Accelerating Quaternion Low-Rank Approximation for Color Images via Complex Embedding

Akshadha A[1], Devika Rani J[2], Sooraj K. Ambat[3]
[1]Department of Computer Science, Cochin University of Science and Technology, Kochi-682 022, India
[2]School of Computing, Amrita Vishwa Vidyapeetham, Kollam-690 525, India
[3]Naval Physical and Oceanographic Laboratory (NPOL), DRDO, Kochi–682 021, India
[1]aksaa0708@gmail.com [2]djrani862000@gmail.com [3]sooraj.npol@gov.in

***Abstract*—Quaternion-based representations are widely used in color image processing due to their ability to jointly model inter-channel correlations. However, low-rank approximation methods in the quaternion domain, such as Quaternion Singular Value Decomposition (QSVD) and Quaternion Bilateral Random Projections (QBRP), are computationally expensive and lack support from optimized numerical libraries. In this work, we propose a complex embedding framework for quaternion low-rank approximation, which maps quaternion matrices into an equivalent complex-valued representation of reduced dimensionality compared to real embeddings. Leveraging this embedding, standard complex SVD and randomized bilateral projection techniques can be directly applied using highly optimized BLAS and LAPACK routines. Experimental results on RGB image compression demonstrate that the proposed approach achieves comparable reconstruction quality to quaternion-domain methods, while reducing computational complexity and runtime, making it particularly advantageous for real-time applications.**

***Index Terms*—Quaternion Singular Value Decomposition, Quaternion Bilateral Random Projections, Complex Embedding**

I. Introduction

Color image processing has witnessed significant progress over the past decades, driven by advances in mathematical modelling and computational efficiency. Conventional real-valued representations typically process the red (R), green (G), and blue (B) channels independently. While effective in many applications, such representations fail to explicitly capture the strong inter-channel correlations that naturally exist among color components. Numerous studies have demonstrated that the RGB channels are highly correlated, and neglecting this dependency may lead to suboptimal performance in color image analysis and restoration tasks [1, 2].

Quaternion algebra provides a natural and powerful framework for modeling color images by encoding the three color channels into a single hypercomplex entity, with the imaginary components corresponding to the R, G, and B channels, thereby enabling unified processing while preserving inter-channel dependencies. Owing to this property, quaternion-based methods have been successfully applied to various color image processing tasks, including denoising, compression, and low-rank approximation.

Singular Value Decomposition (SVD) is a fundamental tool in image processing and data analysis, widely used for tasks such as compression, denoising, and dimensionality reduction. However, for high-dimensional data, deterministic SVD incurs significant computational and memory costs, limiting its applicability in large-scale or real-time scenarios.

To address this issue, randomized low-rank approximation methods have gained considerable attention. In particular, Bilateral Random Projections (BRP) [3] provide an efficient alternative by approximating the underlying low-rank structure using only a small number of matrix multiplications, thereby reducing computational complexity while maintaining competitive accuracy.

In the quaternion domain, the Quaternion Singular Value Decomposition (QSVD) extends SVD to color image representations, but the non-commutative nature of quaternion algebra and the lack of optimized implementations make it computationally expensive. Quaternion extensions of BRP overcome this by enabling efficient low-rank approximation of quaternion matrices, making them well-suited for large-scale color image processing. Such computations can be carried out directly in the quaternion domain or indirectly through real or complex embeddings. Existing real-valued embeddings map an $M \times N$ quaternion matrix into a $4M \times 4N$ real matrix; though effective, this substantially increases dimensionality, raising memory usage and computational overhead. This motivates the question: can quaternion low-rank approximation be performed more efficiently without sacrificing reconstruction quality?

In this work, we answer this question affirmatively by proposing a complex embedding framework for quaternion low-rank approximation. Instead of relying on real-valued representations, the proposed approach embeds a quaternion matrix into a complex matrix of size $2M \times 2N$, reducing both computational and memory costs by a factor of four compared with real embeddings. Moreover, complex arithmetic can directly leverage highly optimized BLAS and LAPACK implementations that are widely supported across modern hardware architectures [4, 5]. This allows the proposed framework to fully exploit optimized numerical linear algebra routines, achieving substantial speedups over both direct quaternion computations and real-valued embeddings.

Based on the proposed complex embedding, we develop efficient SVD and BRP-based low-rank approximation schemes in the complex domain, followed by a reconstruction back to the quaternion representation. Theoretical complexity analysis and extensive experiments on color image datasets demonstrate that the proposed methods achieve comparable or supe-

rior reconstruction quality while significantly reducing computational time and memory consumption.

The main contributions of this paper are:

- We leverage an existing complex embedding of quaternions to address the problem of efficient low-rank approximation. Specifically, we propose a novel method for optimizing quaternion-based low-rank approximations within this complex embedding framework.
- We develop SVD- and BRP-based low-rank approximation schemes operating entirely in the complex domain.
- We provide a comprehensive theoretical complexity comparison among direct quaternion methods, real embeddings, and the proposed complex embedding approach.
- We validate the effectiveness and efficiency of the proposed methods through extensive experiments on color image datasets.

Notations: Throughout this paper, bold uppercase letters denote matrices, lowercase letters denote scalars. Quaternion-valued scalars and matrices are denoted using a dot $(\cdot)$, e.g., $\dot{q}$ and $\dot{\mathbf{Q}}$. $\dot{\mathbf{Q}}(x, y)$ denotes the quaternion-valued entry at position (x, y) of $\dot{\mathbf{Q}}$.

II. Preliminaries

A. Quaternion Algebra and Notation

Quaternion algebra, first introduced by Hamilton in 1843, extends complex numbers to a four-dimensional hypercomplex number system. A quaternion $\dot{q} \in \mathbb{H}$ can be expressed as

$$\dot{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k},$$

where $q_0, q_1, q_2, q_3 \in \mathbb{R}$. Here, q_0 denotes the real part of q, while (q_1, q_2, q_3) constitutes its imaginary components. The imaginary units $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ satisfy the following multiplication rules:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1, \qquad \text{and}$$
$$\mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \quad \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \quad \mathbf{ki} = -\mathbf{ik} = \mathbf{j}.$$

A quaternion matrix $\dot{\mathbf{Q}} \in \mathbb{H}^{M\times N}$ can be represented in component form as

$$\dot{\mathbf{Q}} = \mathbf{Q}_0 + \mathbf{Q}_1\mathbf{i} + \mathbf{Q}_2\mathbf{j} + \mathbf{Q}_3\mathbf{k}, \tag{1}$$

where $\mathbf{Q}_\ell \in \mathbb{R}^{M\times N}$ for $\ell = 0, 1, 2, 3$.

B. Quaternion Representation of RGB Images

In color image processing, an RGB image can be naturally represented using pure quaternions (where the real part is zero), which allows joint processing of color channels while preserving their intrinsic correlations. Specifically, given an RGB image with pixel values $\mathbf{R}(x, y)$, $\mathbf{G}(x, y)$, and $\mathbf{B}(x, y)$ at spatial location (x, y), each color pixel can be encoded as a pure quaternion:

$$\dot{\mathbf{Q}}(x, y) = \mathbf{R}(x, y)\mathbf{i} + \mathbf{G}(x, y)\mathbf{j} + \mathbf{B}(x, y)\mathbf{k}.$$

By stacking all pixels, the color image can be represented as a pure quaternion matrix $\dot{\mathbf{Q}} \in \mathbb{H}^{M\times N}$. This representation enables unified processing of color information and has been shown to be effective in various low-rank approximation and restoration tasks [6].

C. Low-Rank Approximation

Let $\dot{\mathbf{Q}} \in \mathbb{H}^{M\times N}$ be a quaternion matrix. The objective of quaternion low-rank approximation is to find a matrix $\widehat{\dot{\mathbf{Q}}} \in \mathbb{H}^{M\times N}$ of rank at most r, where $r \ll \min(M, N)$, such that $\widehat{\dot{\mathbf{Q}}}$ provides a good approximation to $\dot{\mathbf{Q}}$. This problem can be formulated as

$$\min_{\operatorname{rank}(\widehat{\dot{\mathbf{Q}}})\le r} \left\| \dot{\mathbf{Q}} - \widehat{\dot{\mathbf{Q}}} \right\|_F, \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

The rank of a quaternion matrix is defined in terms of its Quaternion Singular Value Decomposition (QSVD). Specifically, given the QSVD of $\dot{\mathbf{Q}}$, the rank is equal to the number of non-zero singular values. This definition is consistent with the notion of matrix rank in the real and complex domains and remains invariant under equivalent real or complex embeddings of quaternion matrices.

Low-rank approximation based on QSVD provides the optimal solution for (2) in the least-squares sense. However, the direct computation of QSVD is computationally expensive for large-scale quaternion matrices, motivating the development of efficient approximate methods such as randomized algorithms [6].

III. Quaternion based Low-Rank Approximation Methods

A. Quaternion Domain Methods

1) Quaternion Singular Value Decomposition (QSVD): Given a quaternion matrix $\dot{\mathbf{Q}} \in \mathbb{H}^{M\times N}$ obtained from the quaternion representation of an RGB image, the QSVD factorizes $\dot{\mathbf{Q}}$ as

$$\dot{\mathbf{Q}} = \dot{\mathbf{U}}\,\mathbf{\Sigma}\,\dot{\mathbf{V}}^H,$$

where $\dot{\mathbf{U}} \in \mathbb{H}^{M\times M}$ and $\dot{\mathbf{V}} \in \mathbb{H}^{N\times N}$ are unitary quaternion matrices, $\mathbf{\Sigma} \in \mathbb{R}^{M\times N}$ is a diagonal matrix containing the singular values, and $(\cdot)^H$ denotes the quaternion conjugate transpose. A rank-r approximation is obtained by keeping the top-r singular values and associated singular vectors.

2) Quaternion Bilateral Random Projections (QBRP) [6]: To reduce the computational burden of QSVD, QBRP has been proposed as an efficient alternative for quaternion low-rank approximation. Instead of explicitly computing the QSVD, QBRP approximates the column and row subspaces of $\dot{\mathbf{Q}}$ using randomized projections in the quaternion domain. The resulting low-rank approximation can be obtained using only a small number of quaternion matrix multiplications, significantly reducing computational complexity while maintaining satisfactory approximation accuracy.

B. Real-Embedded Quaternion Methods

An alternative approach to quaternion matrix computation is to embed a quaternion matrix into an equivalent real-valued matrix. Given a quaternion matrix $\dot{\mathbf{Q}}$ with a representation as in (1), its real-valued embedding can be expressed as

$$\mathbf{Q}^{\mathbb{R}} = \begin{bmatrix} \mathbf{Q}_0 & -\mathbf{Q}_1 & -\mathbf{Q}_2 & -\mathbf{Q}_3 \\ \mathbf{Q}_1 & \mathbf{Q}_0 & -\mathbf{Q}_3 & \mathbf{Q}_2 \\ \mathbf{Q}_2 & \mathbf{Q}_3 & \mathbf{Q}_0 & -\mathbf{Q}_1 \\ \mathbf{Q}_3 & -\mathbf{Q}_2 & \mathbf{Q}_1 & \mathbf{Q}_0 \end{bmatrix} \in \mathbb{R}^{4M\times 4N}.$$

This embedding preserves the algebraic structure of quaternion matrix operations while expanding the matrix dimension by a factor of four in each direction. Once embedded, standard real-valued low-rank approximation techniques, such as singular value decomposition or randomized projection methods, can be applied directly.

Dimensional Expansion and Limitations: Although real embeddings enable the use of standard real-valued linear algebra routines, the significant dimensional expansion from $M \times N$ to $4M \times 4N$ limits their efficiency for large-scale problems.

C. Real Domain Low-Rank Approximation Methods

As a baseline, low-rank approximation can also be performed directly in the real domain by processing color channels independently. Given an RGB image, the three color components are first separated as $\mathbf{R}, \mathbf{G}, \mathbf{B} \in \mathbb{R}^{M\times N}$. These components can be vertically stacked to form a real-valued matrix $\mathbf{C} = \left[\mathbf{R^T G^T B^T}\right]^T \in \mathbb{R}^{3M\times N}$. Standard real-valued low-rank approximation methods can then be applied to $\mathbf{C}$. While computationally efficient, this approach treats each color channel independently and does not explicitly model inter-channel correlations.

IV. Proposed Methodology

A. Complex Embedding of Quaternion Matrices

To reduce the computational and memory overhead associated with real-valued embeddings, we adopt a complex embedding framework for quaternion matrices. Given a quaternion matrix $\dot{\mathbf{Q}}$ with a representation as in (1), its equivalent complex representation is defined as

$$\mathbf{Q}^{\mathbb{C}} = \begin{bmatrix} \mathbf{Q}_0 + \mathbf{i}\mathbf{Q}_1 & \mathbf{Q}_2 + \mathbf{i}\mathbf{Q}_3 \\ -\mathbf{Q}_2 + \mathbf{i}\mathbf{Q}_3 & \mathbf{Q}_0 - \mathbf{i}\mathbf{Q}_1 \end{bmatrix} \in \mathbb{C}^{2M\times 2N}.$$

Compared with the real embedding $\mathbf{Q}^{\mathbb{R}} \in \mathbb{R}^{4M\times 4N}$, the proposed complex embedding reduces both dimensional expansion and memory usage by a factor of four while preserving the algebraic structure of quaternion matrix operations.

Inverse Complex Embedding: Let $\hat{\mathbf{C}} \in \mathbb{C}^{2M\times 2N}$ be a low-rank approximation of $\mathbf{Q}^{\mathbb{C}}$. Partition $\hat{\mathbf{C}}$ into four $M \times N$ blocks:

$$\hat{\mathbf{C}} = \begin{bmatrix} \hat{\mathbf{C}}_{11} & \hat{\mathbf{C}}_{12} \\ \hat{\mathbf{C}}_{21} & \hat{\mathbf{C}}_{22} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{Q}}_0 + \mathbf{i}\hat{\mathbf{Q}}_1 & \hat{\mathbf{Q}}_2 + \mathbf{i}\hat{\mathbf{Q}}_3 \\ -\hat{\mathbf{Q}}_2 + \mathbf{i}\hat{\mathbf{Q}}_3 & \hat{\mathbf{Q}}_0 - \mathbf{i}\hat{\mathbf{Q}}_1 \end{bmatrix}.$$

where $\hat{\mathbf{Q}}_0, \hat{\mathbf{Q}}_1, \hat{\mathbf{Q}}_2$, and $\hat{\mathbf{Q}}_3$ are the components of the quaternion matrix, and $\mathbf{i}$ denotes the imaginary unit.

Once the complex matrix is split, we reconstruct the quaternion components by isolating the real and imaginary parts from the sub-blocks. The real and imaginary parts of $\hat{\mathbf{C}}_{11}, \hat{\mathbf{C}}_{12}, \hat{\mathbf{C}}_{21}, \hat{\mathbf{C}}_{22}$ will give us estimates for each component of the quaternion matrix. For each quaternion component $\mathbf{Q}_\ell$, the corresponding real and imaginary parts are averaged across the relevant blocks to improve stability and reduce the impact of noise: $\hat{\mathbf{Q}}_0 = \frac{1}{2}\Re(\hat{\mathbf{C}}_{11}+\hat{\mathbf{C}}_{22}), \hat{\mathbf{Q}}_1 = \frac{1}{2}\Im(\hat{\mathbf{C}}_{11}-\hat{\mathbf{C}}_{22}), \hat{\mathbf{Q}}_2 = \frac{1}{2}\Re(\hat{\mathbf{C}}_{12}-\hat{\mathbf{C}}_{21}), \hat{\mathbf{Q}}_3 = \frac{1}{2}\Im(\hat{\mathbf{C}}_{12}+\hat{\mathbf{C}}_{21})$. where $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts, respectively.

Hence, the reconstructed quaternion matrix is $\hat{\dot{\mathbf{Q}}} = \hat{\mathbf{Q}}_0 + \hat{\mathbf{Q}}_1\mathbf{i}+\hat{\mathbf{Q}}_2\mathbf{j}+\hat{\mathbf{Q}}_3\mathbf{k}$. This process of inverse complex embedding allows the recovery of the original quaternion matrix while maintaining the algebraic structure of quaternion matrix operations.

Implementation Advantages: An additional advantage of the complex embedding framework is its implementation efficiency. Modern libraries such as BLAS and LAPACK provide highly optimized support for complex-valued linear algebra, enabling efficient memory access and parallel computation [7]. In contrast, quaternion arithmetic lacks standardized support, making complex-domain formulations computationally more efficient without compromising approximation quality.

B. SVD via Complex-Embedded Quaternion for RGB Image Compression

The scheme for color image compression using complex embedding is given in Algorithm 1. Given an RGB image, we first construct its quaternion representation and then map it to the complex-embedded matrix $\mathbf{Q}^{\mathbb{C}} \in \mathbb{C}^{2M\times 2N}$. A truncated complex SVD (retaining the top-r singular components) is applied to $\mathbf{Q}^{\mathbb{C}}$ to obtain a rank-r approximation, which preserves the dominant spectral components while discarding low-energy information. This yields a compact representation controlled by the selected rank, allowing a direct trade-off between reconstruction quality and compression rate.

After truncation, the compressed complex matrix is converted back to the quaternion domain through inverse complex embedding. Specifically, the recovered blocks are used to reconstruct the quaternion components, producing the estimate $\hat{\dot{\mathbf{Q}}}$, from which the RGB channels are obtained. Because

Algorithm 1 SVD via Complex-Embedded Quaternion for RGB Image Compression

Input: Color Image $\mathbf{X} \in \mathbb{R}^{M\times N\times 3}$, target rank r
Ensure: $r \le \min(M, N)$

1: Form complex embedding

$$\mathbf{C} = \begin{bmatrix} \mathbf{Q}_0 + \mathbf{i}\mathbf{Q}_1 & \mathbf{Q}_2 + \mathbf{i}\mathbf{Q}_3 \\ -\mathbf{Q}_2 + \mathbf{i}\mathbf{Q}_3 & \mathbf{Q}_0 - \mathbf{i}\mathbf{Q}_1 \end{bmatrix},$$

where $\mathbf{Q}_0 = \mathbf{0}$, $\mathbf{Q}_1 = \mathbf{R}$, $\mathbf{Q}_2 = \mathbf{G}$, $\mathbf{Q}_3 = \mathbf{B}$

2: Compute truncated complex SVD of $\mathbf{C}$: $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, $\mathbf{U}_r = \mathbf{U}(:, 1:r)$, $\mathbf{\Sigma}_r = \mathbf{\Sigma}(1:r, 1:r)$, $\mathbf{V}_r = \mathbf{V}(:, 1:r)$, and $\widehat{\mathbf{C}} = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^H$.

3: Recover $\widehat{\dot{\mathbf{Q}}}$ from $\widehat{\mathbf{C}}$ using the inverse complex embedding described in Section IV-A.

4: Extract compressed RGB channels: $\widehat{\mathbf{R}} = \widehat{\mathbf{Q}}_1$, $\widehat{\mathbf{G}} = \widehat{\mathbf{Q}}_2$, $\widehat{\mathbf{B}} = \widehat{\mathbf{Q}}_3$ {Discard $\widehat{\mathbf{Q}}_0$}.

Output: Compressed RGB image

the complex embedding keeps cross-channel coupling in a structured form, the method preserves color consistency better than channel-wise real SVD while requiring substantially less expansion than full real-valued quaternion embeddings.

C. BRP via Complex-Embedded Quaternion for RGB Image Compression

To further reduce computational cost, we replace deterministic SVD with Bilateral Random Projections (BRP) in the same complex-embedded pipeline. Starting from $\mathbf{Q}^{\mathbb{C}}$, BRP computes a low-rank approximation using random left and right projections and small core factorizations instead of a full decomposition. This significantly lowers runtime and memory overhead, especially for large images, while still capturing the principal low-rank structure needed for compression.

The resulting low-rank approximation $\hat{\mathbf{C}} \in \mathbb{C}^{2M\times 2N}$ is then mapped back via inverse complex embedding to reconstruct $\hat{\dot{\mathbf{Q}}}$ and the corresponding RGB image. The complex-embedded quaternion BRP pipeline is summarized in Algorithm 2.

V. Experimental Results

A. Experimental Setup

All experiments used four standard 512 × 512 RGB images widely used in the image processing literature: Lena, Peppers, Airplane (F-16), and Baboon [8]. These span a range of texture complexity, edge structures, and color distributions, providing a comprehensive testbed for evaluating reconstruction quality and computational performance. Each image was represented in the quaternion domain by encoding its three color channels into the imaginary components of a pure quaternion matrix. All methods were evaluated at multiple rank values $r \in \{32, 64, 128\}$ to study reconstruction performance and computational behavior under varying compression levels.

Algorithm 2 BRP via Complex-Embedded Quaternion for RGB Image Compression

Input: Color Image $\mathbf{X} \in \mathbb{R}^{M\times N\times 3}$, target rank r
Ensure: $r \le \min(M, N)$

1: Form complex embedding

$$\mathbf{C} = \begin{bmatrix} \mathbf{Q}_0 + \mathbf{i}\mathbf{Q}_1 & \mathbf{Q}_2 + \mathbf{i}\mathbf{Q}_3 \\ -\mathbf{Q}_2 + \mathbf{i}\mathbf{Q}_3 & \mathbf{Q}_0 - \mathbf{i}\mathbf{Q}_1 \end{bmatrix},$$

where $\mathbf{Q}_0 = \mathbf{0}$, $\mathbf{Q}_1 = \mathbf{R}$, $\mathbf{Q}_2 = \mathbf{G}$, $\mathbf{Q}_3 = \mathbf{B}$

2: Generate Complex Standard Gaussian Random matrices $\mathbf{\Omega}_1 \in \mathbb{C}^{2N\times r}$, $\mathbf{\Omega}_2 \in \mathbb{C}^{2M\times r}$
3: Compute projections $\mathbf{Y}_1 = \mathbf{C}\mathbf{\Omega}_1$, $\mathbf{Y}_2 = \mathbf{C}^H\mathbf{\Omega}_2$
4: Low-rank approximation $\mathbf{C}_r = \mathbf{Y}_1(\mathbf{\Omega}_2^H\mathbf{Y}_1)^{-1}\mathbf{Y}_2^H$
5: Recover $\hat{\dot{\mathbf{Q}}}$ from $\widehat{\mathbf{C}}$ using the inverse complex embedding described in Section IV-A.
6: Extract compressed RGB channels: $\widehat{\mathbf{R}} = \widehat{\mathbf{Q}}_1$, $\widehat{\mathbf{G}} = \widehat{\mathbf{Q}}_2$, $\widehat{\mathbf{B}} = \widehat{\mathbf{Q}}_3$ {Discard $\widehat{\mathbf{Q}}_0$}.

Output: Compressed RGB image

All algorithms were implemented in MATLAB Online (R2025b, v25.2.0) on a 64-bit Ubuntu 22.04 Linux environment (AWS-hosted, Intel Xeon Platinum 8488C). To ensure reproducible runtime evaluation, experiments were restricted to a single thread via *maxNumCompThreads(1)*, eliminating variability from automatic multi-threading. The Linux host reported approximately 124 GB RAM; MATLAB Online session memory limits could not be determined due to unavailable memory-reporting functions. A fixed random seed *(rng(0, 'twister'))* was used for all randomized algorithms to ensure reproducibility. Each experiment was repeated over 20 independent runs, and the reported results are the averages across these runs.

B. Results and Discussion

Table I reports quantitative reconstruction performance (PSNR, SSIM, MSE, and runtime) for Lena, Peppers, Baboon and Airplane images of size 512×512 at ranks 32, 64, and 128. Representative visual reconstruction results are provided in the Supplementary Material owing to space limitations. The results consistently show that the proposed BRP-based formulations deliver a strong efficiency–quality trade-off compared with the Quaternion SVD baseline. For Lena at Rank 64, runtime drops from 10.489 s to 0.053 s (≈99.5% faster), while PSNR decreases slightly from 31.490 to 31.273 (0.69% loss). For Baboon at Rank 64, runtime decreases from 10.717 s to 0.054 s (≈99.5% faster), with PSNR decreasing from 23.271 to 23.144 (0.55% loss). SSIM values remain close to those of Quaternion SVD (from 0.826 to 0.818 for Baboon), indicating that structural content is well preserved. Similar trends are observed across other images, confirming that Q-to-C† BRP maintains reconstruction quality while dramatically reducing runtime.

Across all images, Q-to-C† BRP consistently achieves substantial runtime reductions while preserving reconstruction quality close to its SVD counterpart. At Rank 64, runtime drops from 0.848 s to 0.053 s on Lena (93.75% faster) and from 0.850 s to 0.054 s on Baboon (93.65% faster). Corresponding PSNR reductions are small: 31.490 to 31.273 (0.69%) for Lena and 23.271 to 23.144 (0.55%) for Baboon, with similarly minor SSIM changes.

Compared with Q-to-R* SVD, Q-to-C† BRP remains substantially faster across all images and ranks (e.g., at Rank 64: Lena 2.887 s vs 0.053 s, Baboon 2.946 s vs 0.054 s), while maintaining acceptable quality for compression-oriented scenarios. Compared with Real BRP, Q-to-C† BRP typically provides higher PSNR/SSIM at slightly increased runtime, reflecting a favorable balance between ultra-fast but lower-fidelity schemes and high-fidelity but computationally expensive SVD-type methods. Overall, these results support Q-to-C† BRP as a practical choice for real-time or large-scale applications where both speed and reconstruction quality are critical.

VI. Conclusion

This paper addressed the problem of efficient low-rank approximation for quaternion-valued color images, where di-

TABLE I
COMPARATIVE EVALUATION OF SVD AND BRP-BASED IMAGE RECONSTRUCTION FOR RANKS 32, 64, 128

Image	Method	PSNR	SSIM	MSE ($\times 10^{-3}$)	Time (s)
Lena	Quat. SVD	27.589 | 31.490 | 36.810	0.963 | 0.982 | 0.994	1.742 | 0.710 | 0.208	10.866 | 10.489 | 10.322
	Quat. BRP	26.863 | 30.631 | 34.836	0.957 | 0.979 | 0.991	2.059 | 0.865 | 0.342	0.026 | 0.044 | 0.084
	Q-to-R* SVD	27.589 | 31.490 | 36.810	0.963 | 0.982 | 0.994	1.742 | 0.710 | 0.208	2.986 | 2.887 | 2.927
	Q-to-R* BRP	27.684 | 31.633 | 36.904	0.965 | 0.983 | 0.994	1.704 | 0.687 | 0.204	0.091 | 0.113 | 0.228
	Q-to-C† SVD	27.589 | 31.490 | 36.810	0.963 | 0.982 | 0.994	1.742 | 0.710 | 0.208	0.843 | 0.848 | 0.853
	Q-to-C† BRP	27.390 | 31.273 | 36.484	0.962 | 0.982 | 0.994	1.824 | 0.746 | 0.225	**0.028** | **0.053** | **0.110**
	Real SVD	26.609 | 30.286 | 34.990	0.955 | 0.977 | 0.991	2.183 | 0.936 | 0.317	0.082 | 0.082 | 0.085
	Real BRP	25.528 | 29.098 | 33.668	0.948 | 0.972 | 0.989	2.801 | 1.231 | 0.430	0.004 | 0.007 | 0.012
Peppers	Quat. SVD	27.379 | 31.346 | 35.692	0.961 | 0.982 | 0.993	1.828 | 0.733 | 0.270	7.182 | 7.105 | 7.144
	Quat. BRP	26.577 | 30.480 | 34.480	0.954 | 0.979 | 0.991	2.199 | 0.895 | 0.357	0.014 | 0.024 | 0.049
	Q-to-R* SVD	27.379 | 31.346 | 35.692	0.961 | 0.982 | 0.993	1.828 | 0.733 | 0.270	0.743 | 0.714 | 0.733
	Q-to-R* BRP	27.412 | 31.343 | 35.733	0.962 | 0.983 | 0.993	1.815 | 0.734 | 0.267	0.017 | 0.031 | 0.057
	Q-to-C† SVD	27.379 | 31.346 | 35.692	0.961 | 0.982 | 0.993	1.828 | 0.733 | 0.270	0.246 | 0.242 | 0.250
	Q-to-C† BRP	27.101 | 31.035 | 35.353	0.959 | 0.981 | 0.993	1.949 | 0.788 | 0.292	**0.008** | **0.017** | **0.033**
	Real SVD	25.794 | 29.669 | 33.637	0.946 | 0.975 | 0.989	2.634 | 1.079 | 0.433	0.047 | 0.047 | 0.048
	Real BRP	24.713 | 28.577 | 32.479	0.936 | 0.969 | 0.986	3.379 | 1.388 | 0.565	0.003 | 0.004 | 0.005
Baboon	Quat. SVD	21.131 | 23.271 | 27.171	0.717 | 0.826 | 0.926	7.707 | 4.709 | 1.918	10.063 | 10.717 | 6.995
	Quat. BRP	20.639 | 22.554 | 26.088	0.692 | 0.797 | 0.907	8.632 | 5.554 | 2.462	0.025 | 0.045 | 0.047
	Q-to-R* SVD	21.131 | 23.271 | 27.171	0.717 | 0.826 | 0.926	7.707 | 4.709 | 1.918	2.970 | 2.946 | 0.689
	Q-to-R* BRP	21.219 | 23.473 | 27.417	0.720 | 0.830 | 0.930	7.553 | 4.495 | 1.813	0.061 | 0.114 | 0.048
	Q-to-C† SVD	21.131 | 23.271 | 27.171	0.717 | 0.826 | 0.926	7.707 | 4.709 | 1.918	0.830 | 0.850 | 0.221
	Q-to-C† BRP	21.016 | 23.144 | 26.927	0.710 | 0.818 | 0.922	7.914 | 4.849 | 2.029	**0.027** | **0.054** | **0.031**
	Real SVD	20.703 | 22.634 | 25.967	0.693 | 0.802 | 0.905	8.506 | 5.453 | 2.531	0.081 | 0.083 | 0.046
	Real BRP	20.020 | 21.681 | 24.637	0.663 | 0.763 | 0.875	9.953 | 6.791 | 3.438	0.004 | 0.007 | 0.005
Airplane	Quat. SVD	27.648 | 32.355 | 38.792	0.835 | 0.907 | 0.966	1.719 | 0.581 | 0.132	7.389 | 7.700 | 7.458
	Quat. BRP	26.866 | 31.372 | 34.630	0.787 | 0.877 | 0.898	2.058 | 0.729 | 0.368	0.015 | 0.025 | 0.051
	Q-to-R* SVD	27.648 | 32.355 | 38.792	0.835 | 0.907 | 0.966	1.719 | 0.581 | 0.132	0.817 | 0.874 | 0.858
	Q-to-R* BRP	27.827 | 32.529 | 38.960	0.849 | 0.918 | 0.970	1.649 | 0.559 | 0.127	0.017 | 0.036 | 0.064
	Q-to-C† SVD	27.648 | 32.355 | 38.792	0.835 | 0.907 | 0.966	1.719 | 0.581 | 0.132	0.258 | 0.278 | 0.280
	Q-to-C† BRP	27.481 | 32.134 | 38.510	0.825 | 0.904 | 0.965	1.786 | 0.612 | 0.141	**0.009** | **0.017** | **0.033**
	Real SVD	27.242 | 31.739 | 37.441	0.833 | 0.900 | 0.958	1.887 | 0.670 | 0.180	0.049 | 0.049 | 0.049
	Real BRP	26.065 | 30.410 | 36.057	0.792 | 0.870 | 0.942	2.475 | 0.910 | 0.248	0.003 | 0.004 | 0.006

Note: Values in each cell follow the sequence Rank 32 | 64 | 128.
*Q-to-R: Quaternion-to-Real transformation variant. †Q-to-C: Quaternion-to-Complex transformation variant.

rect quaternion-domain methods, while effective in preserving inter-channel correlations, suffer from high computational cost. To overcome this limitation, we proposed a novel complex embedding framework that maps quaternion matrices into structured complex matrices of reduced dimensionality.

By performing low-rank approximation using SVD and bilateral random projections in the complex domain, the proposed approach significantly reduces both computational complexity and memory requirements compared to existing real-embedded quaternion methods, while maintaining competitive reconstruction quality. The availability of highly optimized complex BLAS and LAPACK implementations further enhances the practical efficiency of the proposed framework.

Extensive experiments on color image datasets demonstrate that the proposed complex-embedded methods achieve a favorable balance between speed and accuracy, making them well suited for large-scale and real-time color image processing tasks.

REFERENCES

[1] S. J. Sangwine and R. E. N. Horne. "Colour Image Processing". In: *The Colour Image Processing Handbook*. Springer, 1999, pp. 68–69.

[2] Dan He et al. "Inter-Channel Correlation Modeling and Improved Skewed Histogram Shifting for Reversible Data Hiding in Color Images". In: *IEEE Transactions on Information Forensics and Security* 15.8 (2020), pp. 2345–2359.

[3] T. Zhou and D. Tao. *Bilateral Random Projections*. 2011. eprint: arXiv:1112.5215.

[4] Jack J. Dongarra et al. "An Extended Set of FORTRAN Basic Linear Algebra Subprograms". In: *ACM Transactions on Mathematical Software* 14.1 (1988), pp. 1–17.

[5] Field G. Van Zee. "Implementing High-Performance Complex Matrix Multiplication via the 1M Method". In: *ACM Transactions on Mathematical Software* 42.4 (2016), pp. 1–29.

[6] Jifei Miao and Kit Ian Kou. *Constrained low-rank quaternion approximation for color image denoising by bilateral random projections*. 2020. eprint: 2006.00749.

[7] Intel Corporation. *Intel oneAPI Math Kernel Library Developer Guide*. 2025.

[8] NYU – EL5123. *Sample Data*. https://eeweb.engineering.nyu.edu/~yao/EL5123/SampleData.html.