Randomized Large-Scale Quaternion Matrix Approximation: Practical Rangefinders and One-Pass Algorithm

Chao Chang* Yuning Yang*†

June 19, 2024

Abstract

Recently, randomized algorithms for low-rank approximation of quaternion matrices have received increasing attention. However, for large-scale problems, existing quaternion orthonormalizations are inefficient, leading to slow rangefinders. To address this, by appropriately leveraging efficient scientific computing libraries in the complex arithmetic, this work devises two practical quaternion rangefinders, one of which is non-orthonormal yet well-conditioned. They are then integrated into the quaternion version of a one-pass algorithm, which originally takes orthonormal rangefinders only. We establish the error bounds and demonstrate that the error is proportional to the condition number of the rangefinder. The probabilistic bounds are exhibited for both quaternion Gaussian and sub-Gaussian embeddings. Numerical experiments demonstrate that the one-pass algorithm with the proposed rangefinders significantly outperforms previous techniques in efficiency. Additionally, we tested the algorithm in a 3D Navier-Stokes equation (5.22GB) and a 4D Lorenz-type chaotic system (5.74GB) data compression, as well as a 31365×27125 image compression to demonstrate its capability for handling 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) [15] 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. [38] developed 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 authors devised also a one-pass algorithm using three sketches and applied it to streaming data [39]. Prior to these work, randomized algorithms have been studied extensively in the literature; see, e.g., [12, 44, 8, 28, 43, 9, 4], and the recent surveys [37, 20, 21, 32, 29].

*College of Mathematics and Information Science, Guangxi University, Nanning, 530004, China
†Corresponding author: Yuning Yang, yyang@gxu.edu.cn.

Despite the noncummutativity in quaternion multiplications, quaternion matrices have been widely used in various applications such as signal processing [11], color image analysis [30, 36], and machine learning [47, 31] in recent years. Randomized quaternion low-rank matrix approximation has garnered increasing attention very recently. Liu et al. [25] 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 [27]. Ren et al. [34] 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-precesion randomized quaternion SVD was studied in [26] and a randomized quaternion UTV decomposition has been proposed in [45].

The framework of the HMT algorithm [15] (also [38, 39], and the quaternion randomized algorithms [25, 34, 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 [15] or approximately [38, 39]. 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 [1] 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) [35] 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 real representation of a quaternion matrix, avoiding quaternion operations and smartly reducing computational complexity. The structure-preserving quaternion Householder QR (QHQR) [19, 24] is numerically stable and accurate; the structure-preserving quaternionic modified Gram-Schimit (QMGS) [41] 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 [25, 23, 34].

1.3 Limitation and motivation

The above quaternionic orthonormalization 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 algorithm mentioned previously[1], is at least an order of magnitude slower than MATLAB's

[1]The code of QHQR was downloaded from Jia's homepage http://maths.jsnu.edu.cn/_upload/article/files/40/5c/0abecd234d2c909be8b4fac9c4ad/1f0499ce-65d1-4364-9000-b8998137e516.zip and that of QMGS was implemented from [41]. QMGS and QHQR both have complexity $O(mn^2)$ for matrices of size $m \times n$ [41, 19, 24, 13]. If the full orthonormal matrix Q is computed, as was done in the implementation of QHQR, then its complexity becomes $O(m^2n)$; see also [13, p. 249]. This is expensive when $m \gg n$, as is usually the size of rangefinders. Nevertheless, empirically we find that even without explicitly forming Q, QHQR is still about twice slower than QMGS, which also confirms the discussions in [13, p. 255].

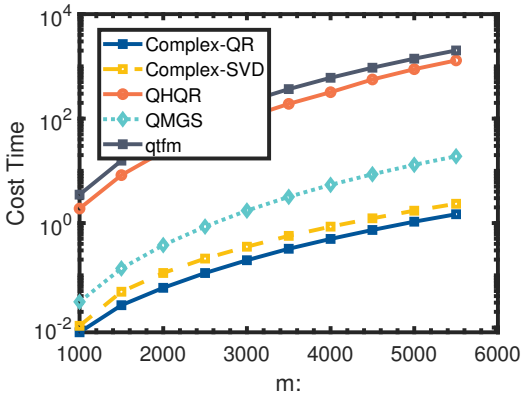


Figure 1: Running time comparisons of QTFM's `qr`, QHQR, QMGS, and MATLAB's `qr` and `svd` (the latter two are applied to the full complex representation; see Sect. 2.2 for its definition). m from 1000 to 5500, $n = 0.1m$.

built in functions `qr` and `svd` for 5000×500 quaternion matrices; see Fig. 1[2]. One of the possible reasons is that 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 LAPACK or 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.

A common design philosophy to gain efficiency 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 [15, 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 to appropriately utilizing mature scientific computing libraries or advanced algorithms in the real/complex arithmetic to accelerate our computations.

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 while maintaining favorable condition numbers. The influence of condition numbers on approximation accuracy will be examined theoretically. Specifically:

[2]MATLAB's `qr` and `svd` are applied to the full complex representation of the quaternion matrix, which is of size $2m \times 2n$. Of course, this is a rough comparison, as complex QR and SVD may not directly generate an orthonormal rangefinder in the quaternion domain; nevertheless, the purpose of this comparison is to show that the speed of quaternion rangefinders still has a large room to improve.

In Section 3, we provide two practical 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-conditioned sketch. Almost all the computations are built upon established scientific computing libraries in the complex arithmetic to ensure their efficiency. Comparisons with previous techniques in terms of time and accuracy are illustrated in Fig. 2 and 3.

In Section 4, the proposed rangefinders are incorporated into the quaternion version of the one-pass framework of [38]. Originally, the algorithm only employs orthonormal rangefinders. Our theoretical findings in Sections 4.2, 4.3, and 4.4 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.
- The probabilistic bounds are exhibited for either quaternion Gaussian or 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 probabilistic bounds rely on the non-asymptotic deviation bounds of extreme singular values of a quaternion random matrix. The quaternion Gaussian case was first studied in [25], while we generalize a result in the real case by Vershynin [40] to quaternions.

Finally, Section 5 evaluated the performance of our algorithm and previous ones through various experiments using synthetic data. To demonstrate the applicability of our approach in large-scale problems, we tested the algorithm in a 3D Navier-Stokes equation (5.22GB) and a 4D Lorenz-type chaotic system (5.74GB) data compression, as well as a 31365×27125 color image compression. Note that existing quaternion randomized algorithms [25, 34, 23, 26, 45] were only reported to handled matrices of size up to a few thousand.

Our implementation in MATLAB is available at `github.com/Mitchell-Cxyk/RQLRMA`.

2 Preliminaries on Quaternions

Throughout this work, quaternion vectors and matrices are written in bold face characters; quaternion scalars and real/complex scalars, vectors, and matrices are written as italic characters.

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 [46]. $\|\mathbf{Q}\|_F := (\sum_{k\in\{w,x,y,z\}} \|Q_k\|_F^2)^{1/2}$.

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 $\mathbf{v}_1, \ldots, \mathbf{v}_r$, they are right linearly independent if for quaternions $k_1, \ldots, k_r$,

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

Most linear algebra concepts and results can be transplanted to the right vector space in parallel [46] and throughout this work, we always omit the prefix "right".

Given a quaternion matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_r]$, its range space is defined as

$$\mathcal{R}(\mathbf{V}) = \operatorname{span}(\mathbf{v}_1, \ldots, \mathbf{v}_r) = \left\{ \sum\nolimits_{i=1}^{r} \mathbf{v}_i k_i \;\mid\; k_i \in \mathbb{Q},\ i = 1, \ldots, r \right\}.$$

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 $\mathbf{u} \perp \mathbf{v}$, and orthonormal if in addition $\langle \mathbf{u}, \mathbf{u} \rangle = \langle \mathbf{v}, \mathbf{v} \rangle = 1$. Given 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 [46]:

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

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

Proposition 2.1 ([46]). *Let $\mathbf{P}, \mathbf{Q}$ be quaternion matrices of proper size. Then*

$$\chi_{k_1\mathbf{P} + k_2\mathbf{Q}} = k_1 \chi_{\mathbf{P}} + k_2 \chi_{\mathbf{Q}} (k_1, k_2 \in \mathbb{R}), \quad \chi_{\mathbf{PQ}} = \chi_{\mathbf{P}} \chi_{\mathbf{Q}},$$
$$\chi_{\mathbf{Q}^*} = \chi_{\mathbf{Q}}{}^*, \quad \chi_{\mathbf{Q}^{-1}} = \chi_{\mathbf{Q}}{}^{-1} \text{ if } \mathbf{Q}^{-1} \text{ exists},$$

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

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

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

We call $\mathbf{Q}_c$ the *compact* complex representation of $\mathbf{Q}$. $\mathbf{Q}_a$ can be generated from $\mathbf{Q}_c$ as $\mathbf{Q}_a = \mathcal{J}\overline{\mathbf{Q}_c}$, where $\mathcal{J} := \begin{bmatrix} 0 & -I_m \\ I_m & 0 \end{bmatrix}$ is the symplectic matrix. 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. *J-adjoint satisfies:*

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

The quaternion Moore-Penrose (MP) inverse can be defined similarly as its real/complex counterpart [41, 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:

$$\mathbf{AXA} = \mathbf{A}, \quad \mathbf{XAX} = \mathbf{X}, \quad (\mathbf{AX})^* = \mathbf{AAX}, \quad (\mathbf{XA})^* = \mathbf{XA}.$$

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. □

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

Lemma 2.4. *Let $\mathbf{A}, \mathbf{B} \in \mathbb{Q}^{m\times n}$. Then $\mathcal{R}(\mathbf{A}) = \mathcal{R}(\mathbf{B})$ if and only if $\mathcal{R}(\chi_{\mathbf{A}}) = \mathcal{R}(\chi_{\mathbf{B}})$.*

One can verify the above lemma as $\mathbf{A}\mathbf{A}^\dagger = \mathbf{B}\mathbf{B}^\dagger \Leftrightarrow \chi_{\mathbf{A}\mathbf{A}^\dagger} = \chi_{\mathbf{B}\mathbf{B}^\dagger} \Leftrightarrow \chi_{\mathbf{A}}\chi_{\mathbf{A}}{}^\dagger = \chi_{\mathbf{B}}\chi_{\mathbf{B}}{}^\dagger$.

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

Theorem 2.1. *(Compact QSVD [46, 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 $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0) \in \mathbb{R}^{n\times n}$ with $\sigma_1 \geq \cdots \geq \sigma_r > 0$, such that $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*$.*

The following property, which can be deduced from [46], 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 $\chi_{\mathbf{A}} = \chi_{\mathbf{U}} \cdot S \cdot \chi_{\mathbf{V}^*}$ with $S = \mathrm{diag}(\Sigma, \Sigma) \in \mathbb{R}^{2n\times 2n}$ is a compact SVD of $\chi_{\mathbf{A}}$, and vice versa.*

Lemma 2.5. *[41, Section 1.6] Under the notations in Theorem 2.1, if $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*$, then $\mathbf{A}^\dagger = \mathbf{V}\Sigma^\dagger\mathbf{U}^*$ with $\Sigma^\dagger = \mathrm{diag}(\sigma_r^{-1}, \ldots, \sigma_1^{-1}, 0, \ldots, 0) \in \mathbb{R}^{n\times n}$.*

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 $s \ll n$ [38] (s is close to the target rank and sometimes can be regarded as a constant), takes a sketch $Y = A\Omega$, and then orthonormalizes it, i.e.,

$$Y = A\Omega \in \mathbb{R}^{m\times s},\ Q = \texttt{orth}(Y),$$

where Q is orthnormal 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 appropriately employ mature libraries such as QR, SVD, and linear equation solvers in complex arithmetic for heavy computations. Finally, we will compare the running time and accuracy of the proposed rangefinders with previous techniques.

3.1 Pseudo-QR

Given a quaternion matrix $\mathbf{X} = X_0 + X_1\mathbf{j} \in \mathbb{Q}^{m\times n}$, its full information has been contained in its compact represtation $\mathbf{X}_c = \begin{bmatrix} X_0 \\ -\overline{X_1} \end{bmatrix} \in \mathbb{C}^{2m\times n}$, while full representation $\chi_{\mathbf{X}}$ futher preserves its

structure as an operator. Although $\mathbf{X}$ 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 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 thin QR in the complex arithmetic can be directly applied to $\mathbf{Y}_c$:

$$\mathbf{Y}_c = QR, \tag{1}$$

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

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

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 $\mathbf{Y}_c$:

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

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

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

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

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:

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

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

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 $\mathbf{Y}$.*

Throughout this work, we respectively denote

$$\sigma_i(\mathbf{H}),\ \sigma_{\max}(\mathbf{H}) = \sigma_1(\mathbf{H}),\ \sigma_{\min}(\mathbf{H}) = \sigma_s(\mathbf{H}),\ \kappa(\mathbf{H}) := \sigma_{\max}(\mathbf{H})/\sigma_{\min}(\mathbf{H}),$$

as the i-th, the largest and the smallest singular value, as well as the condition number of $\mathbf{H}$. The proposition belows shows that, although $\mathbf{H}$ may be non-orthonormal in the quaternion domain, its singular values are structured.

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

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

with $0 \le \mu_i \le 1$, and $\sigma_{\max}(\mathbf{H}) \le \sqrt{2}$.

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

$$\chi_{\mathbf{H}} = \begin{bmatrix} H_0 & H_1 \\ -\overline{H_1} & \overline{H_0} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_c, & \mathbf{H}_a \end{bmatrix} = \begin{bmatrix} \mathbf{H}_c, & \mathcal{J}\overline{\mathbf{H}_c} \end{bmatrix} \in \mathbb{C}^{2m\times 2s}.$$

It follows from (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

$$\chi_{\mathbf{H}}{}^*\chi_{\mathbf{H}} = \begin{bmatrix} I_s & T \\ T^* & I_s \end{bmatrix}. \tag{5}$$

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

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

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 $\sigma_{\max}(\mathbf{H}) \le \sqrt{2}$. □

Remark 3.2. *The two properties above are not enough to control the condition number* $\kappa(\mathbf{H})$. *In fact, the analysis shows that the smallest singular value of* $\mathbf{H}$ *depends on* $T = \mathbf{H}_c^*\mathbf{H}_a$. *If the angle between* $\mathrm{span}(\mathbf{H}_c)$ *and* $\mathrm{span}(\mathbf{H}_a)$ *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* $\mathrm{span}(\mathbf{H}_c)$ *and* $\mathrm{span}(\mathbf{H}_a)$ *are perpendicular to each other, then* T *is exactly* 0 *and the smallest eigevalue of* $\chi_{\mathbf{H}}{}^*\chi_{\mathbf{H}}$ *is* 1. *However, by the construction* (3), $\mathrm{span}(\mathbf{H}_c) = \mathrm{span}(\mathbf{Y}_c)$ *and* $\mathrm{span}(\mathbf{H}_a) = \mathrm{span}(\mathbf{Y}_a)$. *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* $\mathrm{span}(\mathbf{Y}_c)$ *and* $\mathrm{span}(\mathbf{Y}_a)$. *Thus* $\sigma_{\min}(\mathbf{H})$ *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 $\kappa(\mathbf{H})$, it is possible to perform the correction step several times:

$$\mathbf{H} \leftarrow 0.5\mathbf{H}(3I_s - \mathbf{H}^*\mathbf{H}). \tag{6}$$

It was shown in [16, Sect. 8] that if all singular values $\sigma_i(\mathbf{H}) \in [0, \sqrt{3}]$ (which is confirmed by Proposition 3.2), then (6) ensures that all $\sigma_i(\mathbf{H}) \to 1$ quadratically, and hence $\mathbf{H}$ is gradually well-conditioned. However, we empirically find that (6) converges very slowly. Instead, we resort to the following range-preserving correction step:

$$\mathbf{H}_{new} \leftarrow (1-\epsilon)\mathbf{H} + \epsilon(\mathbf{H}^\dagger)^*. \tag{7}$$

The next proposition shows that, if ϵ is chosen close to the smallest singular value of $\mathbf{H}$, and if $\kappa(\mathbf{H})$ is still large, then it will be reduced rapidly.

Proposition 3.3. *Suppose in* (7), *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 $\max\{2\sqrt{2}\delta, 2\delta^2 + 1/2\}$ is 4, i.e, when $\kappa(\mathbf{H}) > 4$, it will be reduced to its square root by (7).

In practice, one can execute the correction step (7) at most 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. Empirically, an $\mathbf{H}$ with $\kappa(\mathbf{H}) < 10$ is enough for practical use.

Corollary 3.1. *Let* $\mathbf{H}_{k+1} \leftarrow (1-\epsilon_k)\mathbf{H}_k + \epsilon_k(\mathbf{H}_k^\dagger)^*$ *with* $\mathbf{H}_0$ *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 Proposition 3.3 and Corollary 3.1 are given in the appendix. The iterative scheme above is essentially the quaternion version of the Newton method for computing polar decomposition [16, Sect. 8] but with more relaxed parameters.

We summarize the computation and analysis of pseudo-QR as follows. The pseudo code is given in Algorithm 1.

Proposition 3.4. *Let* $\mathbf{H}$ *be given by* (1) *and* (2) *and then the correction step* (7) *is excecuted three times. Then* $\mathcal{R}(\mathbf{H}_{new}) = \mathcal{R}(\mathbf{Y})$; *under the setting of Corollary 3.1, if* $\kappa(\mathbf{H}) < 10^8$, *then* $\kappa(\mathbf{H}_{new}) < 10$.

Algorithm 1 (pseudo-QR) quaternion pseudo-QR implementation

Require: Sketch matrix $\mathbf{Y} \in \mathbb{Q}^{m\times s}$.
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 $[U, \sim] = \mathtt{qr}(\mathbf{Y}_c, 0)$
3: Construct $\mathbf{H} = H_0 + H_1\mathbf{j}$ from U with $H_0 = U(1:m,:)$ and $H_1 = -\overline{U(m+1:2m,:)}$.
4: Execute the correction step (7) a few times (often 3 times). Use Algorithm 2 to solve $\chi_{\mathbf{H}^*\mathbf{H}} Z = (\mathbf{H}^*)_c$ to obtain $\mathbf{H}^\dagger$.

The remaining question is how to compute $(\mathbf{H}^\dagger)^*$. [16] computed (7) using QR, which is expensive in the quaternion setting. We convert it to solving linear equations in the complex arithmetic such that highly efficient solvers can be used. To this end, assume that $\mathbf{H}$ 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 [46]. For a general quaternion linear equation $\mathbf{A}\mathbf{X} = \mathbf{B}$ with $\mathbf{A} \in \mathbb{Q}^{m\times n}, \mathbf{X} \in \mathbb{Q}^{n\times s}, \mathbf{B} \in \mathbb{Q}^{m\times s}$, according to Proposition 2.1,

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

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 $\mathbf{X} := \hat{Z}_0 - \overline{\hat{Z}_1}\mathbf{j}$; 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 2 Quaternion linear 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 $\mathbf{X} \in \mathbb{C}^{n_2 \times l}$ satisfy $\mathbf{AX} = \mathbf{B}$
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 $\mathbf{X} = \mathbf{X}_c(1 : n_2, :) - \overline{\mathbf{X}_c(n_2 + 1 : 2n_2, :)}\mathbf{j}$

3.2 Pseudo-SVD

If the orthonormality of $\mathbf{H}$ is absolutely required, the relation 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.

Lemma 3.1 (c.f. [46]). *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 $\mathcal{J}\overline{v}$ is also an eigenvector asssociated to λ. Moreover, $v \perp \mathcal{J}\overline{v}$.*

The above lemma implies that in the ideal situation, if every (normalized) 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 so $\mathbf{H}$ 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 a QSVD of $\mathbf{Y}$ via computing the complex SVD of $\chi_{\mathbf{Y}}$. Let

$$\chi_{\mathbf{Y}} := USV^*, \quad U \in \mathbb{C}^{2m \times 2s},\ V \in \mathbb{C}^{2s \times 2s},\ S = \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma \end{bmatrix} \in \mathbb{R}^{2s \times 2s} \tag{8}$$

be a compact SVD of $\chi_{\mathbf{Y}}$, where Σ is diagonal. Note that due to duplicated singular values, USV^* may not take the structure as that in Proposition 2.2. Nevertheless, this may be constructed. Partition

$$U = \begin{bmatrix} U_{ul} & U_{ur} \\ U_{dl} & U_{dr} \end{bmatrix},\ V = \begin{bmatrix} V_{ul} & V_{ur} \\ V_{dl} & V_{dr} \end{bmatrix},\ \text{with } U_{ul} \in \mathbb{C}^{m \times s},\ V_{ul} \in \mathbb{C}^{s \times s}. \tag{9}$$

Denote

$$\mathbf{H} := U_{ul} - \overline{U_{dl}}\mathbf{j} \in \mathbb{Q}^{m \times s} \quad \text{and} \quad \mathbf{V} := V_{ul} - \overline{V_{dl}}\mathbf{j} \in \mathbb{Q}^{s \times s}. \tag{10}$$

Conditionally, $\mathbf{H}, \Sigma, \mathbf{V}^*$ given by the above approach is a compact QSVD of $\mathbf{Y}$:

Theorem 3.1. *Let $\mathbf{H}, \Sigma, \mathbf{V}$ be given by (8) and (10). If all the singular values of $\mathbf{Y}$ are distinct, i.e., Σ consists of distict singular values, then $\mathbf{H}\Sigma\mathbf{V}^*$ is a compact QSVD of $\mathbf{Y}$, i.e., $\mathbf{Y} = \mathbf{H}\Sigma\mathbf{V}^*$.*

We first present the following lemma.

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

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

Proof of Theorem 3.1. The assumption shows that Σ consists of distinct singular values. By Lemma 2.2, it suffices to prove that

$$\chi_{\mathbf{Y}} = \chi_{\mathbf{H}} S \chi_{\mathbf{V}}{}^* = [\mathbf{H}_c, \mathcal{J}\overline{\mathbf{H}_c}] \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} \mathbf{V}_c^* \\ (\mathcal{J}\overline{\mathbf{V}_c})^* \end{bmatrix} \tag{11}$$

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. The construction of $\mathbf{H}$ and $\mathbf{V}$ obeys $\chi_{\mathbf{Y}}\mathbf{V}_c = \mathbf{H}_c\Sigma$. Then,

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

and so $\chi_{\mathbf{Y}}[\mathbf{V}_c, \mathcal{J}\overline{\mathbf{V}_c}] = [\mathbf{H}_c, \mathcal{J}\overline{\mathbf{H}_c}]\mathrm{diag}(\Sigma, \Sigma)$, which together with the orthonormality of $[\mathbf{V}_c, \mathcal{J}\overline{\mathbf{V}_c}] \in \mathbb{C}^{2s\times 2s}$ yields (11). □

Remark 3.3. *We provide Theorem 3.1 and the proof as we cannot find one in the literature. The proof implies why the assumption is neccessary. Consider the example $\mathbf{Y} = I_2 \in \mathbb{Q}^{2\times 2}$, and so $\chi_{\mathbf{Y}} = I_4 \in \mathbb{C}^{4\times 4}$. Any orthonormal $U = [u_1, \ldots, u_4] \in \mathbb{C}^{4\times 4}$ is a singular vector matrix of $\chi_{\mathbf{Y}}$. If constructing $\mathbf{H}_c = [u_1, u_2]$, then although $u_1 \perp u_2, u_1 \perp \mathcal{J}\overline{u_1}$, as $\mathcal{J}\overline{u_1}$ may not perpendicular to u_2, $\mathbf{H}$ may not be orthonormal and so $\mathbf{H}$ is not a singular vector matrix of $\mathbf{Y} = I_2$.*

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 [38]; 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 (10) 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 these two cases, $\mathbf{H}$ given by (10) is no longer orthonormal and may not even span the correct range.

Fortunately, we find that the following correction step can tackle the above issue. When the above issue occurs, numerically computing an SVD of $\chi_{\mathbf{Y}}$ exhibits the following form:

$$\chi_{\mathbf{Y}} = U\tilde{S}V^*, \quad U \in \mathbb{C}^{2m\times 2s},\ V \in \mathbb{C}^{2s\times 2s},\ \tilde{S} = \begin{bmatrix} S_g & 0 \\ 0 & \Sigma_b \end{bmatrix} \in \mathbb{R}^{2s\times 2s}, \tag{12}$$
$$U^*U = I_{2s}, V^*V = I_{2s}, \Sigma_b \in \mathbb{R}^{2t\times 2t}(t < s), S_g = \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma \end{bmatrix} \in \mathbb{R}^{2(s-t)\times 2(s-t)};$$

i.e., now the singular values $\tilde{S}$ can be partitioned as the "good" part S_g and the "bad" part Σ_d. 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 may be disturbed, as well as the sigular values with multiplicity larger than 2. Besides, there still holds $\mathcal{R}(\chi_{\mathbf{Y}}) = \mathcal{R}(U)$; this is assured for example by using MATLAB's `svd` even with very ill-conditioned $\chi_{\mathbf{Y}}$.

Given $\chi_{\mathbf{Y}} = U\tilde{S}V^*$ as in (12), if still generating $\mathbf{H}$ by (10), then $\mathbf{H}^*\mathbf{H}$ will exhibit the form

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

where "×" is a matrix not equal to identity. Based on this observation, correcting $\mathbf{H}$ can be excecuted 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 (12) denote

$$\chi_{\mathbf{Y}} := Y_g + Y_b, \quad Y_g = U_gS_gV_g^*, \quad Y_b = U_b\Sigma_bV_b^*. \tag{13}$$

Further partition U_g as in (9) and generate $\mathbf{H}_g \in \mathbb{Q}^{m\times(s-t)}$ from U_g as in (10). Since $S_g = \mathrm{diag}(\Sigma, \Sigma)$, Lemma 3.2 ensures that

$$\mathbf{H}_g \text{ is orthonormal and } \mathcal{R}(\chi_{\mathbf{H}_g}) = \mathcal{R}(U_g). \tag{14}$$

For the bad part, we can still select a representation basis from U_b. Write $U_b = [u_1, \dots, u_{2t}] \in \mathbb{C}^{2m\times 2t}$. Then:

Proposition 3.5. *One can find* $U_{\tilde{b}} := [u_{i_1}, \dots, u_{i_t}] \subset U_b = [u_1, \dots, u_{2t}]$, *such that*

$$\operatorname{span}(U_{\tilde{b}}, \mathcal{J}\overline{U_{\tilde{b}}}) = \operatorname{span}(U_b) = 2t. \tag{15}$$

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

Proposition 3.6. *If* $\mathbf{Y}$ *has full column rank, then* $\mathcal{R}([\mathbf{H}_g, \mathbf{H}_{\tilde{b}}]) = \mathcal{R}(\mathbf{Y})$.

Proof. By Lemma 2.4 it suffices to prove $\mathcal{R}(\chi_{[\mathbf{H}_g,\mathbf{H}_{\tilde{b}}]}) = \mathcal{R}(\chi_{\mathbf{Y}})$. Note that $\mathcal{R}(\chi_{[\mathbf{H}_g,\mathbf{H}_{\tilde{b}}]}) = \mathcal{R}([\chi_{\mathbf{H}_g}, \chi_{\mathbf{H}_{\tilde{b}}}]) = \mathcal{R}(\chi_{\mathbf{H}_g}) \otimes \mathcal{R}(\chi_{\mathbf{H}_{\tilde{b}}})$, where the second relation follows from $\mathbf{H}_g \perp \mathbf{H}_{\tilde{b}}$; on the other hand, $\mathcal{R}(\chi_{\mathbf{Y}}) = \mathcal{R}(U) = \mathcal{R}(U_g) \otimes \mathcal{R}(U_b)$. (14), (15), and $\chi_{\mathbf{H}_{\tilde{b}}} = [U_{\tilde{b}}, \mathcal{J}\overline{U_{\tilde{b}}}]$ give the assertion. □

Denote $\mathbf{H}_{new} = [\mathbf{H}_g, \mathbf{H}_{\tilde{b}}]$, it remains to adjust $\mathbf{H}_{new}$ such that it is orthonormal. Owing to (14) and that $\mathbf{H}_g \perp \mathbf{H}_{\tilde{b}}$, we only need to orthonormalize $\mathbf{H}_{\tilde{b}}$, which in fact can be simultaneously done during the selection of $U_{\tilde{b}}$ using (modified) Gram-Schmidt orthogonalization [13].

However, in case that $\mathbf{Y}$ is too ill-conditioning ($\kappa(\mathbf{Y}) > 10^{13}$), the following process for finding $\mathbf{H}_{\tilde{b}}$ is more accurate and efficient. The idea still resorts to complex SVD. Denote $\mathbf{H}_b \in \mathbb{Q}^{m\times 2t}$ such that $\chi_{\mathbf{H}_b} = [U_b, \mathcal{J}\overline{U_b}]$ ($\mathbf{H}_b$ needs not be explicitly constructed). By Lemma 3.1 and the definition of U_b, $\chi_{\mathbf{H}_b}$ spans the same invariant subspace as U_b, i.e., $\mathcal{R}(\chi_{\mathbf{H}_b}) = \mathcal{R}(U_b)$. Additionally, let

$$\mathbf{H}_b \leftarrow \mathbf{H}_b \operatorname{diag}(f),\ f \in \mathbb{R}^{2t} \text{ with } f_j \sim \operatorname{Uniform}(0,1) + 1,\ j = 1, \dots, 2t. \tag{16}$$

This does not change the range of $\mathbf{H}_b$ and empirically, multiplying $\operatorname{diag}(f)$ avoids $\mathbf{H}_b$ to have duplicated singular values. In this case, applying complex SVD to $\chi_{\mathbf{H}_b} \in \mathbb{C}^{2m\times 4t}$ has the form:

$$\chi_{\mathbf{H}_b} = [U_{h1}, U_{h2}, U_{hb}] \operatorname{diag}(\Sigma_h, \Sigma_h, 0) V_h^*,\ \Sigma_h \in \mathbb{R}^{t\times t},\ U_{h1} \in \mathbb{C}^{2m\times t}, \tag{17}$$

where Σ_h are the positive distinct singular values of $\mathbf{H}_b$. Thus (17) reduces to the case of (12). Note that $\mathcal{R}([U_{h1}, U_{h2}]) = \mathcal{R}(\chi_{\mathbf{H}_b}) = \mathcal{R}(U_b)$. We thus 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 $\mathcal{R}(\chi_{\mathbf{H}_{\tilde{b}}}) = \operatorname{span}(U_b)$. Denote $\mathbf{H}_{new} = [\mathbf{H}_g, \mathbf{H}_{\tilde{b}}]$. Finally we summarize that

$$\mathcal{R}(\mathbf{H}_{new}) = \mathcal{R}(\mathbf{Y}) \text{ and } \mathbf{H}_{new} \text{ is orthonormal.}$$

The whole computation is summarized in Algorithm 3.

3.3 Comparsions

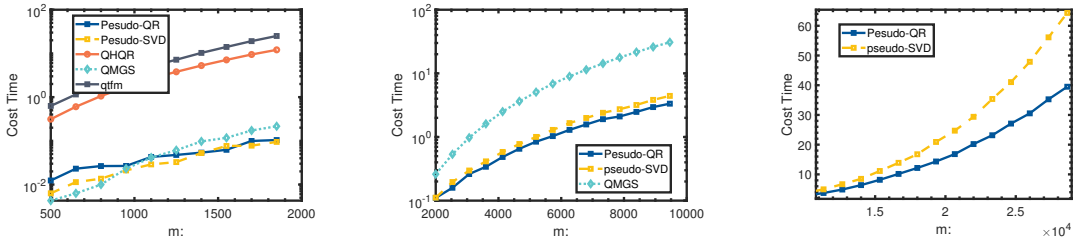
For a sketch $\mathbf{Y} \in \mathbb{Q}^{m\times s}$, the computational complexity of Pseudo-QR and Pseudo-SVD is $O(ms^2)$ (assuming $s \ll m$). In practice, these algorithms are suited to different criteria. Pseudo-QR performs well when the condition number $\kappa(\mathbf{Y}) < 10^8$. Conversely, 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 QTFM's `qr`, the structure-preserving quaternion Householder QR (QHQR) [19], and the structure-preserving modified Gram-Schmidt (QMGS) [41]. Fig. 2a illustrates that our rangefinders exhibit significantly lower computational costs compared to QTFM's `qr` and QHQR. The

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 $[U, \sim, \sim] = \texttt{svd}(\chi_{\mathbf{Y}}, 0)$ 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** $t \lesssim s$ **then**
4: Construct $\mathbf{H}_b$ from (16). Compute $[U_h, \sim, \sim] = \texttt{svd}(\chi_{\mathbf{H}_b}, 0)$ with $U_h = [U_{h1}, U_{h2}, U_{hb}]$ as (17), 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 $u_{i_1}, \ldots, u_{i_t}$ from U_b and simultaneously do orthogonalization such that $u_{i_j} \perp [u_{i_1}, \mathcal{J}\overline{u_{i_1}}, \ldots, u_{i_{j-1}}, \mathcal{J}\overline{u_{i_{j-1}}}]$, $j = 2, \ldots, t$. Let $U_{\tilde{b}} = [u_{i_1,\ldots,u_{i_t}}]$ 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: $\mathbf{H} = [\mathbf{H}_g, \mathbf{H}_{\tilde{b}}]$.



(a) All rangefinders (b) QMGS, our rangefinders (c) our rangefidners

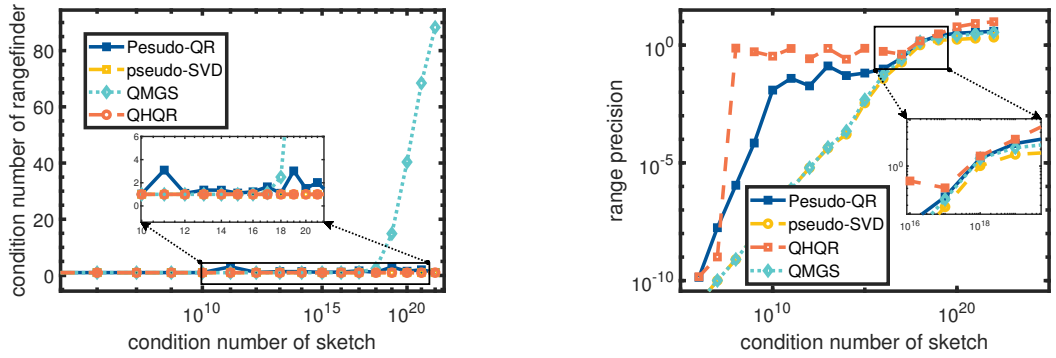
Figure 2: Running Time when $m = 20s$: x-axis is m and y-axis is the running time with log scale (the first two subfigures).

disparity in their time complexity can span two to three orders of magnitude, escalating with an increase in the row size of the sketches from 600 to 2000. As the size continues to grow, Fig. 2b demonstrates that our rangefinders outperform QMGS markedly, while QTFM's `qr` and QHQR become prohibitively time-consuming. Finally, Fig. 2c reveals that Pseudo-QR operates faster than Pseudo-SVD when m 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 QMGS outperform the competetors, while Pesudo-QR is still valuable when $\kappa(\mathbf{Y}) < 10^8$. QHQR performs the worst when $\kappa(\mathbf{Y})$ increases.

3.4 Remarks

Very recently, based on randomized subspace embeddings [29, 33], efficient randomized orthogonalization techniques have been proposed in the real domain [3, 2], which guarantee to generate a non-orthonormal yet well-conditioned matrix with high probability. The design of our non-orthonormal rangefinders is in the same spirit of [3, 2], while note that our computations are deterministic (if $\mathbf{Y}$ is given). It is feasible to combine Pseudo-QR, Pseudo-SVD, and any



(a) y-axis: condition number of the rangefinder

(b) y-axis: range Precision

Figure 3: $m = 1000$, $n = 200$, $\kappa(\mathbf{Y})$ from $1e6$ to $1e22$.

QB approximation with randomized embedding techniques to yield possibly more practical non-orthonormal rangefinders. However, a fast deterministic quaternion QB approximation still serves as a backbone.

4 One-Pass Algorithm

In this section, we consider the one-pass randomized algorithm proposed by Tropp et al. [38] while with a possibly non-orthonormal rangefinder (originally it only takes an 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 [38, 39].

Our theoretical result in Section 4.2 shows that the accuracy loss of the truncation approximation is proportional to the condition number of the rangefinder $\kappa(\mathbf{H})$. Probabilistic bounds with Gaussian and sub-Gaussian embeddings will be given in Sections 4.3 and 4.4.

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 $\mathbf{\Omega} \in \mathbb{Q}^{n\times s}$ and $\mathbf{\Psi} \in \mathbb{Q}^{l\times m}$ independently where the sketch size satisfy: $r \le s \le l \ll \min\{m, n\}$. The main information can be captured by two sketches:

$$\mathbf{Y} = \mathbf{A\Omega} \in \mathbb{Q}^{m\times s} \quad \text{and} \quad \mathbf{W} = \mathbf{\Psi A} \in \mathbb{Q}^{l\times n}, \tag{18}$$

where $\mathbf{Y}$ is used to generate the range $\mathbf{H} \in \mathbb{Q}^{m\times s}$ by using any range-preserving rangefinders such as pesudo-QR or pesudo-SVD. Then, a rank-s approximation of $\mathbf{A}$ is given by

$$\mathbf{A} \approx \mathbf{HX}, \quad \mathbf{X} = (\mathbf{\Psi H})^{\dagger}\,\mathbf{W} \in \mathbb{Q}^{s\times n}. \tag{19}$$

Finally, a truncated QSVD is applied to $\mathbf{X}$ to further obtain the final rank-r approximation. In the whole algorithm, only $\mathbf{Y}$, $\mathbf{W}$, and $\mathbf{\Psi}$ are required, which means that $\mathbf{A}$ is not exposed to the process. This is suitable for the scenorio that $\mathbf{A}$ is too large to be read in memory. The pseudocode is given in Algorithms 4 and 5. Note that when $\mathbf{\Psi}$ takes the identity matrix and $\mathbf{H}$ is orthonormal, the algorithm is exactly the randomized QSVD [25]. That is to say, the one-pass algorithm takes an additional sketch $\mathbf{W} = \mathbf{\Psi A}$ than randomized QSVD, thereby trading accuracy for efficiency as well as the usage in the limit storage scenorio.

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 $\mathbf{\Psi} \in \mathbb{Q}^{l\times m}$,
Ensure: Rank-s approximation of the form $\hat{\mathbf{A}} = \mathbf{HX} \in \mathbb{Q}^{m\times n}$ with $\mathbf{H} \in \mathbb{Q}^{m\times s}$ and $\mathbf{X} \in \mathbb{Q}^{s\times n}$
1: $\mathbf{H} \leftarrow \mathcal{F}(\mathbf{Y})$ ▷ $\mathcal{F}$ is a range-preserving rangefinder map and $\mathbf{H}$ is well-conditioned.
2: $\mathbf{X} \leftarrow (\mathbf{\Psi H}) \backslash \mathbf{W}$ ▷ Solve overdetermined quaternion linear equations using Algorithm 2.
3: return $(\mathbf{H}, \mathbf{X})$

Algorithm 5 Truncation Stage

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

Remark 4.1. *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 [38, 15, 25].*

4.2 Deterministic error

This subsection is concerned with the deterministic error without assuming the distribution of the test matrices at first. Probabilistic bounds will be detailed in the next two subsections. We do not assume a specific rangefinder $\mathbf{H}$, but the following basic requirements on $\mathbf{H}$ and the sketch size are made througout this subsection:

$$\mathcal{R}(\mathbf{H}) = \mathcal{R}(\mathbf{Y}), \text{ and } \mathbf{H} \text{ has full column rank;} \tag{20}$$

$$r \le s \le l \le \min\{m, n\}. \tag{21}$$

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

$$\mathbf{H} = \mathbf{QB}, \ \mathbf{Q} \in \mathbb{Q}^{m\times s}, \ \mathbf{Q}^*\mathbf{Q} = I_s. \tag{22}$$

Under (20), $\mathcal{R}(\mathbf{Q}) = \mathcal{R}(\mathbf{H}) = \mathcal{R}(\mathbf{Y})$ and $\mathbf{B} \in \mathbb{Q}^{s\times s}$ is invertible. Here (22) 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 $\mathbf{Q}_\perp\mathbf{Q}_\perp^* := I_m - \mathbf{QQ}^*$ such that $\mathcal{R}(\mathbf{Q}) \perp \mathcal{R}(\mathbf{Q}_\perp)$. Denote

$$\mathbf{\Psi}_1 = \mathbf{\Psi Q}_\perp \in \mathbb{Q}^{l\times(m-s)}, \quad \mathbf{\Psi}_2 = \mathbf{\Psi Q} \in \mathbb{Q}^{l\times s}. \tag{23}$$

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

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^* = [\mathbf{U}_1 \quad \mathbf{U}_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^* \\ \mathbf{V}_2^* \end{bmatrix}, \ \Sigma_1 \in \mathbb{R}^{r\times r}, \ \mathbf{U}_1 \in \mathbb{Q}^{m\times r}, \ \mathbf{V}_1 \in \mathbb{Q}^{n\times r}; \tag{24}$$

and denoote:

$$\mathbf{\Omega}_1 = \mathbf{V}_1^*\mathbf{\Omega} \in \mathbb{Q}^{r\times s}, \quad \mathbf{\Omega}_2 = \mathbf{V}_2^*\mathbf{\Omega} \in \mathbb{Q}^{(n-r)\times s}. \tag{25}$$

Theorem 4.1 (QB error)**.** *Let* $(\mathbf{H}, \mathbf{X})$ *be generated by Algorithm 4 and* (20) *holds. With notations in* (23) *and* (25)*, if* $\mathbf{\Omega}_1^*$ *and* $\mathbf{\Psi}_2$ *have full column rank, we have the following Frobenius error estimation,*

$$\|\mathbf{A} - \mathbf{H}\mathbf{X}\|_F^2 = \|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|_F^2 + \|\mathbf{\Psi}_2^\dagger \mathbf{\Psi}_1 \left(\mathbf{Q}_\perp^* \mathbf{A}\right)\|_F^2 \tag{26}$$

$$\leq \left(1 + \|\mathbf{\Psi}_2^\dagger \mathbf{\Psi}_1\|_2^2\right)\left(1 + \|\mathbf{\Omega}_2 \mathbf{\Omega}_1^\dagger\|_2^2\right)\|\Sigma_2\|_F^2. \tag{27}$$

(27) is a rough estimation, whose purpose is to demonstrate that the QB error is independent of $\kappa(\mathbf{H})$. More refined probabilistic bounds will be given in Sections 4.3 and 4.4 based on (26).

Recall that $\mathbf{X} = (\mathbf{\Psi}\mathbf{H})^\dagger \mathbf{\Psi}\mathbf{A}$ in Algorithm 4. Denote $\lfloor \mathbf{X} \rfloor_r$ as the best rank-r approximation of $\mathbf{X}$. Analygous to the real/complex counterpart, $\lfloor \mathbf{X} \rfloor_r$ is also given by the rank-r turncated 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* (20) *hold, and* $\lfloor \mathbf{X} \rfloor_r$ *be truncated by Algorithm 5. Then*

$$\|\mathbf{H}\lfloor \mathbf{X} \rfloor_r - \mathbf{H}\mathbf{X}\|_F \leq \kappa(\mathbf{H})\left(\|\mathbf{A} - \mathbf{H}\mathbf{X}\|_F + \|\Sigma_2\|_F\right). \tag{28}$$

To make the proofs of the theorems clear, we devide them 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* $\mathbf{B} = \mathbf{A}^\dagger$.

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, [14] proved that $(EF)^\dagger = F^\dagger E^\dagger$ holds if and only if:

$$E^\dagger EFF^*E^*EFF^\dagger = FF^*E^*E. \tag{29}$$

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. □

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* (22)*, and* $\mathbf{\Psi}$ *is an arbitrary* $l \times m$ $(s \leq l \leq m)$ *quaternion matrix with full row rank. Then we have:*

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

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

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

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

The following lemma is a trivial quaternion version of [38, Lemma A.4].

Lemma 4.3. *Assume that $\mathbf{\Psi}_2$ has full column rank; then*

$$(\mathbf{\Psi Q})^{\dagger}\,\mathbf{\Psi A}-\mathbf{Q}^{*}\mathbf{A}=\mathbf{\Psi}_2^{\dagger}\mathbf{\Psi}_1\,(\mathbf{Q}_{\perp}^{*}\mathbf{A})\,.$$

The follow lemma comes from [25, Section 4.3]; see also [15, Theorem 9.1].

Lemma 4.4. *With the notations in Algorithm 4 and in (25), and $\mathbf{Q}$ is in (22), if $\mathbf{\Omega}_1$ has full row rank, then*

$$\|\mathbf{A}-\mathbf{QQ}^{*}\mathbf{A}\|_F^2=\|\mathbf{A}-\mathbf{YY}^{\dagger}\mathbf{A}\|_F^2\le\|\Sigma_2\|_a^2+\|\Sigma_2\mathbf{\Omega}_2\mathbf{\Omega}_1^{\dagger}\|_F^2 \tag{31}$$

$$\le\left(1+\|\mathbf{\Omega}_2\mathbf{\Omega}_1^{\dagger}\|_2^2\right)\|\Sigma_2\|_F^2. \tag{32}$$

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

$$\|\mathbf{A}-\mathbf{HX}\|_F^2=\|\mathbf{A}-\mathbf{H}\,(\mathbf{\Psi H})^{\dagger}\,\mathbf{\Psi A}\|_F^2=\|\mathbf{A}-\mathbf{Q}(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}\|_F^2. \tag{33}$$

Thus, it suffices to evaluate the right part of (33). By Pythagorean identity,

$$\begin{aligned}\|\mathbf{A}-\mathbf{Q}(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}\|_F^2&=\|\mathbf{A}-\mathbf{QQ}^{*}\mathbf{A}+\mathbf{QQ}^{*}\mathbf{A}-\mathbf{Q}(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}\|_F^2\\&=\|\mathbf{A}-\mathbf{QQ}^{*}\mathbf{A}\|_F^2+\|\mathbf{Q}(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}-\mathbf{QQ}^{*}\mathbf{A}\|_F^2\\&=\|\mathbf{A}-\mathbf{QQ}^{*}\mathbf{A}\|_F^2+\|\mathbf{\Psi}_2^{\dagger}\mathbf{\Psi}_1\,(\mathbf{Q}_{\perp}^{*}\mathbf{A})\,\|_F^2,\end{aligned} \tag{34}$$

where the last equality uses Lemma 4.3. Note that $\|\mathbf{Q}_{\perp}^{*}\mathbf{A}\|_F=\|\mathbf{Q}_{\perp}\mathbf{Q}_{\perp}^{*}\mathbf{A}\|_F=\|\mathbf{A}-\mathbf{QQ}^{*}\mathbf{A}\|_F$; thus $\|\mathbf{\Psi}_2^{\dagger}\mathbf{\Psi}_1\,(\mathbf{Q}_{\perp}^{*}\mathbf{A})\,\|_F\le\|\mathbf{\Psi}_2^{\dagger}\mathbf{\Psi}_1\|_2\|\mathbf{A}-\mathbf{QQ}^{*}\mathbf{A}\|_F$, which together with (34) and Lemma 4.4 yields (27). □

Proof of Theorem 4.2. Let $\mathbf{H}=\mathbf{QB}$ as (22) where $\mathbf{Q}$ is partially orthonormal and $\mathbf{B}$ is invertible. Then $\mathbf{X}=(\mathbf{\Psi H})^{\dagger}\,\mathbf{\Psi A}=(\mathbf{\Psi QB})^{\dagger}\,\mathbf{\Psi A}=\mathbf{B}^{-1}\,(\mathbf{\Psi Q})^{\dagger}\,\mathbf{\Psi A}$, where the last equality follows from the proof of Lemma 4.2 and that a random fat matrix $\mathbf{\Psi}\in\mathbb{Q}^{l\times m}$ drawn from continuous distribution has full row rank genericly. Denote $\mathbf{A}_{in}:=\mathbf{Q}(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}$. We have:

$$\begin{aligned}\|\mathbf{HX}-\mathbf{H}\lfloor\mathbf{X}\rfloor_r\|_F&\le\|\mathbf{H}\|_2\|\mathbf{X}-\lfloor\mathbf{X}\rfloor_r\|_F\\&=\|\mathbf{H}\|_2\|\mathbf{B}^{-1}(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}-\lfloor\mathbf{B}^{-1}(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}\rfloor_r\|_F\\&\le\|\mathbf{H}\|_2\|\mathbf{B}^{-1}(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}-\mathbf{B}^{-1}\lfloor(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}\rfloor_r\|_F\\&\le\|\mathbf{H}\|_2\|\mathbf{B}^{-1}\|_2\cdot\|(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}-\lfloor(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}\rfloor_r\|_F\\&\overset{([38,\ (6.3)],\ \mathbf{Q}\lfloor(\mathbf{\Psi Q})^{\dagger}\mathbf{\Psi A}\rfloor_r=\lfloor\mathbf{A}_{in}\rfloor_r)}{=}\kappa(\mathbf{H})\|\mathbf{A}_{in}-\lfloor\mathbf{A}_{in}\rfloor_r\|_F\ \le\kappa(\mathbf{H})\|\mathbf{A}_{in}-\lfloor\mathbf{A}\rfloor_r\|_F\\&\le\kappa(\mathbf{H})(\|\mathbf{A}_{in}-\mathbf{A}\|_F+\|\mathbf{A}-\lfloor\mathbf{A}\rfloor_r\|_F)\\&=\kappa(\mathbf{H})\,(\|\mathbf{A}-\mathbf{HX}\|_F+\|\Sigma_2\|_F)\,,\end{aligned}$$

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

□

4.3 Probabilistic error: Guassian embedding

We further quantify the probabilistic estimation of the QB error in Theorem 4.1 with Guassian test matrices. The fixed-rank error can be derived accordingly. Using the statistical properties of quaternion Gaussian matrices established in [25], the estimation can be proved using a similar deduction as in [38]. We first recall some results from [25].

Definition 4.1. *(c.f. [25]) Let* $\mathbf{A} \in \mathbb{Q}^{m\times n}$ *and write* $\mathbf{A} = A_w + A_x\mathbf{i} + A_y\mathbf{j} + A_z\mathbf{k}$. $\mathbf{A}$ *is Gaussian if all entries of* $A_k \in \mathbb{R}^{m\times n}, k \in \{w, x, y, z\}$ *are i.i.d standard Gaussian random variables.*

we call a quaternion matrix $\mathbf{A}$ standard Gaussian if it satisfies Definition 4.1.

Lemma 4.5. *([25]) Let* $\mathbf{G} \in \mathbb{Q}^{m\times n}$ *be standard Guassian and* $\mathbf{S} \in \mathbb{Q}^{l\times m}$, $\mathbf{T} \in \mathbb{Q}^{n\times s}$ *be fixed. Then* $\mathbb{E}\|\mathbf{SGT}\|_F^2 = 4\|\mathbf{S}\|_F^2\|\mathbf{T}\|_F^2$. *Furthermore if* $m \le n$, *then* $\mathbb{E}\|\mathbf{G}^\dagger\|_F^2 = \frac{m}{4(n-m)+2}$.

In what follows, we denote $f(n, m) := \frac{4n}{4(m-n)+2}$.

Theorem 4.3 (Probabilistic QB error)**.** *For* $\mathbf{A} \in \mathbb{Q}^{m\times n}(m \ge n)$, *let* $(\mathbf{H}, \mathbf{X})$ *be generated by Algorithm 4 with (20) and (21) hold. If* $\mathbf{\Psi}$ *and* $\mathbf{\Omega}$ *in (18) are standard Gaussian, then*

$$\mathbb{E}\|\mathbf{HX} - \mathbf{A}\|_F^2 \le (1 + f(s, l))(1 + f(r, s))\|\Sigma_2\|_F^2 = \left(\frac{2l+1}{2(l-s)+1}\right)\left(\frac{2s+1}{2(s-r)+1}\right)\|\Sigma_2\|_F^2. \tag{35}$$

Proof. Write $\mathbf{H} = \mathbf{QB}$ as in (22) with $\mathbf{Q}$ orthonormal and $\mathbf{B}$ invertible. From (26) of Theorem 4.1, it suffices to respectively estimate $\|\mathbf{\Psi}_2^\dagger\mathbf{\Psi}_1(\mathbf{Q}_\perp^*\mathbf{A})\|_F^2$ and $\|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|_F^2$. Owing to the marginal property of the standard normal distribution, $\mathbf{\Psi}_2 = \mathbf{\Psi Q} \in \mathbb{Q}^{l\times s}$ and $\mathbf{\Psi}_1 = \mathbf{\Psi Q}_\perp$ are statistically independent standard Guassian matrices [25]. Note that $\mathbf{\Psi}_2^* \in \mathbb{Q}^{s\times l}(s \le l)$ is still standard Gaussian and it holds that $\mathbf{\Psi}_2^\dagger = ((\mathbf{\Psi}_2^\dagger)^*)^* = ((\mathbf{\Psi}_2^*)^\dagger)^*$; thus by Lemma 4.5, $\mathbb{E}\|\mathbf{\Psi}_2^\dagger\|_F^2 = \mathbb{E}\|(\mathbf{\Psi}_2^*)^\dagger\|_F^2 = s/(4(l-s)+2) = f(s, l)/4$. Therefore,

$$\begin{aligned}
\mathbb{E}_{\mathbf{\Psi}}\|\mathbf{\Psi}_2^\dagger\mathbf{\Psi}_1(\mathbf{Q}_\perp^*\mathbf{A})\|_F^2 &= \mathbb{E}_{\mathbf{\Psi}_2}\mathbb{E}_{\mathbf{\Psi}_1}\|\mathbf{\Psi}_2^\dagger\mathbf{\Psi}_1(\mathbf{Q}_\perp^*\mathbf{A})\|_F^2 \\
&\overset{\text{(Lemma 4.5)}}{=} 4\mathbb{E}\|\mathbf{\Psi}_2^\dagger\|_F^2\|\mathbf{Q}_\perp^*\mathbf{A}\|_F^2 \\
&= f(s, l)\|\mathbf{Q}_\perp^*\mathbf{A}\|_F^2 = f(s, l)\|(I - \mathbf{QQ}^*)\mathbf{A}\|_F^2.
\end{aligned} \tag{36}$$

It follows from (26) and the independence of $\mathbf{\Omega}$ and $\mathbf{\Psi}$ that

$$\mathbb{E}\|\mathbf{HX} - \mathbf{A}\|_F^2 = (1 + f(s, l))\,\mathbb{E}_{\mathbf{\Omega}}\|(I - \mathbf{QQ}^*)\mathbf{A}\|_F^2. \tag{37}$$

(32) of Lemma 4.4 shows that

$$\begin{aligned}
\mathbb{E}_{\mathbf{\Omega}}\|(I - \mathbf{QQ}^*)\mathbf{A}\|_F^2 &\le \|\Sigma_2\|_F^2 + \mathbb{E}_{\mathbf{\Omega}}\|\Sigma_2\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F^2 \\
&\le (1 + f(r, s))\,\|\Sigma_2\|_F^2,
\end{aligned} \tag{38}$$

where the deduction of the second inequality is similar to that of (36). Plugging this into (37) yields (35). □

Theorem 4.4 (Probabilistic fixed-rank error)**.** *Under the setting of Theorem 4.3, let* $\hat{\mathbf{A}} = \mathbf{H}\lfloor\mathbf{X}\rfloor_r$ *be generated by Algorithm 5. Then*

$$\mathbb{E}\|\hat{\mathbf{A}} - \mathbf{A}\|_F \le \left((1 + \kappa(\mathbf{H}))\sqrt{(1 + f(s, l)\cdot(1 + f(r, s)))} + \kappa(\mathbf{H})\right)\|\Sigma_2\|_F.$$

Proof. This is obtained by $\|\hat{\mathbf{A}} - \mathbf{A}\|_F \le \|\mathbf{A} - \mathbf{HX}\|_F + \|\mathbf{HX} - \mathbf{H}\lfloor\mathbf{X}\rfloor_r\|_F$ together with Theorems 4.2, 4.3, and Jensen inequality. □

The above two probabilistic errors generalize [38, Theorem 4.3, Corollary 6.4] to the quaternion and non-orthonormal rangefinders settings.

4.4 Probabilistic error: sub-Guassian embedding

A random variable ξ is called sub-Gaussian if it satisfies [40] $\mathbb{P}\{|\xi| > t\} \leq \exp\left(1 - ct^2\right)$ for all $t > 0$ and some constant $c > 0$. The sub-Gaussian norm of X is defined as: $\|\xi\|_{\psi_2} = \sup_{p>=1} p^{-1/2} \left(\mathbb{E}|\xi|^p\right)^{1/p}$. Common sub-Gaussian variables include Gaussian, sparse Gaussian, Radmacher, and bounded random variables. Using sub-Gaussian such as sparse Gaussian or Radmacher gives more flexibility or improves the speed of generating random test matrices and sketches, We define sub-Gaussian quaternion matrices as follows:

Definition 4.2 (Entry-independent sub-Gaussian quaternion matrix)**.** *Let $\mathbf{A} \in \mathbb{Q}^{m\times n}$ and write $\mathbf{A} = A_w + A_x\mathbf{i} + A_y\mathbf{j} + A_z\mathbf{k}$. If all entries of the real matrix $\mathbf{A}_r := \left[A_w^T, A_x^T, A_y^T, A_z^T\right]^T \in \mathbb{R}^{4m\times n}$ are independent sub-Gaussian random variables, we call $\mathbf{A}$ a quaternion sub-Gaussian matrix. If every entry of $\mathbf{A}_r$ is centered and has unit variance, we say that $\mathbf{A}$ is a centered and unit variance quaternion sub-Gaussian matrix.*

We establish the probabilistic QB error with sub-Gaussian test matrices, while the deduction for fixed-rank error is ommited as it is similar to Theorem 4.4.

Theorem 4.5 (Probabilistic QB error)**.** *Assume that the sketch parameters satisfy $r < s < l < min\{m, n\}$. Draw sub-Gaussian quaternion test matrices $\mathbf{\Omega} \in \mathbb{Q}^{n\times s}$ and $\mathbf{\Psi} \in \mathbb{Q}^{l\times m}$ according to Definition 4.2 and assume that every entry of $\mathbf{\Omega}_r$ and $\mathbf{\Psi}_r$ has the same sub-Gaussian norm K. Let $\mathbf{H}, \mathbf{X}$ be generated by Algorithm 4; then*

$$\|\mathbf{HX} - \mathbf{A}\|_F \leq \left(1 + \frac{K\sqrt{l}\sqrt{2\left(\log(\min\{m-s, n\}) + \log(4l) + 1\right)/c_1}}{\sqrt{l} - C_K\sqrt{s} - t}\right) \cdot \left(1 + \frac{K\sqrt{s}\sqrt{2\left(\log(min\{m, n\} - r) + \log(4s) + 1\right)/c_1}}{\sqrt{s} - C_K\sqrt{r} - t}\right)\|\Sigma_2\|_F \tag{39}$$

with probability at least $1 - 2\exp\left(-\frac{2c_2t^2}{K^4}\right) - \frac{1}{\min\{m-s,n\}} - \frac{1}{\min\{m,n\}-r}$, where $c_1 > 0, c_2 > 0$ are absolulte constants and $C_K > 0$ only depends on K.

Comparing with the Gaussian bound (35), an extra factor of order $O\left(\log(\min\{m, n\})\right)$ appears in (39). The main reason is that for quaternion sub-Gaussian, we do not have a sharp estimation of $\|\mathbf{SGT}\|_F$ as that in Lemma 4.5. The main tool for proving Theorem 4.5 is the following result, which generalizes the deviation bound of [40] for extream singular values of a sub-Gaussian real matrix to the quaternion realm.

Theorem 4.6 (Deviation bound)**.** *Let $\mathbf{A} \in \mathbb{Q}^{N\times n}(N > 4n)$. If each row $\mathbf{A}_i$ of $\mathbf{A}$ are independent quaternion sub-Gaussian isotropic vectors, then for every $t > 0$, with probability at least $1 - \exp\left(-ct^2/K^4\right)$ one has*

$$2\sqrt{N} - 2C_K\sqrt{n} - 2t \leq \sigma_{\min}(\mathbf{A}) \leq \sigma_{\max}(\mathbf{A}) \leq 2\sqrt{N} + 2C_K\sqrt{n} + 2t. \tag{40}$$

Here $c > 0$ is an absolulte constant and $C_K > 0$ only depends on the max sub-Gaussian norm $K = \max_i \|\mathbf{A}_i\|_{\psi_2}$ of the rows.

Theorem 4.6 relies on the following definitions; see [40, Def. 5.19] for their real counterparts.

Definition 4.3 (Quaternion sub-Gaussian and isotropic vectors)**.** *A quaternion random vector $\mathbf{v}$ in $\mathbb{Q}^n$ is called a sub-Gaussian vector if the one-demensional marginals $\langle\mathbf{v}_r, x\rangle$ are sub-Gaussian random variables for all $x \in \mathbb{R}^{4n}$, where $\mathbf{v}_r = [v_w^T, v_x^T, v_y^T, v_z^T]^T \in \mathbb{R}^{4n}$. The sub-Gaussian norm of $\mathbf{v}$ is defined as:*

$$\|\mathbf{v}\|_{\psi_2} := \|\mathbf{v}_r\|_{\psi_2} = \sup_{x^Tx=1, x\in\mathbb{R}^{4n}} \|\langle\mathbf{v}_r, x\rangle\|_{\psi_2}.$$

$\mathbf{v}$ is called isotropic if $\mathbb{E}\mathbf{v}_r\mathbf{v}_r^T = I_{4n}$.

The following proposition is also crucial for proving Theorem 4.5.

Proposition 4.1. *Let $\mathbf{A} \in \mathbb{Q}^{N\times m}$, $\mathbf{V} \in \mathbb{Q}^{m\times n}$ $(m \le n)$ be row-orthonormal, and $\mathbf{\Omega} \in \mathbb{Q}^{n\times s}$ be a quaternion centered sub-Gaussian matrix (Def. 4.2) with entries of $\mathbf{\Omega}_r$ all having the same sub-Gaussian norm K. Then*

$$\|\mathbf{AV\Omega}\|_F \le 2\sqrt{st}\|\mathbf{A}\|_F \tag{41}$$

with probability at least $1 - 4e \cdot \exp\left(\log(s\min\{N,m\}) - ct^2/K^2\right)$, where $c > 0$ is an absolulte constant.

The detailed proofs in this subsetion are left to the appendix, and we give a sketch of the proof of Theorem 4.5 here: observing (26), we first derive $\|\mathbf{\Psi}_2^\dagger\mathbf{\Psi}_1\left(\mathbf{Q}_\perp^*\mathbf{A}\right)\|_F \le \|\mathbf{\Psi}_2^\dagger\|_2\|\mathbf{\Psi Q}_\perp\left(\mathbf{Q}_\perp^*\mathbf{A}\right)\|_F$; then respectively bound $\|\mathbf{\Psi}_2^\dagger\|_2$ by Theorem 4.6 and $\|\mathbf{\Psi Q}_\perp\left(\mathbf{Q}_\perp^*\mathbf{A}\right)\|_F \le O(\sqrt{l\log(\min\{m,n\})})\|\mathbf{Q}_\perp^*\mathbf{A}\|_F$ by Proposition 4.1 (observe that $\mathbf{\Psi}$ is sub-Gaussian, $\mathbf{Q}_\perp$ is orthonormal, and $\mathbf{Q}_\perp^*\mathbf{A}$ is fixed). Next, use (31) to obtain $\|\mathbf{Q}_\perp^*\mathbf{A}\|_F^2 = \|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|_F^2 \le \|\Sigma_2\|_F^2 + \|\Sigma_2\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F^2$; the second term $\|\Sigma_2\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F \le \|\Sigma_2\mathbf{V}_2^*\mathbf{\Omega}\|_F\|\mathbf{\Omega}_1^\dagger\|_2$; respectively bound $\|\mathbf{\Omega}_1^\dagger\|_2$ by Theorem 4.6 and $\|\Sigma_2\mathbf{V}_2^*\mathbf{\Omega}\|_F \le O(\sqrt{s\log(\min\{m,n\})})\|\Sigma_2\|_F$ by Proposition 4.1. Finally, combining the above pieces and using union bound or total probability, one obtains (39).

5 Numerical Experiments

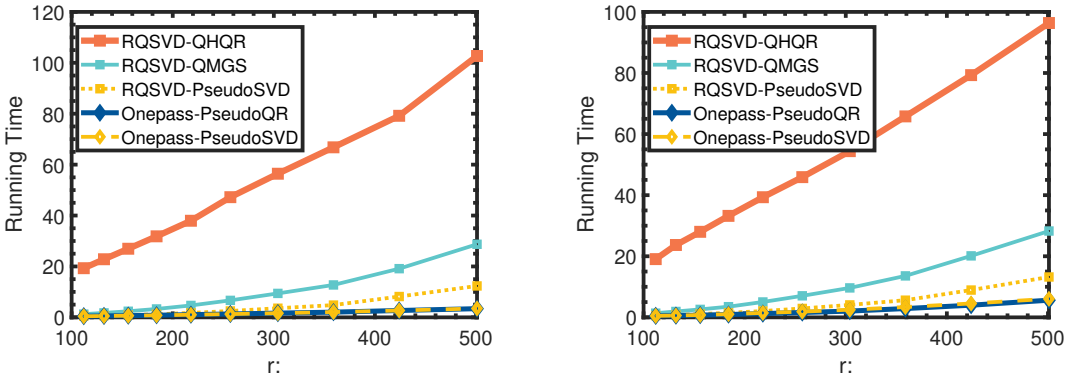
In this section, we tested our quaternion one-pass algorithm with the devised rangefinders on several examples. We used randomized quaternion SVD (RQSVD) [25, Algorithm 3.1] as a baseline to evaluate the performance of our algorithm for synthetic data in example 5.1. We then handled large-scale data processing tasks of scientific simulation output compression and color image processing using our algorithm. All the experiments were carried out in MATLAB 2023b on a personal computer with an Intel(R) CPU i7-12700 of 2.10 GHz and 64GB of RAM and work in double-precesion arithmetic. The QTFM [35] was employed for basic quaternion operations. This section only presents limited selection of results and more detail results are conatined in supplementary materials.

5.1 Synthetic data

Example 5.1. *In this example, we constructed $A = U\Sigma V^* \in \mathbb{Q}^{2000\times 1600}$ where U, V are partial unitary and Σ were generated as follows:*

1. *Low rank plus noise:* $\Sigma = \mathrm{diag}(1,\ldots,1,0,\ldots,0) + \xi n^{-1}\mathbf{E} \in \mathbb{Q}^{1600\times 1600}$. $\mathbf{E}$ *is a quaternion standard Gaussian matrix;*
2. *Polynomial decay spectrum (pds):* $\Sigma = \mathrm{diag}\left(1,\ldots,1,2^{-p},3^{-p},\ldots,(n-R+1)^{-p}\right) \in \mathbb{Q}^{1600\times 1600}$;
3. *Exponential decay spectrum (eds):* $\Sigma = \mathrm{diag}\left(1,\ldots,1,10^{-q},10^{-2q},\ldots,10^{-(n-R)q}\right) \in \mathbb{Q}^{1600\times 1600}$.

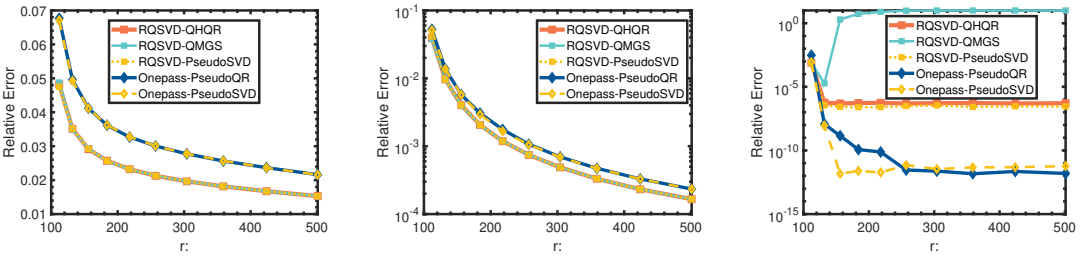
We adjusted the parameter ξ, p, q to obtain different decay rate. For every synthetic data case, we evaluated the performance of RQSVD and one-pass algorithm with different rangefinders: 1) RQSVD with QHQR, 2) RQSVD with QMGS, 3) RQSVD with our pseudo-SVD, 4) one-pass with pseudo-QR, and 5) one-pass with pseudo-SVD. The deterministic QSVD used in RQSVD, as suggested in [25, Remark 3.2], were from [24, 41], whose code is also provided in Jia's



(a) low-rank plus noise and pds (b) eds

Figure 4: $\xi = 0.01, p = 2, q = 0.25$. Time cost of RQSVD (QHQR, QMGS, Pseudo-SVD) and algorithm 5 (pseudo-QR, pseudo-SVD) of synthetic data. x-axis is the target rank; y-axis is the running time in seconds.

A



(a) Low rank plus noise (b) pds (c) eds

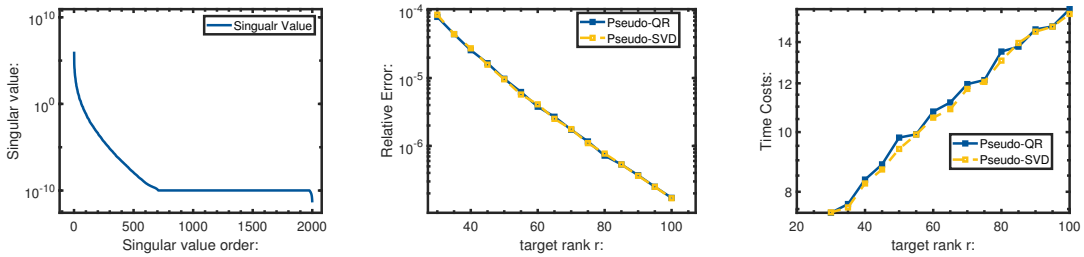
Figure 5: Relative Errors of RQSVD (QHQR, QMGS, Pseudo-SVD) and algorithm 5 (pseudo-QR, pseudo-SVD) of synthetic data. x-label is target rank, y-label is relative $\|A - \hat{A}\|_F / \|A\|_F$.

homepage; that used in one-pass were implemented by ourselves[3]*. We tested different target rank* r *varying from* 100 *to* 500*. For one-pass algorithm we set* $s = r + 5$ *and* $l = 2s$*; the oversampling parameter of RQSVD was set to* 5*, the same as one-pass algorithm. The relative error is defined as* $\|\mathbf{A} - \hat{\mathbf{A}}\|_F / \|\mathbf{A}\|_F$.

Fig. 4 shows that RQSVD with QHQR is the slowest one, followed by that with QMGS; RQSVD with Pseudo-SVD is about two times faster than that with QMGS. This shows that even Pseudo-SVD is useful for accelerating RQSVD. One-Pass with Pseudo-QR and Pseudo-SVD are the most fastest ones, which are about 3.5x faster than RQSVD with Pseudo-SVD and are about 8x faster than RQSVD with QMGS. This is also benifitial from the efficiency of our deterministic QSVD function `ModifiedcsvdQ77`.

In terms of the relative error, Fig. 5 show that one-pass algorithm is worse than RQSVD. This is understandable, as commented in the pagragraph below (19). We also remark that their performance discrepancy diminishes when the spectral decays more rapidly. When the original data is extremely ill-conditioned, as in Fig. 5c, some needs to be interpreted particularly. We

[3]Roughly speaking, our QSVD is based on computing the SVD of the full complex representation; however, as pointed out in Sect. 3.2, we need to correct the approach such that it still works in the scenorio that there exist duplicated singular values and the matrix is very ill-conditioned. The idea is similar to pseudo-SVD but the detail is more involved, which may be reported in an independent manuscript. The code is with the name `ModifiedcsvdQ77` in our github webpage.



(a) Singular values (b) Relative Frobenius Error (c) Running Time

Figure 6: Relative Errors and Running Time of CFD simulation with different target rank k

first observe that RQSVD with QMGS is inaccuracy when r is near 120; this is due to that QMGS's orthonormality is more sensitive to the condition number of the sketch, as illustrated in Fig. 3a. At the same time, RQSVD with QHQR holds the orthogonality while preserving less accurate range (Fig. 3b), and Pseudo-SVD takes both aspects into account, therefore, RQSVD with Pseudo-SVD is slightly better than that with QHQR. On the other hand, it is interesting to see that one-pass with both Pseudo-QR and Pseudo-SVD perform better than the others; in fact, this is due to that in this very ill-conditioned scenorio, our deterministic QSVD function `ModifiedcsvdQ77` is more accurate than that used in RQSVD [25].

5.2 Scientific data

Example 5.2. *In this example, we applied the one-pass algorithm with pseudo-QR and pseudo-SVD to compress the output of a computational fluid dynamics (CFD) simulation. We have obtained a numerical simulation on a finite elements model of the unsteady 3D Navier-Stokes equations for microscopic natural convection about biological research applications using the QuickerSim CFD Toolbox for MATLAB. There are* 20914 *nodes to characteristic the velocity, each node having three directions and can be represented as a pure quaternion. Each element represents the space velocity field by:*

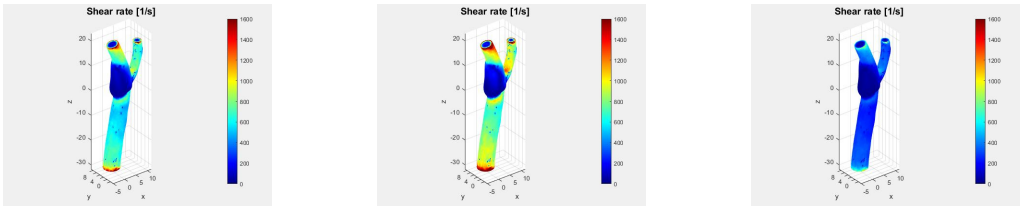
$$a_{p,t} = v_x(p,t)\mathbf{i} + v_y(p,t)\mathbf{j} + v_z(p,t)\mathbf{k}, \tag{42}$$

where $v = (v_x, v_y, v_z)$ *is a velocity vector field in the model. The velocity field is time-dependent and so we collect it at* 20000 *time instants, resulting into a pure quaternion matrix* microConvection *of size* 20914×20000. *Due to the periodicity, the data matrix* microConvection *generated by this type of models has rapidly decay spectrum, as shown in figure 6a.*

Fig. 6 show that the algorithm work efficiently and obtain highly accurate approximation for this example. In particular, if we set $k = 50$, the relative error is under 10^{-5} and is acceptable to be used in PDE numerical solving, while the time costs of the whole proceedure, including sketching and randomized approximation, are less than 10 seconds.

Fig. 7 graphically illustrates that the compression can highly approximate the origin velocity field: the figures in the second row are the shear rates computed from velocity field ($k = 50$), which closely match the real data, as shown in the first row. In this case, our algorithm compressed the original data from 5.22GB to 47.8MB, whose compression ratio is 99.11%.

Example 5.3. *We tested compressing the output of a 4D Lorenz-type chaotic system simulation.*



(a) Origin at 8th step (b) Origin at 14th step (c) Origin at 23rd step

(d) Recovered at 8th step (e) Recovered at 14th step (f) Recovered at 23rd step

Figure 7: Shear rate computed from origin data and compressed data ($k = 50$)

The dynamical system is shown as follows:

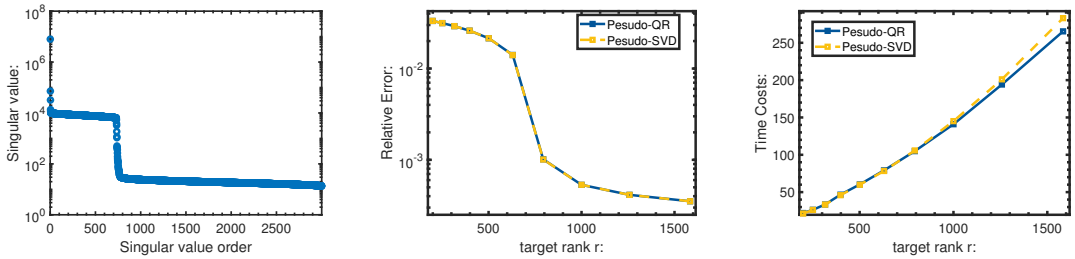
$$\begin{cases} \frac{dx}{dt} = a(y - x), \\ \frac{dy}{dt} = cx - y - xz + w, \\ \frac{dz}{dt} = -bz + xy, \\ \frac{dw}{dt} = (c - 1)y + w - \frac{x^3}{b}, \end{cases}$$

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 $a = 15$, $b = 2$, $c = 28$ and select 10000 initial states (x, y, z, w) randomly selected from sphere $\|(x, y, z, w)\|_2 = 20$. By using `ode45` to compute the system at 20000 time instances, we obtained a 20000×10000 quaternion matrix which records the information of solutions. The distinction from the previous CFD simulation example is that, this 4D Lorenz-type system is a hyperchaotic one, and so the quaternion matrix has more flatten spectrum decay, as shown in figure 8a. Specifically, in this hyperchaotic system, the presence of attractors causes a significant drop in singular values from the 700th to the 800th.

Figure 8a indicates a larger sketch size is necessary to obtain a high accuracy approximation. One-pass algorithm with pseudo-QR and pseudo-SVD have similar performance in accuracy. When the target rank $r > 1000$, the relative error is less than 10^{-3}. Our algorithm compresses the original data form 5.74GB to 998MB and the compression ratio reaches to 85% ($r = 1000$). We can observe that pseduo-QR is faster when the target rank is larger.

5.3 Image compression

Example 5.4. *We compressed a large-scale color image of high-resolution map, which has 31365×27125 pixels and encompasses a diverse range of environments, including clusters of buildings, factories, fields, and bodies of water. Considering the limitation of memory capabilities of local machines, streaming data processing can significantly reduce the consumption of time and space. We have performed linear updates on the sketches similar to [38, Algorithm 2], thus avoiding revisiting the original data and additional transport flops. Through linear updating,*



(a) Singular values (b) Relative Frobenius (c) Running Time

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

r	rangefinder	Relative Error	PSNR	Cost Time
2000	pseudo-QR	0.2613	18.0186	372.62
	pseudo-SVD	0.2614	18.0149	400.71
2900	pseudo-QR	0.2252	19.3095	764.04
	pseudo-SVD	0.2253	19.3066	871.75
3900	pseudo-QR	0.1966	20.4905	1433.66
	pseudo-SVD	0.1966	20.4885	1675.73

Table 1: Experiment of image compression, where $r = 2000, 2900, 3900$, $s = r + 50$, $l = 2s$.

we have obtained two sketches whose sizes are $31365 \times s$ *and* $l \times 27125$ *respectively. One-pass algorithm with pseudoQR or pseudo-SVD was then used to generate a low-rank approximation.*

Table 1 indicates that, as target rank varies from 2000 to 3900, the relative error of the low-rank approximation gradually decreases, and the PSNR (Peak Signal-to-Noise Ratio) and running time progressively increase. Comparing between pseudo-QR and pseudo-SVD, they have similar performance with the one-pass algorithm. However, in most cases, the time consumed by pseudo-QR is less than that by pseudo-SVD. In image compression, pseudo-QR may be a more suitable rangefinder.

Fig. 9 more intuitively reflects the approximation quality of the algorithm. We select two representative area including structures and fields in the original image; see Fig. 9a and 9b. The former contains more architectural details, while the latter is more open. We can see that after approximating with target rank 3900, the main elements of the map are still distinguishable, and the color channels have been well preserved when compression ratio reaches to 71.4%.

6 Conclusions

Existing quaternion rangefinders 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. [38] 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 for both quaternion Gaussian and sub-Gaussian test matrices; in particular, it was demonstrated that the error is proportional to the rangefinder's condition number. The efficiency and effectiveness of the algorithm were verified on large-scale experiments

(a) Original structures in Guangxi Univ.

(b) Original fields

(c) Comressed structures in Guangxi Univ.

(d) Compressed fields

Figure 9: Top two images are two representative areas selected from the orginal iamge *XixiangtangMap* which has 31365 × 27125 pixels. Down two images are corresponding parts of the approximation with target rank $r = 3900$.

such as scientifc data compression and color image compression, which consist of quaternion matrices of dimension > 20000.

Our experiments also demonstrate that the devised rangefinder pseudo-SVD can accelerate randomized QSVD [25]. Other quaternion randomized algorithms [34, 23, 26, 45] may also benefit from pseduo-SVD. Besides, it is also helpful to study how to incorporate a non-orthonormal rangefinder such as pseudo-QR into these algorithms, and it is interesting to devise more practical rangefinders, as that mentioned in Sect. 3.4.

References

[1] 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

[2] O. Balabanov. Randomized Cholesky QR factorizations. *arXiv:2210.09953*, 2022. 13

[3] Oleg Balabanov and Laura Grigori. Randomized Gram–Schmidt process with application to GMRES. *SIAM J. Sci. Comput.*, 44(3):A1450–A1474, 2022. 13

[4] C. Boutsidis, D. P. Woodruff, and P. 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. Association for Computing Machinery, 2016. 1

[5] A. Bunse-Gerstner, R. Byers, and V. Mehrmann. A quaternion QR algorithm. *Numer. Math.*, 55(1):83–95, 1989. 2, 11

[6] J. Chen and M. K. Ng. Color image inpainting via robust pure quaternion matrix completion: Error bound and weighted loss. *SIAM J. Imag. Sci.*, 15(3):1469–1498, 2022. 5

[7] Y. Chen, Z.-G. Jia, Y. Peng, Y.-X. Peng, and D. Zhang. A new structure-preserving quaternion QR decomposition method for color image blind watermarking. *Signal Processing*, 185:108088, 2021. 2

[8] K. L. Clarkson and D. 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. ACM, 2009. 1

[9] M. B. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pages 163—172, New York, NY, USA, 2015. Association for Computing Machinery. 1

[10] Intel Corporation. Intel math kernel library, 2024. 2

[11] T. A. Ell, N. Le Bihan, and S. J. Sangwine. Quaternion Fourier Transforms. In *Quaternion Fourier Transforms for Signal and Image Processing*, pages 35–66. John Wiley & Sons, Ltd, 2014. 2

[12] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, 2004. 1

[13] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, fourth edition edition, 2013. 2, 12

[14] T. N. E. Greville. Note on the generalized inverse of a matrix product. *SIAM Rev.*, 8(4):518–521, 1966. 16

[15] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011. 1, 2, 3, 15, 17

[16] Nicholas J Higham. *Functions of matrices: theory and computation*. SIAM, 2008. 8, 9

[17] Z. Jia, M. K. Ng, and G.-J. Song. Robust quaternion matrix completion with applications to image inpainting. *Numer. Linear Algebra Appl.*, 26(4):e2245, 2019. 5

[18] Z. Jia, M. Wei, and S. Ling. A new structure-preserving method for quaternion Hermitian eigenvalue problems. *J. Comput. Appl. Math.*, 239:12–24, 2013. 2

[19] Z. Jia, M. Wei, M.-X. Zhao, and Y. Chen. A new real structure-preserving quaternion QR algorithm. *J. Comput. Appl. Math.*, 343:26–48, 2018. 2, 12

[20] R. Kannan and S. Vempala. Randomized algorithms in numerical linear algebra. *Acta Numer.*, 26:95–135, 2017. 1

[21] A. Kireeva and J. A. Tropp. Randomized matrix computations: Themes and variations. *arXiv:2402.17873*, 2024. 1

[22] N. Le Bihan and J. Mars. Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing. *Signal Process.*, 84(7):1177–1199, 2004. 10, 16

[23] C. Li, Y. Liu, F. Wu, and M. Che. Randomized block Krylov subspace algorithms for low-rank quaternion matrix approximations. *Numer. Alg.*, 96:687–717, 2023. 2, 4, 26

[24] Y. Li, M. Wei, F. Zhang, and J. Zhao. Real structure-preserving algorithms of Householder based transformations for quaternion matrices. *J. Comput. Appl. Math.*, 305:82–91, 2016. 2, 20

[25] Q. Liu, S. Ling, and Z. Jia. Randomized quaternion singular value decomposition for low-rank matrix approximation. *SIAM J. Sci. Comput.*, 44(2):A870–A900, 2022. 2, 4, 14, 15, 17, 18, 20, 22, 26, 30, 32

[26] Y. Liu, F. Wu, M. Che, and C. Li. Fixed-precision randomized quaternion singular value decomposition algorithm for low-rank quaternion matrix approximations. *Neurocomput.*, 580:127490, 2024. 2, 4, 26

[27] C. Lyu, J. Pan, M. K. Ng, and X. Zhao. Randomized low rank approximation for nonnegative pure quaternion matrices. *Applied Math. Lett.*, 150:108940, 2024. 2

[28] M. W. Mahoney. Randomized algorithms for matrices and data. *Found. Trends Mach. Learn.*, 3(2):123–224, 2011. 1

[29] P.-G. Martinsson and J. A. Tropp. Randomized numerical linear algebra: foundations and algorithms. *Acta Numer.*, 29:403–572, 2020. 1, 13

[30] J. Miao, K. I. Kou, D. Cheng, and W. Liu. Quaternion higher-order singular value decomposition and its applications in color image processing. *Inf. Fusion*, 92:139–153, 2023. 2

[31] T. Minemoto, T. Isokawa, H. Nishimura, and N. Matsui. Feed forward neural network with random quaternionic neurons. *Signal Process.*, 136:59–68, 2017. 2

[32] R. Murray, J. Demmel, M. W. Mahoney, N. B. Erichson, M. Melnichenko, O. A. Malik, L. Grigori, P. Luszczek, M. Dereziński, M. E. Lopes, T. Liang, H. Luo, and J. Dongarra. Randomized numerical linear algebra: A perspective on the field with an eye to software. *arXiv:2302.11474*, 2023. 1

[33] Y. Nakatsukasa and J. A. Tropp. Fast & accurate randomized algorithms for linear systems and eigenvalue problems. *arXiv:2111.00113*, 2022. 13

[34] H. Ren, R.-R. Ma, Q. Liu, and Z.-J. Bai. Randomized quaternion QLP decomposition for low-rank approximation. *J. Sci. Comput.*, 92(3):80, 2022. 2, 4, 26

[35] S. J. Sangwine and N. Le Bihan. Quaternion toolbox for MATLAB, http://qtfm.sourceforge.net. 2, 20

[36] C. Soo, J.-H. Chang, and J.-J. 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, 2003. 2

[37] J. A. Tropp and R. J. Webber. Randomized algorithms for low-rank matrix approximation: Design, analysis, and applications. *arXiv:2306.12418*, 2023. 1

[38] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Practical sketching algorithms for low-rank matrix approximation. *SIAM J. Matrix Anal. & Appl.*, 38(4):1454–1485, 2017. 1, 2, 4, 6, 11, 14, 15, 16, 17, 18, 23, 24

[39] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Streaming low-rank matrix approximation with an application to scientific simulation. *SIAM J. Sci. Comput.*, 41(4):A2430–A2463, 2019. 1, 2, 14

[40] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In G. Kutyniok and Y. C. Eldar, editors, *Compressed Sensing: Theory and Applications*, pages 210–268. Cambridge University Press, 2012. 4, 19, 31, 33

[41] M. Wei, Y. Li, F. Zhang, and J. Zhao. *Quaternion Matrix Computations*. Nova Science Publishers, Incorporated, 2018. 2, 5, 6, 12, 20

[42] N. A. Wiegmann. Some theorems on matrices with real quaternion elements. *Canadian J. Math.*, 7:191–201, 1955. 29

[43] D. P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):1–157, 2014. 1

[44] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert. A fast randomized algorithm for the approximation of matrices. *Applied Computat. Harmon. Anal.*, 25(3):335–366, 2008. 1

[45] R. Xu and Y. Wei. Randomized quaternion matrix UTV decomposition and its application in quaternion matrix optimization. *Pac. J. Optim., to appear*, 2024. 2, 4, 26

[46] F. Zhang. Quaternions and matrices of quaternions. *Linear Algebra Appl.*, 251:21–57, 1997. 4, 5, 6, 9, 10, 30

[47] H. Zhang and H. Lv. Augmented quaternion extreme learning machine. *IEEE Access*, 7:90842–90850, 2019. 2

A Proofs in the Pseudo-QR subsection

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

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

Then (7) shows that $\mathbf{H}_{new}$ has singular values $f(\sigma_i) = (1-\epsilon)\sigma_i + \epsilon {\sigma_i}^{-1}$, $i = 1, \ldots, s$. 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 $f(x) \le \sigma_1$ 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 $\epsilon \ge \sigma_s$ that $f(\sigma_s) \ge (1-\epsilon)\sigma_s + \sigma_s\sigma_s^{-1} \ge 1 = f(1)$, i.e., f is upper bounded by $f(\sigma_s) = (1-\epsilon)\,\sigma_s + \epsilon/\sigma_s$ on $[\sigma_s, 1]$. Furthermore, as $\sigma_s \le 1$ and $\epsilon \le \delta\sigma_s$,

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

The analysis above together with $\sigma_1 \le \sqrt{2}$ in Proposition 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 $x > 0$, $f(x) \ge 2\sqrt{\epsilon(1-\epsilon)}$. The range of ϵ implies $2\sqrt{\epsilon(1-\epsilon)} \ge 2\sqrt{\sigma_s(1-\sigma_s)}$. Thus $f(x) \ge 2\sqrt{\sigma_s(1-\sigma_s)}$.

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

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

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

then

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

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

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

then

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

where the second inequality follows from $\sigma_1 \ge 1$ and $\sigma_s \le 1/(2\delta)$, 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.

□

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

Proof of Proposition 3.5. The idea of the proof essentially follows from [42, Theorem 1]. We first use the induction method to show that $\mathrm{span}(U_{\tilde{b}}, \mathcal{J}\overline{U_{\tilde{b}}}) = 2t$. For $t = 1$, this claim is true as $u_1 \perp \mathcal{J}\overline{u_1}$. Suppose now that we have found $U_k := [u_{i_1}, \ldots, u_{i_k}] \in U_b$ $(k < t)$ such that $\mathrm{span}(U_k, \mathcal{J}\overline{U_k})$ has dimension $2k$. Note that U_b is partially orthonormal and $\mathrm{span}(U_b)$ has dimension $2t > 2k$, and so there always exists at least a $u_{i_{k+1}} \in U_b \setminus U_k$, such that

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

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

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

deducing a contradiction.

Next, we will prove that $\mathrm{span}(U_k, \mathcal{J}\overline{U_k}, u_{i_{k+1}}, \mathcal{J}\overline{u_{i_{k+1}}})$ 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 $\mathrm{span}(M)$ and $\mathrm{span}(M)^\perp$, respectively. Then $u_{i_{k+1}}$ can be divided into two part:

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

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

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

with the second equality from Lemma 2.3.

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

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

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

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

also has dimension $2(k+1)$. Thus the induction method shows that there exists $U_{\tilde{b}} = [u_{i_1}, \ldots, u_{i_t}] \in U_b$, such that $\mathrm{span}(U_{\tilde{b}}, \mathcal{J}U_{\tilde{b}})$ 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 $\chi_{\mathbf{Y}}$, $\mathcal{J}\overline{u}$ is also corresponding to σ. which means that $\mathcal{R}(\mathcal{J}\overline{U_{\tilde{b}}}) \subset \mathcal{R}(U_b)$. Recall that $\mathcal{R}(U_b)$ also has dimension $2t$; thus $\mathcal{R}(U_{\tilde{b}}, \mathcal{J}\overline{U_{\tilde{b}}}) = \mathcal{R}(U_b)$. □

B Proofs in the sub-Gaussian subsection

B.1 Real representation

The analysis relies on the real representation of a quaternion matrix. Similar to the complex representation, a quaternion matrix $\mathbf{Q} = Q_w + Q_x\mathbf{i} + Q_y\mathbf{j} + Q_z\mathbf{k} \in \mathbb{Q}^{m\times n}$ has a (full) real representation [46, 25]:

$$\Upsilon_{\mathbf{Q}} := \begin{bmatrix} Q_w & -Q_x & -Q_y & -Q_z \\ Q_x & Q_w & -Q_z & Q_y \\ Q_y & Q_z & Q_w & -Q_x \\ Q_z & -Q_y & Q_x & Q_w \end{bmatrix} \in \mathbb{R}^{4m\times 4n}.$$

Similar to the complex representation $\chi_{\mathbf{Q}}$, $\Upsilon_{\mathbf{Q}}$ also keeps several basic properties of $\mathbf{Q}$, such as

$$\mathbf{C} = \mathbf{AB} \Leftrightarrow \Upsilon_{\mathbf{C}} = \Upsilon_{\mathbf{A}}\Upsilon_{\mathbf{B}};$$

$\mathbf{Q}$ is row/column-orthonormal if and only if $\Upsilon_{\mathbf{Q}}$ is row/column-orthonormal.

For convenience, we use $\mathbf{Q}_r \in \mathbb{R}^{4m\times n}$ to denote the first column block of $\Upsilon_{\mathbf{Q}}$ as the compact real representation of $\mathbf{Q}$, i.e.,

$$\mathbf{Q}_r := \begin{bmatrix} Q_w \\ Q_x \\ Q_y \\ Q_z \end{bmatrix} \in \mathbb{R}^{4m\times n}.$$

Spectral and Frobenius norms of a quaternion matrix $\mathbf{Q}$ can be represented by the real representations as below:

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

B.2 Technical lemmas

Lemma B.1. *([40, Lemma 5.24]) Let $x_1, \dots, x_N$ be real independent centered sub-Gaussian random variables. Then $x = [x_1, \dots, x_N]^T$ is a centered sub-Gaussian random vector in $\mathbb{R}^n$, and*

$$\|x\|_{\psi_2} \leq C \max_{1 \leq i \leq N} \|x_i\|_{\psi_2},$$

where C is an absolute constant.

Proposition B.1. *(Hoeffding-type inequality, [40, Proposition 5.10]) Let $x_1, \dots, x_N$ be real independent centered sub-Gaussian random variables, and let $K = \max_i \|x_i\|_{\psi_2}$. Then for every $a = [a_1, \dots, a_N]^T \in \mathbb{R}^N$ and every $t \geq 0$, we have*

$$\mathbb{P}\left\{\left|\sum_{i=1}^N a_i x_i\right| \geq t\right\} \leq e \cdot \exp\left(-\frac{ct^2}{K^2\|a\|^2}\right),$$

where $c > 0$ is an absolute constant.

Definition B.1. *(sub-exponential random variable, [40]) A real 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:

$$\|x\|_{\psi_1} = \sup_{p \geq 1} p^{-1} (\mathbb{E}|x|^p)^{1/p}.$$

Lemma B.2. *([40, Corollary 5.17]) Let $x_1, \dots, x_N$ be real independent centered sub-exponential random real variables, and let $K = \max_i \|x_i\|_{\psi_1}$. Then, for every $\varepsilon \geq 0$, we have:*

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

where $c > 0$ is an absolulte constant.

Lemma B.3. *([40, Lemma 5.36]) Consider a real matrix M that satisfies*

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

for some $\delta > 0$. Then

$$1 - \delta \leq s_{\min}(M) \leq s_{\max}(M) \leq 1 + \delta. \tag{49}$$

Conversely, if M satisfies (49) for some $\delta > 0$ then $\|M^T M - I\| \leq 3\max(\delta, \delta^2)$.

Lemma B.4. *Let x, y be two real 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 B.5. *Let x, y be positive random variables; we have:*

$$\mathbb{P}\{xy > \varepsilon^2\} \leq \mathbb{P}\{x > \varepsilon\} + \mathbb{P}\{y > \varepsilon\}$$

Proof. This is by setting $x_1 = \log(x)$, $y_1 = \log(y)$ and using Lemma B.4. □

B.3 Proof of Proposition 4.1

Proof of Proposition 4.1. Let $\Upsilon_{\mathbf{A}} \in \mathbb{R}^{4N\times 4m}$ be the real representation of $\mathbf{A}$. Let the full SVD of $\Upsilon_{\mathbf{A}}$ be $\Upsilon_{\mathbf{A}} = U_{\Upsilon_{\mathbf{A}}}\Sigma V_{\Upsilon_{\mathbf{A}}}^T$ with $U_{\Upsilon_{\mathbf{A}}} \in \mathbb{R}^{4N\times 4N}, \Sigma \in \mathbb{R}^{4N\times 4m}, V_{\Upsilon_{\mathbf{A}}} \in \mathbb{R}^{4m\times 4m}$. Denote $\hat{V} = V_{\Upsilon_{\mathbf{A}}}^T \Upsilon_{\mathbf{V}}$. Since $\mathbf{V}$ is row-orthonormal, so is $\Upsilon_{\mathbf{V}}$; then $\hat{V}$ is also row-orthonormal. We have:

$$\|\mathbf{AV\Omega}\|_F^2 = \|\Upsilon_{\mathbf{A}}\Upsilon_{\mathbf{V}}\mathbf{\Omega}_r\|_F^2 = \|U_{\Upsilon_{\mathbf{A}}}\Sigma V_{\Upsilon_{\mathbf{A}}}^*\Upsilon_{\mathbf{V}}\mathbf{\Omega}_r\|_F^2 = \|\Sigma\hat{V}\mathbf{\Omega}_r\|_F^2. \tag{50}$$

Write

$$\|\Sigma\hat{V}\mathbf{\Omega}_r\|_F^2 = \sum_{1\le i\le \min\{4N,4m\}, 1\le j\le s} \Sigma_{i,i}^2 \left(\hat{V}\mathbf{\Omega}_r\right)_{i,j}^2 (I_s)_{j,j}^2, \tag{51}$$

where I_s is the identity matrix of size $s\times s$. Then

$$\|\Sigma\hat{V}\mathbf{\Omega}_r\|_F^2 \le \|\Sigma\|_F^2 \max_{i,j} |(\hat{V}\mathbf{\Omega}_r)_{i,j}|^2 \|I\|_F^2 = s\|\Sigma\|_F^2 \max_{i,j} |(\hat{V}\mathbf{\Omega}_r)_{i,j}|^2. \tag{52}$$

For any fixed i, j,

$$\left|(\hat{V}\mathbf{\Omega}_r)_{i,j}\right| = \left|\sum_k \hat{V}_{i,k}(\mathbf{\Omega}_r)_{k,j}\right|; \tag{53}$$

since each $(\mathbf{\Omega}_r)_{k,j}$ is an independent centered sub-gaussian random variable with the same sub-Gaussian norm K, by Hoeffding-type inequality (Proposition B.1),

$$\mathbb{P}\left\{\left|\sum_k \hat{V}_{i,k}(\mathbf{\Omega}_r)_{k,j}\right| > t\right\} \le e\cdot\exp\left(-\frac{ct^2}{K^2\sum_k \hat{V}_{i,k}^2}\right) = e\cdot\exp\left(-\frac{ct^2}{K^2}\right), \tag{54}$$

where the equality is because $\hat{V}$ is real partially orthonormal. Then we union bound for all $i = 1,\dots,\min\{4N,4m\}$, $j = 1,\dots,s$:

$$\mathbb{P}\left\{\max_{i,j}\left|\sum_k \hat{V}_{i,k}(\mathbf{\Omega}_r)_{k,j}\right| > t\right\} \le 4e\cdot s\min\{N,m\}\exp\left(-ct^2/K^2\right) \tag{55}$$

$$= 4e\cdot\exp\left(\log(s\min\{N,m\}) - ct^2/K^2\right). \tag{56}$$

Combining inequalities above and the fact that $4\|\mathbf{A}\|_F^2 = \|\Upsilon_{\mathbf{A}}\|_F^2 = \|\Sigma\|_F^2$, the following inequality holds:

$$\mathbb{P}\left\{\|\mathbf{AV\Omega}\|_F^2 > 4st^2\|\mathbf{A}\|_F^2\right\} = \mathbb{P}\left\{\max_{i,j}\left|\sum_k \hat{V}_{i,k}(\mathbf{\Omega}_r)_{k,j}\right|^2 > t^2\right\} \tag{57}$$

$$\le 4e\cdot\exp\left(\log(s\min\{N,m\}) - ct^2/K^2\right), \tag{58}$$

i.e.,

$$\mathbb{P}\left\{\|\mathbf{AV\Omega}\|_F > 2\sqrt{st}\|\mathbf{A}\|_F\right\} \le 4e\cdot\exp\left(\log(s\min\{N,m\}) - ct^2/K^2\right). \tag{59}$$

□

B.4 Proof of Theorem 4.6

Proof of Theorem 4.6. First, from [25], $\sigma_{\min}(\Upsilon_{\mathbf{A}}) = \sigma_{\min}(\mathbf{A})$ and $\sigma_{\max}(\Upsilon_{\mathbf{A}}) = \sigma_{\max}(\mathbf{A})$. Thus we focus on the estimation of $\sigma_{\min}(\Upsilon_{\mathbf{A}})$ and $\sigma_{\max}(\Upsilon_{\mathbf{A}})$. The conclusion is to prove that

$$1 - C_K\sqrt{\frac{n}{N}} - \frac{t}{\sqrt{N}} \le \sigma_{\min}\left(\frac{\Upsilon_{\mathbf{A}}}{\sqrt{4N}}\right) \le \sigma_{\max}\left(\frac{\Upsilon_{\mathbf{A}}}{\sqrt{4N}}\right) \le 1 + C_K\sqrt{\frac{n}{N}} + \frac{t}{\sqrt{N}};$$

applying Lemma B.3 to $M := \Upsilon_{\mathbf{A}}/\sqrt{4N}$, it suffices to prove that

$$\|\frac{1}{4N}\Upsilon_{\mathbf{A}}^T\Upsilon_{\mathbf{A}} - I\|_2 \le \max(\delta,\delta^2) =: \varepsilon \quad \text{where} \quad \delta = C\sqrt{\frac{n}{N}} + \frac{t}{\sqrt{N}}. \tag{60}$$

We can evaluate the spectral norm $\|\frac{1}{4N}\Upsilon_{\mathbf{A}}^T\Upsilon_{\mathbf{A}} - I\|_2$ on a $\frac{1}{4}$-net $\mathcal{N}$ of the unit sphere $S^{4n-1} = \{x \in \mathbb{R}^{4n} \mid x^T x = 1\}$:

$$\left\|\frac{1}{4N}\Upsilon_{\mathbf{A}}^T\Upsilon_{\mathbf{A}} - I\right\|_2 \leq 2\max_{x\in\mathcal{N}}\left|\left\langle(\frac{1}{4N}\Upsilon_{\mathbf{A}}^T\Upsilon_{\mathbf{A}} - I)x, x\right\rangle\right| = 2\max_{x\in\mathcal{N}}\left|\frac{1}{4N}\|\Upsilon_A x\|_2^2 - 1\right|. \tag{61}$$

Write $\Upsilon_{\mathbf{A}}$ as a block matrix $\Upsilon_{\mathbf{A}} = \begin{bmatrix} A_w & -A_x & -A_y & -A_z \\ A_x & A_w & -A_z & A_y \\ A_y & A_z & A_w & -A_x \\ A_z & -A_y & A_x & A_w \end{bmatrix} =: \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix}$ with $B_i \in \mathbb{R}^{N\times 4n}$, $i = 1, 2, 3, 4$. Since each row of $\mathbf{A}$ is independent and isotropic (Def. 4.2), each row of B_i is an independent sub-Gaussian isotropic real vector.

Fix any real vector $x \in S^{4n-1}$, we will upper bound $P\left\{\left|\frac{1}{4N}\|B_i x\|_2 - \frac{1}{4}\right| > \frac{\varepsilon}{2}\right\}$ for each fix i. The idea comes from the proof of [40, Thm. 5.39]. First denote

$$\|B_i x\|^2 = \sum_{j=1}^{N}((B_i)_j x)^2 =: \sum_{j=1}^{N} Z_j^2,$$

where $(B_i)_j$ represents the j-th row of B_i. As $(B_i)_j$, $j = 1, \ldots, N$ are independent rows, $Z_j = (B_i)_j x$ are independent sub-Gaussian random variables; one can compute $\mathbb{E}Z_j^2 = 1$ and

$$\|Z_j\|_{\psi_2} = \|(B_i)_j x\|_{\psi_2} \leq \sup_{y\in S^{4n-1}}\|(B_i)_j y\|_{\psi_2} = \|(B_i)_j\|_{\psi_2} \leq \max_{1\leq k\leq N}\|\mathbf{A}_k\|_{\psi_2} = K,$$

where the second inequality follows from the definition of the sub-Gaussian norm of the quaternion vector $\mathbf{A}_k$ in Def. 4.3. Thus $Z_j^2 - 1$ are independent centered sub-exponential random variables. Using Lemma B.2 to give:

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

where c_1 is an absolute constant and the last two equalities come from the definition of δ in (60). By Lemma B.4,

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

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 $C \geq K^2\sqrt{\frac{1}{c_1}\left(\ln 9 + \frac{\ln 8}{n}\right)}$.

Using (61), we have:

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

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 $\delta = C\sqrt{\frac{n}{N}} + \frac{t}{\sqrt{N}}$ in (60). As noted at the beginning of the proof, this completes the proof of the theorem. □

Remark B.1. *The right-hand side of inequality (40) 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* $\mathbf{A}^*$.

B.5 Proof of Theorem 4.5

The following lemma is useful for proving Theorem 4.5.

Lemma B.6. *Let $\mathbf{\Psi} \in \mathbb{Q}^{l\times m}$ be a quaternion centered sub-Gaussian matrix (Def. 4.2) with entries of $\mathbf{\Psi}_r$ all having the same sub-Gaussian norm K; $\mathbf{Q} \in \mathbb{Q}^{m\times s}$ is column-orthnormal. Then $\mathbf{\Phi} = \mathbf{\Psi Q}$ has independent rows, each of which is quaternion sub-Gaussian isotropic and has sub-Gaussian norm CK for some constant $C > 0$ (c.f. Def. 4.3).*

Proof. Let $\mathbf{\Psi}_i, \mathbf{\Phi}_i$ respectively be the i-th row of $\mathbf{\Psi}$ and $\mathbf{\Phi}$; then $\mathbf{\Phi}_i = \mathbf{\Psi}_i\mathbf{Q}$, $i = 1, \ldots, l$ are independent due to the independence of entries of $\mathbf{\Psi}$.

For notational convenience, we next use row vectors $\mathbf{v} \in \mathbb{Q}^{1\times s}, \mathbf{w} \in \mathbb{Q}^{1\times m}$ to respectively represent $\mathbf{\Phi}_i$ and $\mathbf{\Psi}_i$, i.e., $\mathbf{v} = \mathbf{wQ}$. We still denote $\mathbf{v}_r = [v_w, v_x, v_y, v_z] \in \mathbb{R}^{1\times 4s}$; via real representaion, $\mathbf{v} = \mathbf{wQ}$ is equivalent to $\mathbf{v}_r = \mathbf{w}_r\Upsilon_{\mathbf{Q}}$. We show that $\mathbf{v}$ is a quaternion sub-Gaussian and isotropic vector. For any real column vector $x \in \mathbb{R}^{4s}$, let $y = \Upsilon_{\mathbf{Q}}x \in \mathbb{R}^{4m}$. Then

$$\mathbf{v}_r x = \mathbf{w}_r\Upsilon_{\mathbf{Q}}x = \mathbf{w}_r y.$$

Since $\mathbf{w}_r$ is a real row vector of independent sub-Gaussian entries with zero mean, $\mathbf{w}_r y$ is a linear combination of the entries of $\mathbf{w}_r$, and so $\mathbf{w}_r y$, namely, $\mathbf{v}_r x$, is still a sub-Gaussian random variable. To show that $\mathbf{v}$ is isotropc, again by Def. 4.3, we compute

$$\mathbb{E}_{\mathbf{v}}\left(\mathbf{v}_r^T\mathbf{v}_r\right) = \mathbb{E}_{\mathbf{w}}\left(\Upsilon_{\mathbf{Q}}^T\mathbf{w}_r^T\mathbf{w}_r\Upsilon_{\mathbf{Q}}\right) = \Upsilon_{\mathbf{Q}}^T\mathbb{E}_{\mathbf{w}}\left(\mathbf{w}_r^T\mathbf{w}_r\right)\Upsilon_{\mathbf{Q}}. \tag{62}$$

By assumption of $\mathbf{\Psi}$, every entries of $\mathbf{w}_r$ have zero mean and unit variance and are independent. Thus

$$\mathbb{E}_{\mathbf{w}}\left(\mathbf{w}_r^T\mathbf{w}_r\right) = I_{4m} \quad \text{and} \quad \mathbb{E}_{\mathbf{v}}\left(\mathbf{v}_r^T\mathbf{v}_r\right) = \Upsilon_{\mathbf{Q}}^T\Upsilon_{\mathbf{Q}} = I_{4s}, \tag{63}$$

where the last equality is because $\Upsilon_{\mathbf{Q}}^{4m\times 4s}$ is also column-orthonormal (by the assumption on $\mathbf{Q}$). Thus $\mathbf{v}$ is isotropc.

Last, we estiamte the sub-Gaussian norm of the row $\mathbf{v}$: $\|\mathbf{v}\|_{\psi_2}$. We have

$$\begin{aligned}\|\mathbf{v}\|_{\psi_2} = \|\mathbf{v}_r\|_{\psi_2} &= \sup_{x^Tx=1, x\in\mathbb{R}^{4s}} \|\mathbf{v}_r x\|_{\psi_2} \\ &= \sup_{x^Tx=1, x\in\mathbb{R}^{4s}} \|\mathbf{w}_r\Upsilon_{\mathbf{Q}}x\|_{\psi_2} \\ &\leq \sup_{y^Ty=1, y\in\mathbb{R}^{4m}} \|\mathbf{w}_r y\|_{\psi_2} = \|\mathbf{w}_r\|_{\psi_2},\end{aligned}$$

where the inequality comes from that $y^Ty = x^T\Upsilon_{\mathbf{Q}}^T\Upsilon_{\mathbf{Q}}x = x^Tx = 1$, and that y is in an s-dimensional subspace of $\mathbb{R}^{4m}$. Finally, since every entries of $\mathbf{w}_r$ has the same sub-Gaussian norm K, by Lemma B.1, $\|\mathbf{w}_r\|_{\psi_2} \leq C\max_{j\leq 4m}\|(\mathbf{w}_r)_j\|_{\psi_2} = CK$ for some absolulte constant C. Thus $\|\mathbf{v}\|_{\psi_2} \leq CK$. This completes the proof. □

Now we prove Theorem 4.5 based on Theorems 4.1 and 4.6.

Proof of Theorem 4.5. Recall that (26) of Theorem 4.1 implies

$$\|\mathbf{HX} - \mathbf{A}\|_F \leq \|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|_F + \|\mathbf{\Psi}_2^\dagger\mathbf{\Psi}_1\left(\mathbf{Q}_\perp^*\mathbf{A}\right)\|_F, \tag{64}$$

where $\mathbf{\Psi}_2 = \mathbf{\Psi Q} \in \mathbb{Q}^{l\times s}$, $\mathbf{\Psi}_1 = \mathbf{\Psi Q}_\perp \in \mathbb{Q}^{l\times(m-s)}$. On the other hand, (32) of Lemma 4.4 implies

$$\|\mathbf{A} - \mathbf{QQ}^*\mathbf{A}\|_F \leq \|\Sigma_2\|_F + \|\Sigma_2\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F, \tag{65}$$

where $\mathbf{\Omega}_1 = \mathbf{V}_1^*\mathbf{\Omega} \in \mathbb{Q}^{r\times s}$, $\mathbf{\Omega}_2 = \mathbf{V}_2^*\mathbf{\Omega} \in \mathbb{Q}^{(n-r)\times s}$. Recall that $\mathbf{\Psi}$ and $\mathbf{\Omega}$ are quaternion matrices; each entry of $\mathbf{\Psi}_r$ and $\mathbf{\Omega}_r$ is independent centered sub-Gaussian random variable with unit variance, all having the same sub-Gaussian norm K. Some deviation bounds will be given first.

By the property of the spectral norm, we have following estimations:

$$\|\mathbf{\Psi}_2^\dagger\mathbf{\Psi}_1\left(\mathbf{Q}_\perp^*\mathbf{A}\right)\|_F \le \|\mathbf{\Psi}_2^\dagger\|_2\|\mathbf{\Psi}_1\left(\mathbf{Q}_\perp^*\mathbf{A}\right)\|_F = \|\mathbf{\Psi}_2^\dagger\|_2\|\mathbf{\Psi}\mathbf{Q}_\perp\left(\mathbf{Q}_\perp^*\mathbf{A}\right)\|_F \tag{66}$$

$$\|\Sigma_2\mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\|_F \le \|\mathbf{\Omega}_1^\dagger\|_2\|\Sigma_2\mathbf{\Omega}_2\|_F = \|\mathbf{\Omega}_1^\dagger\|_2\|\Sigma_2\mathbf{V}_2^*\mathbf{\Omega}\|_F \tag{67}$$

We first bound $\|\mathbf{\Psi}_2^\dagger\|_2$ and $\|\mathbf{\Omega}_1^\dagger\|_2$ using Theorem 4.6. By definition, $\|\mathbf{\Psi}_2^\dagger\|_2 = 1/\sigma_{\min}\left(\mathbf{\Psi}_2\right)$ and $\|\mathbf{\Omega}_1^\dagger\|_2 = 1/\sigma_{\min}\left(\mathbf{\Omega}_1\right) = 1/\sigma_{\min}\left(\mathbf{\Omega}_1^*\right)$. By Lemma B.6, $\mathbf{\Psi}_2 \in \mathbb{Q}^{l\times s}$ $(s < l)$ and $\mathbf{\Omega}_1^* \in \mathbb{Q}^{s\times r}$ $(r < s)$ all has independent rows, which are all isotropic having sub-Gaussian norm CK for some constant $C > 0$. Thus, by Theorem 4.6, we have:

$$\mathbb{P}\left\{\|\mathbf{\Psi}_2^\dagger\|_2 > \frac{1}{2\sqrt{l} - 2C_K\sqrt{s} - 2t}\right\} \le \exp\left(-\frac{c_2t^2}{K^4}\right); \tag{68}$$

$$\mathbb{P}\left\{\|\mathbf{\Omega}_1^\dagger\|_2 > \frac{1}{2\sqrt{s} - 2C_K\sqrt{r} - 2t}\right\} \le \exp\left(-\frac{c_2t^2}{K^4}\right) \tag{69}$$

for some absolulte constant $c_2 > 0$ and C_K only dependes on the sub-Gaussian norm CK.

We next bound $\|\mathbf{\Psi}\mathbf{Q}_\perp\left(\mathbf{Q}_\perp^*\mathbf{A}\right)\|_F$ and $\|\Sigma_2\mathbf{V}_2^*\mathbf{\Omega}\|_F$ by Proposition 4.1. First conjugate transpose such that $\|\mathbf{\Psi}\mathbf{Q}_\perp\left(\mathbf{Q}_\perp^*\mathbf{A}\right)\|_F = \|\left(\mathbf{Q}_\perp^*\mathbf{A}\right)^*\mathbf{Q}_\perp^*\mathbf{\Psi}^*\|_F$; now $\left(\mathbf{Q}_\perp^*\mathbf{A}\right)^* \in \mathbb{Q}^{n\times(m-s)}$, $\mathbf{Q}_\perp^* \in \mathbb{Q}^{(m-s)\times m}$, and $\mathbf{\Psi}^* \in \mathbb{Q}^{m\times l}$. Since $\mathbf{Q}_\perp^* \in \mathbb{Q}^{(m-s)\times m}$ is row-orthonormal and $\mathbf{\Psi}^*$ is a quaternion sub-Gaussian matrix with every entries of $\mathbf{\Psi}_r^*$ having the same sub-Gaussian norm K, applying Proposition 4.1 with

$$t_1 := K\sqrt{\frac{2\log\left(\min\{m-s,n\}\right) + \log(4l) + 1}{c_1}},$$

where $c_1 > 0$ is the absolulte constant in Proposition 4.1, we have

$$\begin{aligned}
&\mathbb{P}\left\{\|\left(\mathbf{Q}_\perp^*\mathbf{A}\right)^*\mathbf{Q}_\perp^*\mathbf{\Psi}^*\|_F > 2\sqrt{l}t_1\|\mathbf{Q}_\perp^*\mathbf{A}\|_F\right\}\\
\le& 4e\cdot\exp\left(\log(l\min\{m-s,n\}) - c_1t_1^2/K^2\right)\\
=& 4e\cdot\exp\left(\log(l\min\{m-s,n\}) - 2\log(\min\{m-s,n\}) - \log(4l) - 1\right)\\
=& \frac{1}{\min\{m-s,n\}}.
\end{aligned}$$

Note that

$$\|\mathbf{Q}_\perp^*\mathbf{A}\|_F = \|\mathbf{Q}_\perp\mathbf{Q}_\perp^*\mathbf{A}\|_F = \|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|_F; \tag{70}$$

thus

$$\mathbb{P}\left\{\mathbf{\Psi}\mathbf{Q}_\perp\left(\mathbf{Q}_\perp^*\mathbf{A}\right)\|_F > 2\sqrt{l}t_1\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\|_F\right\} \le \frac{1}{\min\{m-s,n\}}. \tag{71}$$

Similarly, for $\|\Sigma_2\mathbf{V}_2^*\mathbf{\Omega}\|_F$, $\Sigma_2 \in \mathbb{Q}^{(\min\{m,n\}-r)\times(\min\{m,n\}-r)}$, $\mathbf{V}_2^* \in \mathbb{Q}^{(\min\{m,n\}-r)\times n}$, $\mathbf{\Omega} \in \mathbb{Q}^{n\times s}$. Applying Proposition 4.1 with

$$t_2 := K\sqrt{\frac{2\log(\min\{m,n\} - r) + \log(4s) + 1}{c_1}},$$

we have

$$\begin{aligned}
&\mathbb{P}\left\{\|\Sigma_2\mathbf{V}_2^*\mathbf{\Omega}\|_F > 2\sqrt{s}t_2\|\Sigma_2\|_F\right\}\\
\le& 4e\cdot\exp\left(\log\left(s(\min\{m,n\} - r)\right) - c_1t_2^2/K^2\right)\\
=& 4e\cdot\exp\left(\log\left(s(\min\{m,n\} - r)\right) - 2\log(\min\{m,n\} - r) - \log(4s) - 1\right)\\
=& \frac{1}{\min\{m,n\} - r}.
\end{aligned} \tag{72}$$

Finally, using (64), (66), (68), (71), and Lemma B.5, we have

$$\begin{aligned}&\mathbb{P}\left\{\|\mathbf{HX}-\mathbf{A}\|_F \le \left(1+\frac{2\sqrt{l}t_1}{2\sqrt{l}-2C_K\sqrt{s}-t}\right)\|\mathbf{A}-\mathbf{QQ}^*\mathbf{A}\|_F\right\}\\ \ge& 1-\exp\left(-\frac{c_2t^2}{K^4}\right)-\frac{1}{\min\{m-s,n\}}.\end{aligned}$$

(65), (67), (69), (72) together with Lemma B.5 gives

$$\mathbb{P}\left\{\|\mathbf{A}-\mathbf{QQ}^*\mathbf{A}\|_F \le \left(1+\frac{2\sqrt{s}t_2}{2\sqrt{s}-2C_K\sqrt{r}-t}\right)\|\Sigma_2\|_F^2\right\} \ge 1-\exp\left(-\frac{c_2t^2}{K^4}\right)-\frac{1}{\min\{m,n\}-r}.$$

Combining the above two inequalities and using again Lemma B.5 finally yields that

$$\begin{aligned}&\|\mathbf{HX}-\mathbf{A}\|_F\\ \le& \left(1+\frac{K\sqrt{l}\sqrt{2\log\left(\min\{m-s,n\}\right)+\log(4l)+1}}{(\sqrt{l}-C_K\sqrt{s}-t)\sqrt{c_1}}\right)\left(1+\frac{K\sqrt{s}\sqrt{2\log(\min\{m,n\}-r)+\log(4s)+1}}{(\sqrt{s}-C_K\sqrt{r}-t)\sqrt{c_1}}\right)\|\Sigma_2\|_F\end{aligned}$$

with probability at least $1-\exp\left(-\frac{c_2t^2}{K^4}\right)-\frac{1}{\min\{m-s,n\}}-\exp\left(-\frac{c_2t^2}{K^4}\right)-\frac{1}{\min\{m,n\}-r}$. □