathbf{Q}^* \mathbf{A}\|_a + \|\Psi_2^\dagger \Psi_1 (\mathbf{Q}_\perp^* \mathbf{A})\|_a \quad (25)$$

$$\leq \left(1 + \|\Psi_2^\dagger\|_2 \|\Psi_1\|_2\right) \left(1 + \|\Omega_2\|_2 \|\Omega_1^\dagger\|_2\right) \|\Sigma_2\|_a; \quad (26)$$

in particular,

$$\|\mathbf{A} - (\Psi \mathbf{Q})^\dagger \Psi \mathbf{A}\|_F^2 = \|\mathbf{A} - \mathbf{Q}\mathbf{Q}^* \mathbf{A}\|_F^2 + \|\Psi_2^\dagger \Psi_1 (\mathbf{Q}_\perp^* \mathbf{A})\|_F^2. \quad (27)$$

Recall that $\mathbf{X} = (\Psi \mathbf{H})^\dagger \Psi \mathbf{A}$ in Algorithm 4. Denote $[\mathbf{X}]_r$ as the best rank- r approximation of $\mathbf{X}$. Analogous to the real/complex counterpart, $[\mathbf{X}]_r$ is also given by the rank- r truncated QSVD [22]. The truncation error is estimated in the following theorem:

Theorem 4.2 (Truncation error). *Let $\mathbf{H}$, $\mathbf{X}$ be generated by Algorithm 4 with (19) hold, and $\lfloor \mathbf{X} \rfloor_r$ be truncated by Algorithm 5. Then*

$$\| \mathbf{H} \lfloor \mathbf{X} \rfloor_r - \mathbf{H} \mathbf{X} \|_a \leq \kappa(\mathbf{H}) (\| \mathbf{A} - \mathbf{H} \mathbf{X} \|_a + \tau_r(\mathbf{A})). \quad (28)$$

To make the proof of Theorem 4.1 clear, we devide it in a series of lemmas.

Lemma 4.1. *Let $\mathbf{A} \in \mathbb{Q}^{m \times n}$ and $\mathbf{B} \in \mathbb{Q}^{n \times p}$; then $(\mathbf{A}\mathbf{B})^\dagger \neq \mathbf{B}^\dagger \mathbf{A}^\dagger$ in general. However, we have the following sufficient conditions for $(\mathbf{A}\mathbf{B})^\dagger = \mathbf{B}^\dagger \mathbf{A}^\dagger$:*

1. $\mathbf{A}$ has orthonormal columns ($\mathbf{A}^* \mathbf{A} = \mathbf{A}^\dagger \mathbf{A} = I$) or
2. $\mathbf{B}$ has orthonormal rows ($\mathbf{B} \mathbf{B}^* = \mathbf{B} \mathbf{B}^\dagger = I$) or
3. $\mathbf{A}$ has full column rank ($\mathbf{A}^\dagger \mathbf{A} = I$) and $\mathbf{B}$ has full row rank ($\mathbf{B} \mathbf{B}^\dagger = I$) or
4. $\mathbf{B} = \mathbf{A}^*$ or
5. $\mathbf{B} = \mathbf{A}^\dagger$.

Proof. By Lemma 2.2, it suffices to prove $(\chi_{\mathbf{A}} \chi_{\mathbf{B}})^\dagger = \chi_{\mathbf{B}}^\dagger \chi_{\mathbf{A}}^\dagger$. For two complex matrices E and F , [13] proved that $(EF)^\dagger = F^\dagger E^\dagger$ holds if and only if:

$$E^\dagger E F F^* E^* E F F^\dagger = F F^* E^* E. \quad (29)$$

Let $E = \chi_{\mathbf{A}}$, $F = \chi_{\mathbf{B}}$. Using the properties of generalized inverse and complex representation of quaternion matrices, one can check that each condition in this lemma can make (29) hold. $\square$

Lemma 4.2. *Let $\mathbf{H} \in \mathbb{Q}^{m \times s}$ ($m \geq s$) have full column rank, $\mathbf{Q} \in \mathbb{Q}^{m \times s}$ is given by (21), and Ψ is an arbitrary $l \times m$ ($s \leq l \leq m$) quaternion matrix with full row rank. Then we have:*

$$\mathbf{H} \mathbf{X} = \mathbf{H} (\Psi \mathbf{H})^\dagger \Psi \mathbf{A} = \mathbf{Q} (\Psi \mathbf{Q})^\dagger \Psi \mathbf{A}. \quad (30)$$

Proof. Let $\mathbf{H} = \mathbf{Q} \mathbf{B}$ where $\mathbf{B} \in \mathbb{Q}^{s \times s}$ is invertible. Then

$$\begin{aligned} \mathbf{H} (\Psi \mathbf{H})^\dagger \Psi \mathbf{A} &= \mathbf{Q} \mathbf{B} (\Psi \mathbf{Q} \mathbf{B})^\dagger \Psi \mathbf{A} \\ &\stackrel{(\text{Lemma 4.1, point 3})}{=} \mathbf{Q} \mathbf{B} \mathbf{B}^{-1} (\Psi \mathbf{Q})^\dagger \Psi \mathbf{A} \\ &= \mathbf{Q} (\Psi \mathbf{Q})^\dagger \Psi \mathbf{A}. \end{aligned}$$

The second identity is from Lemma 4.1 because $\Psi \mathbf{Q} \in \mathbb{Q}^{l \times s}$ ($l \geq s$) has full column rank (Ψ^* and $\mathbf{Q}$ are both of full column rank) and $\mathbf{B}$ is invertible. $\square$

The following lemma is a trivial quaternion version of [41, Lemma A.4].

Lemma 4.3. *Assume that Ψ_2 has full column rank; then*

$$(\Psi \mathbf{Q})^\dagger \Psi \mathbf{A} - \mathbf{Q}^* \mathbf{A} = \Psi_2^\dagger \Psi_1 (\mathbf{Q}_\perp^* \mathbf{A}).$$

The follow lemma comes from [27, Section 4.3]; see also [14, Theorem 9.1].

Lemma 4.4 (deterministic error bound). *With the notations in Algorithm 4 and in (24), and $\mathbf{Q}$ is as in Lemma 4.2,*

$$\begin{aligned} \| \mathbf{A} - \mathbf{Q} \mathbf{Q}^* \mathbf{A} \|_a^2 &\leq \| \Sigma_2 \|_a^2 + \| \Sigma_2 \Omega_2^* (\Omega_1^*)^\dagger \|_a^2 \\ &\leq \left(\| \Sigma_2 \|_a + \| \Sigma_2 \Omega_2^* (\Omega_1^*)^\dagger \|_a \right)^2. \end{aligned}$$

Now we prove Theorem 4.1. Note that $(\mathbf{X}^\dagger)^* = (\mathbf{X}^*)^\dagger$ will be frequently used in the proof.

Proof of Theorem 4.1. By Lemma 4.2, it holds that

$$\|\mathbf{A} - \mathbf{H}\mathbf{X}\|_a = \|\mathbf{A} - \mathbf{H}(\Psi\mathbf{H})^\dagger \Psi\mathbf{A}\|_a = \|\mathbf{A} - \mathbf{Q}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}\|_a. \quad (31)$$

Thus, it suffices to evaluate the right part of (31). Then,

$$\begin{aligned} \|\mathbf{A} - \mathbf{Q}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}\|_a &\leq \|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|_a + \|\mathbf{Q}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|_a \\ &= \|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|_a + \|\Psi_2^\dagger \Psi_1 (\mathbf{Q}_\perp^* \mathbf{A})\|_a, \end{aligned}$$

where the equality uses that both Frobenius and spectral norm are unitary invariant and by Lemma 4.3. Note that

$$\|\Psi_2^\dagger \Psi_1 (\mathbf{Q}_\perp^* \mathbf{A})\|_a \leq \|\Psi_2^\dagger\|_2 \|\Psi_1\|_2 \|\mathbf{Q}_\perp^* \mathbf{A}\|_a \quad (32)$$

and

$$\|\mathbf{Q}_\perp^* \mathbf{A}\|_a = \|\mathbf{Q}_\perp \mathbf{Q}_\perp^* \mathbf{A}\|_a = \|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|_a;$$

it follows from Lemma 4.4 that

$$\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|_a \leq \left(1 + \|(\Omega_1^*)^\dagger\|_2 \|\Omega_2^*\|_2\right) \|\Sigma_2\|_a.$$

In particular, using $\Psi_2^\dagger \Psi_1 (\mathbf{Q}_\perp^* \mathbf{A}) = \mathbf{Q}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}$ and Pythagorean identity,

$$\begin{aligned} \|\mathbf{A} - \mathbf{Q}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}\|_F^2 &= \|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A} + \mathbf{Q}\mathbf{Q}^*\mathbf{A} - \mathbf{Q}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}\|_F^2 \\ &= \|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|_F^2 + \|\Psi_2^\dagger \Psi_1 (\mathbf{Q}_\perp^* \mathbf{A})\|_F^2. \end{aligned}$$

□

Proof of Theorem 4.2. Let $\mathbf{H} = \mathbf{Q}\mathbf{B}$ as (21) where $\mathbf{Q}$ is partially orthonormal and $\mathbf{B}$ is invertible. Then $\mathbf{X} = (\Psi\mathbf{H})^\dagger \Psi\mathbf{A} = (\Psi\mathbf{Q}\mathbf{B})^\dagger \Psi\mathbf{A} = \mathbf{B}^{-1}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}$, where the last equality follows from the proof of Lemma 4.2 and that a random fat $\Psi \in \mathbb{Q}^{l \times m}$ drawn from continuous distribution has full row rank generically. Denote $\mathbf{A}_{in} := \mathbf{Q}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}$. We have:

$$\begin{aligned} \|\mathbf{H}\mathbf{X} - \mathbf{H}\lfloor\mathbf{X}\rfloor_r\|_a &\leq \|\mathbf{H}\|_2 \|\mathbf{X} - \lfloor\mathbf{X}\rfloor_r\|_a \\ &= \|\mathbf{H}\|_2 \|\mathbf{B}^{-1}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A} - \lfloor\mathbf{B}^{-1}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}\rfloor_r\|_a \\ &\leq \|\mathbf{H}\|_2 \|\mathbf{B}^{-1}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A} - \mathbf{B}^{-1}\lfloor(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}\rfloor_r\|_a \\ &\leq \|\mathbf{H}\|_2 \|\mathbf{B}^{-1}\|_2 \cdot \|(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A} - \lfloor(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}\rfloor_r\|_a \\ &\stackrel{([41], \mathbf{Q}\lfloor(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}\rfloor_r = \mathbf{A}_{in})}{=} \kappa(\mathbf{H}) \|\mathbf{A}_{in} - \lfloor\mathbf{A}_{in}\rfloor_r\|_a \leq \kappa(\mathbf{H}) \|\mathbf{A}_{in} - \lfloor\mathbf{A}\rfloor_r\|_a \\ &\leq \kappa(\mathbf{H})(\|\mathbf{A}_{in} - \mathbf{A}\|_a + \|\mathbf{A} - \lfloor\mathbf{A}\rfloor_r\|_a) \\ &= \kappa(\mathbf{H})(\|\mathbf{A} - \mathbf{H}\mathbf{X}\|_a + \tau_r(\mathbf{A})), \end{aligned}$$

where the second inequality is because that $\mathbf{B}^{-1}\lfloor(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}\rfloor_r$ has rank at most r , which is no closer than $\lfloor\mathbf{B}^{-1}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}\rfloor_r$ to $\mathbf{B}^{-1}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A}$. The last equality uses Lemma 4.2 that $\mathbf{H}\mathbf{X} = \mathbf{H}(\Psi\mathbf{H})^\dagger \Psi\mathbf{A} = \mathbf{Q}(\Psi\mathbf{Q})^\dagger \Psi\mathbf{A} = \mathbf{A}_{in}$.

□

4.3 Selection of test matrices

There are advantages and disadvantages to using test matrices from various probability distributions and structure in creating sketches. Such distributions influence parameter selection, randomization and computation expenses, storage space complexity, trade cost in streaming and distributed systems, as well as the numerical stability and error bounds.

- **Gaussian.** Quaternion Gaussian matrices have similar properties to real or complex Gaussian matrices. In addition to practical performance, unitary invariance also enables more accurate a priori error bounds, which will be discussed in Section 5.
- **Radmacher.** Rademacher matrices has similar behavior to quaternion gaussian matrix. But it cost less in storage and arithmetic.
- **Sparse.** Sparse structural random matrices require much less storage and arithmetic costs compared to matrices with no structure. However, they are less reliable and numerically stable and may need more oversampling in QB stage.
- **Rectangular.** Tall random matrices act as approximate isometries. If we allow for a more flexible selection of sketch size parameters, the distribution may not be as crucial. However, to provide a priori error bound, assumptions of distribution and independence are also needed. It as randomized embedding will be discussed in Section 6.

5 Gaussian Test Matrices

We further quantify the probabilistic estimation of the QB error in Theorem 4.1 with Gaussian test matrices. The probabilistic truncation error can be then derived from Theorem 4.2 accordingly. Using the statistical properties of quaternion Gaussian matrices established in [27], the estimation can be derived using a similar deduction as in [41]. We first recall some results from [27].

Lemma 5.1. ([27]) *Let $\mathbf{G} \in \mathbb{Q}^{m \times n}$ be a quaternion Gaussian matrix and $\mathbf{S} \in \mathbb{Q}^{l \times m}$, $\mathbf{T} \in \mathbb{Q}^{n \times s}$ be fixed. Then:*

$$\begin{aligned} \mathbb{E} \|\mathbf{SGT}\|_F^2 &= 4 \|\mathbf{S}\|_F^2 \|\mathbf{T}\|_F^2, \\ \mathbb{E} \|\mathbf{SGT}\|_2 &\leq 3 (\|\mathbf{S}\|_2 \|\mathbf{T}\|_F + \|\mathbf{S}\|_F \|\mathbf{T}\|_2). \end{aligned}$$

Lemma 5.2. ([27]) *Let $\mathbf{G} \in \mathbb{Q}^{m \times n}$ ($m \leq n$) be a quaternion Gaussian matrix. Then*

$$\mathbb{E} \|\mathbf{G}^\dagger\|_F^2 = \frac{m}{4(n-m)+2}, \quad \text{and} \quad \mathbb{E} \|\mathbf{G}^\dagger\|_2 = \frac{e\sqrt{4n+2}}{2n-2m+2}.$$

Theorem 5.1 (Probabilistic QB error). *For $\mathbf{A} \in \mathbb{Q}^{m \times n}$ ($m \geq n$), let $(\mathbf{H}, \mathbf{X})$ be generated by Algorithm 4 with (19) hold, and let $\mathbf{H} = \mathbf{QB}$ as in (21). If Ψ and Ω in (17) are quaternion Gaussian, for any $k < s$, we have*

$$\mathbb{E} \|\mathbf{HX} - \mathbf{A}\|_F^2 \leq \left(\frac{4l+2}{4(l-s)+2} \right) \left(\frac{4s+2}{4(s-k)+2} \right) \|\Sigma_2\|_F^2. \quad (33)$$

Proof. For the Frobenius norm, from (27) of Theorem 4.1, it suffices to respectively estimate $\|\Psi_2^\dagger \Psi_1 (\mathbf{Q}_\perp^* \mathbf{A})\|_F^2$ and $\|\mathbf{A} - \mathbf{QQ}^* \mathbf{A}\|_F^2$. Owing to the marginal property of the standard normal distribution, $\Psi_2 = \Psi \mathbf{Q} \in \mathbb{Q}^{l \times s}$ and $\Psi_1 = \Psi \mathbf{Q}_\perp$ are statistically independent gaussian

matrices. We thus have

$$\begin{aligned}
\mathbb{E}_{\Psi} \|\Psi_2^\dagger \Psi_1 (\mathbf{Q}_\perp^* \mathbf{A})\|_F^2 &= \mathbb{E}_{\Psi_2} \mathbb{E}_{\Psi_1} \|\Psi_2^\dagger \Psi_1 (\mathbf{Q}_\perp^* \mathbf{A})\|_F^2 \\
&\stackrel{(\text{Lemma 5.1})}{=} 4 \mathbb{E} \|\Psi_2^\dagger\|_F^2 \|\mathbf{Q}_\perp^* \mathbf{A}\|_F^2 \\
&\stackrel{(\text{Lemma 5.2})}{\leq} \frac{4s}{4(l-s)+2} \|\mathbf{Q}_\perp^* \mathbf{A}\|_F^2 \\
&= \frac{4s}{4(l-s)+2} \|(I - \mathbf{Q}\mathbf{Q}^*)\mathbf{A}\|_F^2.
\end{aligned} \tag{34}$$

From (27) and the independence of Ω and Ψ ,

$$\mathbb{E} \|\mathbf{H}\mathbf{X} - \mathbf{A}\|_F^2 = \mathbb{E}_\Omega \left(1 + \frac{4s}{4(l-s)+2} \right) \|(I - \mathbf{Q}\mathbf{Q}^*)\mathbf{A}\|_F^2. \tag{35}$$

Lemma 4.4 shows that

$$\begin{aligned}
\mathbb{E}_\Omega \|(I - \mathbf{Q}\mathbf{Q}^*)\mathbf{A}\|_F^2 &\leq \|\Sigma_2\|_F^2 + \mathbb{E}_\Omega \|\Sigma_2 \Omega_2^* (\Omega_1^*)^\dagger\|_F^2 \\
&\leq \left(1 + \frac{4k}{4(s-k)+2} \right) \|\Sigma_2\|_F^2,
\end{aligned} \tag{36}$$

where the deduction of the second inequality is similar to (34). Plugging this into (35) yields (33). $\square$

6 Sub-Gaussian Test Matrices

This section establish the probabilistic QB error Theorem 4.1 with sug-Gaussian test matrices:

Theorem 6.1 (Probabilistic error). *Assume that the sketch parameters satisfy the $k \ll s \ll l \ll \min\{m, n\}$. Draw random test matrices $\Omega \in \mathbb{Q}^{n \times s}$ and $\Psi \in \mathbb{Q}^{l \times m}$ such that the rows of Ω^* and Ψ are independent sub-gaussian isotropic random vectors. Let $\mathbf{H}, \mathbf{X}$ be generated by Algorithm 4. Then we have*

$$\|\mathbf{H}\mathbf{X} - \mathbf{A}\|_a \leq \left(1 + \frac{2\sqrt{l} + 2C_K\sqrt{m-s} + t}{2\sqrt{l} - 2C_K\sqrt{s} - t} \right) \left(1 + \frac{2\sqrt{s} + 2C_K\sqrt{n-k} + t}{2\sqrt{s} - 2C_K\sqrt{k} - t} \right) \|\Sigma_2\|_a$$

with probability at least $1 - 2 \exp\left(-\frac{2c_K t^2}{K^4}\right)$, where C_K and c_K are only depend on the sub-gaussian norm K .

The main tool to prove the above theorem is Theorem 6.2, which gives a deviation bound for extremal singular values of a quaternion sub-Gaussian matrix. To achieve this, we need to use real representation of a quaternion matrix as a bridge, such that the results in [43] can be applied.

Similar to complex representation, a quaternion matrix $\mathbf{Q} = Q_w + Q_x \mathbf{i} + Q_y \mathbf{j} + Q_z \mathbf{k}$ has real representation [27]:

$$\mathbf{Q}_r = \begin{bmatrix} Q_w \\ Q_x \\ Q_y \\ Q_z \end{bmatrix} \text{ and } \Upsilon_Q = [\mathcal{J}_0 \mathbf{Q}_r, \mathcal{J}_1 \mathbf{Q}_r, \mathcal{J}_2 \mathbf{Q}_r, \mathcal{J}_3 \mathbf{Q}_r]$$

where $\mathcal{J}_0 = I_{4m}$ and

$$\mathcal{J}_1 = \begin{bmatrix} -e_2^T \\ e_1^T \\ e_4^T \\ -e_3^T \end{bmatrix} \otimes I_m, \quad \mathcal{J}_2 = \begin{bmatrix} -e_3^T \\ -e_4^T \\ e_1^T \\ e_2^T \end{bmatrix} \otimes I_m, \quad \mathcal{J}_3 = \begin{bmatrix} -e_4^T \\ e_3^T \\ -e_2^T \\ e_1^T \end{bmatrix} \otimes I_m.$$

All of $\mathcal{J}_i (i = 0, 1, 2, 3)$ are orthogonal. Spectral and Frobenius norms of a quaternion matrix $\mathbf{Q}$ can be represented by those of real matrices as below:

$$\|\mathbf{Q}\|_2 = \|\Upsilon_{\mathbf{Q}}\|_2 \geq \|\mathbf{Q}_r\|_2, \quad \|\mathbf{Q}\|_F = \frac{1}{2} \|\Upsilon_{\mathbf{Q}}\|_F = \|\mathbf{Q}_r\|_F.$$

Subsequently, we shall delineate certain fundamental definitions and lemmas pertaining to real probabilistic theory. These can be extended to the realm of quaternions by leveraging the correspondence between quaternion matrices and their real representations.

Definition 6.1. (sub-gaussian random variable, [43]) A random variable X is called a sub-gaussian random variable if satisfying:

$$\mathbb{P}\{|X| > t\} \leq \exp(1 - ct^2)$$

for all $t > 0$. And the sub-gaussian norm of X , denoted $\|X\|_{\Psi_2}$, is defined as:

$$\|X\|_{\Psi_2} = \sup_{p \geq 1} p^{-1/2} (\mathbb{E}|X|^p)^{1/p}$$

Definition 6.2. (sub-exponential random variable, [43]) A random variable X is called a sub-exponential random variable if satisfying

$$\mathbb{P}\{|X| > t\} \leq \exp(1 - t/K_1)$$

for all $t > 0$. And the sub-exponential norm of X , denoted $\|X\|_{\Psi_1}$, is defined as:

$$\|X\|_{\Psi_1} = \sup_{p \geq 1} p^{-1} (\mathbb{E}|X|^p)^{1/p}.$$

Definition 6.3. (sub-gaussian vector, [43]) A random vector X in $\mathbb{R}^n$ is called a sub-gaussian vector if the one-dimensional marginals (X, x) are sub-gaussian random variables for all $x \in \mathbb{R}^n$. The sub-gaussian norm of X is defined as:

$$\|X\|_{\Psi_2} = \sup_{x \in S^{n-1}} \|\langle X, x \rangle\|_{\Psi_2}.$$

Definition 6.4. (isotropic vector, [43]) A random vector $X \in \mathbb{R}^n$ is called isotropic if $\Sigma(X) = I$. Equivalently, X is isotropic if

$$\mathbb{E}\langle X, x \rangle^2 = \|x\|_2^2 \quad \text{for all } x \in \mathbb{R}^n.$$

Definition 6.5. (quaternion sub-gaussian random variable (matrix)) A quaternion random variable $x = x_w + x_x \mathbf{i} + x_y \mathbf{j} + x_z \mathbf{k}$ follows quaternion sub-gaussian distribution if x_w, x_x, x_y, x_z are randomly and independently drawn from a real sub-gaussian distribution. If all entries of the quaternion matrix A are independent and identically distributed to sub-gaussian distribution with sub-gaussian norm $\|A\|_{\Psi_2} = K$, we call the A is a sub-gaussian matrix with sub-gaussian norm $\|A\|_{\Psi_2} = K$.

Definition 6.6. (Quaternion isotropic vectors) Let $\mathbf{\Omega}$ be a quaternion isotropic vector in $\mathbb{Q}^n$ if its real column representation satisfies:

$$\mathbb{E} \mathbf{\Omega}_r \mathbf{\Omega}_r^T = I_{4n}$$

Lemma 6.1. ([43]) Let $X_1, \dots, X_N$ be independent centered sub-exponential random variables, and let $K = \max_i \|X_i\|_{\Psi_1}$. Then, for every $\varepsilon \geq 0$, we have:

$$\mathbb{P}\left\{\left|\sum_{i=1}^N X_i\right| \geq \varepsilon N\right\} \leq 2 \exp\left[-c \min\left(\frac{\varepsilon^2}{K^2}, \frac{\varepsilon}{K}\right) N\right].$$

Lemma 6.2. ([43]) Consider a real matrix B that satisfies

$$\|B^T B - I\| \leq \max(\delta, \delta^2)$$

for some $\delta > 0$. Then

$$1 - \delta \leq s_{\min}(B) \leq s_{\max}(B) \leq 1 + \delta. \quad (37)$$

Conversely, if B satisfies (37) for some $\delta > 0$ then $\|B^T B - I\| \leq 3 \max(\delta, \delta^2)$.

Lemma 6.3. Let X, Y be two random variables; we have:

$$\mathbb{P}\{X + Y > 2\varepsilon\} \leq \mathbb{P}\{X > \varepsilon\} + \mathbb{P}\{Y > \varepsilon\}$$

Proof. Using formula of total probability,

$$\begin{aligned} \mathbb{P}\{x + y > 2\varepsilon\} &= \mathbb{P}\{x + y > 2\varepsilon | y > \varepsilon\} \mathbb{P}\{y > \varepsilon\} + \mathbb{P}\{x + y > 2\varepsilon | y \leq \varepsilon\} \mathbb{P}\{y \leq \varepsilon\} \\ &\leq \mathbb{P}\{y > \varepsilon\} + \mathbb{P}\{x + y > 2\varepsilon | y \leq \varepsilon\} \\ &\leq \mathbb{P}\{y > \varepsilon\} + \mathbb{P}\{x > \varepsilon\} \end{aligned}$$

□

Lemma 6.4. Let X, Y be positive random variables; we have:

$$\mathbb{P}\{XY > \varepsilon^2\} \leq \mathbb{P}\{X > \varepsilon\} + \mathbb{P}\{Y > \varepsilon\}$$

Proof. Consider the random variables $X_1 = \ln(X)$, $Y_1 = \ln(Y)$. Then using Lemma 6.3 gives

$$\mathbb{P}\{X_1 + Y_1 > 2 \ln \varepsilon\} \leq \mathbb{P}\{X_1 > \ln \varepsilon\} + \mathbb{P}\{Y_1 > \ln \varepsilon\}.$$

□

The deviation bound for extreme singular values of a quaternion sub-Gaussian matrix is given as follows. The idea of the proof follows from [43, Theorem 39].

Theorem 6.2 (Deviation bound). Let $\mathbf{A} \in \mathbb{Q}^{N \times n}$ ($N > 4n$) be a quaternion matrix whose rows A_i are independent sub-Gaussian isotropic random quaternion vectors, Then for every $t \geq 0$, with probability at least $1 - \exp\left(-\frac{c_K t^2}{K^4}\right)$ one has

$$2\sqrt{N} - 2C\sqrt{n} - t \leq \sigma_{\min}(\mathbf{A}) \leq \sigma_{\max}(\mathbf{A}) \leq 2\sqrt{N} + 2C\sqrt{n} + t. \quad (38)$$

Here C_K and c_K only depend on the sub-Gaussian norm $K = \max_i \|A_i\|_{\Psi_2}$ of the rows.

Proof. First, from [27], $\sigma_{\min}(\Upsilon_{\mathbf{A}}) = \sigma_{\min}(\mathbf{A})$ and $\sigma_{\max}(\Upsilon_{\mathbf{A}}) = \sigma_{\max}(\mathbf{A})$. Thus we focus on the estimation of $\sigma_{\min}(\Upsilon_{\mathbf{A}})$ and $\sigma_{\max}(\Upsilon_{\mathbf{A}})$. Applying Lemma 6.2 for $B := \Upsilon_{\mathbf{A}}/\sqrt{4N}$, the conclusion is equivalent to

$$\left\|\frac{1}{4N} \Upsilon_{\mathbf{A}}^* \Upsilon_{\mathbf{A}} - I\right\|_2 \leq \max(\delta, \delta^2) =: \varepsilon \quad \text{where} \quad \delta = C\sqrt{\frac{4n}{4N}} + \frac{t}{\sqrt{4N}}. \quad (39)$$

We can evaluate the operator norm $\|\frac{1}{4N}\Upsilon_{\mathbf{A}}^*\Upsilon_{\mathbf{A}} - I\|_2$ in (40) on a $\frac{1}{4}$ -net $\mathcal{N}$ of the unit sphere S_{4n-1} :

$$\left\| \frac{1}{4N}\Upsilon_{\mathbf{A}}^*\Upsilon_{\mathbf{A}} - I \right\|_2 \leq 2 \max_{x \in \mathcal{N}} \left| \left\langle \left(\frac{1}{4N}\Upsilon_{\mathbf{A}}^*\Upsilon_{\mathbf{A}} - I \right) x, x \right\rangle \right| = 2 \max_{x \in \mathcal{N}} \left| \frac{1}{4N} \|\Upsilon_{\mathbf{A}} x\|_2^2 - 1 \right| \quad (40)$$

Write $\Upsilon_{\mathbf{A}}$ as $\Upsilon_{\mathbf{A}} = \begin{bmatrix} \Upsilon_{\mathbf{A}1} \\ \Upsilon_{\mathbf{A}2} \\ \Upsilon_{\mathbf{A}3} \\ \Upsilon_{\mathbf{A}4} \end{bmatrix}$ with $\Upsilon_{\mathbf{A}i} \in \mathbb{R}^{N \times 4n}$, $i = 1, 2, 3, 4$; here we call $\Upsilon_{\mathbf{A}i}$ the i -th block row of the real counterpart $\Upsilon_{\mathbf{A}}$. By its structure and that each row of $\mathbf{A}$ is independent and isotropic, each row of $\Upsilon_{\mathbf{A}i}$ is an independent sub-Gaussian isotropic real vector.

Fix any vector $x \in S^{4n-1}$, we will upper bound $P\left\{\left|\frac{1}{4N} \|\Upsilon_{\mathbf{A}i} x\|_2^2 - 1\right| > \frac{\varepsilon}{2}\right\}$ for each fix i . The idea comes from the proof of [43, Theorem 39]. First denote

$$\|\Upsilon_{\mathbf{A}i} x\|_2^2 = \sum_{j=1}^N \langle \Upsilon_{\mathbf{A}i}^j, x \rangle^2 =: \sum_{j=1}^N Z_j^2,$$

where $\Upsilon_{\mathbf{A}i}^j$ represents the j -th row of $\Upsilon_{\mathbf{A}i}$. $Z_j = \langle \Upsilon_{\mathbf{A}i}^j, x \rangle$ are independent sub-Gaussian random variables with $\mathbb{E}Z_j^2 = 1$ and $\|Z_j\|_{\Psi_2} \leq K$. Thus $Z_j^2 - 1$ are independent centered sub-exponential random variables. Using Lemma 6.1 to give:

$$\begin{aligned} \mathbb{P}\left\{\left|\frac{1}{N} \|\Upsilon_{\mathbf{A}i} x\|_2^2 - 1\right| \geq \frac{\varepsilon}{2}\right\} &= \mathbb{P}\left\{\left|\frac{1}{N} \sum_{j=1}^N Z_j^2 - 1\right| \geq \frac{\varepsilon}{2}\right\} \leq 2 \exp\left[-\frac{c_1}{K^4} \min(\varepsilon^2, \varepsilon)N\right] \\ &= 2 \exp\left[-\frac{c_1}{K^4} \delta^2 N\right] \leq 2 \exp\left[-\frac{c_1}{K^4} (C^2 n + t^2)\right], \end{aligned}$$

where c_1 is an absolute constant. By Lemma 6.3,

$$\begin{aligned} P\left\{\left|\frac{1}{4N} \|\Upsilon_{\mathbf{A}} x\|_2^2 - 1\right| > \frac{\varepsilon}{2}\right\} &\leq \sum_{i=1}^4 P\left\{\left|\frac{1}{4N} \|\Upsilon_{\mathbf{A}i} x\|_2^2 - 1\right| > \frac{\varepsilon}{2}\right\} \\ &\leq 8 \exp\left[-\frac{c_1}{K^4} (C^2 n + t^2)\right]. \end{aligned}$$

Taking the union bound over all vectors x in the net $\mathcal{N}$ of cardinality $|\mathcal{N}| \leq 9^n$, we obtain:

$$\mathbb{P}\left\{\max_{x \in \mathcal{N}} \left|\frac{1}{4N} \|\Upsilon_{\mathbf{A}} x\|_2^2 - 1\right| \geq \frac{\varepsilon}{2}\right\} \leq 9^n \cdot 8 \exp\left[-\frac{c_1}{K^4} (C^2 n + t^2)\right] \leq \exp\left(-\frac{c_1 t^2}{K^4}\right),$$

where the last inequality holds when $C \geq K^2 \sqrt{\frac{1}{c_1} (\ln 9 + \frac{\ln 8}{n})}$.

Back to (40), we have:

$$\mathbb{P}\left\{\left\|\frac{1}{4N}\Upsilon_{\mathbf{A}}^*\Upsilon_{\mathbf{A}} - I\right\|_2 \geq \varepsilon\right\} \leq \exp\left(-\frac{c_1 t^2}{K^4}\right)$$

It means that at least with probability $1 - \exp\left(-\frac{c_1 t^2}{K^4}\right)$, we have:

$$1 - \delta \leq \sigma_{\min}\left(\frac{1}{2\sqrt{N}}\Upsilon_{\mathbf{A}}\right) \leq \sigma_{\max}\left(\frac{1}{2\sqrt{N}}\Upsilon_{\mathbf{A}}\right) \leq 1 + \delta$$

where $\delta = C\sqrt{\frac{4n}{4N}} + \frac{t}{\sqrt{4N}}$ in (39). As noted at the beginning of the proof, this completes the proof of the theorem. $\square$

Remark 6.1. The right-hand side of inequality (38) still holds when $N \leq 4n$. However, the left-hand side may yield a trivial result.

If $\mathbf{A}$ has independent quaternion isotropic columns, the largest singular value can be estimated by using conjugate transposition $\mathbf{A}^*$.

Now we prove Theorem 6.1 based on Theorems 4.1 and 6.2.

Proof of theorem 6.1. Recall that Theorem 4.1 gives

$$\|\mathbf{H}\mathbf{X} - \mathbf{A}\|_a \leq \left(1 + \|\Psi_2^\dagger\|_2 \|\Psi_1\|_2\right) \left(1 + \|\Omega_2^*\|_2 \left(\Omega_1^\dagger\right)^* \right) \|\Sigma_2\|_a,$$

where $\Psi_2 = \Psi\mathbf{Q} \in \mathbb{Q}^{l \times s}$, $\Psi_1 = \Psi\mathbf{Q}_\perp \in \mathbb{Q}^{l \times (m-s)}$, $\Omega_1^* = \Omega^*\mathbf{V}_1 \in \mathbb{Q}^{s \times k}$, $\Omega_2^* = \Omega\mathbf{V}_2^* \in \mathbb{Q}^{s \times (n-k)}$ with independent isotropic sub-gaussian rows. Some deviation bounds will be given first.

By definition, $\|\Psi_2^\dagger\|_2 = 1/\sigma_{\min}(\Psi_2)$, $\|\Psi_1\|_2 = \sigma_{\max}(\Psi_1)$, $\|\Omega_2^*\|_2 = \sigma_{\max}(\Omega_2)$ and $\left(\Omega_1^\dagger\right)^* \|\Omega_1\|_2 = \sigma_{\min}(\Omega_1)$. Thus, by Lemma 6.4 and Theorem 6.2, we have:

$$\begin{aligned} \mathbb{P} \left\{ 1 + \|\Psi_2^\dagger\|_2 \|\Psi_1\|_2 > 1 + \frac{2\sqrt{l} + 2C_K\sqrt{m-s} + t}{2\sqrt{l} - 2C_K\sqrt{s} - t} \right\} &\leq \exp\left(-\frac{c_K t^2}{K^4}\right); \\ \mathbb{P} \left\{ 1 + \|\Omega_2^*\|_2 \left(\Omega_1^\dagger\right)^* > 1 + \frac{2\sqrt{s} + 2C_K\sqrt{n-k} + t}{2\sqrt{s} - 2C_K\sqrt{k} - t} \right\} &\leq \exp\left(-\frac{c_K t^2}{K^4}\right). \end{aligned}$$

Again by Lemma 6.4,

$$\begin{aligned} \mathbb{P} \left\{ \|\mathbf{H}\mathbf{X} - \mathbf{A}\|_a \geq \left(1 + \frac{2\sqrt{l} + 2C_K\sqrt{m-s} + t}{2\sqrt{l} - 2C_K\sqrt{s} - t}\right) \left(1 + \frac{2\sqrt{s} + 2C_K\sqrt{n-k} + t}{2\sqrt{s} - 2C_K\sqrt{k} - t}\right) \|\Sigma_2\|_a \right\} \\ \leq 2 \cdot \exp\left(-\frac{c_K t^2}{K^4}\right). \end{aligned}$$

□

7 Numerical Experiments

In this section, we test the practical algorithm by following examples. All the experiments are carried out in MATLAB 2021a on a personal computer with an Intel(R) CPU i7-12700 of 2.10 GHz and 32GB of RAM. All example use the one-pass randomized quaternion low-rank approximation algorithm with pseduo-QR and pseudo-SVD.

For Algorithm 4, the approximation error and relative error are defined as :

$$\text{relative error} = \frac{\|\mathbf{A} - \hat{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F} \quad (41)$$

7.1 Synthetic data

This subsection will be provided in the next version.

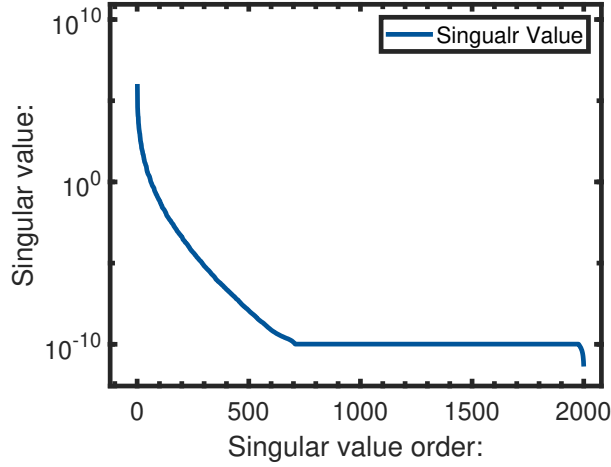


Figure 4: Singular values of *microConvection*

7.2 Image compression

This subsection will be provided in the next version.

7.3 Scientific data

Example 7.1. *In this example, we test the compression of the output of a computational fluid dynamics simulation by quaternion low-rank approximation. We have obtained a numerical simulation on a mesh of the 3D Navier–Stokes equations for microscopic natural convection for biological research applications. Velocity and pressure field are computed by QuickerSim CFD Toolbox for MATLAB. Then shear rate on each node can be computed by velocity field.*

*The origin data contains 20914 nodes and 2000 time instants. The elements of quaternion matrices *microConvection* represent the space velocity field by:*

$$a_{p,t} = v_x(p, t)\mathbf{i} + v_y(p, t)\mathbf{j} + v_z(p, t)\mathbf{k} \quad (42)$$

where $v = (v_x, v_y, v_z)$ is velocity vector field on the $\mathcal{M} \times \mathbb{R}$, $p \in \mathcal{M}$ is discrete point on vessel surface. Original data has rapidly decay spectrum as shown in figure 4.

Figure 5 show that our algorithm work efficiently and obtain high accuracy when origin data have rapidly decay spectrum. In particular, if we set $k = 60$, the relative Frobenius error is under 10^{-5} while the compression cost less than 2 second. It means that our algorithm can complete compression process before next input, which allows us compress on-the-fly large-scale quaternion output of scientific simulations.

Figure 6 illustrates that our compression can highly approximate the origin velocity field and compute the shear rates closely match the real data even if algorithm compress the original data from 256.81MB to 26.28MB.

Example 7.2. *In this section, we will compress the output of a 4D Lorenz-type chaotic system simulation by our randomized quaternion low-rank approximation algorithms. The chaotic*

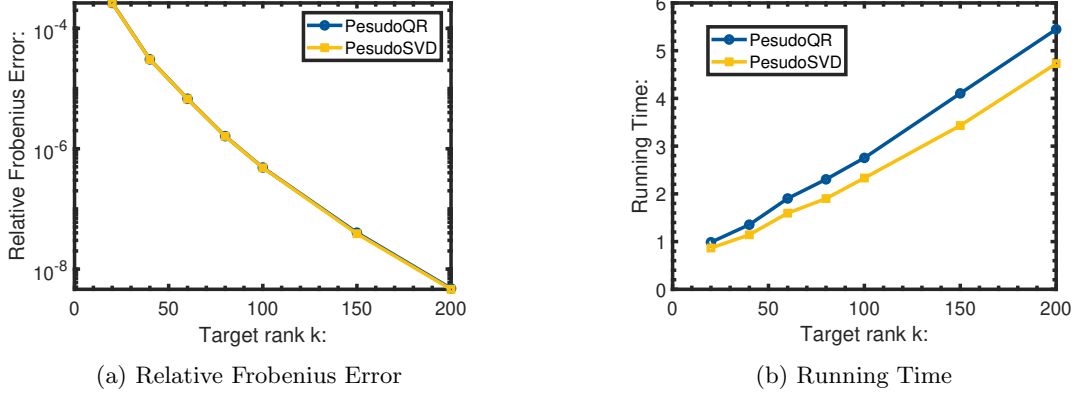


Figure 5: Relative Errors and Running Time of CFD simulation with different target rank k

system is as follows:

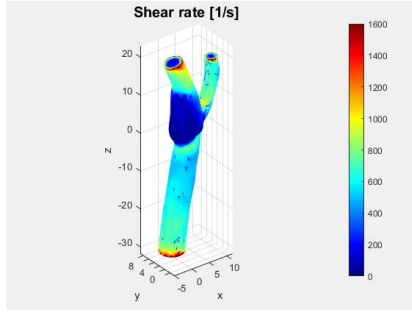
$$\begin{cases} \frac{dx}{dt} = a(y - x) - ew, \\ \frac{dy}{dt} = xz - hy, \\ \frac{dz}{dt} = b - xy - cz, \\ \frac{dw}{dt} = ky - dw, \end{cases}$$

where x, y, z and w are state variables and a, b, c, d, e, h are positive parameters of system. In our simulation, we set $a = 5, b = 20, c = 1, d = 0.1, k = 0.1, e = 20.6, h = 1$. And we set 10000 initial state (x, y, z, w) randomly from sphere $\|(x, y, z, w)\|_2 = 20$. Choosing 2000 time instance, we can obtain a 10000×2000 quaternion matrix which record the information of solutions. The quaternion matrix has slower spectrum decay shown in figure 7.

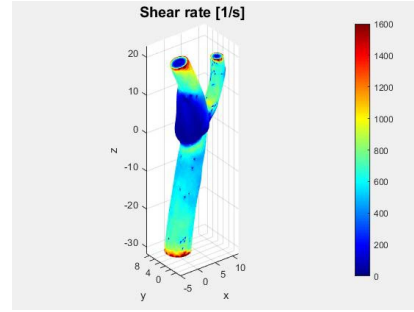
Figure 8 show that larger sketch size is necessary to obtain a high accuracy when input data have low spectrum decay. One-pass algorithms using pseudo-QR and pseduo-SVD have similar accuracy with the same sketch size. However, pseduo-QR may cost less time with larger sketch size.

8 Conclusions

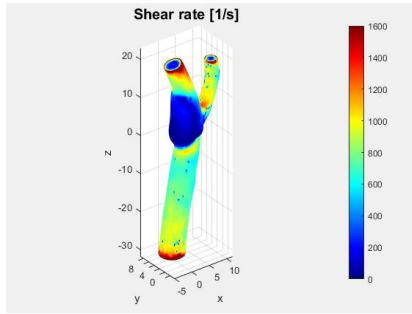
Existing quaternion rangefinders, which are based on quaternion orthogonalization, may be inefficient for large-scale problems. Based on the strategy of trading accuracy or space for speed, this work presented two practical rangefinders, which may not be orthonormal but still well-conditioned. The proposed rangefinders were then incorporated into the the one-pass algorithm proposed by Tropp et al. [41] for low-rank approximation to quaternion matrices. Throughout the whole algorithm, heavy quaternion computations has been transformed to QR, SVD, and solving linear equations in the complex arithmetic, such that mature scientific computing libraries or advanced algorithms can be employed to accelerate the computations. Theoretically, the probabilistic error bound was established; in particular, it was shown that the condition number of the rangefinder is proportional to the error, giving validity of using a non-orthonormal yet well-conditioned rangefinder. In addition, we established a deviation bound for the extreme singular values of a quaternion sub-Gaussian matrix, giving theoretical support of using a sub-Gaussian test matrix. Numerical experiments demonstrate that our algorithms work efficiently with less storage costs. Finally, we tested two practical numerical examples, including image dimensionality reduction and compression of large-scale scientific simulation data, to verify the



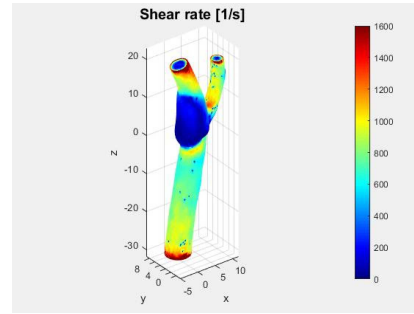
(a) Origin at 8th step



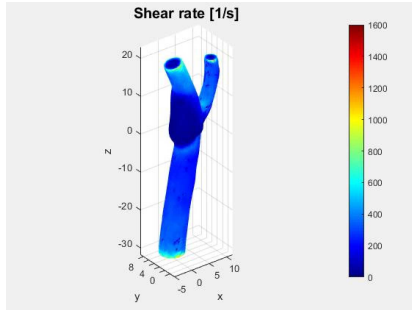
(b) Recovered at 8th step



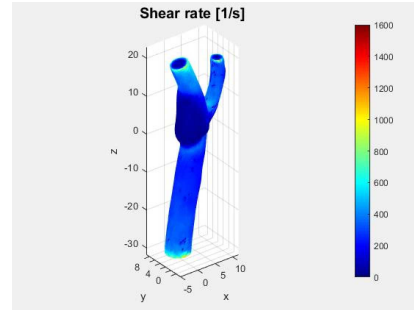
(c) Origin at 14th step



(d) Recovered at 14th step



(e) Origin at 23rd step



(f) Recovered at 23rd step

Figure 6: Shear rate computed from origin data and compressed data

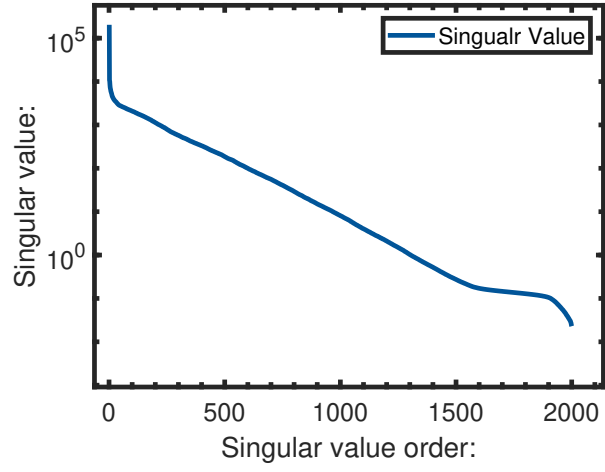
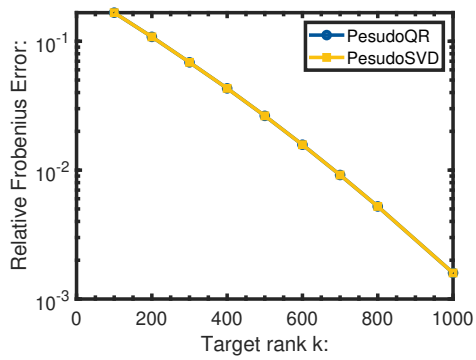
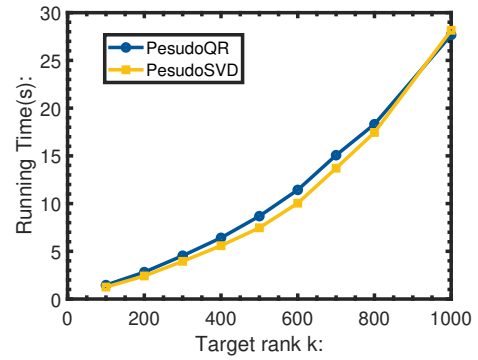


Figure 7: Singular values of 4D Lorenz system



(a) Relative Frobenius Error



(b) Running Time

Figure 8: Relative Errors and Running Time of Lorenz system with different target rank k

effectiveness of the algorithm.

References

- [1] Fast monte-carlo algorithms for finding low-rank approximations | Journal of the ACM. <https://dl.acm.org/doi/abs/10.1145/1039488.1039494>. 2
- [2] *Quaternion Matrix Computation*. 2, 5, 6, 13
- [3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. 2
- [4] Christos Boutsidis, David P. Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 236–249, New York, NY, USA, June 2016. Association for Computing Machinery. 2
- [5] Angelika Bunse-Gerstner, Ralph Byers, and Volker Mehrmann. A quaternion QR algorithm. 55(1):83–95. 2, 11
- [6] Junren Chen and Michael K. Ng. Color image inpainting via robust pure quaternion matrix completion: Error bound and weighted loss. *SIAM Journal on Imaging Sciences*, 15(3):1469–1498, 2022. 5
- [7] Yong Chen, Zhi-Gang Jia, Yan Peng, Ya-Xin Peng, and Dan Zhang. A new structure-preserving quaternion QR decomposition method for color image blind watermarking. 185:108088. 2
- [8] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, pages 205–214, Bethesda MD USA, May 2009. ACM. 2
- [9] Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, page 163–172, New York, NY, USA, 2015. Association for Computing Machinery. 2
- [10] Intel Corporation. Intel math kernel library (version x.x). <https://software.intel.com/content/www/us/en/develop/tools/math-kernel-library.html>, 20xx. Accessed: yyyy-mm-dd. 2
- [11] Todd A. Ell, Nicolas Le Bihan, and Stephen J. Sangwine. Quaternion Fourier Transforms. In *Quaternion Fourier Transforms for Signal and Image Processing*, pages 35–66. John Wiley & Sons, Ltd. 2
- [12] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, fourth edition edition. 12
- [13] T. N. E. Greville. Note on the Generalized Inverse of a Matrix Product. 8(4):518–521. 16
- [14] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. 53(2):217–288. 1, 2, 3, 15, 16
- [15] Nicholas J. Higham. Computing the Polar Decomposition—with Applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1160–1174, 1986. 9

- [16] Zhigang Jia, Michael K. Ng, and Guang-Jing Song. Lanczos method for large-scale quaternion singular value decomposition. *Numerical Algorithms*, 82(2):699–717, 2019. [14](#)
- [17] Zhigang Jia, Michael K. Ng, and Guang-Jing Song. Robust quaternion matrix completion with applications to image inpainting. *Numerical Linear Algebra with Applications*, 26(4):e2245, 2019. [5](#)
- [18] Zhigang Jia, Musheng Wei, and Sitao Ling. A new structure-preserving method for quaternion Hermitian eigenvalue problems. 239:12–24. [2](#)
- [19] Zhigang Jia, Musheng Wei, Mei-Xiang Zhao, and Yong Chen. A new real structure-preserving quaternion QR algorithm. 343(C):26–48. [2](#), [13](#)
- [20] Ravindran Kannan and Santosh Vempala. Randomized algorithms in numerical linear algebra. *Acta Numerica*, 26:95–135, May 2017. [2](#)
- [21] Anastasia Kireeva and Joel A. Tropp. Randomized matrix computations: Themes and variations, February 2024. [2](#)
- [22] Nicolas Le Bihan and Jérôme Mars. Singular value decomposition of quaternion matrices: A new tool for vector-sensor signal processing. *Signal Processing*, 84(7):1177–1199, 2004. [10](#), [15](#)
- [23] Chaoqian Li, Yonghe Liu, Fengsheng Wu, and Maolin Che. Randomized block Krylov subspace algorithms for low-rank quaternion matrix approximations. [2](#)
- [24] Ying Li, Musheng Wei, Fengxia Zhang, and Jianli Zhao. Real structure-preserving algorithms of Householder based transformations for quaternion matrices. 305:82–91. [2](#)
- [25] Ying Li, Musheng Wei, Fengxia Zhang, and Jianli Zhao. A fast structure-preserving method for computing the singular value decomposition of quaternion matrices. *Applied Mathematics and Computation*, 235:157–167, 2014. [14](#)
- [26] Ying Li, Musheng Wei, Fengxia Zhang, and Jianli Zhao. Real structure-preserving algorithms of Householder based transformations for quaternion matrices. *Journal of Computational and Applied Mathematics*, 305:82–91, October 2016. [2](#), [14](#)
- [27] Qiaohua Liu, Sitao Ling, and Zhigang Jia. Randomized Quaternion Singular Value Decomposition for Low-Rank Matrix Approximation. 44(2):A870–A900. [2](#), [4](#), [15](#), [16](#), [18](#), [19](#), [21](#)
- [28] Yonghe Liu, Fengsheng Wu, Maolin Che, and Chaoqian Li. Fixed-precision randomized quaternion singular value decomposition algorithm for low-rank quaternion matrix approximations. *Neurocomputing*, 580:127490, May 2024. [2](#)
- [29] Chengyao Lyu, Junjun Pan, Michael K. Ng, and Xile Zhao. Randomized low rank approximation for nonnegative pure quaternion matrices. 150:108940. [2](#)
- [30] Michael W. Mahoney. Randomized Algorithms for Matrices and Data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, November 2011. [2](#)
- [31] Per-Gunnar Martinsson and Joel Tropp. Randomized Numerical Linear Algebra: Foundations & Algorithms, March 2021. [2](#)
- [32] Per-Gunnar Martinsson and Joel A. Tropp. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, May 2020. [2](#)
- [33] Toshifumi Minemoto, Tejiro Isokawa, Haruhiko Nishimura, and Nobuyuki Matsui. Feed forward neural network with random quaternionic neurons. 136:59–68. [2](#)

- [34] Riley Murray, James Demmel, Michael W. Mahoney, N. Benjamin Erichson, Maksim Melnichenko, Osman Asif Malik, Laura Grigori, Piotr Luszczek, Michał Dereziński, Miles E. Lopes, Tianyu Liang, Hengrui Luo, and Jack Dongarra. Randomized Numerical Linear Algebra : A Perspective on the Field With an Eye to Software, April 2023. 2
- [35] Huan Ren, Ru-Ru Ma, Qiaohua Liu, and Zheng-Jian Bai. Randomized Quaternion QLP Decomposition for Low-Rank Approximation. 92(3):80. 2
- [36] Stephen J. Sangwine and Nicolas Le Bihan. Quaternion toolbox for MATLAB. <http://qtfm.sourceforge.net/>. 2
- [37] Stephen J. Sangwine and Nicolas Le Bihan. Quaternion singular value decomposition based on bidiagonalization to a real or complex matrix using quaternion Householder transformations. *Applied Mathematics and Computation*, 182(1):727–738, November 2006. 14
- [38] Soo-ChangPei, Ja-Han Chang, and Jian-Jiun Ding. Quaternion matrix singular value decomposition and its applications for color image processing. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, volume 1, pages I–805–8. IEEE. 2
- [39] Yanfeng Sun, Shangyou Chen, and Baocai Yin. Color face recognition based on quaternion matrix representation. 32(4):597–605. 2
- [40] Joel A. Tropp and Robert J. Webber. Randomized algorithms for low-rank matrix approximation: Design, analysis, and applications, September 2023. 2
- [41] Joel A. Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Practical Sketching Algorithms for Low-Rank Matrix Approximation. 38(4):1454–1485. 1, 2, 4, 6, 11, 14, 15, 16, 17, 18, 25
- [42] Joel A. Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Streaming Low-Rank Matrix Approximation with an Application to Scientific Simulation. 41(4):A2430–A2463. 2, 14
- [43] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Gitta Kutyniok and Yonina C. Eldar, editors, *Compressed Sensing: Theory and Applications*, pages 210–268. Cambridge University Press. 4, 19, 20, 21, 22
- [44] Musheng Wei, Ying Li, Fengxia Zhang, and Jianli Zhao. *Quaternion Matrix Computations*. Nova Science Publishers, Incorporated, 2018. 2
- [45] David P. Woodruff. Sketching as a Tool for Numerical Linear Algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1-2):1–157, 2014. 2
- [46] Franco Woolfe, Edo Liberty, Vladimir Rokhlin, and Mark Tygert. A fast randomized algorithm for the approximation of matrices. *Applied and Computational Harmonic Analysis*, 25(3):335–366, November 2008. 2
- [47] Fuzhen Zhang. Quaternions and matrices of quaternions. 251:21–57. 4, 5, 6, 9, 10
- [48] Huisheng Zhang and Hui Lv. Augmented Quaternion Extreme Learning Machine. 7:90842–90850. 2

Proof of Theorem 3.3. Let $\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^*$ be a compact QSVD of $\mathbf{H}$ where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_s)$ with σ_i arranged in a decreasing order. Denote

$$f(x) := (1 - \epsilon)x + \epsilon x^{-1}.$$

Then (6) shows that $\mathbf{H}_{new}$ has singular values $f(\sigma_i) = (1 - \epsilon)\sigma_i + \epsilon\sigma_i^{-1}$, $i = 1, \dots, s$. We first consider the upper bound of f on $[\sigma_s, 1]$ and $[1, \sigma_1]$, respectively.

For any ϵ and $x \in [1, \sigma_1]$,

$$f(x) = (1 - \epsilon)x + \epsilon x^{-1} \leq (1 - \epsilon)x + \epsilon x \leq x,$$

meaning that $f(x) \leq \sigma_1$ on $[1, \sigma_1]$.

Next, we consider $x \in [\sigma_s, 1]$. Observe that $f(x)$ is convex when $x > 0$, which achieves the maximal value on the boundary. It follows from $\epsilon \geq \sigma_s$ that $f(\sigma_s) \geq (1 - \epsilon)\sigma_s + \sigma_s \sigma_s^{-1} \geq 1 = f(1)$, i.e., f is upper bounded by $f(\sigma_s) = (1 - \epsilon)\sigma_s + \epsilon/\sigma_s$ on $[\sigma_s, 1]$. Furthermore, as $\sigma_s \leq 1$ and $\epsilon \leq \delta\sigma_s$,

$$f(x) \leq (1 - \epsilon)\sigma_s + \epsilon/\sigma_s \leq (1 - \delta\sigma_s)\sigma_s + \delta\sigma_s/\sigma_s = \delta + \sigma_s - \delta\sigma_s^2 < \delta + 1/(4\delta).$$

The analysis above together with $\sigma_1 \leq \sqrt{2}$ in Theorem 3.2 shows that the largest singular value of $\mathbf{H}_{new}$ cannot exceed $\max\{\sqrt{2}, \delta + 1/(4\delta)\}$.

On the other hand, it follows from the convexity of $f(x)$ that when $x > 0$, $f(x) \geq 2\sqrt{\epsilon(1 - \epsilon)}$. The range of ϵ implies $2\sqrt{\epsilon(1 - \epsilon)} \geq 2\sqrt{\sigma_s(1 - \sigma_s)}$. Thus $f(x) \geq 2\sqrt{\sigma_s(1 - \sigma_s)}$.

We also need a lower bound on σ_s . When $\kappa(H) > \max\{2\sqrt{2}\delta, 2\delta^2 + 1/2\} = 2\delta \max\{\sqrt{2}, \delta + 1/(4\delta)\} \geq 2\delta\sigma_1$, we obtain that $\sigma_s \leq 1/(2\delta)$.

Comparing the two upper bounds on f obtained previously, if $\sigma_1 \geq \delta + 1/(4\delta)$,

$$\kappa(\mathbf{H}_{new}) \leq \frac{\sigma_1}{2\sqrt{\sigma_s(1 - \sigma_s)}}; \quad (43)$$

then

$$\frac{\kappa(\mathbf{H}_{new})}{\sqrt{\kappa(\mathbf{H})}} \leq \frac{\sqrt{\sigma_1}}{2\sqrt{(1 - \sigma_s)}} \leq \sqrt{\sigma_1}/\sqrt{2} \leq \sqrt{\max\{\sqrt{2}, \delta + 1/(4\delta)\}/\sqrt{2}} \leq 1, \quad (44)$$

where the second inequality follows from $\sigma_s \leq 1/(2\delta) < 1/2$, and the last one comes from the range of δ . Similarly, if $\sigma_1 < \delta + 1/(4\delta)$,

$$\kappa(\mathbf{H}_{new}) \leq \frac{\delta + 1/(4\delta)}{2\sqrt{\sigma_s(1 - \sigma_s)}}; \quad (45)$$

then

$$\frac{\kappa(\mathbf{H}_{new})}{\sqrt{\kappa(\mathbf{H})}} \leq \frac{\delta + 1/(4\delta)}{2\sqrt{\sigma_1(1 - \sigma_s)}} \leq \frac{\delta + 1/(4\delta)}{2\sqrt{1 - 1/(2\delta)}} < 1, \quad (46)$$

where the second inequality follows from $\sigma_1 \geq 1$ and $\sigma_s \leq 1/(2\delta)$, while the last one comes from that $\frac{\delta + 1/(4\delta)}{2\sqrt{1 - 1/(2\delta)}}$ is non-decreasing on $[1, \sqrt{7}/2]$. The result follows. $\square$

Proof of Corollary 3.1. Using the analysis in Theorem 3.3, one can use the induction method to show that if all the singular values $\sigma_i(\mathbf{H}_k)$ lies in $[2\sqrt{\sigma_s(\mathbf{H}_{k-1})(1 - \sigma_s(\mathbf{H}_{k-1}))}, \max\{\sqrt{2}, \delta + 1/(4\delta)\}]$, then all $\sigma_i(\mathbf{H}_{k+1})$ also lie in $[2\sqrt{\sigma_s(\mathbf{H}_k)(1 - \sigma_s(\mathbf{H}_k))}, \max\{\sqrt{2}, \delta + 1/(4\delta)\}]$. Similar to the proof of Theorem 3.3, $\kappa(\mathbf{H}_{k+1}) < \sqrt{\kappa(\mathbf{H}_k)}$. $\square$

Proof of Proposition 3.1. We first use the induction method to show that $\text{span}(U_b, \mathcal{J}\overline{U_b}) = 2t$. For $t = 1$, this claim is true as $u_1 \perp \mathcal{J}\overline{u_1}$. Suppose now that we have found $U_k := [u_{i_1}, \dots, u_{i_k}] \in U_b$ ($k < t$) such that $\text{span}(U_k, \mathcal{J}\overline{U_k})$ has dimension $2k$. Note that U_b is partially orthonormal and $\text{span}(U_b)$ has dimension $2t > 2k$, and so there always exists at least a $u_{i_{k+1}} \in U_b \setminus U_k$, such that

$$u_{i_{k+1}} \notin \text{span}(U_k, \mathcal{J}\overline{U_k}). \quad (47)$$

We first show that (47) is equivalent to $\mathcal{J}\overline{u_{i_{k+1}}} \notin \text{span}(U_k, \mathcal{J}\overline{U_k})$. Suppose on the contrary that $\mathcal{J}\overline{u_{i_{k+1}}} \in \text{span}(U_k, \mathcal{J}\overline{U_k})$, which by Lemma 2.1 is equivalent to

$$\overline{u_{i_{k+1}}} = \mathcal{J}^* \mathcal{J}\overline{u_{i_{k+1}}} \in \text{span}(\mathcal{J}^* U_k, \mathcal{J}^* \mathcal{J}\overline{U_k}) \Leftrightarrow u_{i_{k+1}} \in \text{span}(\mathcal{J}\overline{U_k}, U_k),$$

deducing a contradiction.

Next, we will prove that $\text{span}(U_k, \overline{\mathcal{J}U_k}, u_{i_{k+1}}, \overline{\mathcal{J}u_{i_{k+1}}})$ has dimension $2(k+1)$. Denote $M := [U_k, \mathcal{J}U_k]$, $P_M := MM^\dagger$, and $P_{M^\perp} := I_{2m} - MM^\dagger$ the orthogonal projection onto $\text{span}(M)$ and $\text{span}(M)^\perp$, respectively. Then $u_{i_{k+1}}$ can be divided into two part:

$$P_{M^\perp} u_{i_{k+1}} \perp \text{span}(M) \quad \text{and} \quad P_M u_{i_{k+1}} \in \text{span}(M). \quad (48)$$

Correspondingly, $\overline{\mathcal{J}u_{i_{k+1}}}$ can be divided into $P_{M^\perp} \overline{\mathcal{J}u_{i_{k+1}}}$ and $P_M \overline{\mathcal{J}u_{i_{k+1}}}$, where we also notice that

$$\overline{\mathcal{J}P_{M^\perp} u_{i_{k+1}}} = \overline{\mathcal{J}u_{i_{k+1}}} - \overline{\mathcal{J}P_M u_{i_{k+1}}} = \overline{\mathcal{J}u_{i_{k+1}}} - P_M \overline{\mathcal{J}u_{i_{k+1}}} = P_{M^\perp} \overline{\mathcal{J}u_{i_{k+1}}},$$

with the second equality from Lemma 2.3.

By Lemma 2.1, $P_{M^\perp} u_{i_{k+1}} \perp \overline{\mathcal{J}P_{M^\perp} u_{i_{k+1}}}$, and due to the above relation, $\overline{\mathcal{J}P_{M^\perp} u_{i_{k+1}}} \perp M$. Thus,

$$\text{span}(M, P_{M^\perp} u_{i_{k+1}}, P_{M^\perp} \overline{\mathcal{J}u_{i_{k+1}}}) = \text{span}(M, P_{M^\perp} u_{i_{k+1}}, \overline{\mathcal{J}P_{M^\perp} u_{i_{k+1}}})$$

has dimension $2(k+1)$. Therefore,

$$\begin{aligned} & \text{span}(U_k, \overline{\mathcal{J}U_k}, u_{i_{k+1}}, \overline{\mathcal{J}u_{i_{k+1}}}) \\ &= \text{span}(M, P_{M^\perp} u_{i_{k+1}} + P_M u_{i_{k+1}}, P_M \overline{\mathcal{J}u_{i_{k+1}}} + P_{M^\perp} \overline{\mathcal{J}u_{i_{k+1}}}) \end{aligned}$$

also has dimension $2(k+1)$. Thus the induction method shows that there exists $U_{\tilde{b}} = [u_{i_1}, \dots, u_{i_t}] \in U_b$, such that $\text{span}(U_{\tilde{b}}, \mathcal{J}U_{\tilde{b}})$ has dimension $2t$.

It is obvious that $\mathcal{R}(U_{\tilde{b}}) \subset \mathcal{R}(U_b)$. Furthermore, for any u in $U_{\tilde{b}}$ corresponding to singular value σ of χ_Y , $\mathcal{J}u$ is also corresponding to σ . which means that $\mathcal{R}(\mathcal{J}U_{\tilde{b}}) \subset \mathcal{R}(U_b)$. Recall that $\mathcal{R}(U_b)$ also has dimension $2t$; thus $\mathcal{R}(U_{\tilde{b}}, \mathcal{J}U_{\tilde{b}}) = \mathcal{R}(U_b)$. $\square$